In [26]:

```python
import numpy as np
import pandas as pd
import xlsxwriter as ExcelWriter
import statsmodels.tsa.api as smt
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from matplotlib.dates import DateFormatter
import seaborn as sns
import datetime as dt
from pandas.plotting import register_matplotlib_converters # Handle date-time
register_matplotlib_converters()                            #conversions between
                                                            #pandas and matplotlib
%matplotlib inline
import linearmodels as plm
import statsmodels.api as sm
import statsmodels.formula.api as smf
import statsmodels.tsa.api as smt
import statsmodels.stats.outliers_influence as smo
from statsmodels.tsa.api import VAR
from statsmodels.stats.outliers_influence import variance_inflation_factor

from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.tsa.arima_process import ArmaProcess
from statsmodels.tsa.stattools import pacf
from statsmodels.regression.linear_model import yule_walker
from statsmodels.tsa.stattools import adfuller
import matplotlib.pyplot as plt
import numpy as np
```

```
# Panel Dataset:
Date = Start 2020-03-04, End 2021-12-31
Month = 1 if Jan., 2 if Feb., 3 if Mar., and so forth.
Day = 1 if yyyy-mm-01, 2 if yyyy-mm-02, 3 if yyyy-mm-03, and so forth.
DateWk = String for names of each weekday, Monday thru Sunday
DSP = 1 if Apple Music, 2 if SoundCloud, 3 if Spotify;
Streams = Daily # of streams from associated Digital Streaming
Platform(DSP)
Listeners = Daily # of listeners from associated Digital Streaming
Platform(DSP)
CMEngage = ChartMetric growth in Engagement Ranking
CMCrossP = ChartMetric growth in Cross-Platform Performance
TWFollowers = Daily count of Twitter Followers
TWRetweets = Daily # of Twitter Retweets
IGReach = Daily # of unique accounts reached by Instagrame Story
```

In [27]:

```python
Panel = pd.read_csv("/Users"
                    "/andrew7"
                    "/Desktop/"
                    "[TONE_DA]/"
                    "Estimation 22/"
                    "Consolidated Data 01 2022"
                    "/Tone_Est_Paneled II.csv",
                   parse_dates = [0])
Panel
```

Out[27]:

	Date	Month	Day	DateWk	Year	DSP	Streams	Listeners	CMEngage	CMCrossP	TWFollowers	TV

	Date	Month	Day	DateWk	Year	DSP	Streams	Listeners	CMEngage	CMCrossP	TWFollowers	TV
0	2020-03-04	3	4	Wednesday	2020	1	7	7	379454	465864	226	
1	2020-03-05	3	5	Thursday	2020	1	58	7	404875	487377	226	
2	2020-03-06	3	6	Friday	2020	1	59	8	407748	494572	227	
3	2020-03-07	3	7	Saturday	2020	1	57	4	413262	498931	227	
4	2020-03-08	3	8	Sunday	2020	1	19	5	411552	497810	227	
...	...	...	...	...	...	...	...	...	...	...	...	
1998	2021-12-27	12	27	Monday	2021	3	5430	4137	131006	160044	381	
1999	2021-12-28	12	28	Tuesday	2021	3	5917	4513	128797	157601	381	
2000	2021-12-29	12	29	Wednesday	2021	3	6479	4782	117195	143391	381	
2001	2021-12-30	12	30	Thursday	2021	3	5854	4571	116277	142352	381	
2002	2021-12-31	12	31	Friday	2021	3	5476	4283	114926	140372	382	

2003 rows × 13 columns

In [28]:

```
#Panel['lnDiffEGR']        = np.log(Panel['CMEngage'].diff()).bfill().ffill()
#Panel['lnDiffIGR']    = np.log(Panel['IGReach'].diff()).bfill().ffill()
#Panel['lnDiffS']    = np.log(Panel['Streams'].diff()).bfill().ffill()
#Panel['lnDiffL'] = np.log(Panel['Listeners'].diff()).bfill().ffill()
#Panel['lnDiffTWR']        = np.log(Panel['TWRetweets'].diff()).bfill().ffill()
#Panel['lnDiffTWF']        = np.log(Panel['TWFollowers'].diff()).bfill().ffill()
#Panel[['lnDiffIGR','lnDiffEGR','lnDiffS','lnDiffL','lnDiffTWR','lnDiffTWF']]
```

```
# Mapping dicts of strings for the following:
    Day of the Week;
    Digital Streaming Platform(DSP)
```

In [29]:

```
DayWkdict = {'Sunday': 1,
             'Monday': 2,
             'Tuesday': 3,
             'Wednesday': 4,
             'Thursday': 5,
             'Friday': 6,
             'Saturday': 7}
Panel['DayWk'] = Panel['DateWk'].map(DayWkdict)
DSPdict = {1:'Apple Music',
           2:'SoundCloud',
           3:'Spotify'}
Panel['DSPstr'] = Panel['DSP'].map(DSPdict)
Panel
```

Out[29]:

	Date	Month	Day	DateWk	Year	DSP	Streams	Listeners	CMEngage	CMCrossP	TWFollowers	TV
0	2020-03-04	3	4	Wednesday	2020	1	7	7	379454	465864	226	
1	2020-03-05	3	5	Thursday	2020	1	58	7	404875	487377	226	
2	2020-03-06	3	6	Friday	2020	1	59	8	407748	494572	227	
3	2020-03-07	3	7	Saturday	2020	1	57	4	413262	498931	227	
4	2020-03-08	3	8	Sunday	2020	1	19	5	411552	497810	227	
...	...	...	...	...	...	...	...	...	...	...	...	
1998	2021-12-27	12	27	Monday	2021	3	5430	4137	131006	160044	381	
1999	2021-12-28	12	28	Tuesday	2021	3	5917	4513	128797	157601	381	
2000	2021-12-29	12	29	Wednesday	2021	3	6479	4782	117195	143391	381	
2001	2021-12-30	12	30	Thursday	2021	3	5854	4571	116277	142352	381	
2002	2021-12-31	12	31	Friday	2021	3	5476	4283	114926	140372	382	

2003 rows × 15 columns

```
    # Mapping a new dictionary, "Data":
        Date = yyyy-mm-dd
        Platform = Apple Music(=1), SoundCloud(=2), Spotify(=3)
        DateWk = Day of the week, ie., Monday(=1), Tuesday(=2), and so
forth.
        Month = [mm] Jan.(=1), Feb.(=2), Mar(=3), and so forth.
        Year = [yyyy], 2020 - 2021
        S = Streams
        L = Listeners
        TWF = Twitter Followers
        TWR = Twitter Retweets
        EGR = Engagement Ranking
        CPP = Cross-Platform Performance Ranking
        IGR = Instagram Story 'Reach'
        DSP_str = Apple Music, ..., Spotify (For Indexing)
```

In [30]:

```
Data_dict = {'Date': Panel['Date'],
             'Platform': Panel['DSPstr'],
             'DateWk': Panel['DateWk'],
             'DayWk': Panel['DayWk'],
             'Month': Panel['Month'],
             'Year': Panel['Year'],
             'S': Panel['Streams'],
             'L': Panel['Listeners'],
```

```
                'TWF': Panel['TWFollowers'],
                'TWR': Panel['TWRetweets'],
                'EGR': Panel['CMEngage'],
                'CPP': Panel['CMCrossP'],
                'IGR': Panel['IGReach'],
                'DSP_str': Panel['DSPstr']}
Data = pd.DataFrame(Data_dict)
Data[['S','L','EGR','IGR','TWR','TWF','CPP']].astype(float)
Data.replace([-np.inf, np.inf], np.nan, inplace=True)
Data['IGR'] = Data['IGR'].bfill().ffill()
Data['S'] = Data['S'].bfill().ffill()
Data['TWR'] = Data['TWR'].bfill().ffill()
Data['L'] = Data['L'].bfill().ffill()
Data['TWF'] = Data['TWF'].bfill().ffill()
with pd.option_context('display.max_rows', None, 'display.max_columns', None):
    print(Data[['S','L','EGR','IGR','TWR','TWF','CPP']])
```

```
            S    L     EGR         IGR  TWR  TWF     CPP
0           7    7  379454  186.250000    0  226  465864
1          58    7  404875  168.500000    0  226  487377
2          59    8  407748  164.666667    0  227  494572
3          57    4  413262  179.000000    0  227  498931
4          19    5  411552  180.666667    0  227  497810
5          38   11  415817  225.250000    0  227  501227
6          40    9  414664  190.000000    0  226  500469
7           7    7  414341  180.000000    0  226  500399
8          10    7  413918  207.666667    0  226  500335
9           8    7  415214  169.000000    0  227  501318
10          7    3  413591    0.000000    0  226  500659
11          7    5  408760  171.500000    0  226  496706
12          6    4  399549    0.000000    0  226  484701
13          5    5  399135  193.285714    0  226  484657
14          8    8  406383  161.666667    0  225  488540
15          5    5  417056  148.333333    0  225  494006
16          2    2  418699  148.600000    0  225  494503
17          5    4  426139  123.000000    0  225  498981
18          1    1  429750    0.000000    0  225  501102
19          2    2  438659    0.000000    0  225  507319
20          5    4  445913  160.222222    0  225  511525
21         28   20  448515  105.500000    0  225  512635
22         45   27  452730  127.000000    0  226  514821
23         17    9  341602    0.000000    0  225  382407
24         13   11  343616    0.000000    0  226  384935
25         15   10  333774  138.000000    0  226  374229
26          4    4  322209  146.000000    0  227  366514
27         16   12  312061  178.200000    0  227  362294
28         13    9  324869  162.800000    0  227  371120
29         27   16  282651  160.600000    0  227  334575
30         18   15  285552  196.571429    0  227  336553
31         17    9  290998  165.909091    2  227  341518
32         20   13  288818  175.500000    3  227  338022
33          6    6  283728  203.666667    3  227  332444
34         10    8  288997  218.666667    3  228  334539
35         10    9  296078  168.750000    3  228  341988
36          0    0  299220  177.833333    3  228  345197
37          6    6  448821  168.333333    3  227  502606
38          7    4  472256  159.333333    0  228  522166
39         12    7  481960  165.000000    0  229  530635
40          6    4  505548  139.750000    0  227  547701
41          4    4  539644    0.000000    0  227  572423
42          3    3  547905  165.800000    0  226  578003
43          4    4  562240  162.000000    0  226  594268
44          4    3  555210  142.666667    0  225  589751
45          8    8  551205    0.000000    0  226  588103
46          3    2  559018  211.750000    0  226  595743
```

47	2	1	554701	164.400000	0	226	592746
48	5	5	555634	174.250000	0	226	594567
49	10	8	538266	170.500000	2	226	581016
50	9	5	546391	139.000000	2	226	588651
51	3	3	533536	198.000000	2	225	580193
52	13	9	541360	0.000000	2	225	584867
53	8	8	549850	194.000000	2	225	591210
54	5	4	551360	205.000000	2	225	591896
55	3	3	544636	234.571429	2	224	586274
56	16	9	568271	275.125000	0	224	606940
57	14	9	563626	250.142857	0	224	600868
58	8	5	555423	212.600000	0	224	595268
59	9	4	564996	201.000000	0	224	602651
60	12	8	559317	208.000000	1	224	595769
61	5	5	561622	187.000000	1	225	597060
62	3	3	556652	0.000000	1	224	594814
63	9	8	553142	245.500000	1	225	592939
64	60	29	553023	0.000000	1	225	590759
65	29	21	544634	0.000000	2	226	587015
66	31	15	536673	0.000000	2	226	571040
67	38	23	550856	0.000000	0	226	581556
68	35	15	557842	236.600000	0	226	581556
69	39	21	537363	173.000000	0	226	564287
70	32	21	538170	0.000000	0	227	562225
71	55	26	540942	240.250000	0	227	560625
72	25	18	524966	201.333333	4	228	546174
73	33	17	516319	214.571429	6	228	540673
74	22	12	511468	0.000000	6	230	533818
75	46	22	515156	198.333333	5	231	535432
76	46	25	513002	145.000000	5	231	534768
77	36	21	520109	201.125000	5	231	538280
78	33	18	519319	189.500000	5	232	536826
79	22	16	538287	255.166667	1	231	549608
80	23	15	529271	219.333333	1	232	542717
81	27	21	514177	224.000000	2	234	533284
82	20	16	520627	256.166667	2	235	537875
83	28	18	525801	216.000000	2	235	540350
84	23	15	526663	272.333333	3	235	542294
85	27	19	522011	212.000000	3	235	537581
86	20	16	523749	218.166667	1	235	539246
87	36	18	518884	197.250000	5	242	535509
88	15	14	520384	175.666667	5	242	536111
89	37	23	514266	260.000000	7	243	533633
90	51	30	503459	206.200000	7	243	523928
91	40	29	304795	235.000000	7	244	332081
92	29	20	296312	250.500000	7	243	321938
93	23	13	302868	262.400000	6	243	330055
94	38	19	303630	245.800000	1	243	331511
95	19	12	303743	238.400000	2	242	331555
96	34	18	296275	0.000000	1	243	324253
97	42	17	298039	0.000000	1	243	325322
98	39	21	296472	0.000000	2	243	324116
99	37	19	298975	0.000000	2	243	326462
100	72	23	301194	0.000000	2	243	330363
101	29	18	291989	209.333333	2	243	319283
102	57	19	294750	223.818182	2	243	321391
103	48	24	298780	170.000000	2	242	324731
104	35	21	302653	342.000000	2	242	328982
105	18	16	300100	322.500000	2	242	327346
106	22	10	298970	253.200000	2	243	326690
107	586	155	291076	232.777778	11	246	318276
108	758	151	296041	243.166667	14	249	322192
109	507	150	294639	247.352941	14	251	320722
110	564	162	293293	245.333333	15	252	319200
111	429	159	289527	241.000000	15	252	313790
112	376	145	289245	252.285714	15	253	313114

113	414	126	287850	304.400000	15	255	311772
114	252	111	288704	244.333333	1	257	312846
115	249	109	289748	238.666667	1	256	313547
116	177	91	289305	224.000000	1	258	313285
117	202	91	286865	265.400000	0	259	310034
118	238	102	286719	0.000000	0	259	309043
119	232	110	319810	273.666667	0	260	346724
120	213	105	346051	268.500000	0	260	372202
121	197	101	346047	246.500000	1	261	371050
122	235	103	343872	236.666667	1	262	369772
123	156	81	344405	281.000000	1	263	369834
124	160	76	343862	265.250000	1	263	369326
125	196	84	346059	231.333333	1	263	370906
126	157	81	349684	284.000000	2	267	375800
127	138	78	360898	0.000000	2	268	382880
128	144	73	386061	268.500000	5	268	391894
129	150	69	391598	385.333333	5	268	389624
130	113	63	394795	157.000000	5	268	389928
131	101	58	405264	0.000000	5	269	394679
132	115	64	414517	298.500000	5	269	410553
133	121	74	417999	328.800000	0	269	414197
134	136	72	422942	321.888889	0	269	428815
135	103	68	425865	283.833333	0	269	443553
136	95	60	436288	0.000000	1	271	450759
137	134	64	443198	341.000000	1	271	425344
138	103	69	403181	0.000000	1	271	414166
139	113	70	327410	0.000000	1	272	350773
140	96	62	317937	0.000000	1	272	346522
141	106	65	395475	0.000000	1	272	415089
142	119	68	400203	297.750000	1	271	419045
143	98	60	405490	237.000000	0	271	423495
144	103	64	405219	308.250000	0	271	424297
145	112	69	405969	0.000000	0	271	424953
146	90	57	406543	0.000000	0	271	425964
147	147	60	413620	257.000000	0	271	429003
148	126	60	418538	0.000000	0	271	424274
149	105	61	426097	0.000000	1	271	429792
150	146	61	410436	311.333333	1	271	426634
151	92	52	412520	261.000000	3	271	425788
152	100	63	410440	0.000000	3	272	425318
153	89	65	428508	283.000000	3	272	440638
154	136	57	429377	304.000000	3	272	440474
155	99	60	428296	368.333333	3	272	443558
156	127	65	428448	221.000000	2	273	443859
157	90	61	429076	0.000000	2	273	443646
158	75	58	468058	280.000000	0	273	470433
159	89	54	460322	0.000000	0	274	468464
160	120	59	483726	297.250000	0	274	482386
161	73	54	483667	267.000000	0	276	482486
162	108	59	483748	269.000000	0	277	482538
163	103	62	482615	326.000000	0	278	481650
164	72	53	450708	316.000000	0	278	446642
165	84	43	492936	359.666667	1	280	491073
166	103	60	489948	276.000000	1	279	489249
167	68	50	434017	271.000000	2	279	435777
168	88	57	391444	241.250000	2	280	402606
169	112	68	359109	0.000000	2	279	379892
170	69	44	341031	243.800000	4	279	366392
171	92	49	341960	209.000000	12	279	367293
172	88	55	342191	0.000000	12	279	367981
173	100	45	320266	246.000000	12	279	348232
174	92	55	322116	266.800000	12	279	349562
175	119	59	298233	0.000000	12	279	329730
176	108	59	307554	283.000000	4	279	336751
177	125	56	306545	274.000000	4	279	335365
178	60	45	302433	0.000000	4	279	332512

179	68	45	318140	277.000000	4	279	346529
180	87	43	344569	0.000000	19	279	370574
181	70	49	346617	326.500000	19	280	371511
182	96	54	363454	297.750000	19	280	381413
183	61	40	351333	239.000000	19	280	377496
184	192	91	300267	209.666667	19	281	325742
185	192	67	246090	304.000000	19	281	277955
186	109	58	210302	299.000000	11	282	243481
187	144	67	176641	406.500000	11	282	208842
188	131	69	177477	332.222222	11	283	210718
189	126	61	177484	317.125000	11	284	210535
190	126	75	153947	295.500000	11	284	184902
191	163	76	146085	401.000000	11	285	174231
192	101	55	144346	362.000000	11	286	172816
193	121	59	146917	366.333333	11	286	175702
194	124	57	138904	274.000000	11	287	167285
195	119	64	134556	282.000000	11	288	161989
196	84	55	130237	295.600000	11	289	157344
197	80	51	131819	289.400000	11	289	159573
198	100	59	126596	247.333333	11	290	153081
199	153	56	135085	322.000000	11	291	161660
200	113	47	104512	335.800000	11	291	127138
201	407	153	105466	340.800000	11	292	130204
202	1586	382	109977	321.250000	11	293	134714
203	1797	448	116828	349.571429	11	293	141243
204	1582	428	217453	342.714286	11	294	219782
205	1493	434	83294	392.000000	12	295	84028
206	1305	393	168118	327.875000	12	300	169789
207	1219	347	156445	334.333333	14	301	158675
208	1165	384	153901	608.000000	14	300	156071
209	1093	361	81179	335.600000	14	299	82756
210	1063	351	80903	288.666667	6	298	82616
211	797	317	72195	355.250000	6	298	75459
212	746	305	72189	317.500000	6	300	75488
213	546	268	67869	512.500000	5	301	71181
214	536	264	67557	476.777778	2	301	70805
215	541	252	67827	421.000000	3	300	71125
216	481	247	73344	365.000000	3	302	76556
217	428	242	72847	0.000000	7	300	76133
218	454	249	78202	402.500000	7	302	80852
219	476	238	80717	0.000000	7	303	82991
220	440	225	83704	393.333333	7	304	84969
221	381	220	83581	356.000000	10	304	84899
222	409	221	100334	355.000000	9	304	101265
223	426	225	83775	0.000000	9	305	85045
224	406	210	84469	451.000000	9	305	85410
225	444	222	84318	346.333333	8	306	85348
226	513	226	84332	0.000000	8	306	85392
227	389	229	84334	370.000000	8	306	85405
228	333	203	84448	0.000000	4	306	85519
229	312	185	84485	359.000000	5	305	85603
230	383	205	84623	0.000000	2	305	85684
231	405	218	84594	300.000000	2	307	85662
232	408	219	84559	0.000000	2	309	85688
233	451	221	84647	338.666667	2	310	85764
234	345	200	84562	451.500000	2	310	85692
235	302	180	84472	361.000000	2	311	85676
236	321	188	84721	317.000000	2	312	85838
237	378	219	84769	301.000000	2	312	85867
238	369	222	83840	378.000000	2	312	85403
239	464	234	84668	295.666667	2	313	85884
240	435	226	84659	305.600000	5	314	85885
241	353	198	84581	243.000000	6	316	85830
242	292	173	84597	389.000000	6	316	85852
243	423	228	81765	0.000000	4	315	83779
244	429	213	82769	593.000000	7	318	84434

245	414	232	82132	348.750000	8	318	84115
246	434	239	86016	372.000000	11	318	86785
247	406	215	83756	369.000000	9	317	85547
248	385	182	80396	389.000000	13	318	82960
249	284	172	79956	351.000000	13	319	82590
250	361	187	79071	0.000000	13	318	81831
251	319	158	80134	429.500000	9	321	82787
252	355	175	79797	418.000000	6	321	82410
253	356	168	74877	270.600000	6	321	90182
254	378	184	79293	0.000000	0	322	93858
255	299	184	68613	412.000000	0	322	82709
256	241	136	64688	430.000000	0	322	79075
257	327	159	69767	0.000000	0	321	83645
258	352	164	74374	492.000000	5	321	88270
259	239	158	72278	360.000000	8	321	85491
260	269	163	67766	278.000000	9	321	81554
261	354	180	72594	300.000000	9	321	86682
262	278	169	77077	0.000000	9	321	90728
263	275	175	78359	0.000000	9	323	91986
264	290	168	77416	0.000000	9	324	91402
265	337	186	79071	0.000000	3	324	92734
266	438	193	77361	0.000000	0	324	91465
267	475	189	75410	0.000000	0	325	90048
268	372	168	45644	0.000000	0	326	57869
269	437	179	51472	0.000000	3	326	63678
270	520	194	48379	0.000000	3	326	60840
271	333	181	73087	441.000000	3	326	86798
272	306	182	71852	439.000000	3	326	85545
273	356	182	74298	395.000000	3	325	87662
274	358	205	74700	400.333333	5	327	88098
275	317	175	80506	0.000000	5	327	93917
276	386	167	56367	347.333333	2	327	68739
277	339	165	53648	0.000000	4	327	66156
278	282	158	54734	323.666667	4	327	67437
279	274	171	474438	402.500000	4	327	474638
280	314	165	86078	387.500000	4	326	99358
281	323	178	85292	365.000000	4	327	98254
282	269	171	79047	620.000000	4	327	92475
283	263	170	81394	566.500000	4	329	94672
284	275	175	74827	486.000000	2	329	88902
285	265	169	146415	0.000000	5	329	155077
286	306	179	489974	565.000000	11	327	490071
287	227	128	390456	419.500000	14	327	390707
288	302	167	390393	359.000000	12	328	390729
289	267	163	454868	0.000000	12	328	455351
290	361	175	455749	0.000000	12	327	455789
291	255	148	443108	0.000000	12	326	444004
292	302	159	444982	471.000000	9	327	445170
293	317	150	442715	452.666667	1	328	443624
294	294	151	436807	0.000000	0	328	437346
295	375	160	410615	0.000000	0	329	407557
296	303	130	406252	0.000000	1	329	406498
297	344	163	137501	595.000000	1	329	138523
298	397	184	405398	561.000000	1	328	405896
299	339	147	405649	341.000000	1	328	406237
300	355	150	391967	0.000000	3	328	392593
301	298	155	392317	432.000000	3	328	392979
302	385	158	389303	0.000000	2	332	389671
303	335	155	131185	336.500000	2	332	128200
304	230	141	87725	0.000000	2	332	88743
305	271	142	93311	465.000000	2	333	94242
306	298	161	92886	0.000000	2	333	93893
307	307	156	93130	406.000000	0	334	94087
308	284	165	106199	389.000000	0	335	106344
309	286	150	105800	0.000000	0	335	105617
310	308	139	104196	0.000000	0	335	105128

311	299	144	90598	0.000000	0	335	91376
312	257	144	90853	449.500000	0	335	91589
313	305	157	90654	508.000000	0	337	91402
314	311	153	376423	384.000000	0	338	377020
315	295	168	379606	0.000000	0	337	379854
316	278	167	379950	418.000000	0	336	380270
317	257	144	376015	424.500000	0	337	377551
318	238	136	388479	0.000000	0	337	389748
319	247	137	386878	425.000000	0	337	388137
320	192	118	391298	0.000000	0	337	391892
321	223	155	390244	0.000000	0	337	390988
322	253	144	390109	424.000000	0	336	390812
323	257	156	394280	569.000000	0	336	394523
324	317	184	394347	0.000000	0	336	394601
325	239	148	394451	0.000000	0	335	394690
326	257	132	394532	0.000000	0	334	394769
327	286	152	394606	0.000000	0	334	394857
328	309	142	394714	0.000000	0	334	394974
329	280	154	394813	0.000000	0	334	395060
330	291	179	394777	532.333333	0	334	395056
331	316	151	394907	0.000000	0	336	395185
332	283	157	394836	653.500000	0	337	395125
333	266	142	394976	0.000000	0	337	395255
334	290	148	392975	0.000000	0	337	393730
335	287	166	392828	490.333333	0	337	393648
336	255	168	394892	449.000000	0	337	395344
337	251	157	371312	400.000000	0	338	373179
338	249	158	369610	298.000000	0	338	370202
339	230	152	367034	0.000000	1	338	369376
340	210	140	366199	467.500000	1	338	368415
341	198	135	365574	0.000000	1	338	367673
342	225	145	365041	0.000000	1	338	366754
343	225	143	374382	0.000000	1	340	376394
344	229	141	139948	0.000000	1	340	159312
345	249	151	135604	0.000000	1	340	160070
346	246	135	479620	418.000000	0	340	480199
347	209	139	478705	511.000000	0	340	478590
348	167	119	480301	0.000000	3	339	480841
349	176	126	479880	0.000000	3	340	480418
350	201	134	479150	489.666667	3	341	479552
351	194	129	255638	341.000000	3	340	255776
352	199	130	255713	0.000000	3	341	255856
353	194	122	255800	0.000000	3	341	255954
354	186	112	255918	0.000000	3	340	256059
355	250	153	255994	515.000000	0	340	256124
356	192	133	120997	485.000000	0	340	128588
357	219	153	256177	407.000000	1	340	256308
358	262	144	256178	510.500000	1	340	256307
359	253	137	256213	256.000000	1	339	256350
360	188	131	256032	484.000000	1	340	256172
361	180	127	256330	0.000000	1	338	256463
362	209	126	256400	531.333333	1	338	256541
363	205	139	256523	425.000000	0	339	256674
364	157	115	256757	423.000000	4	341	256907
365	175	109	256768	391.500000	6	340	256868
366	151	94	256507	398.000000	6	341	256612
367	163	120	256747	0.000000	6	341	256848
368	137	100	256915	0.000000	6	343	256974
369	202	130	256804	518.000000	11	342	256953
370	202	104	251280	471.333333	11	342	251477
371	182	113	248548	371.000000	5	344	248742
372	156	107	125536	0.000000	5	345	126702
373	217	121	211712	0.000000	5	344	212943
374	170	104	192846	0.000000	5	344	194268
375	183	96	191590	367.000000	5	344	192593
376	135	92	207119	620.000000	0	345	208547

377	171	116	204629	0.000000	0	344	206141
378	172	104	197569	542.000000	0	343	198671
379	150	100	203459	613.000000	1	343	204867
380	214	125	126430	386.500000	4	345	127226
381	178	108	273698	297.500000	4	345	274046
382	211	111	273031	0.000000	4	345	273451
383	173	117	278761	501.000000	4	346	279410
384	145	108	156151	567.000000	4	346	166819
385	164	104	110515	565.333333	4	346	111887
386	178	107	79881	461.000000	3	346	81606
387	183	112	80435	0.000000	1	346	81771
388	216	116	79983	0.000000	3	348	81414
389	384	309	271420	0.000000	3	348	272709
390	274	166	270655	0.000000	3	348	272020
391	214	115	275886	337.500000	3	348	276341
392	174	109	83187	0.000000	3	348	84374
393	165	119	272007	0.000000	3	348	273396
394	152	103	269272	0.000000	3	348	270475
395	167	111	171768	705.500000	1	348	146495
396	178	114	135110	684.000000	2	347	136226
397	176	107	132527	440.500000	2	345	133685
398	139	102	104630	381.500000	10	342	105796
399	201	115	115840	411.333333	12	342	116954
400	194	110	105027	376.600000	13	342	105770
401	157	110	133189	345.500000	12	341	134800
402	177	107	132302	0.000000	12	341	133853
403	151	95	130351	544.000000	14	341	131985
404	149	109	150495	389.500000	14	344	162668
405	164	120	152541	333.000000	9	344	161286
406	210	107	154716	573.500000	8	344	169757
407	126	91	93046	484.666667	7	344	93927
408	344	149	81542	414.375000	10	345	82472
409	324	156	148644	321.000000	13	346	164471
410	214	106	143247	418.000000	10	346	160029
411	452	165	144610	453.500000	10	345	160399
412	433	150	136486	393.000000	7	348	147009
413	429	152	143148	592.000000	7	349	158677
414	317	138	140789	511.000000	7	350	156182
415	411	142	136205	0.000000	4	350	152013
416	311	131	137048	0.000000	0	351	153117
417	329	124	139537	0.000000	0	351	155707
418	381	137	138363	478.500000	0	351	155280
419	353	123	74974	0.000000	0	351	86951
420	360	134	133509	0.000000	0	351	148445
421	284	129	131246	678.500000	0	352	149303
422	230	123	137594	545.500000	0	352	149248
423	277	118	128995	478.666667	0	354	144237
424	201	111	131142	395.000000	0	354	146172
425	259	139	127286	407.400000	0	355	142567
426	327	137	131724	386.000000	0	357	146274
427	393	132	129964	0.000000	0	359	142071
428	248	143	126700	392.666667	0	359	139067
429	225	143	128650	0.000000	1	359	141048
430	254	106	123625	284.000000	1	359	136816
431	203	109	126834	484.000000	1	358	139761
432	233	160	128868	448.000000	1	357	141862
433	407	307	125037	371.000000	1	357	141271
434	312	198	121657	349.000000	1	357	140935
435	278	131	210626	0.000000	1	357	200335
436	277	144	207869	490.000000	0	358	197464
437	214	109	215188	356.000000	0	357	212084
438	199	121	216360	363.200000	2	358	211717
439	234	127	214338	0.000000	3	356	204617
440	197	104	213310	526.000000	3	356	213465
441	227	132	218547	341.600000	5	356	202373
442	211	135	223983	325.000000	10	354	206637

443	185	124	225413	0.000000	10	355	205982
444	167	108	227835	679.000000	10	355	236828
445	144	95	247239	405.500000	8	355	244274
446	196	115	249043	372.500000	8	356	239129
447	214	127	252074	0.000000	7	357	256071
448	185	117	202575	0.000000	5	356	216693
449	180	116	200993	0.000000	1	357	215754
450	174	96	204089	409.500000	1	357	218364
451	172	99	203267	0.000000	1	358	213418
452	136	93	202685	374.000000	0	357	213428
453	183	102	214882	0.000000	0	356	223061
454	188	108	213361	0.000000	0	356	221820
455	157	106	212867	338.000000	0	356	221742
456	154	101	208083	0.000000	0	356	217384
457	194	110	205480	0.000000	0	356	215695
458	179	101	204942	0.000000	0	356	215519
459	114	78	206213	723.000000	0	356	216698
460	158	91	206925	0.000000	0	355	217314
461	181	105	202998	456.000000	0	355	210101
462	134	95	201336	307.000000	0	355	208523
463	174	106	202520	0.000000	0	354	213401
464	127	80	206397	0.000000	0	353	217183
465	144	98	209917	0.000000	0	354	221719
466	187	92	206312	0.000000	0	354	218000
467	177	104	205545	586.000000	0	354	213308
468	148	87	200304	620.000000	0	354	208799
469	154	113	196062	567.000000	0	354	205583
470	155	105	193449	656.000000	0	355	206939
471	164	93	198829	470.000000	0	355	208047
472	128	86	199593	0.000000	0	355	208852
473	137	92	196498	0.000000	0	355	206458
474	120	79	192456	568.000000	0	355	203043
475	132	86	196850	583.000000	0	355	206785
476	143	106	196593	455.000000	0	355	206641
477	138	91	106171	531.250000	0	359	119038
478	179	96	196080	353.600000	0	359	205839
479	128	94	198032	571.500000	1	359	211118
480	90	81	191821	0.000000	1	359	205129
481	128	87	191425	560.500000	1	359	201896
482	104	83	191678	415.750000	1	359	205514
483	123	84	191748	424.000000	8	359	202477
484	145	101	193442	462.500000	8	359	203946
485	165	110	195785	0.000000	8	359	205524
486	169	94	198361	428.000000	7	360	211446
487	105	71	197539	522.000000	7	360	214899
488	116	90	194216	473.500000	7	361	214339
489	121	80	193181	0.000000	7	361	213470
490	203	105	193933	524.000000	0	361	209600
491	130	80	195556	0.000000	0	362	211529
492	128	84	186517	0.000000	0	363	206082
493	82	72	179335	0.000000	0	363	200023
494	95	73	184140	599.500000	0	363	204068
495	112	82	183054	389.000000	0	363	203282
496	120	94	198804	0.000000	3	364	218310
497	147	77	194315	531.500000	3	365	214971
498	145	108	198659	512.666667	5	365	218717
499	154	88	201418	386.166667	5	365	220296
500	136	96	205633	0.000000	5	366	224171
501	170	88	208727	578.666667	5	366	226775
502	152	76	208636	403.000000	5	367	226672
503	163	87	196470	0.000000	2	367	216215
504	245	154	203680	0.000000	2	366	222188
505	274	143	197581	0.000000	0	367	217078
506	248	108	196171	0.000000	0	367	215880
507	154	92	195006	0.000000	0	368	213605
508	186	85	197290	0.000000	0	367	215599

509	226	104	197707	0.000000	0	367	216407
510	149	101	194372	0.000000	0	367	214011
511	177	119	193151	0.000000	0	366	213490
512	189	102	207535	505.000000	0	366	226366
513	168	99	208233	416.500000	0	366	226834
514	201	87	199393	0.000000	1	367	219295
515	130	76	202501	0.000000	1	368	222335
516	178	101	198891	0.000000	1	367	219410
517	146	99	196651	0.000000	1	367	217766
518	124	79	193603	0.000000	1	367	215172
519	125	94	191422	0.000000	6	367	213120
520	147	95	178585	0.000000	6	367	200714
521	122	94	176328	0.000000	5	367	198909
522	106	78	173750	0.000000	5	367	196521
523	129	88	182049	315.750000	5	367	203594
524	140	93	180046	0.000000	5	367	201658
525	161	85	178864	487.333333	5	367	200640
526	174	102	179979	387.500000	0	366	201731
527	142	83	179326	0.000000	0	366	201045
528	112	85	180194	0.000000	0	367	201582
529	109	85	188933	0.000000	0	367	211050
530	104	72	186785	0.000000	0	367	209433
531	128	87	184783	418.000000	0	367	208131
532	159	103	182659	0.000000	1	367	207083
533	125	96	185255	0.000000	1	367	209219
534	126	92	188019	0.000000	1	368	211647
535	93	77	189455	0.000000	1	368	212742
536	102	79	188008	0.000000	1	367	211463
537	93	74	185769	0.000000	1	367	209366
538	105	77	182183	519.000000	0	367	206069
539	135	109	184224	0.000000	0	367	207421
540	104	72	188417	379.000000	0	367	210772
541	130	102	167429	767.000000	0	367	191301
542	170	95	167371	0.000000	0	366	191400
543	125	86	171055	0.000000	0	366	194453
544	143	82	170906	0.000000	0	366	194640
545	132	86	170680	0.000000	0	366	194550
546	123	90	183220	585.000000	0	366	206789
547	127	90	185749	0.000000	0	366	209485
548	129	86	195792	333.666667	0	366	218675
549	100	73	193640	0.000000	0	366	217194
550	129	87	190217	0.000000	0	365	214396
551	91	75	188106	0.000000	0	365	212330
552	107	75	192868	0.000000	0	365	216490
553	112	91	191441	0.000000	0	366	214965
554	150	96	188660	0.000000	0	366	212279
555	148	106	186542	0.000000	0	366	211061
556	116	71	192326	0.000000	0	366	215632
557	105	78	176393	0.000000	0	366	200355
558	128	88	178235	564.000000	0	366	201898
559	121	95	180723	0.000000	0	366	203865
560	214	113	182655	0.000000	1	365	205795
561	138	92	176690	0.000000	1	365	200426
562	143	103	172898	0.000000	2	366	196684
563	193	107	176476	0.000000	2	366	199554
564	148	74	195542	0.000000	2	367	217194
565	195	107	184383	628.000000	2	366	208283
566	190	103	185287	0.000000	0	368	209004
567	166	99	186172	0.000000	1	368	209553
568	151	107	179959	0.000000	2	368	203856
569	161	106	184686	0.000000	4	369	207904
570	135	106	176942	0.000000	4	369	201215
571	146	85	179493	0.000000	4	368	203810
572	169	103	179679	0.000000	4	368	203713
573	156	99	179606	0.000000	4	370	203351
574	148	107	183315	0.000000	4	370	206703

575	152	110	163769	0.000000	0	370	188628
576	167	103	163972	0.000000	0	370	188083
577	190	108	163364	0.000000	0	370	186669
578	106	69	165714	0.000000	0	370	188744
579	134	100	164531	0.000000	0	370	187582
580	176	112	70581	0.000000	0	370	88448
581	192	118	68621	496.000000	0	370	83620
582	180	105	93105	0.000000	0	370	109935
583	217	116	95780	524.000000	0	370	112442
584	115	83	84317	0.000000	0	370	102075
585	153	103	85672	0.000000	0	370	102921
586	179	116	178674	0.000000	0	370	200873
587	161	93	175569	0.000000	0	369	198229
588	160	114	174595	0.000000	0	368	197171
589	118	93	177983	0.000000	0	367	199986
590	162	115	172809	0.000000	0	368	195333
591	99	76	170240	0.000000	0	368	192329
592	107	75	156524	0.000000	0	368	178840
593	122	95	158883	0.000000	0	367	180989
594	178	116	160917	0.000000	0	367	182756
595	136	106	161596	0.000000	0	367	183014
596	162	99	155106	0.000000	0	367	178234
597	159	105	159061	0.000000	0	367	181421
598	112	83	154096	0.000000	0	368	177380
599	109	86	154098	0.000000	0	368	177339
600	124	80	169973	0.000000	0	367	193002
601	126	96	90094	0.000000	0	366	106612
602	127	92	91363	0.000000	0	366	108366
603	147	96	87277	0.000000	0	367	104554
604	161	93	86850	243.500000	0	367	103855
605	113	84	76326	226.000000	0	367	93772
606	116	84	77747	0.000000	0	367	95009
607	122	89	80237	0.000000	0	367	97949
608	141	100	160527	0.000000	0	368	180884
609	132	84	154773	524.000000	0	368	175905
610	162	99	142524	578.000000	0	366	164592
611	106	76	152527	327.000000	0	366	173068
612	107	73	152973	426.000000	0	366	173101
613	117	72	157836	0.000000	0	366	177303
614	147	103	153075	0.000000	0	366	177454
615	145	96	153240	0.000000	0	366	173565
616	123	84	159900	0.000000	0	366	182754
617	124	76	157000	338.000000	0	367	180826
618	128	97	157231	0.000000	0	367	179706
619	98	69	149995	0.000000	0	367	174156
620	67	52	165640	0.000000	0	367	190712
621	131	82	167088	0.000000	0	367	192152
622	158	108	160550	0.000000	0	367	187333
623	111	83	158182	0.000000	0	366	190443
624	143	106	161890	286.500000	0	366	193309
625	130	100	164682	0.000000	0	366	196097
626	117	79	164845	301.666667	0	367	195957
627	127	89	166158	231.500000	0	367	197279
628	115	80	147652	0.000000	0	367	177915
629	124	92	148909	0.000000	0	367	179348
630	136	84	146850	0.000000	0	367	177138
631	84	57	150009	0.000000	0	367	178997
632	95	65	144635	0.000000	0	367	174217
633	116	72	145801	0.000000	0	367	174965
634	83	66	144494	0.000000	0	367	173293
635	116	87	146153	0.000000	0	367	175380
636	102	85	145220	0.000000	0	367	174476
637	109	75	144031	0.000000	3	367	173381
638	137	93	147408	445.666667	0	368	175850
639	157	96	148046	0.000000	3	368	176578
640	92	70	147921	0.000000	3	368	176678

641	114	84	163246	0.000000	3	367	192698
642	127	83	164039	0.000000	3	367	193488
643	110	89	164348	376.250000	13	368	193759
644	94	76	161081	0.000000	11	370	190975
645	95	78	160728	465.000000	12	370	190149
646	199	114	145553	402.000000	16	371	174517
647	117	76	140862	0.000000	16	372	170358
648	112	71	157238	612.000000	16	372	187196
649	249	126	137573	515.000000	15	373	167402
650	185	90	135038	0.000000	5	373	164676
651	123	95	137122	0.000000	5	373	166167
652	108	77	138403	0.000000	0	374	166467
653	133	85	141684	412.000000	0	374	168679
654	222	103	140256	367.666667	0	375	167178
655	153	106	136841	0.000000	0	375	163913
656	186	107	137575	0.000000	0	375	164482
657	187	95	135642	0.000000	0	376	162632
658	196	129	129767	360.000000	0	377	157028
659	208	112	141365	0.000000	0	378	170130
660	195	117	139969	0.000000	0	379	168785
661	154	96	135621	0.000000	0	380	164893
662	253	175	132797	0.000000	0	381	161959
663	349	203	131006	291.500000	2	381	160044
664	306	163	128797	0.000000	2	381	157601
665	396	148	117195	0.000000	2	381	143391
666	313	153	116277	0.000000	2	381	142352
667	312	151	114926	559.000000	2	382	140372
668	51	39	379454	186.250000	0	226	465864
669	85	69	404875	168.500000	0	226	487377
670	116	38	407748	164.666667	0	227	494572
671	39	21	413262	179.000000	0	227	498931
672	19	54	411552	180.666667	0	227	497810
673	49	28	415817	225.250000	0	227	501227
674	35	33	414664	190.000000	0	226	500469
675	50	26	414341	180.000000	0	226	500399
676	36	12	413918	207.666667	0	226	500335
677	31	35	415214	169.000000	0	227	501318
678	17	201	413591	0.000000	0	226	500659
679	20	175	408760	171.500000	0	226	496706
680	36	30	399549	0.000000	0	226	484701
681	62	15	399135	193.285714	0	226	484657
682	138	18	406383	161.666667	0	225	488540
683	240	20	417056	148.333333	0	225	494006
684	20	41	418699	148.600000	0	225	494503
685	27	14	426139	123.000000	0	225	498981
686	13	16	429750	0.000000	0	225	501102
687	17	10	438659	0.000000	0	225	507319
688	24	60	445913	160.222222	0	225	511525
689	46	55	448515	105.500000	0	225	512635
690	33	98	452730	127.000000	0	226	514821
691	24	68	341602	0.000000	0	225	382407
692	12	33	343616	0.000000	0	226	384935
693	19	39	333774	138.000000	0	226	374229
694	23	37	322209	146.000000	0	227	366514
695	39	28	312061	178.200000	0	227	362294
696	35	29	324869	162.800000	0	227	371120
697	19	38	282651	160.600000	0	227	334575
698	53	25	285552	196.571429	0	227	336553
699	191	63	290998	165.909091	2	227	341518
700	226	55	288818	175.500000	3	227	338022
701	153	47	283728	203.666667	3	227	332444
702	249	27	288997	218.666667	3	228	334539
703	202	27	296078	168.750000	3	228	341988
704	94	27	299220	177.833333	3	228	345197
705	81	26	448821	168.333333	3	227	502606
706	165	23	472256	159.333333	0	228	522166

707	106	25	481960	165.000000	0	229	530635
708	108	18	505548	139.750000	0	227	547701
709	164	20	539644	0.000000	0	227	572423
710	144	18	547905	165.800000	0	226	578003
711	162	27	562240	162.000000	0	226	594268
712	163	34	555210	142.666667	0	225	589751
713	100	31	551205	0.000000	0	226	588103
714	82	42	559018	211.750000	0	226	595743
715	97	49	554701	164.400000	0	226	592746
716	108	36	555634	174.250000	0	226	594567
717	175	23	538266	170.500000	2	226	581016
718	107	9	546391	139.000000	2	226	588651
719	158	17	533536	198.000000	2	225	580193
720	94	48	541360	0.000000	2	225	584867
721	120	39	549850	194.000000	2	225	591210
722	134	31	551360	205.000000	2	225	591896
723	160	43	544636	234.571429	2	224	586274
724	159	15	568271	275.125000	0	224	606940
725	126	22	563626	250.142857	0	224	600868
726	91	25	555423	212.600000	0	224	595268
727	160	11	564996	201.000000	0	224	602651
728	117	15	559317	208.000000	1	224	595769
729	129	58	561622	187.000000	1	225	597060
730	147	76	556652	0.000000	1	224	594814
731	145	30	553142	245.500000	1	225	592939
732	4385	73	553023	0.000000	1	225	590759
733	2682	51	544634	0.000000	2	226	587015
734	2470	47	536673	0.000000	2	226	571040
735	2555	55	550856	0.000000	0	226	581556
736	2271	52	557842	236.600000	0	226	581556
737	2218	60	537363	173.000000	0	226	564287
738	2202	59	538170	0.000000	0	227	562225
739	2050	52	540942	240.250000	0	227	560625
740	2262	76	524966	201.333333	4	228	546174
741	1623	67	516319	214.571429	6	228	540673
742	1821	59	511468	0.000000	6	230	533818
743	1570	85	515156	198.333333	5	231	535432
744	1709	88	513002	145.000000	5	231	534768
745	985	91	520109	201.125000	5	231	538280
746	926	95	519319	189.500000	5	232	536826
747	956	109	538287	255.166667	1	231	549608
748	717	81	529271	219.333333	1	232	542717
749	748	71	514177	224.000000	2	234	533284
750	876	51	520627	256.166667	2	235	537875
751	938	82	525801	216.000000	2	235	540350
752	879	69	526663	272.333333	3	235	542294
753	699	86	522011	212.000000	3	235	537581
754	686	133	523749	218.166667	1	235	539246
755	694	98	518884	197.250000	5	242	535509
756	568	83	520384	175.666667	5	242	536111
757	568	56	514266	260.000000	7	243	533633
758	578	115	503459	206.200000	7	243	523928
759	570	112	304795	235.000000	7	244	332081
760	585	219	296312	250.500000	7	243	321938
761	595	312	302868	262.400000	6	243	330055
762	733	306	303630	245.800000	1	243	331511
763	586	279	303743	238.400000	2	242	331555
764	571	272	296275	0.000000	1	243	324253
765	686	308	298039	0.000000	1	243	325322
766	880	335	296472	0.000000	2	243	324116
767	715	239	298975	0.000000	2	243	326462
768	939	205	301194	0.000000	2	243	330363
769	694	109	291989	209.333333	2	243	319283
770	699	202	294750	223.818182	2	243	321391
771	791	385	298780	170.000000	2	242	324731
772	660	296	302653	342.000000	2	242	328982

773	708	175	300100	322.500000	2	242	327346
774	872	185	298970	253.200000	2	243	326690
775	1413	571	291076	232.777778	11	246	318276
776	1409	536	296041	243.166667	14	249	322192
777	1382	562	294639	247.352941	14	251	320722
778	1046	570	293293	245.333333	15	252	319200
779	1254	562	289527	241.000000	15	252	313790
780	1256	518	289245	252.285714	15	253	313114
781	1044	528	287850	304.400000	15	255	311772
782	1027	535	288704	244.333333	1	257	312846
783	948	431	289748	238.666667	1	256	313547
784	1087	409	289305	224.000000	1	258	313285
785	705	460	286865	265.400000	0	259	310034
786	704	444	286719	0.000000	0	259	309043
787	698	439	319810	273.666667	0	260	346724
788	653	499	346051	268.500000	0	260	372202
789	714	558	346047	246.500000	1	261	371050
790	768	460	343872	236.666667	1	262	369772
791	726	478	344405	281.000000	1	263	369834
792	750	616	343862	265.250000	1	263	369326
793	707	633	346059	231.333333	1	263	370906
794	668	554	349684	284.000000	2	267	375800
795	600	564	360898	0.000000	2	268	382880
796	664	509	386061	268.500000	5	268	391894
797	600	442	391598	385.333333	5	268	389624
798	663	401	394795	157.000000	5	268	389928
799	683	408	405264	0.000000	5	269	394679
800	930	344	414517	298.500000	5	269	410553
801	1001	325	417999	328.800000	0	269	414197
802	747	277	422942	321.888889	0	269	428815
803	516	283	425865	283.833333	0	269	443553
804	617	247	436288	0.000000	1	271	450759
805	509	259	443198	341.000000	1	271	425344
806	650	246	403181	0.000000	1	271	414166
807	698	266	327410	0.000000	1	272	350773
808	377	284	317937	0.000000	1	272	346522
809	507	251	395475	0.000000	1	272	415089
810	384	245	400203	297.750000	1	271	419045
811	399	222	405490	237.000000	0	271	423495
812	441	243	405219	308.250000	0	271	424297
813	365	237	405969	0.000000	0	271	424953
814	334	261	406543	0.000000	0	271	425964
815	484	275	413620	257.000000	0	271	429003
816	435	272	418538	0.000000	0	271	424274
817	474	266	426097	0.000000	1	271	429792
818	346	228	410436	311.333333	1	271	426634
819	342	224	412520	261.000000	3	271	425788
820	388	293	410440	0.000000	3	272	425318
821	349	283	428508	283.000000	3	272	440638
822	367	256	429377	304.000000	3	272	440474
823	394	251	428296	368.333333	3	272	443558
824	433	252	428448	221.000000	2	273	443859
825	425	206	429076	0.000000	2	273	443646
826	378	235	468058	280.000000	0	273	470433
827	404	184	460322	0.000000	0	274	468464
828	371	191	483726	297.250000	0	274	482386
829	303	201	483667	267.000000	0	276	482486
830	353	196	483748	269.000000	0	277	482538
831	477	199	482615	326.000000	0	278	481650
832	443	239	450708	316.000000	0	278	446642
833	429	247	492936	359.666667	1	280	491073
834	309	1007	489948	276.000000	1	279	489249
835	433	1056	434017	271.000000	2	279	435777
836	380	810	391444	241.250000	2	280	402606
837	361	724	359109	0.000000	2	279	379892
838	455	657	341031	243.800000	4	279	366392

839	444	645	341960	209.000000	12	279	367293
840	321	651	342191	0.000000	12	279	367981
841	384	289	320266	246.000000	12	279	348232
842	342	217	322116	266.800000	12	279	349562
843	352	185	298233	0.000000	12	279	329730
844	346	234	307554	283.000000	4	279	336751
845	278	224	306545	274.000000	4	279	335365
846	294	197	302433	0.000000	4	279	332512
847	299	189	318140	277.000000	4	279	346529
848	308	284	344569	0.000000	19	279	370574
849	307	362	346617	326.500000	19	280	371511
850	321	317	363454	297.750000	19	280	381413
851	344	282	351333	239.000000	19	280	377496
852	509	373	300267	209.666667	19	281	325742
853	357	387	246090	304.000000	19	281	277955
854	354	387	210302	299.000000	11	282	243481
855	299	367	176641	406.500000	11	282	208842
856	349	279	177477	332.222222	11	283	210718
857	298	303	177484	317.125000	11	284	210535
858	264	296	153947	295.500000	11	284	184902
859	340	290	146085	401.000000	11	285	174231
860	301	272	144346	362.000000	11	286	172816
861	270	296	146917	366.333333	11	286	175702
862	307	549	138904	274.000000	11	287	167285
863	331	547	134556	282.000000	11	288	161989
864	371	486	130237	295.600000	11	289	157344
865	367	452	131819	289.400000	11	289	159573
866	357	678	126596	247.333333	11	290	153081
867	347	588	135085	322.000000	11	291	161660
868	240	504	104512	335.800000	11	291	127138
869	669	642	105466	340.800000	11	292	130204
870	2302	1421	109977	321.250000	11	293	134714
871	2241	1483	116828	349.571429	11	293	141243
872	2383	1510	217453	342.714286	11	294	219782
873	2263	1548	83294	392.000000	12	295	84028
874	1796	1522	168118	327.875000	12	300	169789
875	1464	1368	156445	334.333333	14	301	158675
876	1709	1446	153901	608.000000	14	300	156071
877	1656	1355	81179	335.600000	14	299	82756
878	1438	1325	80903	288.666667	6	298	82616
879	1466	1360	72195	355.250000	6	298	75459
880	1444	1366	72189	317.500000	6	300	75488
881	1179	1263	67869	512.500000	5	301	71181
882	1215	1161	67557	476.777778	2	301	70805
883	1134	1143	67827	421.000000	3	300	71125
884	1463	1154	73344	365.000000	3	302	76556
885	1148	1121	72847	0.000000	7	300	76133
886	1293	1099	78202	402.500000	7	302	80852
887	1429	1084	80717	0.000000	7	303	82991
888	1694	1035	83704	393.333333	7	304	84969
889	2116	1026	83581	356.000000	10	304	84899
890	2177	2147	100334	355.000000	9	304	101265
891	2210	2317	83775	0.000000	9	305	85045
892	2164	2064	84469	451.000000	9	305	85410
893	1715	1940	84318	346.333333	8	306	85348
894	2605	1932	84332	0.000000	8	306	85392
895	1740	1877	84334	370.000000	8	306	85405
896	1670	1731	84448	0.000000	4	306	85519
897	2897	1207	84485	359.000000	5	305	85603
898	2810	1081	84623	0.000000	2	305	85684
899	3027	1093	84594	300.000000	2	307	85662
900	1902	1101	84559	0.000000	2	309	85688
901	2241	1139	84647	338.666667	2	310	85764
902	1997	1122	84562	451.500000	2	310	85692
903	2240	1032	84472	361.000000	2	311	85676
904	1635	2544	84721	317.000000	2	312	85838

905	3168	2570	84769	301.000000	2	312	85867
906	4571	2289	83840	378.000000	2	312	85403
907	6247	2235	84668	295.666667	2	313	85884
908	2850	2255	84659	305.600000	5	314	85885
909	3579	2298	84581	243.000000	6	316	85830
910	5709	2028	84597	389.000000	6	316	85852
911	4381	1624	81765	0.000000	4	315	83779
912	4392	1512	82769	593.000000	7	318	84434
913	3780	1547	82132	348.750000	8	318	84115
914	5055	1523	86016	372.000000	11	318	86785
915	5823	1553	83756	369.000000	9	317	85547
916	3207	1373	80396	389.000000	13	318	82960
917	4599	1254	79956	351.000000	13	319	82590
918	6185	1373	79071	0.000000	13	318	81831
919	6306	1350	80134	429.500000	9	321	82787
920	7532	1356	79797	418.000000	6	321	82410
921	7512	1299	74877	270.600000	6	321	90182
922	4774	1289	79293	0.000000	0	322	93858
923	4409	1195	68613	412.000000	0	322	82709
924	4005	1150	64688	430.000000	0	322	79075
925	6340	1226	69767	0.000000	0	321	83645
926	5385	1230	74374	492.000000	5	321	88270
927	4846	1217	72278	360.000000	8	321	85491
928	2869	1150	67766	278.000000	9	321	81554
929	5304	1230	72594	300.000000	9	321	86682
930	4750	1288	77077	0.000000	9	321	90728
931	4502	1246	78359	0.000000	9	323	91986
932	4694	1413	77416	0.000000	9	324	91402
933	4337	1451	79071	0.000000	3	324	92734
934	3901	1425	77361	0.000000	0	324	91465
935	3703	1337	75410	0.000000	0	325	90048
936	3854	1387	45644	0.000000	0	326	57869
937	3719	1436	51472	0.000000	3	326	63678
938	3685	1301	48379	0.000000	3	326	60840
939	4447	1352	73087	441.000000	3	326	86798
940	3124	1263	71852	439.000000	3	326	85545
941	3068	1304	74298	395.000000	3	325	87662
942	3698	1276	74700	400.333333	5	327	88098
943	3302	1349	80506	0.000000	5	327	93917
944	3508	1333	56367	347.333333	2	327	68739
945	3523	1298	53648	0.000000	4	327	66156
946	3757	3294	54734	323.666667	4	327	67437
947	3436	3435	474438	402.500000	4	327	474638
948	4477	2913	86078	387.500000	4	326	99358
949	3588	2652	85292	365.000000	4	327	98254
950	3767	2538	79047	620.000000	4	327	92475
951	3610	2540	81394	566.500000	4	329	94672
952	3387	2464	74827	486.000000	2	329	88902
953	2441	3783	146415	0.000000	5	329	155077
954	2583	3454	489974	565.000000	11	327	490071
955	2624	3169	390456	419.500000	14	327	390707
956	2596	3002	390393	359.000000	12	328	390729
957	2541	2870	454868	0.000000	12	328	455351
958	2385	2772	455749	0.000000	12	327	455789
959	2039	2724	443108	0.000000	12	326	444004
960	2673	1929	444982	471.000000	9	327	445170
961	1894	1611	442715	452.666667	1	328	443624
962	2107	1535	436807	0.000000	0	328	437346
963	2094	1477	410615	0.000000	0	329	407557
964	2502	1307	406252	0.000000	1	329	406498
965	1757	1426	137501	595.000000	1	329	138523
966	1866	1413	405398	561.000000	1	328	405896
967	1926	2175	405649	341.000000	1	328	406237
968	2842	2244	391967	0.000000	3	328	392593
969	2138	2004	392317	432.000000	3	328	392979
970	1466	1884	389303	0.000000	2	332	389671

971	1538	1751	131185	336.500000	2	332	128200
972	1855	1716	87725	0.000000	2	332	88743
973	2451	1795	93311	465.000000	2	333	94242
974	1858	1613	92886	0.000000	2	333	93893
975	1594	1443	93130	406.000000	0	334	94087
976	1933	1493	106199	389.000000	0	335	106344
977	2015	1447	105800	0.000000	0	335	105617
978	2203	1383	104196	0.000000	0	335	105128
979	2242	1404	90598	0.000000	0	335	91376
980	1714	1276	90853	449.500000	0	335	91589
981	1663	1417	90654	508.000000	0	337	91402
982	1830	1283	376423	384.000000	0	338	377020
983	1405	1291	379606	0.000000	0	337	379854
984	1848	1304	379950	418.000000	0	336	380270
985	1916	1347	376015	424.500000	0	337	377551
986	1830	1143	388479	0.000000	0	337	389748
987	1758	1169	386878	425.000000	0	337	388137
988	1016	1197	391298	0.000000	0	337	391892
989	1253	1145	390244	0.000000	0	337	390988
990	1486	1262	390109	424.000000	0	336	390812
991	1454	1285	394280	569.000000	0	336	394523
992	1609	1327	394347	0.000000	0	336	394601
993	1198	1220	394451	0.000000	0	335	394690
994	1063	1124	394532	0.000000	0	334	394769
995	1726	1178	394606	0.000000	0	334	394857
996	1222	1307	394714	0.000000	0	334	394974
997	1403	1272	394813	0.000000	0	334	395060
998	2164	1335	394777	532.333333	0	334	395056
999	1229	1344	394907	0.000000	0	336	395185
1000	1595	1223	394836	653.500000	0	337	395125
1001	1389	1134	394976	0.000000	0	337	395255
1002	1389	1222	392975	0.000000	0	337	393730
1003	1288	1287	392828	490.333333	0	337	393648
1004	1377	1258	394892	449.000000	0	337	395344
1005	1089	1176	371312	400.000000	0	338	373179
1006	1089	1207	369610	298.000000	0	338	370202
1007	1132	1252	367034	0.000000	1	338	369376
1008	1066	1133	366199	467.500000	1	338	368415
1009	1031	1194	365574	0.000000	1	338	367673
1010	1331	1295	365041	0.000000	1	338	366754
1011	957	1254	374382	0.000000	1	340	376394
1012	1021	1243	139948	0.000000	1	340	159312
1013	1113	1247	135604	0.000000	1	340	160070
1014	1349	1266	479620	418.000000	0	340	480199
1015	1043	1168	478705	511.000000	0	340	478590
1016	866	1289	480301	0.000000	3	339	480841
1017	1080	1192	479880	0.000000	3	340	480418
1018	1115	1206	479150	489.666667	3	341	479552
1019	956	1194	255638	341.000000	3	340	255776
1020	802	1200	255713	0.000000	3	341	255856
1021	744	1238	255800	0.000000	3	341	255954
1022	603	1031	255918	0.000000	3	340	256059
1023	961	1384	255994	515.000000	0	340	256124
1024	855	1453	120997	485.000000	0	340	128588
1025	956	1299	256177	407.000000	1	340	256308
1026	1112	1296	256178	510.500000	1	340	256307
1027	1018	1318	256213	256.000000	1	339	256350
1028	776	1252	256032	484.000000	1	340	256172
1029	780	1152	256330	0.000000	1	338	256463
1030	879	1072	256400	531.333333	1	338	256541
1031	859	952	256523	425.000000	0	339	256674
1032	1221	855	256757	423.000000	4	341	256907
1033	982	924	256768	391.500000	6	340	256868
1034	815	879	256507	398.000000	6	341	256612
1035	743	955	256747	0.000000	6	341	256848
1036	735	879	256915	0.000000	6	343	256974

1037	782	932	256804	518.000000	11	342	256953
1038	926	931	251280	471.333333	11	342	251477
1039	838	1064	248548	371.000000	5	344	248742
1040	1030	1006	125536	0.000000	5	345	126702
1041	839	905	211712	0.000000	5	344	212943
1042	827	873	192846	0.000000	5	344	194268
1043	809	792	191590	367.000000	5	344	192593
1044	829	1080	207119	620.000000	0	345	208547
1045	795	1153	204629	0.000000	0	344	206141
1046	759	1047	197569	542.000000	0	343	198671
1047	677	1127	203459	613.000000	1	343	204867
1048	704	1046	126430	386.500000	4	345	127226
1049	726	937	273698	297.500000	4	345	274046
1050	700	862	273031	0.000000	4	345	273451
1051	786	1003	278761	501.000000	4	346	279410
1052	975	956	156151	567.000000	4	346	166819
1053	785	861	110515	565.333333	4	346	111887
1054	957	925	79881	461.000000	3	346	81606
1055	952	969	80435	0.000000	1	346	81771
1056	680	904	79983	0.000000	3	348	81414
1057	603	859	271420	0.000000	3	348	272709
1058	686	1178	270655	0.000000	3	348	272020
1059	769	1168	275886	337.500000	3	348	276341
1060	800	1122	83187	0.000000	3	348	84374
1061	887	1216	272007	0.000000	3	348	273396
1062	852	1252	269272	0.000000	3	348	270475
1063	743	1091	171768	705.500000	1	348	146495
1064	607	1105	135110	684.000000	2	347	136226
1065	574	2275	132527	440.500000	2	345	133685
1066	650	2246	104630	381.500000	10	342	105796
1067	669	1971	115840	411.333333	12	342	116954
1068	610	1886	105027	376.600000	13	342	105770
1069	612	1844	133189	345.500000	12	341	134800
1070	606	1717	132302	0.000000	12	341	133853
1071	796	1594	130351	544.000000	14	341	131985
1072	668	1841	150495	389.500000	14	344	162668
1073	636	1632	152541	333.000000	9	344	161286
1074	627	1551	154716	573.500000	8	344	169757
1075	637	1447	93046	484.666667	7	344	93927
1076	26140	1648	81542	414.375000	10	345	82472
1077	43838	1598	148644	321.000000	13	346	164471
1078	38368	2015	143247	418.000000	10	346	160029
1079	38108	3427	144610	453.500000	10	345	160399
1080	37158	3123	136486	393.000000	7	348	147009
1081	34162	2888	143148	592.000000	7	349	158677
1082	34945	2754	140789	511.000000	7	350	156182
1083	37892	2650	136205	0.000000	4	350	152013
1084	39777	2640	137048	0.000000	0	351	153117
1085	36106	2296	139537	0.000000	0	351	155707
1086	16500	2237	138363	478.500000	0	351	155280
1087	39037	1671	74974	0.000000	0	351	86951
1088	72896	2561	133509	0.000000	0	351	148445
1089	80209	2436	131246	678.500000	0	352	149303
1090	102071	2615	137594	545.500000	0	352	149248
1091	127266	1503	128995	478.666667	0	354	144237
1092	119897	1513	131142	395.000000	0	354	146172
1093	105879	2463	127286	407.400000	0	355	142567
1094	120036	1914	131724	386.000000	0	357	146274
1095	104580	1653	129964	0.000000	0	359	142071
1096	77515	1845	126700	392.666667	0	359	139067
1097	65102	1906	128650	0.000000	1	359	141048
1098	67692	1986	123625	284.000000	1	359	136816
1099	55163	1952	126834	484.000000	1	358	139761
1100	48530	2239	128868	448.000000	1	357	141862
1101	47694	2245	125037	371.000000	1	357	141271
1102	39608	1718	121657	349.000000	1	357	140935

1103	52361	1532	210626	0.000000	1	357	200335
1104	17397	1428	207869	490.000000	0	358	197464
1105	2577	1313	215188	356.000000	0	357	212084
1106	1550	1351	216360	363.200000	2	358	211717
1107	1940	1382	214338	0.000000	3	356	204617
1108	1825	1388	213310	526.000000	3	356	213465
1109	1862	1343	218547	341.600000	5	356	202373
1110	2072	1268	223983	325.000000	10	354	206637
1111	1694	1374	225413	0.000000	10	355	205982
1112	4686	1302	227835	679.000000	10	355	236828
1113	6209	1243	247239	405.500000	8	355	244274
1114	8820	1480	249043	372.500000	8	356	239129
1115	8500	1399	252074	0.000000	7	357	256071
1116	4092	1299	202575	0.000000	5	356	216693
1117	1332	1223	200993	0.000000	1	357	215754
1118	1281	1159	204089	409.500000	1	357	218364
1119	1160	1109	203267	0.000000	1	358	213418
1120	11235	1220	202685	374.000000	0	357	213428
1121	12779	2297	214882	0.000000	0	356	223061
1122	13107	2339	213361	0.000000	0	356	221820
1123	10408	2095	212867	338.000000	0	356	221742
1124	11142	1821	208083	0.000000	0	356	217384
1125	8936	1668	205480	0.000000	0	356	215695
1126	4375	1547	204942	0.000000	0	356	215519
1127	1619	1534	206213	723.000000	0	356	216698
1128	2031	1445	206925	0.000000	0	355	217314
1129	2086	1252	202998	456.000000	0	355	210101
1130	2106	1226	201336	307.000000	0	355	208523
1131	2374	1279	202520	0.000000	0	354	213401
1132	1977	1175	206397	0.000000	0	353	217183
1133	1743	1145	209917	0.000000	0	354	221719
1134	1715	1327	206312	0.000000	0	354	218000
1135	1523	3556	205545	586.000000	0	354	213308
1136	1447	3341	200304	620.000000	0	354	208799
1137	1675	2754	196062	567.000000	0	354	205583
1138	1880	2463	193449	656.000000	0	355	206939
1139	2356	2507	198829	470.000000	0	355	208047
1140	2255	2265	199593	0.000000	0	355	208852
1141	1589	2150	196498	0.000000	0	355	206458
1142	1824	2106	192456	568.000000	0	355	203043
1143	1939	1784	196850	583.000000	0	355	206785
1144	2772	1627	196593	455.000000	0	355	206641
1145	3966	1579	106171	531.250000	0	359	119038
1146	2124	1500	196080	353.600000	0	359	205839
1147	1766	1449	198032	571.500000	1	359	211118
1148	1525	1298	191821	0.000000	1	359	205129
1149	1574	1361	191425	560.500000	1	359	201896
1150	2228	1326	191678	415.750000	1	359	205514
1151	2409	1311	191748	424.000000	8	359	202477
1152	2775	1284	193442	462.500000	8	359	203946
1153	3098	1216	195785	0.000000	8	359	205524
1154	3442	1176	198361	428.000000	7	360	211446
1155	2716	1187	197539	522.000000	7	360	214899
1156	2340	1968	194216	473.500000	7	361	214339
1157	2354	2038	193181	0.000000	7	361	213470
1158	2444	1727	193933	524.000000	0	361	209600
1159	3069	1655	195556	0.000000	0	362	211529
1160	3622	1565	186517	0.000000	0	363	206082
1161	2218	1496	179335	0.000000	0	363	200023
1162	2227	1400	184140	599.500000	0	363	204068
1163	2635	1208	183054	389.000000	0	363	203282
1164	2485	1176	198804	0.000000	3	364	218310
1165	3054	1132	194315	531.500000	3	365	214971
1166	7048	1105	198659	512.666667	5	365	218717
1167	5593	1124	201418	386.166667	5	365	220296
1168	4286	1090	205633	0.000000	5	366	224171

1169	6992	1075	208727	578.666667	5	366	226775
1170	8327	1266	208636	403.000000	5	367	226672
1171	7454	1225	196470	0.000000	2	367	216215
1172	10488	1125	203680	0.000000	2	366	222188
1173	10503	1161	197581	0.000000	0	367	217078
1174	13900	1066	196171	0.000000	0	367	215880
1175	13995	1063	195006	0.000000	0	368	213605
1176	9447	1143	197290	0.000000	0	367	215599
1177	9447	1327	197707	0.000000	0	367	216407
1178	6464	1204	194372	0.000000	0	367	214011
1179	5958	1195	193151	0.000000	0	366	213490
1180	4375	1100	207535	505.000000	0	366	226366
1181	2728	1064	208233	416.500000	0	366	226834
1182	1727	1213	199393	0.000000	1	367	219295
1183	1532	1168	202501	0.000000	1	368	222335
1184	1660	2864	198891	0.000000	1	367	219410
1185	1613	2993	196651	0.000000	1	367	217766
1186	1592	2516	193603	0.000000	1	367	215172
1187	1489	2230	191422	0.000000	6	367	213120
1188	1546	2175	178585	0.000000	6	367	200714
1189	1436	2032	176328	0.000000	5	367	198909
1190	1401	1918	173750	0.000000	5	367	196521
1191	1314	1373	182049	315.750000	5	367	203594
1192	1540	1128	180046	0.000000	5	367	201658
1193	1357	1098	178864	487.333333	5	367	200640
1194	1521	1063	179979	387.500000	0	366	201731
1195	1513	1057	179326	0.000000	0	366	201045
1196	1297	992	180194	0.000000	0	367	201582
1197	1141	1038	188933	0.000000	0	367	211050
1198	1353	1074	186785	0.000000	0	367	209433
1199	1204	1041	184783	418.000000	0	367	208131
1200	1185	1068	182659	0.000000	1	367	207083
1201	1336	1019	185255	0.000000	1	367	209219
1202	1334	1028	188019	0.000000	1	368	211647
1203	1093	1131	189455	0.000000	1	368	212742
1204	1047	1065	188008	0.000000	1	367	211463
1205	1028	1240	185769	0.000000	1	367	209366
1206	1187	1223	182183	519.000000	0	367	206069
1207	1148	1072	184224	0.000000	0	367	207421
1208	1049	1181	188417	379.000000	0	367	210772
1209	1204	1088	167429	767.000000	0	367	191301
1210	1172	1077	167371	0.000000	0	366	191400
1211	936	946	171055	0.000000	0	366	194453
1212	1104	1083	170906	0.000000	0	366	194640
1213	1170	1097	170680	0.000000	0	366	194550
1214	965	1225	183220	585.000000	0	366	206789
1215	1134	1187	185749	0.000000	0	366	209485
1216	1010	1213	195792	333.666667	0	366	218675
1217	924	1176	193640	0.000000	0	366	217194
1218	879	1185	190217	0.000000	0	365	214396
1219	1102	1262	188106	0.000000	0	365	212330
1220	1051	1244	192868	0.000000	0	365	216490
1221	1064	1192	191441	0.000000	0	366	214965
1222	1280	1127	188660	0.000000	0	366	212279
1223	1270	1124	186542	0.000000	0	366	211061
1224	948	1089	192326	0.000000	0	366	215632
1225	912	1192	176393	0.000000	0	366	200355
1226	977	1466	178235	564.000000	0	366	201898
1227	1046	1421	180723	0.000000	0	366	203865
1228	1219	1550	182655	0.000000	1	365	205795
1229	1209	1471	176690	0.000000	1	365	200426
1230	1223	1359	172898	0.000000	2	366	196684
1231	900	1268	176476	0.000000	2	366	199554
1232	792	1250	195542	0.000000	2	367	217194
1233	1085	1325	184383	628.000000	2	366	208283
1234	1106	1321	185287	0.000000	0	368	209004

1235	1406	1387	186172	0.000000	1	368	209553
1236	1081	1344	179959	0.000000	2	368	203856
1237	1125	1322	184686	0.000000	4	369	207904
1238	739	1164	176942	0.000000	4	369	201215
1239	770	1238	179493	0.000000	4	368	203810
1240	885	1129	179679	0.000000	4	368	203713
1241	1004	1146	179606	0.000000	4	370	203351
1242	984	1236	183315	0.000000	4	370	206703
1243	880	1195	163769	0.000000	0	370	188628
1244	811	1334	163972	0.000000	0	370	188083
1245	609	1194	163364	0.000000	0	370	186669
1246	544	1153	165714	0.000000	0	370	188744
1247	688	1819	164531	0.000000	0	370	187582
1248	852	1672	70581	0.000000	0	370	88448
1249	632	1726	68621	496.000000	0	370	83620
1250	753	1666	93105	0.000000	0	370	109935
1251	795	1684	95780	524.000000	0	370	112442
1252	808	1543	84317	0.000000	0	370	102075
1253	598	1540	85672	0.000000	0	370	102921
1254	739	1706	178674	0.000000	0	370	200873
1255	889	1669	175569	0.000000	0	369	198229
1256	900	1676	174595	0.000000	0	368	197171
1257	851	1608	177983	0.000000	0	367	199986
1258	942	1501	172809	0.000000	0	368	195333
1259	716	1550	170240	0.000000	0	368	192329
1260	718	1484	156524	0.000000	0	368	178840
1261	674	2167	158883	0.000000	0	367	180989
1262	734	2062	160917	0.000000	0	367	182756
1263	650	1947	161596	0.000000	0	367	183014
1264	720	1886	155106	0.000000	0	367	178234
1265	690	1912	159061	0.000000	0	367	181421
1266	590	1858	154096	0.000000	0	368	177380
1267	513	1735	154098	0.000000	0	368	177339
1268	964	1784	169973	0.000000	0	367	193002
1269	948	1591	90094	0.000000	0	366	106612
1270	773	1580	91363	0.000000	0	366	108366
1271	830	1520	87277	0.000000	0	367	104554
1272	738	1502	86850	243.500000	0	367	103855
1273	728	1440	76326	226.000000	0	367	93772
1274	883	1388	77747	0.000000	0	367	95009
1275	883	1812	80237	0.000000	0	367	97949
1276	817	1681	160527	0.000000	0	368	180884
1277	856	1651	154773	524.000000	0	368	175905
1278	861	1783	142524	578.000000	0	366	164592
1279	846	1648	152527	327.000000	0	366	173068
1280	629	1480	152973	426.000000	0	366	173101
1281	552	1428	157836	0.000000	0	366	177303
1282	797	1413	153075	0.000000	0	366	177454
1283	2783	1404	153240	0.000000	0	366	173565
1284	3561	1456	159900	0.000000	0	366	182754
1285	2584	1509	157000	338.000000	0	367	180826
1286	920	1414	157231	0.000000	0	367	179706
1287	806	1402	149995	0.000000	0	367	174156
1288	613	1269	165640	0.000000	0	367	190712
1289	856	1371	167088	0.000000	0	367	192152
1290	972	1436	160550	0.000000	0	367	187333
1291	818	1360	158182	0.000000	0	366	190443
1292	699	1430	161890	286.500000	0	366	193309
1293	868	1501	164682	0.000000	0	366	196097
1294	984	1498	164845	301.666667	0	367	195957
1295	1193	1560	166158	231.500000	0	367	197279
1296	1207	1752	147652	0.000000	0	367	177915
1297	1025	1705	148909	0.000000	0	367	179348
1298	1098	1745	146850	0.000000	0	367	177138
1299	984	1579	150009	0.000000	0	367	178997
1300	1434	1566	144635	0.000000	0	367	174217

1301	2150	1658	145801	0.000000	0	367	174965
1302	1831	1567	144494	0.000000	0	367	173293
1303	2056	2259	146153	0.000000	0	367	175380
1304	1954	2221	145220	0.000000	0	367	174476
1305	2037	2035	144031	0.000000	3	367	173381
1306	2082	2052	147408	445.666667	0	368	175850
1307	2095	2023	148046	0.000000	3	368	176578
1308	1044	1925	147921	0.000000	3	368	176678
1309	590	1668	163246	0.000000	3	367	192698
1310	725	2001	164039	0.000000	3	367	193488
1311	743	1789	164348	376.250000	13	368	193759
1312	1159	1783	161081	0.000000	11	370	190975
1313	1081	1814	160728	465.000000	12	370	190149
1314	877	1859	145553	402.000000	16	371	174517
1315	760	1755	140862	0.000000	16	372	170358
1316	759	1926	157238	612.000000	16	372	187196
1317	901	2937	137573	515.000000	15	373	167402
1318	933	3254	135038	0.000000	5	373	164676
1319	850	2792	137122	0.000000	5	373	166167
1320	870	2840	138403	0.000000	0	374	166467
1321	786	3380	141684	412.000000	0	374	168679
1322	672	3657	140256	367.666667	0	375	167178
1323	588	3414	136841	0.000000	0	375	163913
1324	594	3487	137575	0.000000	0	375	164482
1325	678	3448	135642	0.000000	0	376	162632
1326	754	3231	129767	360.000000	0	377	157028
1327	687	3324	141365	0.000000	0	378	170130
1328	619	3429	139969	0.000000	0	379	168785
1329	657	2700	135621	0.000000	0	380	164893
1330	611	3301	132797	0.000000	0	381	161959
1331	655	4137	131006	291.500000	2	381	160044
1332	723	4513	128797	0.000000	2	381	157601
1333	752	4782	117195	0.000000	2	381	143391
1334	721	4571	116277	0.000000	2	381	142352
1335	820	4360	114926	559.000000	2	382	140372
1336	48	39	379454	186.250000	0	226	465864
1337	78	69	404875	168.500000	0	226	487377
1338	50	38	407748	164.666667	0	227	494572
1339	37	21	413262	179.000000	0	227	498931
1340	63	54	411552	180.666667	0	227	497810
1341	37	28	415817	225.250000	0	227	501227
1342	38	33	414664	190.000000	0	226	500469
1343	38	26	414341	180.000000	0	226	500399
1344	20	12	413918	207.666667	0	226	500335
1345	40	35	415214	169.000000	0	227	501318
1346	222	201	413591	0.000000	0	226	500659
1347	178	175	408760	171.500000	0	226	496706
1348	61	30	399549	0.000000	0	226	484701
1349	16	15	399135	193.285714	0	226	484657
1350	19	18	406383	161.666667	0	225	488540
1351	23	20	417056	148.333333	0	225	494006
1352	44	41	418699	148.600000	0	225	494503
1353	15	14	426139	123.000000	0	225	498981
1354	17	16	429750	0.000000	0	225	501102
1355	16	10	438659	0.000000	0	225	507319
1356	69	60	445913	160.222222	0	225	511525
1357	80	55	448515	105.500000	0	225	512635
1358	121	98	452730	127.000000	0	226	514821
1359	78	68	341602	0.000000	0	225	382407
1360	45	33	343616	0.000000	0	226	384935
1361	49	39	333774	138.000000	0	226	374229
1362	54	37	322209	146.000000	0	227	366514
1363	53	28	312061	178.200000	0	227	362294
1364	34	29	324869	162.800000	0	227	371120
1365	122	38	282651	160.600000	0	227	334575
1366	81	25	285552	196.571429	0	227	336553

1367	77	63	290998	165.909091	2	227	341518
1368	105	55	288818	175.500000	3	227	338022
1369	56	47	283728	203.666667	3	227	332444
1370	36	27	288997	218.666667	3	228	334539
1371	39	27	296078	168.750000	3	228	341988
1372	33	27	299220	177.833333	3	228	345197
1373	30	26	448821	168.333333	3	227	502606
1374	59	23	472256	159.333333	0	228	522166
1375	27	25	481960	165.000000	0	229	530635
1376	20	18	505548	139.750000	0	227	547701
1377	35	20	539644	0.000000	0	227	572423
1378	20	18	547905	165.800000	0	226	578003
1379	33	27	562240	162.000000	0	226	594268
1380	35	34	555210	142.666667	0	225	589751
1381	40	31	551205	0.000000	0	226	588103
1382	52	42	559018	211.750000	0	226	595743
1383	58	49	554701	164.400000	0	226	592746
1384	40	36	555634	174.250000	0	226	594567
1385	31	23	538266	170.500000	2	226	581016
1386	15	9	546391	139.000000	2	226	588651
1387	18	17	533536	198.000000	2	225	580193
1388	52	48	541360	0.000000	2	225	584867
1389	44	39	549850	194.000000	2	225	591210
1390	33	31	551360	205.000000	2	225	591896
1391	46	43	544636	234.571429	2	224	586274
1392	18	15	568271	275.125000	0	224	606940
1393	25	22	563626	250.142857	0	224	600868
1394	27	25	555423	212.600000	0	224	595268
1395	26	11	564996	201.000000	0	224	602651
1396	35	15	559317	208.000000	1	224	595769
1397	67	58	561622	187.000000	1	225	597060
1398	82	76	556652	0.000000	1	224	594814
1399	30	30	553142	245.500000	1	225	592939
1400	132	73	553023	0.000000	1	225	590759
1401	90	51	544634	0.000000	2	226	587015
1402	107	47	536673	0.000000	2	226	571040
1403	111	55	550856	0.000000	0	226	581556
1404	103	52	557842	236.600000	0	226	581556
1405	95	60	537363	173.000000	0	226	564287
1406	144	59	538170	0.000000	0	227	562225
1407	93	52	540942	240.250000	0	227	560625
1408	150	76	524966	201.333333	4	228	546174
1409	111	67	516319	214.571429	6	228	540673
1410	118	59	511468	0.000000	6	230	533818
1411	147	85	515156	198.333333	5	231	535432
1412	140	88	513002	145.000000	5	231	534768
1413	133	91	520109	201.125000	5	231	538280
1414	115	95	519319	189.500000	5	232	536826
1415	133	109	538287	255.166667	1	231	549608
1416	109	81	529271	219.333333	1	232	542717
1417	134	71	514177	224.000000	2	234	533284
1418	77	51	520627	256.166667	2	235	537875
1419	110	82	525801	216.000000	2	235	540350
1420	141	69	526663	272.333333	3	235	542294
1421	136	86	522011	212.000000	3	235	537581
1422	186	133	523749	218.166667	1	235	539246
1423	145	98	518884	197.250000	5	242	535509
1424	115	83	520384	175.666667	5	242	536111
1425	75	56	514266	260.000000	7	243	533633
1426	183	115	503459	206.200000	7	243	523928
1427	163	112	304795	235.000000	7	244	332081
1428	315	219	296312	250.500000	7	243	321938
1429	393	312	302868	262.400000	6	243	330055
1430	655	306	303630	245.800000	1	243	331511
1431	441	279	303743	238.400000	2	242	331555
1432	432	272	296275	0.000000	1	243	324253

1433	465	308	298039	0.000000	1	243	325322
1434	479	335	296472	0.000000	2	243	324116
1435	408	239	298975	0.000000	2	243	326462
1436	360	205	301194	0.000000	2	243	330363
1437	248	109	291989	209.333333	2	243	319283
1438	345	202	294750	223.818182	2	243	321391
1439	595	385	298780	170.000000	2	242	324731
1440	539	296	302653	342.000000	2	242	328982
1441	411	175	300100	322.500000	2	242	327346
1442	359	185	298970	253.200000	2	243	326690
1443	3104	571	291076	232.777778	11	246	318276
1444	2530	536	296041	243.166667	14	249	322192
1445	1879	562	294639	247.352941	14	251	320722
1446	2099	570	293293	245.333333	15	252	319200
1447	1773	562	289527	241.000000	15	252	313790
1448	1608	518	289245	252.285714	15	253	313114
1449	1658	528	287850	304.400000	15	255	311772
1450	1411	535	288704	244.333333	1	257	312846
1451	970	431	289748	238.666667	1	256	313547
1452	854	409	289305	224.000000	1	258	313285
1453	810	460	286865	265.400000	0	259	310034
1454	1120	444	286719	0.000000	0	259	309043
1455	1212	439	319810	273.666667	0	260	346724
1456	982	499	346051	268.500000	0	260	372202
1457	961	558	346047	246.500000	1	261	371050
1458	718	460	343872	236.666667	1	262	369772
1459	920	478	344405	281.000000	1	263	369834
1460	1070	616	343862	265.250000	1	263	369326
1461	1022	633	346059	231.333333	1	263	370906
1462	970	554	349684	284.000000	2	267	375800
1463	845	564	360898	0.000000	2	268	382880
1464	858	509	386061	268.500000	5	268	391894
1465	1004	442	391598	385.333333	5	268	389624
1466	624	401	394795	157.000000	5	268	389928
1467	580	408	405264	0.000000	5	269	394679
1468	546	344	414517	298.500000	5	269	410553
1469	573	325	417999	328.800000	0	269	414197
1470	610	277	422942	321.888889	0	269	428815
1471	544	283	425865	283.833333	0	269	443553
1472	543	247	436288	0.000000	1	271	450759
1473	562	259	443198	341.000000	1	271	425344
1474	462	246	403181	0.000000	1	271	414166
1475	586	266	327410	0.000000	1	272	350773
1476	511	284	317937	0.000000	1	272	346522
1477	362	251	395475	0.000000	1	272	415089
1478	339	245	400203	297.750000	1	271	419045
1479	340	222	405490	237.000000	0	271	423495
1480	380	243	405219	308.250000	0	271	424297
1481	442	237	405969	0.000000	0	271	424953
1482	404	261	406543	0.000000	0	271	425964
1483	565	275	413620	257.000000	0	271	429003
1484	424	272	418538	0.000000	0	271	424274
1485	434	266	426097	0.000000	1	271	429792
1486	347	228	410436	311.333333	1	271	426634
1487	346	224	412520	261.000000	3	271	425788
1488	425	293	410440	0.000000	3	272	425318
1489	439	283	428508	283.000000	3	272	440638
1490	372	256	429377	304.000000	3	272	440474
1491	393	251	428296	368.333333	3	272	443558
1492	346	252	428448	221.000000	2	273	443859
1493	405	206	429076	0.000000	2	273	443646
1494	377	235	468058	280.000000	0	273	470433
1495	289	184	460322	0.000000	0	274	468464
1496	301	191	483726	297.250000	0	274	482386
1497	362	201	483667	267.000000	0	276	482486
1498	305	196	483748	269.000000	0	277	482538

1499	341	199	482615	326.000000	0	278	481650
1500	371	239	450708	316.000000	0	278	446642
1501	349	247	492936	359.666667	1	280	491073
1502	1456	1007	489948	276.000000	1	279	489249
1503	1462	1056	434017	271.000000	2	279	435777
1504	1183	810	391444	241.250000	2	280	402606
1505	1047	724	359109	0.000000	2	279	379892
1506	963	657	341031	243.800000	4	279	366392
1507	881	645	341960	209.000000	12	279	367293
1508	941	651	342191	0.000000	12	279	367981
1509	447	289	320266	246.000000	12	279	348232
1510	341	217	322116	266.800000	12	279	349562
1511	321	185	298233	0.000000	12	279	329730
1512	377	234	307554	283.000000	4	279	336751
1513	357	224	306545	274.000000	4	279	335365
1514	330	197	302433	0.000000	4	279	332512
1515	311	189	318140	277.000000	4	279	346529
1516	392	284	344569	0.000000	19	279	370574
1517	522	362	346617	326.500000	19	280	371511
1518	453	317	363454	297.750000	19	280	381413
1519	408	282	351333	239.000000	19	280	377496
1520	670	373	300267	209.666667	19	281	325742
1521	613	387	246090	304.000000	19	281	277955
1522	718	387	210302	299.000000	11	282	243481
1523	830	367	176641	406.500000	11	282	208842
1524	525	279	177477	332.222222	11	283	210718
1525	640	303	177484	317.125000	11	284	210535
1526	645	296	153947	295.500000	11	284	184902
1527	591	290	146085	401.000000	11	285	174231
1528	481	272	144346	362.000000	11	286	172816
1529	613	296	146917	366.333333	11	286	175702
1530	852	549	138904	274.000000	11	287	167285
1531	870	547	134556	282.000000	11	288	161989
1532	787	486	130237	295.600000	11	289	157344
1533	757	452	131819	289.400000	11	289	159573
1534	1054	678	126596	247.333333	11	290	153081
1535	927	588	135085	322.000000	11	291	161660
1536	839	504	104512	335.800000	11	291	127138
1537	1180	642	105466	340.800000	11	292	130204
1538	5450	1421	109977	321.250000	11	293	134714
1539	5651	1483	116828	349.571429	11	293	141243
1540	5687	1510	217453	342.714286	11	294	219782
1541	5706	1548	83294	392.000000	12	295	84028
1542	5314	1522	168118	327.875000	12	300	169789
1543	4586	1368	156445	334.333333	14	301	158675
1544	4125	1446	153901	608.000000	14	300	156071
1545	3453	1355	81179	335.600000	14	299	82756
1546	3325	1325	80903	288.666667	6	298	82616
1547	3206	1360	72195	355.250000	6	298	75459
1548	2900	1366	72189	317.500000	6	300	75488
1549	2814	1263	67869	512.500000	5	301	71181
1550	2505	1161	67557	476.777778	2	301	70805
1551	2750	1143	67827	421.000000	3	300	71125
1552	2743	1154	73344	365.000000	3	302	76556
1553	2242	1121	72847	0.000000	7	300	76133
1554	2130	1099	78202	402.500000	7	302	80852
1555	2172	1084	80717	0.000000	7	303	82991
1556	2125	1035	83704	393.333333	7	304	84969
1557	1957	1026	83581	356.000000	10	304	84899
1558	3791	2147	100334	355.000000	9	304	101265
1559	3893	2317	83775	0.000000	9	305	85045
1560	3796	2064	84469	451.000000	9	305	85410
1561	3518	1940	84318	346.333333	8	306	85348
1562	3482	1932	84332	0.000000	8	306	85392
1563	3411	1877	84334	370.000000	8	306	85405
1564	3092	1731	84448	0.000000	4	306	85519

1565	2307	1207	84485	359.000000	5	305	85603
1566	2253	1081	84623	0.000000	2	305	85684
1567	2167	1093	84594	300.000000	2	307	85662
1568	2203	1101	84559	0.000000	2	309	85688
1569	2295	1139	84647	338.666667	2	310	85764
1570	2646	1122	84562	451.500000	2	310	85692
1571	2154	1032	84472	361.000000	2	311	85676
1572	3984	2544	84721	317.000000	2	312	85838
1573	4284	2570	84769	301.000000	2	312	85867
1574	4099	2289	83840	378.000000	2	312	85403
1575	4031	2235	84668	295.666667	2	313	85884
1576	3990	2255	84659	305.600000	5	314	85885
1577	4078	2298	84581	243.000000	6	316	85830
1578	3663	2028	84597	389.000000	6	316	85852
1579	3289	1624	81765	0.000000	4	315	83779
1580	3441	1512	82769	593.000000	7	318	84434
1581	3245	1547	82132	348.750000	8	318	84115
1582	3102	1523	86016	372.000000	11	318	86785
1583	2987	1553	83756	369.000000	9	317	85547
1584	2860	1373	80396	389.000000	13	318	82960
1585	2662	1254	79956	351.000000	13	319	82590
1586	2363	1373	79071	0.000000	13	318	81831
1587	2279	1350	80134	429.500000	9	321	82787
1588	2356	1356	79797	418.000000	6	321	82410
1589	2034	1299	74877	270.600000	6	321	90182
1590	2240	1289	79293	0.000000	0	322	93858
1591	2244	1195	68613	412.000000	0	322	82709
1592	2142	1150	64688	430.000000	0	322	79075
1593	2053	1226	69767	0.000000	0	321	83645
1594	1911	1230	74374	492.000000	5	321	88270
1595	2094	1217	72278	360.000000	8	321	85491
1596	1883	1150	67766	278.000000	9	321	81554
1597	2292	1230	72594	300.000000	9	321	86682
1598	2065	1288	77077	0.000000	9	321	90728
1599	2329	1246	78359	0.000000	9	323	91986
1600	2688	1413	77416	0.000000	9	324	91402
1601	2577	1451	79071	0.000000	3	324	92734
1602	2564	1425	77361	0.000000	0	324	91465
1603	2599	1337	75410	0.000000	0	325	90048
1604	2536	1387	45644	0.000000	0	326	57869
1605	2537	1436	51472	0.000000	3	326	63678
1606	2301	1301	48379	0.000000	3	326	60840
1607	2574	1352	73087	441.000000	3	326	86798
1608	2353	1263	71852	439.000000	3	326	85545
1609	2862	1304	74298	395.000000	3	325	87662
1610	2544	1276	74700	400.333333	5	327	88098
1611	2254	1349	80506	0.000000	5	327	93917
1612	2509	1333	56367	347.333333	2	327	68739
1613	2355	1298	53648	0.000000	4	327	66156
1614	4837	3294	54734	323.666667	4	327	67437
1615	5478	3435	474438	402.500000	4	327	474638
1616	4731	2913	86078	387.500000	4	326	99358
1617	4066	2652	85292	365.000000	4	327	98254
1618	4127	2538	79047	620.000000	4	327	92475
1619	4212	2540	81394	566.500000	4	329	94672
1620	4001	2464	74827	486.000000	2	329	88902
1621	5543	3783	146415	0.000000	5	329	155077
1622	5294	3454	489974	565.000000	11	327	490071
1623	4967	3169	390456	419.500000	14	327	390707
1624	4632	3002	390393	359.000000	12	328	390729
1625	4594	2870	454868	0.000000	12	328	455351
1626	4571	2772	455749	0.000000	12	327	455789
1627	4498	2724	443108	0.000000	12	326	444004
1628	3283	1929	444982	471.000000	9	327	445170
1629	2700	1611	442715	452.666667	1	328	443624
1630	2808	1535	436807	0.000000	0	328	437346

1631	2478	1477	410615	0.000000	0	329	407557
1632	2471	1307	406252	0.000000	1	329	406498
1633	2739	1426	137501	595.000000	1	329	138523
1634	2298	1413	405398	561.000000	1	328	405896
1635	3437	2175	405649	341.000000	1	328	406237
1636	3974	2244	391967	0.000000	3	328	392593
1637	3744	2004	392317	432.000000	3	328	392979
1638	3684	1884	389303	0.000000	2	332	389671
1639	2900	1751	131185	336.500000	2	332	128200
1640	2963	1716	87725	0.000000	2	332	88743
1641	3091	1795	93311	465.000000	2	333	94242
1642	2838	1613	92886	0.000000	2	333	93893
1643	2435	1443	93130	406.000000	0	334	94087
1644	2346	1493	106199	389.000000	0	335	106344
1645	2353	1447	105800	0.000000	0	335	105617
1646	2298	1383	104196	0.000000	0	335	105128
1647	2249	1404	90598	0.000000	0	335	91376
1648	2074	1276	90853	449.500000	0	335	91589
1649	2130	1417	90654	508.000000	0	337	91402
1650	2039	1283	376423	384.000000	0	338	377020
1651	2064	1291	379606	0.000000	0	337	379854
1652	2045	1304	379950	418.000000	0	336	380270
1653	2046	1347	376015	424.500000	0	337	377551
1654	1780	1143	388479	0.000000	0	337	389748
1655	1918	1169	386878	425.000000	0	337	388137
1656	1902	1197	391298	0.000000	0	337	391892
1657	2256	1145	390244	0.000000	0	337	390988
1658	2152	1262	390109	424.000000	0	336	390812
1659	2458	1285	394280	569.000000	0	336	394523
1660	2325	1327	394347	0.000000	0	336	394601
1661	2406	1220	394451	0.000000	0	335	394690
1662	1968	1124	394532	0.000000	0	334	394769
1663	2027	1178	394606	0.000000	0	334	394857
1664	2274	1307	394714	0.000000	0	334	394974
1665	2095	1272	394813	0.000000	0	334	395060
1666	2226	1335	394777	532.333333	0	334	395056
1667	2474	1344	394907	0.000000	0	336	395185
1668	2349	1223	394836	653.500000	0	337	395125
1669	2052	1134	394976	0.000000	0	337	395255
1670	1868	1222	392975	0.000000	0	337	393730
1671	2073	1287	392828	490.333333	0	337	393648
1672	1939	1258	394892	449.000000	0	337	395344
1673	1917	1176	371312	400.000000	0	338	373179
1674	2004	1207	369610	298.000000	0	338	370202
1675	1887	1252	367034	0.000000	1	338	369376
1676	1742	1133	366199	467.500000	1	338	368415
1677	1948	1194	365574	0.000000	1	338	367673
1678	2023	1295	365041	0.000000	1	338	366754
1679	2072	1254	374382	0.000000	1	340	376394
1680	2130	1243	139948	0.000000	1	340	159312
1681	1994	1247	135604	0.000000	1	340	160070
1682	2044	1266	479620	418.000000	0	340	480199
1683	1854	1168	478705	511.000000	0	340	478590
1684	2012	1289	480301	0.000000	3	339	480841
1685	1745	1192	479880	0.000000	3	340	480418
1686	1742	1206	479150	489.666667	3	341	479552
1687	2044	1194	255638	341.000000	3	340	255776
1688	1930	1200	255713	0.000000	3	341	255856
1689	1885	1238	255800	0.000000	3	341	255954
1690	1578	1031	255918	0.000000	3	340	256059
1691	2084	1384	255994	515.000000	0	340	256124
1692	2144	1453	120997	485.000000	0	340	128588
1693	2050	1299	256177	407.000000	1	340	256308
1694	2196	1296	256178	510.500000	1	340	256307
1695	2065	1318	256213	256.000000	1	339	256350
1696	1973	1252	256032	484.000000	1	340	256172

1697	1780	1152	256330	0.000000	1	338	256463
1698	1707	1072	256400	531.333333	1	338	256541
1699	1413	952	256523	425.000000	0	339	256674
1700	1255	855	256757	423.000000	4	341	256907
1701	1426	924	256768	391.500000	6	340	256868
1702	1366	879	256507	398.000000	6	341	256612
1703	1421	955	256747	0.000000	6	341	256848
1704	1211	879	256915	0.000000	6	343	256974
1705	1324	932	256804	518.000000	11	342	256953
1706	1325	931	251280	471.333333	11	342	251477
1707	1473	1064	248548	371.000000	5	344	248742
1708	1440	1006	125536	0.000000	5	345	126702
1709	1314	905	211712	0.000000	5	344	212943
1710	1341	873	192846	0.000000	5	344	194268
1711	1207	792	191590	367.000000	5	344	192593
1712	1566	1080	207119	620.000000	0	345	208547
1713	1671	1153	204629	0.000000	0	344	206141
1714	1547	1047	197569	542.000000	0	343	198671
1715	1691	1127	203459	613.000000	1	343	204867
1716	1722	1046	126430	386.500000	4	345	127226
1717	1453	937	273698	297.500000	4	345	274046
1718	1298	862	273031	0.000000	4	345	273451
1719	1547	1003	278761	501.000000	4	346	279410
1720	1487	956	156151	567.000000	4	346	166819
1721	1299	861	110515	565.333333	4	346	111887
1722	1481	925	79881	461.000000	3	346	81606
1723	1365	969	80435	0.000000	1	346	81771
1724	1302	904	79983	0.000000	3	348	81414
1725	1242	859	271420	0.000000	3	348	272709
1726	1736	1178	270655	0.000000	3	348	272020
1727	1855	1168	275886	337.500000	3	348	276341
1728	1719	1122	83187	0.000000	3	348	84374
1729	1767	1216	272007	0.000000	3	348	273396
1730	1721	1252	269272	0.000000	3	348	270475
1731	1500	1091	171768	705.500000	1	348	146495
1732	1567	1105	135110	684.000000	2	347	136226
1733	3129	2275	132527	440.500000	2	345	133685
1734	3373	2246	104630	381.500000	10	342	105796
1735	3094	1971	115840	411.333333	12	342	116954
1736	2776	1886	105027	376.600000	13	342	105770
1737	2814	1844	133189	345.500000	12	341	134800
1738	2630	1717	132302	0.000000	12	341	133853
1739	2517	1594	130351	544.000000	14	341	131985
1740	2595	1841	150495	389.500000	14	344	162668
1741	2351	1632	152541	333.000000	9	344	161286
1742	2232	1551	154716	573.500000	8	344	169757
1743	2132	1447	93046	484.666667	7	344	93927
1744	2757	1648	81542	414.375000	10	345	82472
1745	2600	1598	148644	321.000000	13	346	164471
1746	3229	2015	143247	418.000000	10	346	160029
1747	10019	3427	144610	453.500000	10	345	160399
1748	6035	3123	136486	393.000000	7	348	147009
1749	6585	2888	143148	592.000000	7	349	158677
1750	6284	2754	140789	511.000000	7	350	156182
1751	6405	2650	136205	0.000000	4	350	152013
1752	8428	2640	137048	0.000000	0	351	153117
1753	3842	2296	139537	0.000000	0	351	155707
1754	6357	2237	138363	478.500000	0	351	155280
1755	2905	1671	74974	0.000000	0	351	86951
1756	7427	2561	133509	0.000000	0	351	148445
1757	7421	2436	131246	678.500000	0	352	149303
1758	5561	2615	137594	545.500000	0	352	149248
1759	2305	1503	128995	478.666667	0	354	144237
1760	2367	1513	131142	395.000000	0	354	146172
1761	5744	2463	127286	407.400000	0	355	142567
1762	4846	1914	131724	386.000000	0	357	146274

1763	2656	1653	129964	0.000000	0	359	142071
1764	6516	1845	126700	392.666667	0	359	139067
1765	5035	1906	128650	0.000000	1	359	141048
1766	5465	1986	123625	284.000000	1	359	136816
1767	6980	1952	126834	484.000000	1	358	139761
1768	5779	2239	128868	448.000000	1	357	141862
1769	3514	2245	125037	371.000000	1	357	141271
1770	2631	1718	121657	349.000000	1	357	140935
1771	2350	1532	210626	0.000000	1	357	200335
1772	2225	1428	207869	490.000000	0	358	197464
1773	2344	1313	215188	356.000000	0	357	212084
1774	2101	1351	216360	363.200000	2	358	211717
1775	2085	1382	214338	0.000000	3	356	204617
1776	2155	1388	213310	526.000000	3	356	213465
1777	2086	1343	218547	341.600000	5	356	202373
1778	1854	1268	223983	325.000000	10	354	206637
1779	2038	1374	225413	0.000000	10	355	205982
1780	2058	1302	227835	679.000000	10	355	236828
1781	1805	1243	247239	405.500000	8	355	244274
1782	2204	1480	249043	372.500000	8	356	239129
1783	2081	1399	252074	0.000000	7	357	256071
1784	1926	1299	202575	0.000000	5	356	216693
1785	1891	1223	200993	0.000000	1	357	215754
1786	1732	1159	204089	409.500000	1	357	218364
1787	1740	1109	203267	0.000000	1	358	213418
1788	1731	1220	202685	374.000000	0	357	213428
1789	3139	2297	214882	0.000000	0	356	223061
1790	3296	2339	213361	0.000000	0	356	221820
1791	2980	2095	212867	338.000000	0	356	221742
1792	2591	1821	208083	0.000000	0	356	217384
1793	2433	1668	205480	0.000000	0	356	215695
1794	2342	1547	204942	0.000000	0	356	215519
1795	2371	1534	206213	723.000000	0	356	216698
1796	2170	1445	206925	0.000000	0	355	217314
1797	1986	1252	202998	456.000000	0	355	210101
1798	1874	1226	201336	307.000000	0	355	208523
1799	1890	1279	202520	0.000000	0	354	213401
1800	1733	1175	206397	0.000000	0	353	217183
1801	1978	1327	206312	0.000000	0	354	218000
1802	4834	3556	205545	586.000000	0	354	213308
1803	4838	3341	200304	620.000000	0	354	208799
1804	4217	2754	196062	567.000000	0	354	205583
1805	3873	2463	193449	656.000000	0	355	206939
1806	3889	2507	198829	470.000000	0	355	208047
1807	3557	2265	199593	0.000000	0	355	208852
1808	3214	2150	196498	0.000000	0	355	206458
1809	3083	2106	192456	568.000000	0	355	203043
1810	2708	1784	196850	583.000000	0	355	206785
1811	2560	1627	196593	455.000000	0	355	206641
1812	2421	1579	106171	531.250000	0	359	119038
1813	2304	1500	196080	353.600000	0	359	205839
1814	2184	1449	198032	571.500000	1	359	211118
1815	2097	1298	191821	0.000000	1	359	205129
1816	2084	1361	191425	560.500000	1	359	201896
1817	1930	1326	191678	415.750000	1	359	205514
1818	1955	1311	191748	424.000000	8	359	202477
1819	1874	1284	193442	462.500000	8	359	203946
1820	1737	1216	195785	0.000000	8	359	205524
1821	1672	1176	198361	428.000000	7	360	211446
1822	1613	1187	197539	522.000000	7	360	214899
1823	2742	1968	194216	473.500000	7	361	214339
1824	3032	2038	193181	0.000000	7	361	213470
1825	2507	1727	193933	524.000000	0	361	209600
1826	2588	1655	195556	0.000000	0	362	211529
1827	2407	1565	186517	0.000000	0	363	206082
1828	2394	1496	179335	0.000000	0	363	200023

1829	2231	1400	184140	599.500000	0	363	204068
1830	2010	1208	183054	389.000000	0	363	203282
1831	1852	1176	198804	0.000000	3	364	218310
1832	1714	1132	194315	531.500000	3	365	214971
1833	1708	1105	198659	512.666667	5	365	218717
1834	1672	1124	201418	386.166667	5	365	220296
1835	1676	1090	205633	0.000000	5	366	224171
1836	1604	1075	208727	578.666667	5	366	226775
1837	1980	1266	208636	403.000000	5	367	226672
1838	2091	1225	196470	0.000000	2	367	216215
1839	1858	1125	203680	0.000000	2	366	222188
1840	1875	1161	197581	0.000000	0	367	217078
1841	1656	1066	196171	0.000000	0	367	215880
1842	1820	1063	195006	0.000000	0	368	213605
1843	1682	1143	197290	0.000000	0	367	215599
1844	1868	1327	197707	0.000000	0	367	216407
1845	1841	1204	194372	0.000000	0	367	214011
1846	1838	1195	193151	0.000000	0	366	213490
1847	1617	1100	207535	505.000000	0	366	226366
1848	1592	1064	208233	416.500000	0	366	226834
1849	1700	1213	199393	0.000000	1	367	219295
1850	1613	1168	202501	0.000000	1	368	222335
1851	3999	2864	198891	0.000000	1	367	219410
1852	4176	2993	196651	0.000000	1	367	217766
1853	3826	2516	193603	0.000000	1	367	215172
1854	3479	2230	191422	0.000000	6	367	213120
1855	3268	2175	178585	0.000000	6	367	200714
1856	2961	2032	176328	0.000000	5	367	198909
1857	2881	1918	173750	0.000000	5	367	196521
1858	2105	1373	182049	315.750000	5	367	203594
1859	1684	1128	180046	0.000000	5	367	201658
1860	1689	1098	178864	487.333333	5	367	200640
1861	1524	1063	179979	387.500000	0	366	201731
1862	1548	1057	179326	0.000000	0	366	201045
1863	1519	992	180194	0.000000	0	367	201582
1864	1547	1038	188933	0.000000	0	367	211050
1865	1537	1074	186785	0.000000	0	367	209433
1866	1441	1041	184783	418.000000	0	367	208131
1867	1530	1068	182659	0.000000	1	367	207083
1868	1481	1019	185255	0.000000	1	367	209219
1869	1465	1028	188019	0.000000	1	368	211647
1870	1567	1131	189455	0.000000	1	368	212742
1871	1467	1065	188008	0.000000	1	367	211463
1872	1660	1240	185769	0.000000	1	367	209366
1873	1776	1223	182183	519.000000	0	367	206069
1874	1499	1072	184224	0.000000	0	367	207421
1875	1633	1181	188417	379.000000	0	367	210772
1876	1465	1088	167429	767.000000	0	367	191301
1877	1529	1077	167371	0.000000	0	366	191400
1878	1305	946	171055	0.000000	0	366	194453
1879	1444	1083	170906	0.000000	0	366	194640
1880	1478	1097	170680	0.000000	0	366	194550
1881	1780	1225	183220	585.000000	0	366	206789
1882	1627	1187	185749	0.000000	0	366	209485
1883	1654	1213	195792	333.666667	0	366	218675
1884	1549	1176	193640	0.000000	0	366	217194
1885	1647	1185	190217	0.000000	0	365	214396
1886	1681	1262	188106	0.000000	0	365	212330
1887	1707	1244	192868	0.000000	0	365	216490
1888	1720	1192	191441	0.000000	0	366	214965
1889	1575	1127	188660	0.000000	0	366	212279
1890	1661	1124	186542	0.000000	0	366	211061
1891	1643	1089	192326	0.000000	0	366	215632
1892	1590	1192	176393	0.000000	0	366	200355
1893	2008	1466	178235	564.000000	0	366	201898
1894	2166	1421	180723	0.000000	0	366	203865

1895	2246	1550	182655	0.000000	1	365	205795
1896	2190	1471	176690	0.000000	1	365	200426
1897	1920	1359	172898	0.000000	2	366	196684
1898	1877	1268	176476	0.000000	2	366	199554
1899	1880	1250	195542	0.000000	2	367	217194
1900	1802	1325	184383	628.000000	2	366	208283
1901	1828	1321	185287	0.000000	0	368	209004
1902	1926	1387	186172	0.000000	1	368	209553
1903	1856	1344	179959	0.000000	2	368	203856
1904	1835	1322	184686	0.000000	4	369	207904
1905	1717	1164	176942	0.000000	4	369	201215
1906	1715	1238	179493	0.000000	4	368	203810
1907	1562	1129	179679	0.000000	4	368	203713
1908	1735	1146	179606	0.000000	4	370	203351
1909	1884	1236	183315	0.000000	4	370	206703
1910	1730	1195	163769	0.000000	0	370	188628
1911	1851	1334	163972	0.000000	0	370	188083
1912	1690	1194	163364	0.000000	0	370	186669
1913	1674	1153	165714	0.000000	0	370	188744
1914	2499	1819	164531	0.000000	0	370	187582
1915	2388	1672	70581	0.000000	0	370	88448
1916	2474	1726	68621	496.000000	0	370	83620
1917	2433	1666	93105	0.000000	0	370	109935
1918	2406	1684	95780	524.000000	0	370	112442
1919	2148	1543	84317	0.000000	0	370	102075
1920	2122	1540	85672	0.000000	0	370	102921
1921	2352	1706	178674	0.000000	0	370	200873
1922	2472	1669	175569	0.000000	0	369	198229
1923	2396	1676	174595	0.000000	0	368	197171
1924	2637	1608	177983	0.000000	0	367	199986
1925	2315	1501	172809	0.000000	0	368	195333
1926	2301	1550	170240	0.000000	0	368	192329
1927	2112	1484	156524	0.000000	0	368	178840
1928	3057	2167	158883	0.000000	0	367	180989
1929	2954	2062	160917	0.000000	0	367	182756
1930	2901	1947	161596	0.000000	0	367	183014
1931	2842	1886	155106	0.000000	0	367	178234
1932	2806	1912	159061	0.000000	0	367	181421
1933	2895	1858	154096	0.000000	0	368	177380
1934	2602	1735	154098	0.000000	0	368	177339
1935	2470	1784	169973	0.000000	0	367	193002
1936	2385	1591	90094	0.000000	0	366	106612
1937	2275	1580	91363	0.000000	0	366	108366
1938	2225	1520	87277	0.000000	0	367	104554
1939	2216	1502	86850	243.500000	0	367	103855
1940	2069	1440	76326	226.000000	0	367	93772
1941	1981	1388	77747	0.000000	0	367	95009
1942	2465	1812	80237	0.000000	0	367	97949
1943	2331	1681	160527	0.000000	0	368	180884
1944	2340	1651	154773	524.000000	0	368	175905
1945	2532	1783	142524	578.000000	0	366	164592
1946	2281	1648	152527	327.000000	0	366	173068
1947	2166	1480	152973	426.000000	0	366	173101
1948	1923	1428	157836	0.000000	0	366	177303
1949	1940	1413	153075	0.000000	0	366	177454
1950	2115	1404	153240	0.000000	0	366	173565
1951	1853	1456	159900	0.000000	0	366	182754
1952	2024	1509	157000	338.000000	0	367	180826
1953	2213	1414	157231	0.000000	0	367	179706
1954	1892	1402	149995	0.000000	0	367	174156
1955	1669	1269	165640	0.000000	0	367	190712
1956	1851	1371	167088	0.000000	0	367	192152
1957	1934	1436	160550	0.000000	0	367	187333
1958	1806	1360	158182	0.000000	0	366	190443
1959	1985	1430	161890	286.500000	0	366	193309
1960	2038	1501	164682	0.000000	0	366	196097

```
1961    1977  1498  164845  301.666667    0  367  195957
1962    2043  1560  166158  231.500000    0  367  197279
1963    2497  1752  147652    0.000000    0  367  177915
1964    2442  1705  148909    0.000000    0  367  179348
1965    2499  1745  146850    0.000000    0  367  177138
1966    2360  1579  150009    0.000000    0  367  178997
1967    2219  1566  144635    0.000000    0  367  174217
1968    2289  1658  145801    0.000000    0  367  174965
1969    2290  1567  144494    0.000000    0  367  173293
1970    3084  2259  146153    0.000000    0  367  175380
1971    3282  2221  145220    0.000000    0  367  174476
1972    2925  2035  144031    0.000000    3  367  173381
1973    2932  2052  147408  445.666667    0  368  175850
1974    2886  2023  148046    0.000000    3  368  176578
1975    2797  1925  147921    0.000000    3  368  176678
1976    2541  1668  163246    0.000000    3  367  192698
1977    3148  2001  164039    0.000000    3  367  193488
1978    2669  1789  164348  376.250000   13  368  193759
1979    2481  1783  161081    0.000000   11  370  190975
1980    2712  1814  160728  465.000000   12  370  190149
1981    2817  1859  145553  402.000000   16  371  174517
1982    2761  1755  140862    0.000000   16  372  170358
1983    2846  1926  157238  612.000000   16  372  187196
1984    4286  2937  137573  515.000000   15  373  167402
1985    4613  3254  135038    0.000000    5  373  164676
1986    3883  2792  137122    0.000000    5  373  166167
1987    3900  2840  138403    0.000000    0  374  166467
1988    4543  3380  141684  412.000000    0  374  168679
1989    4862  3657  140256  367.666667    0  375  167178
1990    4602  3414  136841    0.000000    0  375  163913
1991    4732  3487  137575    0.000000    0  375  164482
1992    4828  3448  135642    0.000000    0  376  162632
1993    4510  3231  129767  360.000000    0  377  157028
1994    4539  3324  141365    0.000000    0  378  170130
1995    4644  3429  139969    0.000000    0  379  168785
1996    3643  2700  135621    0.000000    0  380  164893
1997    4458  3301  132797    0.000000    0  381  161959
1998    5430  4137  131006  291.500000    2  381  160044
1999    5917  4513  128797    0.000000    2  381  157601
2000    6479  4782  117195    0.000000    2  381  143391
2001    5854  4571  116277    0.000000    2  381  142352
2002    5476  4283  114926  559.000000    2  382  140372
```

In [31]:

```
Data_n = pd.DataFrame()
Data_n.index = Data.index
Data_n['nSt'] = Data['S'].to_numpy() / Data['S'].to_numpy()[0]
Data_n['nLi'] = Data['L'].to_numpy() / Data['L'].to_numpy()[0]
Data_n['nInR'] = Data['IGR'].to_numpy() / Data['IGR'].to_numpy()[0]
Data_n['nTwF'] = Data['TWF'].to_numpy() / Data['TWF'].to_numpy()[0]
Data_n['nChC'] = Data['CPP'].to_numpy() / Data['CPP'].to_numpy()[0]
Data_n['nChE'] = Data['EGR'].to_numpy() / Data['EGR'].to_numpy()[0]
Data_n
```

Out[31]:

	nSt	nLi	nInR	nTwF	nChC	nChE
0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1	8.285714	1.000000	0.904698	1.000000	1.046179	1.066994
2	8.428571	1.142857	0.884116	1.004425	1.061623	1.074565
3	8.142857	0.571429	0.961074	1.004425	1.070980	1.089096
4	2.714286	0.714286	0.970022	1.004425	1.068574	1.084590
...	...	...	...	...	...	...

	nSt	nLi	nInR	nTwF	nChC	nChE
1998	775.714286	591.000000	1.565101	1.685841	0.343542	0.345249
1999	845.285714	644.714286	0.000000	1.685841	0.338298	0.339427
2000	925.571429	683.142857	0.000000	1.685841	0.307796	0.308852
2001	836.285714	653.000000	0.000000	1.685841	0.305566	0.306432
2002	782.285714	611.857143	3.001342	1.690265	0.301315	0.302872

2003 rows × 6 columns

In [32]:

```
Data
```

Out[32]:

	Date	Platform	DateWk	DayWk	Month	Year	S	L	TWF	TWR	EGR	CPP	IG
0	2020-03-04	Apple Music	Wednesday	4	3	2020	7	7	226	0	379454	465864	186.25000
1	2020-03-05	Apple Music	Thursday	5	3	2020	58	7	226	0	404875	487377	168.50000
2	2020-03-06	Apple Music	Friday	6	3	2020	59	8	227	0	407748	494572	164.66666
3	2020-03-07	Apple Music	Saturday	7	3	2020	57	4	227	0	413262	498931	179.00000
4	2020-03-08	Apple Music	Sunday	1	3	2020	19	5	227	0	411552	497810	180.66666
...	...	...	...	...	...	...	...	...	...	...	...	...	.
1998	2021-12-27	Spotify	Monday	2	12	2021	5430	4137	381	2	131006	160044	291.50000
1999	2021-12-28	Spotify	Tuesday	3	12	2021	5917	4513	381	2	128797	157601	0.00000
2000	2021-12-29	Spotify	Wednesday	4	12	2021	6479	4782	381	2	117195	143391	0.00000
2001	2021-12-30	Spotify	Thursday	5	12	2021	5854	4571	381	2	116277	142352	0.00000
2002	2021-12-31	Spotify	Friday	6	12	2021	5476	4283	382	2	114926	140372	559.00000

2003 rows × 14 columns

In [33]:

```
DataIn = Data[['Platform', 'Date']]
Data.index = pd.MultiIndex.from_frame(DataIn)
type(Data.index)
```

Out[33]:

```
pandas.core.indexes.multi.MultiIndex
```

```
# Checking the Correlation Coefficients on the above variables;
```

In [34]:

```
Data[['S','L','EGR','CPP','IGR','TWR','TWF']].corr()
```

Out[34]:

	S	L	EGR	CPP	IGR	TWR	TWF
S	1.000000	0.281764	-0.143763	-0.147351	0.075017	-0.022556	0.134985
L	0.281764	1.000000	-0.399742	-0.413992	0.054268	0.022768	0.493749
EGR	-0.143763	-0.399742	1.000000	0.993352	-0.082037	-0.152661	-0.676003
CPP	-0.147351	-0.413992	0.993352	1.000000	-0.105323	-0.164579	-0.703309
IGR	0.075017	0.054268	-0.082037	-0.105323	1.000000	0.218519	-0.011468
TWR	-0.022556	0.022768	-0.152661	-0.164579	0.218519	1.000000	-0.091411
TWF	0.134985	0.493749	-0.676003	-0.703309	-0.011468	-0.091411	1.000000

```
# Checking the length of the 'Panel' Dataset
```

In [35]:

```
Data_DSP = pd.DataFrame(Data[['S','L','EGR','CPP','IGR','TWR','TWF']])
Data_DSPk = Data_DSP.shape[0]
n_sample = int(Data_DSPk)
print(n_sample)
type(n_sample)
```

```
2003
```

Out[35]:

```
int
```

```
# Isolating the quantitative variables we are interested, isolated by DSP
```

In [36]:

```
AM = Data.query("Platform == 'Apple Music'")
SC = Data.query("Platform == 'SoundCloud'")
SP = Data.query("Platform == 'Spotify'")


AM_e = AM[['S','L','EGR','CPP','IGR','TWR','TWF']]
SC_e = SC[['S','L','EGR','CPP','IGR','TWR','TWF']]
SP_e = SP[['S','L','EGR','CPP','IGR','TWR','TWF']]
AM_e = AM[['S','L','EGR','CPP','IGR','TWR','TWF']]
SC_e = SC[['S','L','EGR','CPP','IGR','TWR','TWF']]
SP_e = SP[['S','L','EGR','CPP','IGR','TWR','TWF']]
```

```
# Checking the VIF's of 'Data_DSP'
```

In [37]:

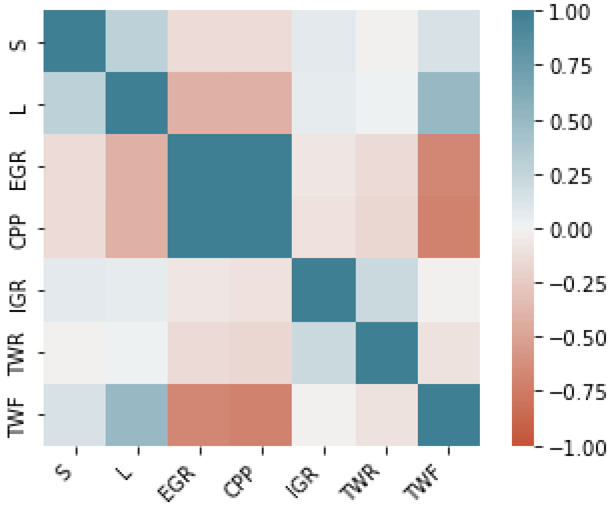
```
print(f"Corr. Coeff.'s \n{Data_DSP.corr()}\n")
corrData1 = Data_DSP.corr()
ax = sns.heatmap(
    corrData1,
    vmin=-1, vmax=1, center=0,
    cmap=sns.diverging_palette(20, 220, n=200),
    square=True)
ax.set_xticklabels(
    ax.get_xticklabels(),
    rotation=45,
    horizontalalignment='right')
```

```
Corr. Coeff.'s
            S         L       EGR       CPP       IGR       TWR       TWF
S    1.000000  0.281764 -0.143763 -0.147351  0.075017 -0.022556  0.134985
```

```
L     0.281764  1.000000 -0.399742 -0.413992  0.054268  0.022768  0.493749
EGR -0.143763 -0.399742  1.000000  0.993352 -0.082037 -0.152661 -0.676003
CPP -0.147351 -0.413992  0.993352  1.000000 -0.105323 -0.164579 -0.703309
IGR  0.075017  0.054268 -0.082037 -0.105323  1.000000  0.218519 -0.011468
TWR -0.022556  0.022768 -0.152661 -0.164579  0.218519  1.000000 -0.091411
TWF  0.134985  0.493749 -0.676003 -0.703309 -0.011468 -0.091411  1.000000
```

Out[37]:

```
[Text(0.5, 0, 'S'),
 Text(1.5, 0, 'L'),
 Text(2.5, 0, 'EGR'),
 Text(3.5, 0, 'CPP'),
 Text(4.5, 0, 'IGR'),
 Text(5.5, 0, 'TWR'),
 Text(6.5, 0, 'TWF')]
```



Plotting function:
Time-Series plot
Histogram
Autocorrelation Function(ACF)
Partial Autocorrelation Function(pACF)

In [38]:

```
# Source:
    # https://github.com/SimiY...
    # .../pydata-sf-2016-arima-tutorial/blob/master/Section_3_ARIMA_Modeling_tutorial.ipyı
#Choose the number of lags to display the sample ACF and PACF;
n_lag=25
#graph_title='Series 1'

# Writing a function to plot this specific sub-section of data;
#Make sure the tsplot() function is defined; #**Function: See dji_citi_sent file;
def tsplot(y1, y2, lags=None, title='', figsize=(14,8)):
    '''Examine the patterns of ACF and PACF, along with the time series plots and histogra

    Original source: https://tomaugspurger.github.io/modern-7-timeseries.html
    '''
    # Plotting layout
    fig = plt.figure(figsize=figsize)
    layout = (2,2)
    ts_ax   = plt.subplot2grid(layout, (0,0))
    hist_ax = plt.subplot2grid(layout, (0,1))
    acf_ax  = plt.subplot2grid(layout, (1,0))
    pacf_ax = plt.subplot2grid(layout, (1,1))

    #Plotting
    y1.plot(ax=ts_ax)
    ts_ax.set_title(title)
    y2.plot(ax=hist_ax, kind='hist', bins=25)
```

```python
    # Function for rotating tick labels on x-axis only
    for tick in ts_ax.get_xticklabels():
        tick.set_rotation(45)
    ts_ax.legend(loc='upper left')
    hist_ax.set_title('Histogram')
    smt.graphics.plot_acf(y2, lags=lags, ax=acf_ax)
    smt.graphics.plot_pacf(y2, lags=lags, ax=pacf_ax)
    [ax.set_xlim(0) for ax in [acf_ax, pacf_ax]]
    sns.despine()
    plt.tight_layout()

    return ts_ax, acf_ax, pacf_ax
```

Time-Series, Histogram, ACF and pACF isolated by DSP

In [39]:

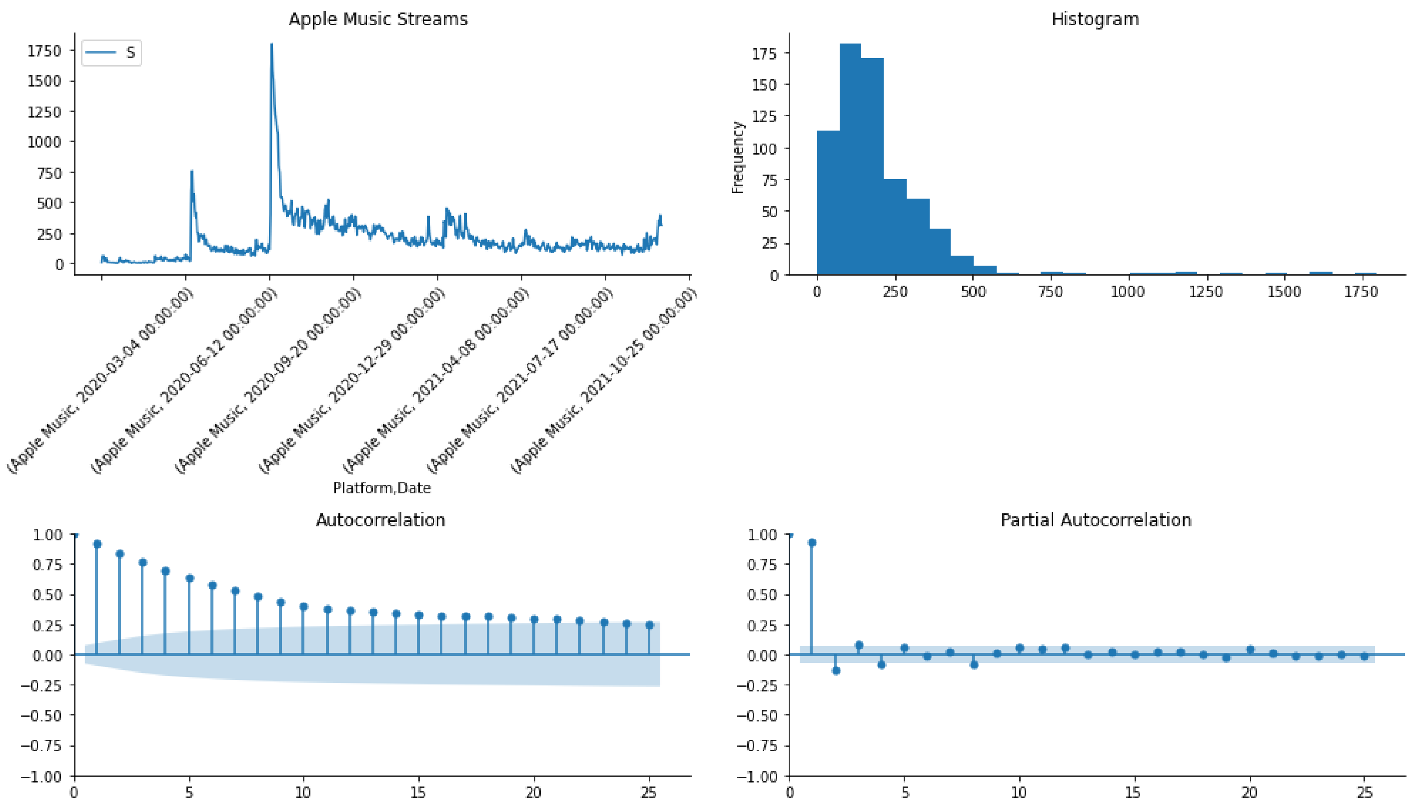
```python
#Run command;
tsplot(AM['S'],
       AM['S'],
       title='Apple Music Streams ' ,
       lags=n_lag)

#Run command;
tsplot(AM['S'],
       AM['S'],
       title='Apple Music Listeners ' ,
       lags=n_lag)

#Run command;
tsplot(SC['S'],
       SC['S'],
       title='SoundCloud Streams ' ,
       lags=n_lag)

#Run command;
tsplot(SC['S'],
       SC['S'],
       title='SoundCloud Listeners ' ,
       lags=n_lag)

#Run command;
tsplot(SP['S'],
       SP['S'],
       title='Spotify Streams ' ,
       lags=n_lag)

#Run command;
tsplot(SP['S'],
       SP['S'],
       title='Spotify Listeners ' ,
       lags=n_lag)
```

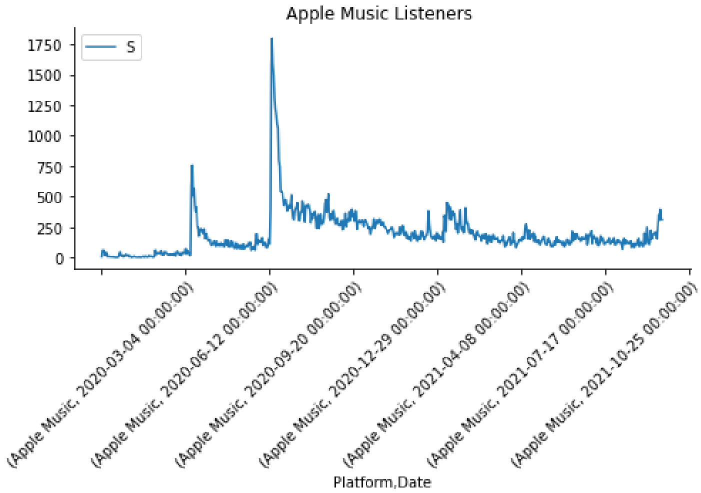
```
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/graphics/tsaplots.py:348: FutureWar
ning: The default method 'yw' can produce PACF values outside of the [-1,1] interval. Afte
r 0.13, the default will change tounadjusted Yule-Walker ('ywm'). You can use this method
now by setting method='ywm'.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/graphics/tsaplots.py:348: FutureWar
ning: The default method 'yw' can produce PACF values outside of the [-1,1] interval. Afte
r 0.13, the default will change tounadjusted Yule-Walker ('ywm'). You can use this method
now by setting method='ywm'.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/graphics/tsaplots.py:348: FutureWar
ning: The default method 'yw' can produce PACF values outside of the [-1,1] interval. Afte
```

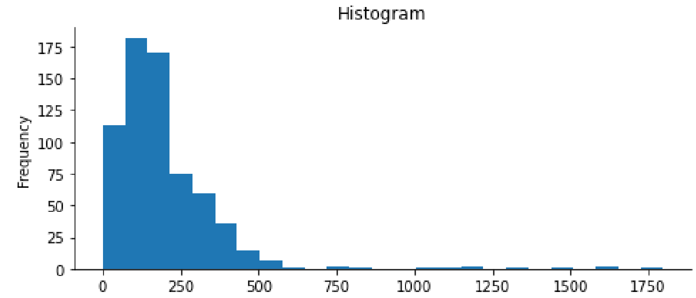
```
r 0.13, the default will change tounadjusted Yule-Walker ('ywm'). You can use this method
now by setting method='ywm'.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/graphics/tsaplots.py:348: FutureWar
ning: The default method 'yw' can produce PACF values outside of the [-1,1] interval. Afte
r 0.13, the default will change tounadjusted Yule-Walker ('ywm'). You can use this method
now by setting method='ywm'.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/graphics/tsaplots.py:348: FutureWar
ning: The default method 'yw' can produce PACF values outside of the [-1,1] interval. Afte
r 0.13, the default will change tounadjusted Yule-Walker ('ywm'). You can use this method
now by setting method='ywm'.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/graphics/tsaplots.py:348: FutureWar
ning: The default method 'yw' can produce PACF values outside of the [-1,1] interval. Afte
r 0.13, the default will change tounadjusted Yule-Walker ('ywm'). You can use this method
now by setting method='ywm'.
  warnings.warn(
```

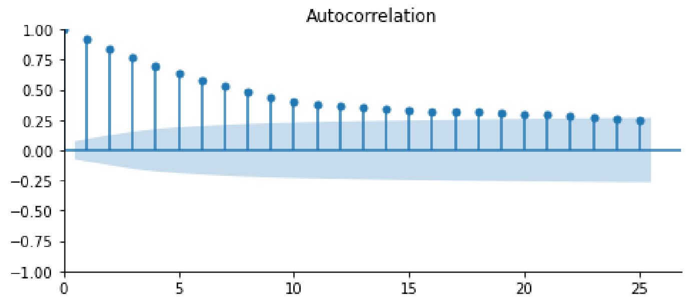
Out[39]:

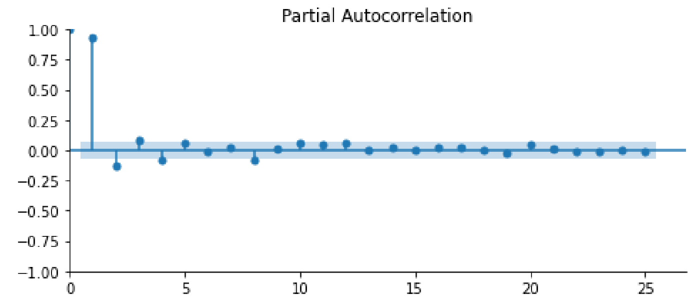
```
(<AxesSubplot:title={'center':'Spotify Listeners '}, xlabel='Platform,Date'>,
 <AxesSubplot:title={'center':'Autocorrelation'}>,
 <AxesSubplot:title={'center':'Partial Autocorrelation'}>)
```

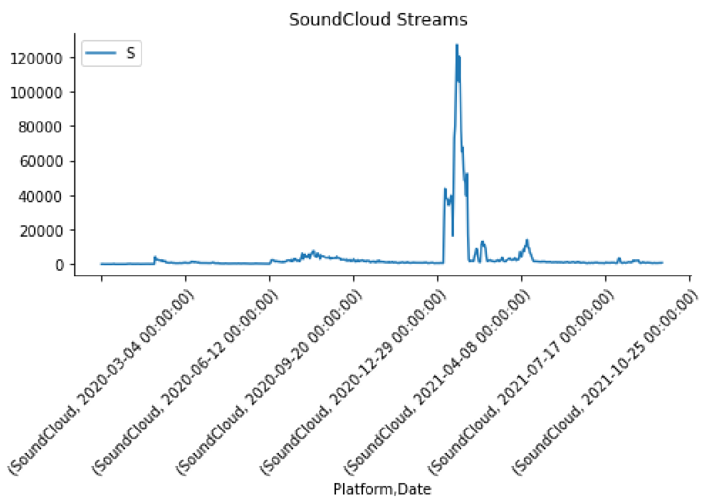


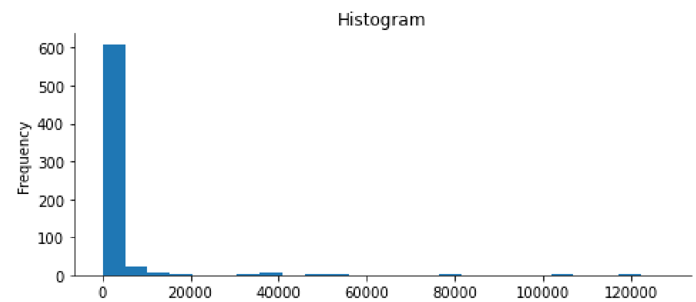

Apple Music Listeners
S
0
250
500
750
1000
1250
1500
1750
(Apple Music, 2020-03-04 00:00:00)
(Apple Music, 2020-06-12 00:00:00)
(Apple Music, 2020-09-20 00:00:00)
(Apple Music, 2020-12-29 00:00:00)
(Apple Music, 2021-04-08 00:00:00)
(Apple Music, 2021-07-17 00:00:00)
(Apple Music, 2021-10-25 00:00:00)
Platform,Date

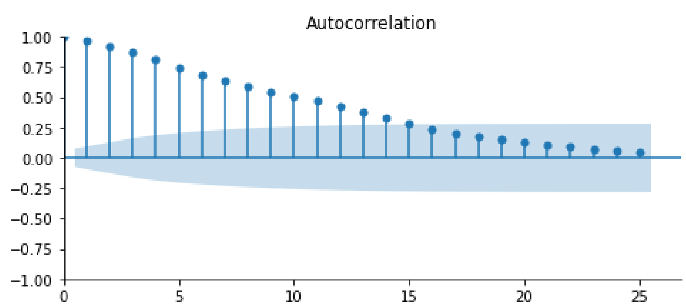

Histogram
Frequency
0
25
50
75
100
125
150
175
0
250
500
750
1000
1250
1500
1750

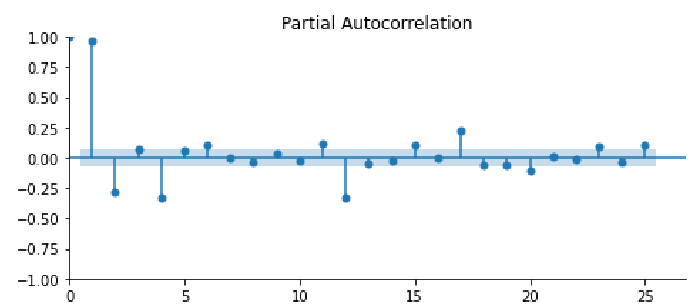

Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25

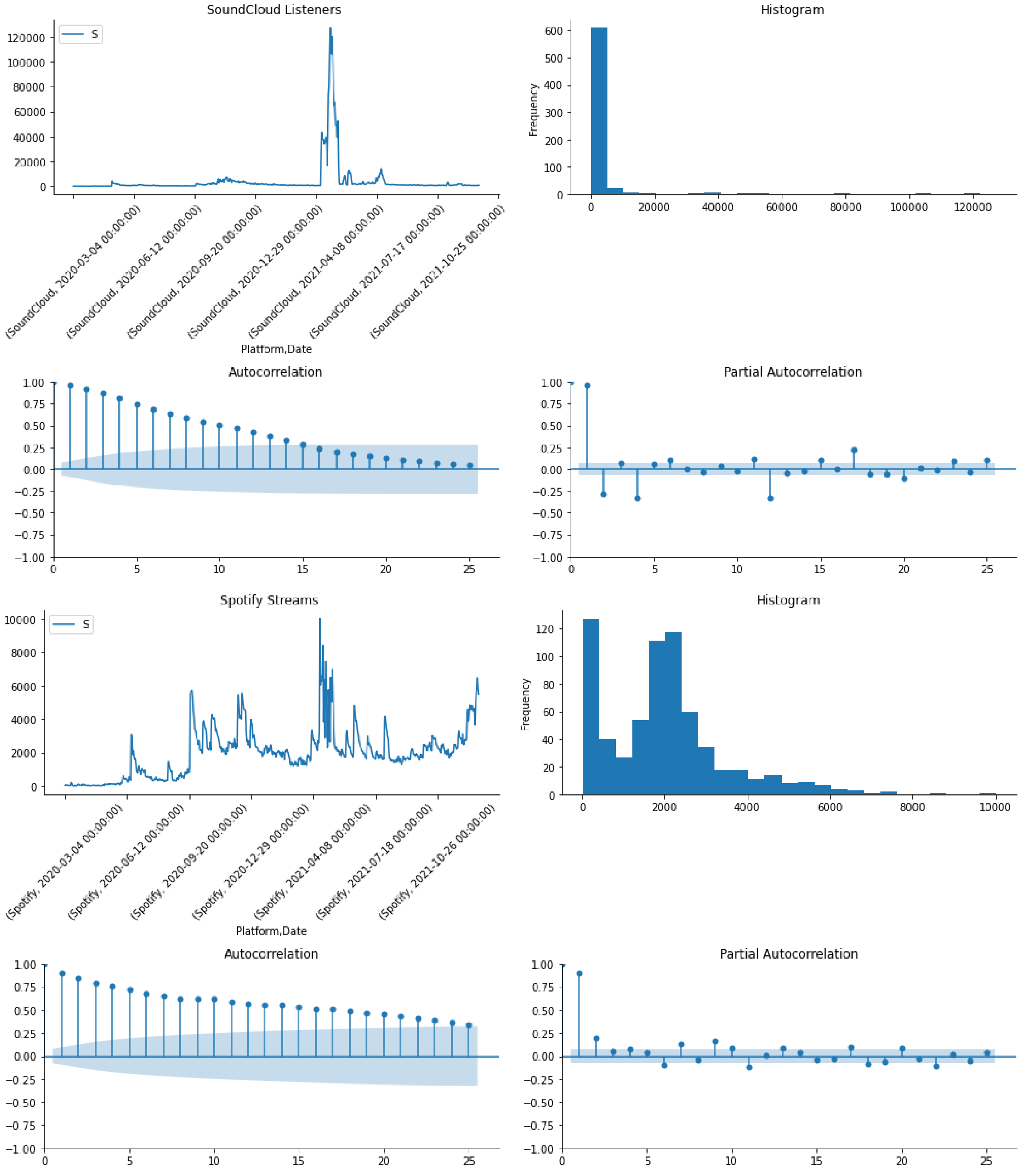

Partial Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25

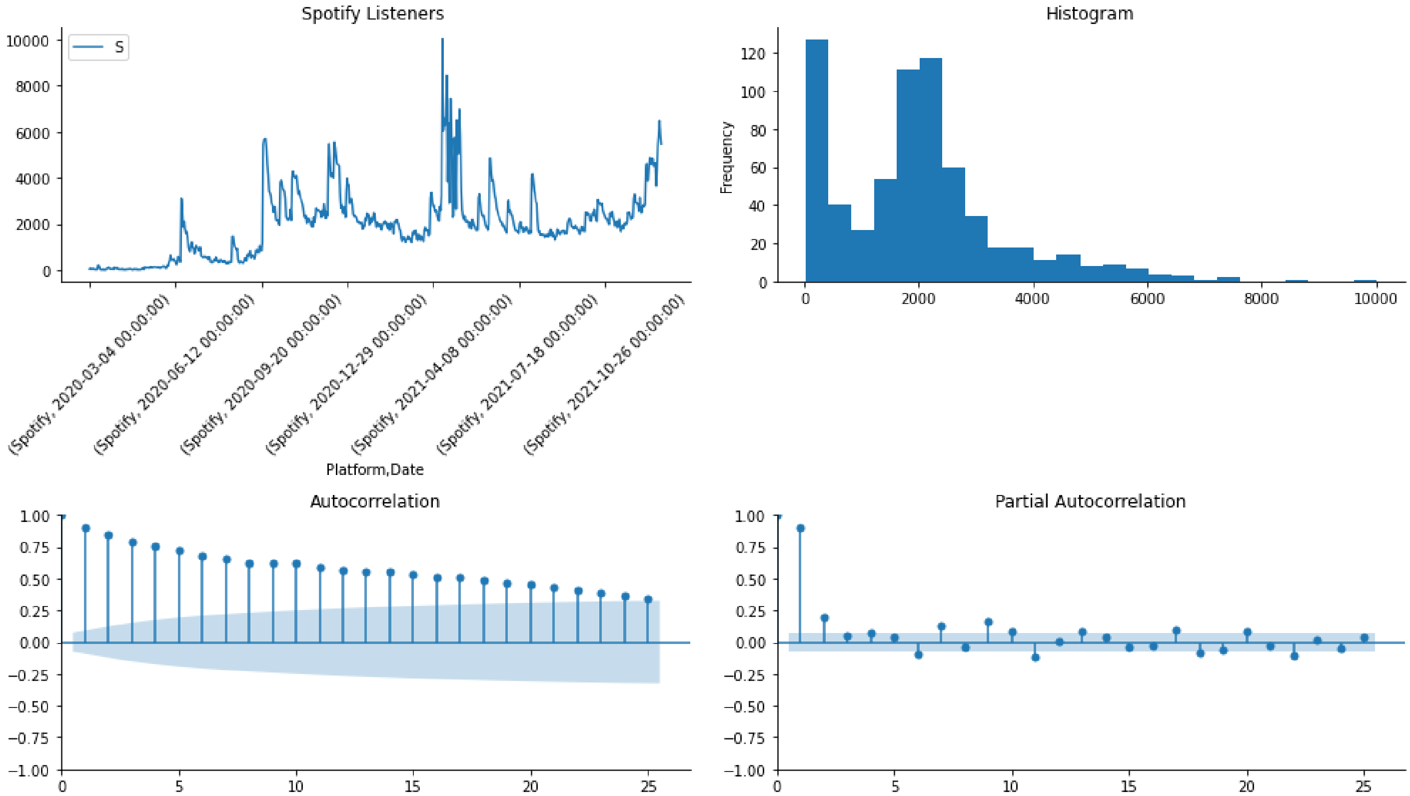

SoundCloud Streams
S
0
20000
40000
60000
80000
100000
120000
(SoundCloud, 2020-03-04 00:00:00)
(SoundCloud, 2020-06-12 00:00:00)
(SoundCloud, 2020-09-20 00:00:00)
(SoundCloud, 2020-12-29 00:00:00)
(SoundCloud, 2021-04-08 00:00:00)
(SoundCloud, 2021-07-17 00:00:00)
(SoundCloud, 2021-10-25 00:00:00)
Platform,Date


Histogram
Frequency
0
100
200
300
400
500
600
0
20000
40000
60000
80000
100000
120000


Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25


Partial Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25


SoundCloud Listeners
S
(SoundCloud, 2020-03-04 00:00:00)
(SoundCloud, 2020-06-12 00:00:00)
(SoundCloud, 2020-09-20 00:00:00)
(SoundCloud, 2020-12-29 00:00:00)
(SoundCloud, 2021-04-08 00:00:00)
(SoundCloud, 2021-07-17 00:00:00)
(SoundCloud, 2021-10-25 00:00:00)
Platform,Date
Histogram
Frequency
Autocorrelation
Partial Autocorrelation
Spotify Streams
S
(Spotify, 2020-03-04 00:00:00)
(Spotify, 2020-06-12 00:00:00)
(Spotify, 2020-09-20 00:00:00)
(Spotify, 2020-12-29 00:00:00)
(Spotify, 2021-04-08 00:00:00)
(Spotify, 2021-07-18 00:00:00)
(Spotify, 2021-10-26 00:00:00)
Platform,Date
Histogram
Frequency
Autocorrelation
Partial Autocorrelation

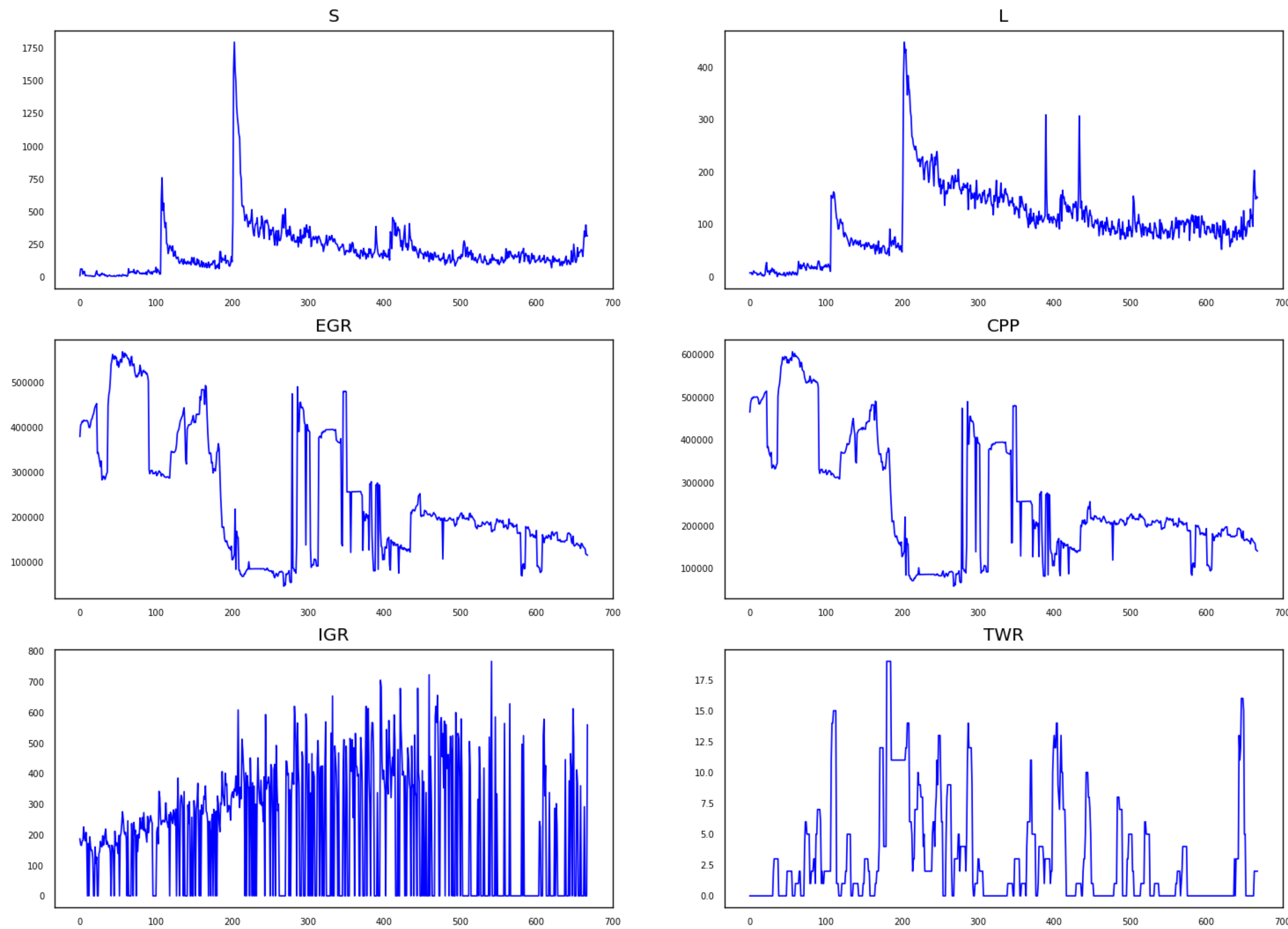


Time-Series plots of each variable we are interested in, isolated by DSP

In [40]:

```
fig, axes = plt.subplots(nrows=3, ncols=2, dpi=120, figsize=(15,11))
fig.suptitle('Apple Music:')
for i, ax in enumerate(axes.flatten()):
    data = AM_e[AM_e.columns[i]]
    ax.plot(data.to_numpy(), color='b', linewidth=1)
    # Decorations
    ax.set_title(AM_e.columns[i])
    ax.xaxis.set_ticks_position('none')
    ax.yaxis.set_ticks_position('none')
    ax.spines["top"].set_alpha(1)
    ax.tick_params(labelsize=6)
```

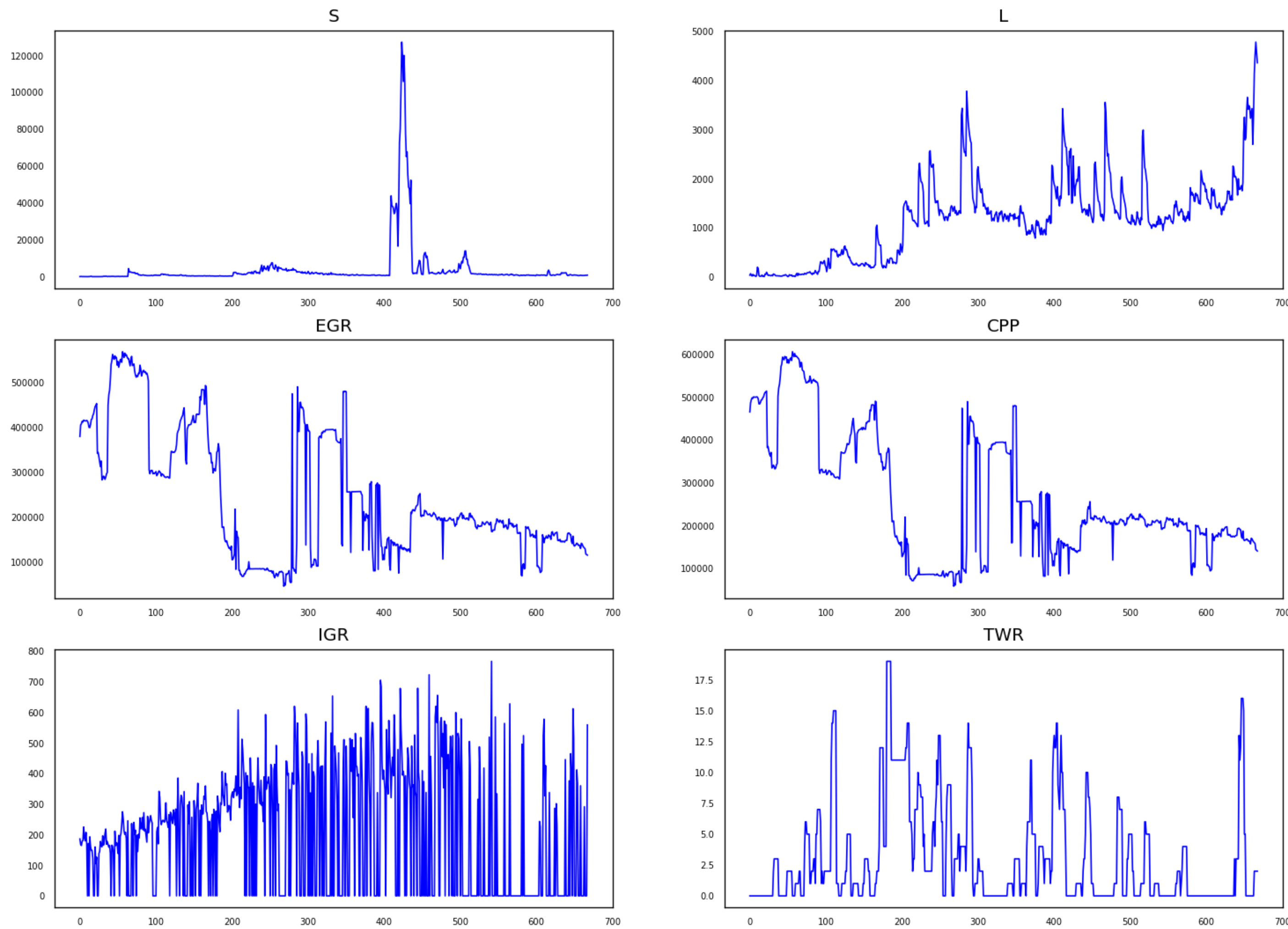
Apple Music:



In [41]:

```
fig, axes = plt.subplots(nrows=3, ncols=2, dpi=120, figsize=(15,11))
fig.suptitle('SoundCloud:')
for i, ax in enumerate(axes.flatten()):
    data = SC_e[SC_e.columns[i]]
    ax.plot(data.to_numpy(), color='b', linewidth=1)
    # Decorations
    ax.set_title(SC_e.columns[i])
    ax.xaxis.set_ticks_position('none')
    ax.yaxis.set_ticks_position('none')
    ax.spines["top"].set_alpha(1)
    ax.tick_params(labelsize=6)
```

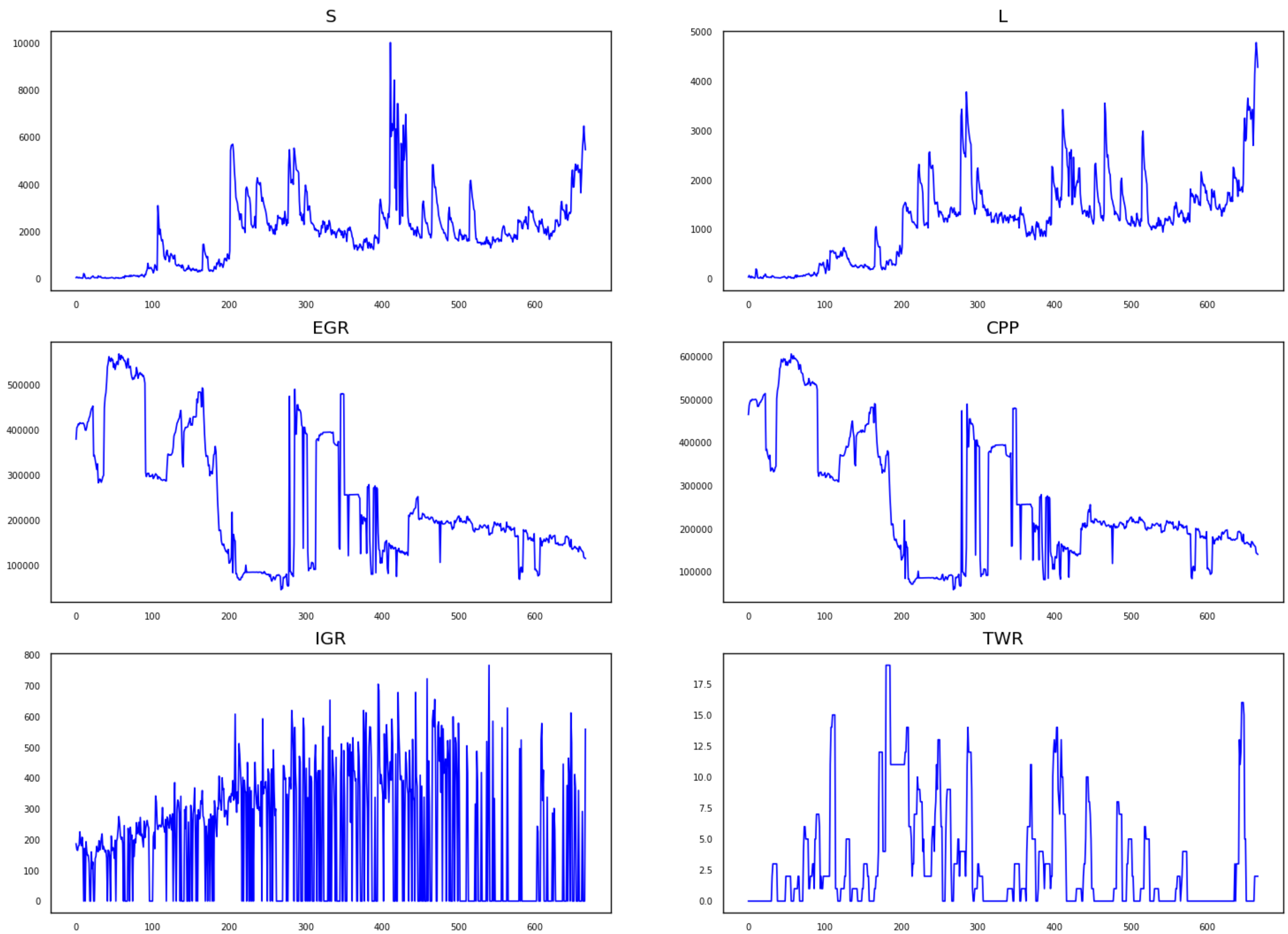
SoundCloud:



In [42]:

```
fig, axes = plt.subplots(nrows=3, ncols=2, dpi=120, figsize=(15,11))
fig.suptitle('Spotify:')
for i, ax in enumerate(axes.flatten()):
    data = SP_e[SP_e.columns[i]]
    ax.plot(data.to_numpy(), color='b', linewidth=1)
    # Decorations
    ax.set_title(SP_e.columns[i])
    ax.xaxis.set_ticks_position('none')
    ax.yaxis.set_ticks_position('none')
    ax.spines["top"].set_alpha(1)
    ax.tick_params(labelsize=6)
```

Spotify:

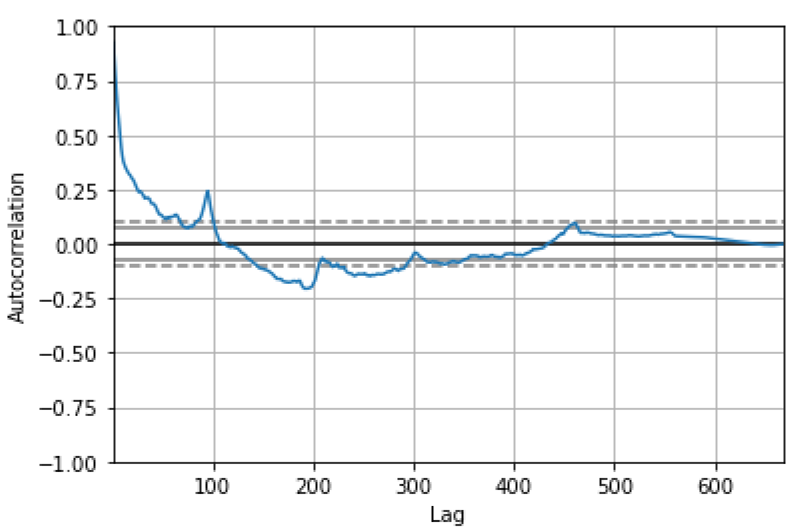



```
In [43]: from pandas.plotting import autocorrelation_plot
```

Apple Music

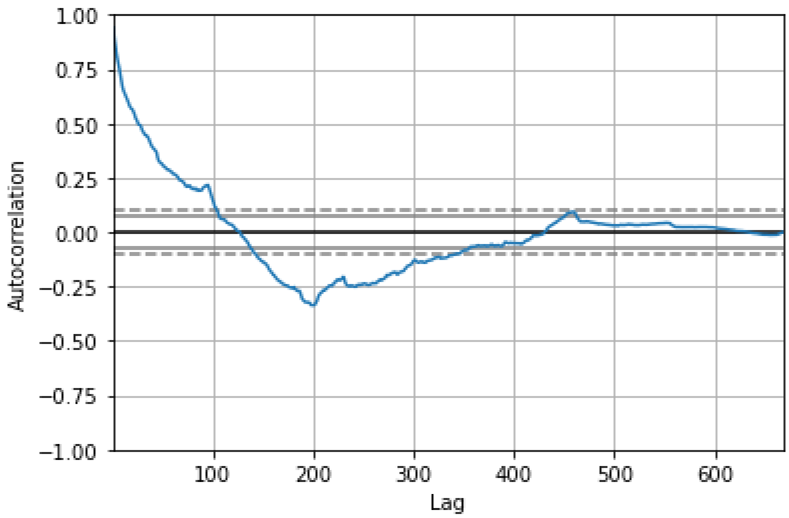
```
In [44]: autocorrelation_plot(AM_e['S'])
```

Out[44]: <AxesSubplot:xlabel='Lag', ylabel='Autocorrelation'>



In [45]:
```
autocorrelation_plot(AM_e['L'])
```

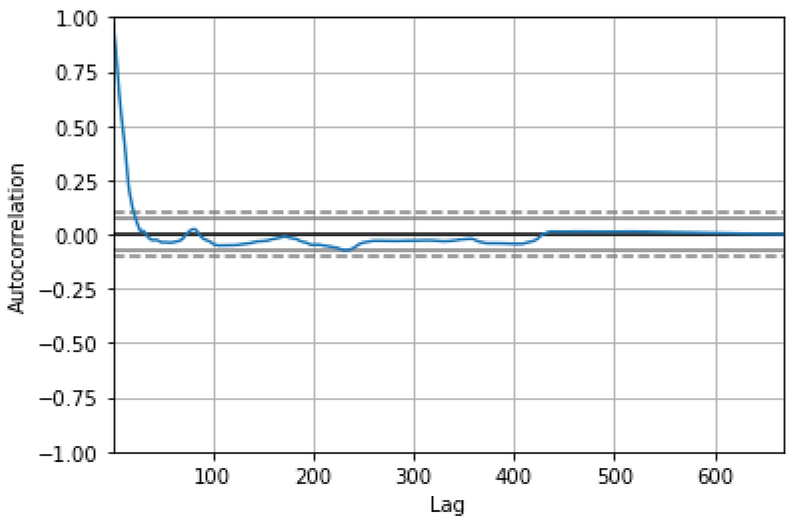
Out[45]: <AxesSubplot:xlabel='Lag', ylabel='Autocorrelation'>



SoundCloud:

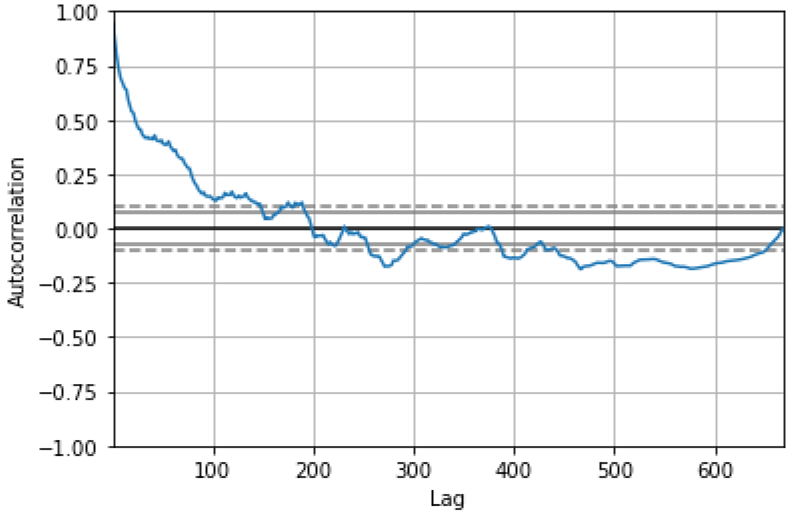
In [46]:
```
autocorrelation_plot(SC_e['S'])
```

Out[46]: <AxesSubplot:xlabel='Lag', ylabel='Autocorrelation'>



In [47]:
```
autocorrelation_plot(SC_e['L'])
```

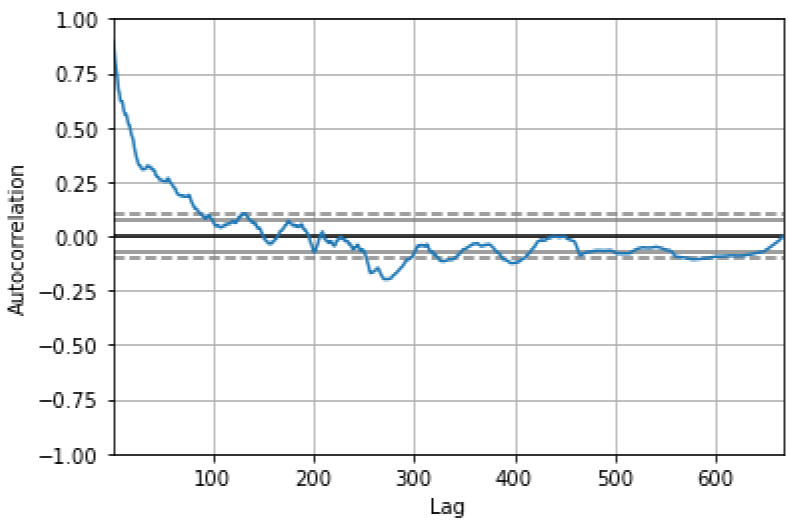
Out[47]: <AxesSubplot:xlabel='Lag', ylabel='Autocorrelation'>



Spotify:

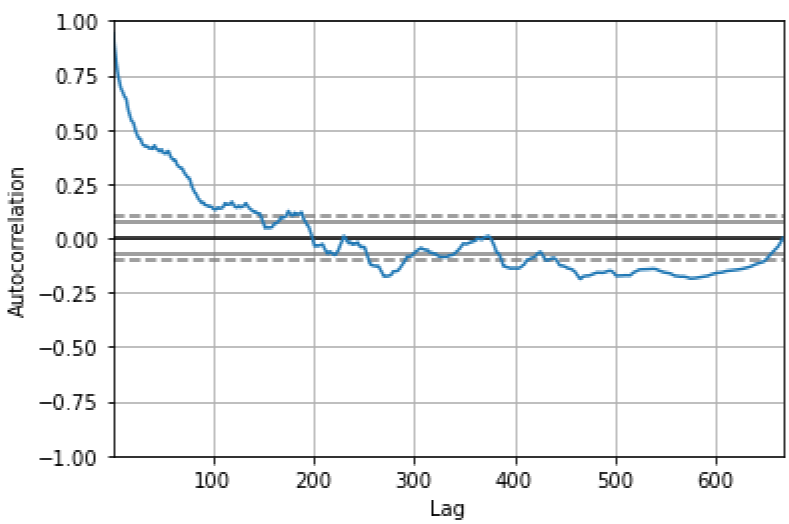
In [48]:
```
autocorrelation_plot(SP_e['S'])
```

Out[48]: <AxesSubplot:xlabel='Lag', ylabel='Autocorrelation'>



In [49]:
```
autocorrelation_plot(SP_e['L'])
```

Out[49]: <AxesSubplot:xlabel='Lag', ylabel='Autocorrelation'>

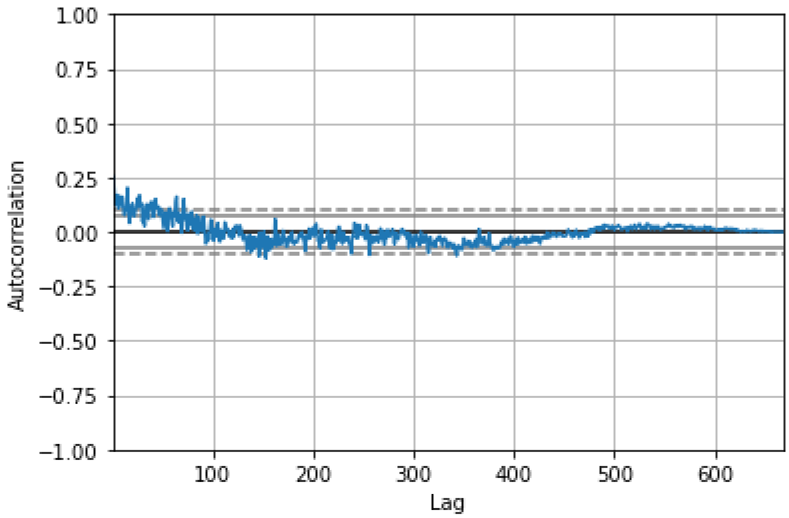


Panel Dataset structure: NOTE Social Media Metrics are the same for each DSP

In [50]:

```
autocorrelation_plot(AM_e['IGR'])
```

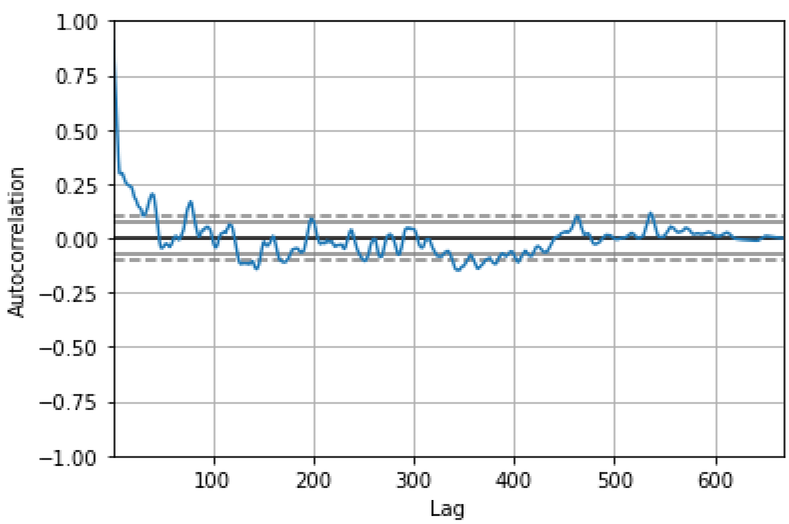
Out[50]: <AxesSubplot:xlabel='Lag', ylabel='Autocorrelation'>



In [51]:

```
autocorrelation_plot(AM_e['TWR'])
```

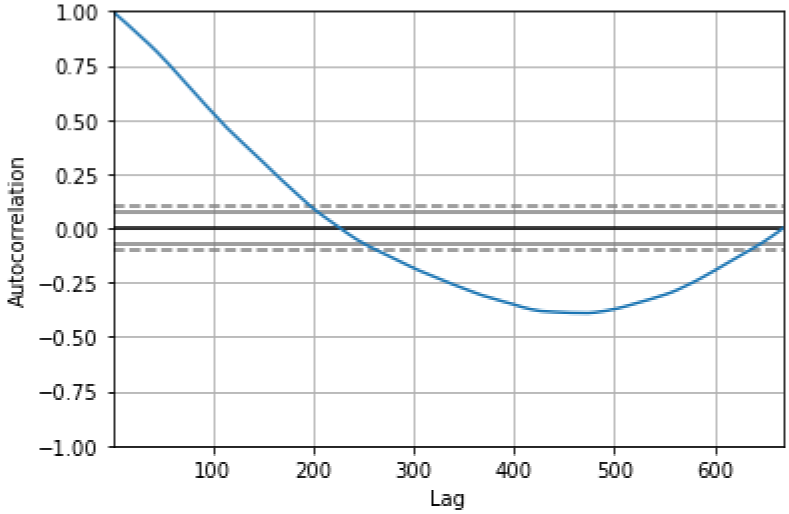
Out[51]: <AxesSubplot:xlabel='Lag', ylabel='Autocorrelation'>



In [52]:

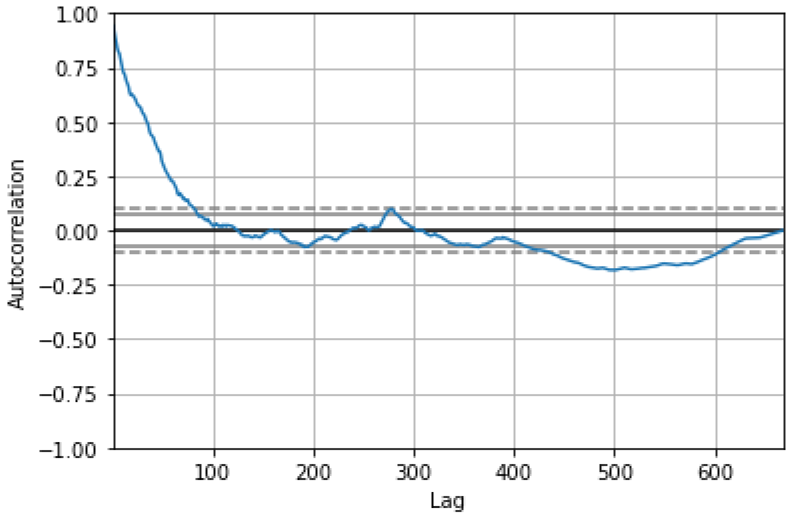
```
autocorrelation_plot(AM_e['TWF'])
```

Out[52]: <AxesSubplot:xlabel='Lag', ylabel='Autocorrelation'>



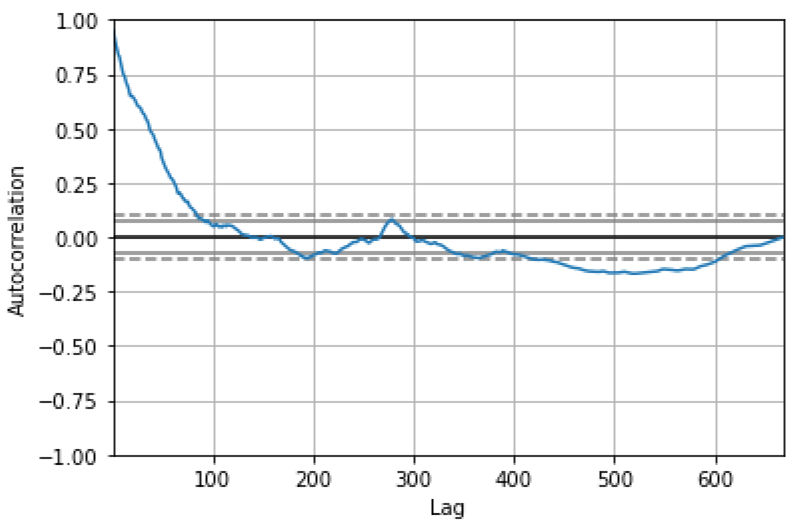
In [53]:
```
autocorrelation_plot(AM_e['EGR'])
```

Out[53]: <AxesSubplot:xlabel='Lag', ylabel='Autocorrelation'>



In [54]:
```
autocorrelation_plot(AM_e['CPP'])
```

Out[54]: <AxesSubplot:xlabel='Lag', ylabel='Autocorrelation'>




```
# Yule-Walker Estimates, lag order of 2, for each set of variables we are
interested in isolated by DSP
```

In [55]:
```
print("Yule-Walker Estimates: Apple Music")
for i,col in zip(AM_e, AM_e.columns):
    rho, sigma = yule_walker(AM_e[i], 2, method='mle')
    print(col)
    print(f'rho: {-rho}')
    print(f'sigma: {sigma}')
```

```
Yule-Walker Estimates: Apple Music
S
rho: [-1.04574226  0.12909676]
sigma: 69.6621848455076
L
rho: [-0.87424984 -0.07028027]
sigma: 22.831740135352174
EGR
rho: [-0.76309646 -0.191057  ]
sigma: 45830.689954129004
CPP
rho: [-0.76484551 -0.19378794]
sigma: 45325.426903731764
IGR
rho: [-0.22085666 -0.13416891]
sigma: 193.05862878050834
TWR
rho: [-1.01429734  0.12168587]
sigma: 1.7776190348604675
TWF
rho: [-0.99746586  0.00141916]
sigma: 4.4351325876760495
```

In [56]:
```
print("Yule-Walker Estimates: SoundCloud")
for i,col in zip(SC_e, SC_e.columns):
    rho, sigma = yule_walker(SC_e[i], 2, method='mle')
    print(col)
    print(f'rho: {-rho}')
    print(f'sigma: {sigma}')
```

```
Yule-Walker Estimates: SoundCloud
S
rho: [-1.22564128  0.26704937]
sigma: 3335.6806844446073
L
rho: [-1.0032068   0.06221035]
sigma: 277.7653067915043
EGR
rho: [-0.76309646 -0.191057  ]
sigma: 45830.689954129004
CPP
rho: [-0.76484551 -0.19378794]
sigma: 45325.426903731764
IGR
rho: [-0.22085666 -0.13416891]
sigma: 193.05862878050834
TWR
rho: [-1.01429734  0.12168587]
sigma: 1.7776190348604675
TWF
rho: [-0.99746586  0.00141916]
sigma: 4.4351325876760495
```

In [57]:
```
print("Yule-Walker Estimates: Spotify")
for i,col in zip(SP_e, SP_e.columns):
    rho, sigma = yule_walker(SP_e[i], 2, method='mle')
```

```
    print(col)
    print(f'rho: {-rho}')
    print(f'sigma: {sigma}')
```

```
Yule-Walker Estimates: Spotify
S
rho: [-0.72223545 -0.196242  ]
sigma: 638.4953807007424
L
rho: [-1.00661023  0.06534222]
sigma: 276.75936839691843
EGR
rho: [-0.76308747 -0.19106258]
sigma: 45864.53546384646
CPP
rho: [-0.76482398 -0.19380649]
sigma: 45358.48850991298
IGR
rho: [-0.22005651 -0.13336461]
sigma: 193.13465541977112
TWR
rho: [-1.01425106  0.12172991]
sigma: 1.7789202001689406
TWF
rho: [-0.99746069  0.00141601]
sigma: 4.437930995008709
```

Augmented Dickey-Fuller Test Results for each variable we are interested in, isolated by DSP

In [58]:

```
print("Augmented Dickey-Fuller Results, Apple Music")
for i,col in zip(AM_e, AM_e.columns):
    ad_fullerAM = adfuller(AM_e[i])
    print(col)
    print(f'ADF Statistic: {ad_fullerAM[0]}')
    print(f'p-value: {ad_fullerAM[1]}\n')
```

```
Augmented Dickey-Fuller Results, Apple Music
S
ADF Statistic: -4.937462452462523
p-value: 2.9468272780574956e-05

L
ADF Statistic: -3.7378466124229353
p-value: 0.0036112276226828335

EGR
ADF Statistic: -2.069552521285494
p-value: 0.2569325717213009

CPP
ADF Statistic: -2.146096882158832
p-value: 0.2263870939802673

IGR
ADF Statistic: -3.6998367388235267
p-value: 0.004115451139504546

TWR
ADF Statistic: -3.5622550808946536
p-value: 0.006525009362436266

TWF
ADF Statistic: -2.225054644810526
```

```
p-value: 0.1972780439230385
```

In [59]:

```python
print("Augmented Dickey-Fuller Results, SoundCloud")
for i,col in zip(SC_e, SC_e.columns):
    ad_fullerSC = adfuller(SC_e[i])
    print(col)
    print(f'ADF Statistic: {ad_fullerSC[0]}')
    print(f'p-value: {ad_fullerSC[1]}\n')
```

```
Augmented Dickey-Fuller Results, SoundCloud
S
ADF Statistic: -4.2523585136523545
p-value: 0.0005366019929134689

L
ADF Statistic: -0.9073730785243389
p-value: 0.7855346607055639

EGR
ADF Statistic: -2.069552521285494
p-value: 0.2569325717213009

CPP
ADF Statistic: -2.146096882158832
p-value: 0.2263870939802673

IGR
ADF Statistic: -3.6998367388235267
p-value: 0.004115451139504546

TWR
ADF Statistic: -3.5622550808946536
p-value: 0.006525009362436266

TWF
ADF Statistic: -2.225054644810526
p-value: 0.1972780439230385
```

In [60]:

```python
print("Augmented Dickey-Fuller Results, Spotify")
for i,col in zip(SP_e, SP_e.columns):
    ad_fullerSP = adfuller(SP_e[i])
    print(col)
    print(f'ADF Statistic: {ad_fullerSP[0]}')
    print(f'p-value: {ad_fullerSP[1]}\n')
```

```
Augmented Dickey-Fuller Results, Spotify
S
ADF Statistic: -2.2156345064134757
p-value: 0.20062004609886225

L
ADF Statistic: -0.9715781759237457
p-value: 0.7634888721856858

EGR
ADF Statistic: -2.068728698603671
p-value: 0.2572733859527792

CPP
ADF Statistic: -2.144844093972699
p-value: 0.22686885366652376
```

```
IGR
ADF Statistic: -3.760751013655978
p-value: 0.0033354010072072005

TWR
ADF Statistic: -3.562414955438831
p-value: 0.006521588745435436

TWF
ADF Statistic: -2.217067513736407
p-value: 0.20010935147968678
```

Granger's Causality Matrices for each variable we are interested in, isolated by DSP

In [ ]:

In [61]:

```
# Source:
    #https://www.machinelearningplus.com/time-series/vector-autoregression-examples-python
from statsmodels.tsa.stattools import grangercausalitytests
maxlag=12
test = 'ssr_chi2test'
def grangers_causation_matrix(data, variables, test='ssr_chi2test', verbose=False):
    """Check Granger Causality of all possible combinations of the Time series.
    The rows are the response variable, columns are predictors. The values in the table
    are the P-Values. P-Values lesser than the significance level (0.05), implies
    the Null Hypothesis that the coefficients of the corresponding past values is
    zero, that is, the X does not cause Y can be rejected.

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

In [62]:

```
print("Granger's Causality Matrix: Apple Music")
grangers_causation_matrix(AM_e, variables = AM_e.columns)
```

Granger's Causality Matrix: Apple Music

Out[62]:

	S_x	L_x	EGR_x	CPP_x	IGR_x	TWR_x	TWF_x
S_y	1.0000	0.0836	0.0077	0.0067	0.1089	0.0287	0.1952
L_y	0.0000	1.0000	0.0000	0.0000	0.2362	0.0823	0.0930
EGR_y	0.1012	0.0178	1.0000	0.0166	0.1076	0.0384	0.0001
CPP_y	0.0619	0.0084	0.0126	1.0000	0.0956	0.0469	0.0001
IGR_y	0.0002	0.0003	0.0604	0.0329	1.0000	0.0000	0.7974
TWR_y	0.0007	0.0111	0.4613	0.4696	0.0276	1.0000	0.3982

	S_x	L_x	EGR_x	CPP_x	IGR_x	TWR_x	TWF_x
TWF_y	0.0000	0.0000	0.0063	0.0019	0.0536	0.0000	1.0000

In [63]:

```
print("Granger's Causality Matrix: SoundCloud")
grangers_causation_matrix(SC_e, variables = SC_e.columns)
```

```
Granger's Causality Matrix: SoundCloud
```

Out[63]:

	S_x	L_x	EGR_x	CPP_x	IGR_x	TWR_x	TWF_x
S_y	1.0000	0.0009	0.1565	0.1749	0.0050	0.1190	0.5848
L_y	0.0051	1.0000	0.0001	0.0001	0.2209	0.4791	0.0001
EGR_y	0.6531	0.0000	1.0000	0.0166	0.1076	0.0384	0.0001
CPP_y	0.6233	0.0000	0.0126	1.0000	0.0956	0.0469	0.0001
IGR_y	0.0159	0.1431	0.0604	0.0329	1.0000	0.0000	0.7974
TWR_y	0.2680	0.2800	0.4613	0.4696	0.0276	1.0000	0.3982
TWF_y	0.0006	0.0260	0.0063	0.0019	0.0536	0.0000	1.0000

In [64]:

```
print("Granger's Causaility Matrix: Spotify")
grangers_causation_matrix(SP_e, variables = SP_e.columns)
```

```
Granger's Causaility Matrix: Spotify
```

Out[64]:

	S_x	L_x	EGR_x	CPP_x	IGR_x	TWR_x	TWF_x
S_y	1.0000	0.0000	0.0000	0.0000	0.0431	0.1447	0.0000
L_y	0.0003	1.0000	0.0001	0.0001	0.0090	0.4626	0.0001
EGR_y	0.0052	0.0000	1.0000	0.0167	0.1085	0.0385	0.0001
CPP_y	0.0042	0.0000	0.0127	1.0000	0.0958	0.0469	0.0001
IGR_y	0.0105	0.1431	0.0652	0.0320	1.0000	0.0000	0.8138
TWR_y	0.5505	0.2804	0.4615	0.4696	0.0283	1.0000	0.4007
TWF_y	0.0002	0.0258	0.0064	0.0019	0.0551	0.0000	1.0000

Johanon's Cointegration tests for each variable we are interested in, isolated by DSP

In [65]:

```
from statsmodels.tsa.vector_ar.vecm import coint_johansen

def cointegration_test(df, alpha=0.05):
    """Perform Johanson's Cointegration Test and Report Summary"""
    out = coint_johansen(df,-1,5)
    d = {'0.90':0, '0.95':1, '0.99':2}
    traces = out.lr1
    cvts = out.cvt[:, d[str(1-alpha)]]
    def adjust(val, length= 6): return str(val).ljust(length)

    # Summary
    print('Name   ::  Test Stat > C(95%)    =>   Signif  \n', '--'*20)
    for col, trace, cvt in zip(df.columns, traces, cvts):
        print(adjust(col), ':: ', adjust(round(trace,2), 9), ">", adjust(cvt, 8), ' =>  '
```

In [66]:
```python
print('Cointegration Test: \nApple Music\n')
cointegration_test(AM_e)
```

```
Cointegration Test:
Apple Music

Name   ::  Test Stat > C(95%)    =>   Signif
 ----------------------------------------
S      ::  333.59    > 111.7797  =>   True
L      ::  225.53    > 83.9383   =>   True
EGR    ::  129.45    > 60.0627   =>   True
CPP    ::  73.27     > 40.1749   =>   True
IGR    ::  37.54     > 24.2761   =>   True
TWR    ::  15.96     > 12.3212   =>   True
TWF    ::  6.29      > 4.1296    =>   True
```

In [67]:
```python
print('Cointegration Test: \nSoundCloud\n')
cointegration_test(SC_e)
```

```
Cointegration Test:
SoundCloud

Name   ::  Test Stat > C(95%)    =>   Signif
 ----------------------------------------
S      ::  245.87    > 111.7797  =>   True
L      ::  144.38    > 83.9383   =>   True
EGR    ::  85.34     > 60.0627   =>   True
CPP    ::  60.02     > 40.1749   =>   True
IGR    ::  36.7      > 24.2761   =>   True
TWR    ::  17.96     > 12.3212   =>   True
TWF    ::  5.49      > 4.1296    =>   True
```

In [68]:
```python
print('Cointegration Test: \nSpotify\n')
cointegration_test(SP_e)
```

```
Cointegration Test:
Spotify

Name   ::  Test Stat > C(95%)    =>   Signif
 ----------------------------------------
S      ::  267.35    > 111.7797  =>   True
L      ::  161.7     > 83.9383   =>   True
EGR    ::  102.59    > 60.0627   =>   True
CPP    ::  52.61     > 40.1749   =>   True
IGR    ::  31.66     > 24.2761   =>   True
TWR    ::  14.06     > 12.3212   =>   True
TWF    ::  3.44      > 4.1296    =>   False
```

Performing train/test splits on each set of variables we are interested in, isolated by DSP

In [69]:
```python
AM = Data.query("Platform == 'Apple Music'")
SC = Data.query("Platform == 'SoundCloud'")
SP = Data.query("Platform == 'Spotify'")
#AM.head(),SC.head(),SP.head()
AMk = AM.shape[0]
SCk = SC.shape[0]
SPk = SP.shape[0]


n_sampleAM = int(AMk)
n_sampleSC = int(SCk)
```

```python
n_sampleSP = int(SPk)

#print(n_sampleAM)
#print(n_sampleSC)
#print(n_sampleSP)

n_trainAM = int(0.90*n_sampleAM)+1
n_testAM = int(n_sampleAM - n_trainAM)+1
n_forecastAM = n_sampleAM - n_trainAM
print(f'Apple Music_Train: {n_trainAM}')
print(f'Apple Music_Test: {n_testAM}')
print(f'Apple Music_Forecast: {n_forecastAM}')

n_train = int(0.95* 2003)+1
n_forecast = n_sample - n_train
n_train

n_trainSC = int(0.90*n_sampleSC)+1
n_testSC = int(n_sampleSC - n_trainSC)+1
n_forecastSC = n_sampleSC - n_trainSC
print(f'SoundCloud_Train: {n_trainSC}')
print(f'SoundCloud_Test: {n_testSC}')
print(f'SoundCloud_Forecast: {n_forecastSC}')

n_trainSP = int(0.90*n_sampleSP)+1
n_testSP = int(n_sampleSP - n_trainSP)+1
n_forecastSP = n_sampleSP - n_trainSP
print(f'Spotify_Train: {n_trainSP}')
print(f'Spotify_Test: {n_testSP}')
print(f'Spotify_Forecast: {n_forecastSP}')

ts_trainAM = AM[['S','L','EGR','CPP','IGR','TWR','TWF']].iloc[:601]
ts_trainSC = SC[['S','L','EGR','CPP','IGR','TWR','TWF']].iloc[:601]
ts_trainSP = SP[['S','L','EGR','CPP','IGR','TWR','TWF']].iloc[:600]
ts_testAM = AM[['S','L','EGR','CPP','IGR','TWR','TWF']].iloc[601:]
ts_testSC = SC[['S','L','EGR','CPP','IGR','TWR','TWF']].iloc[601:]
ts_testSP = SP[['S','L','EGR','CPP','IGR','TWR','TWF']].iloc[600:]

#ts_df
ts_train = pd.concat([ts_trainAM, ts_trainSC, ts_trainSP])
ts_test = pd.concat([ts_testAM, ts_testSC, ts_testSP])

print(ts_train.shape)
print(ts_test.shape)
print(f'Train.head(): \n{ts_train.head()}\n')
print(f'Train.tail(): \n{ts_train.tail()}\n')
print(f'Test.head(): \n{ts_test.head()}\n')
print(f'Test.tail(): \n{ts_test.tail()}\n')
#ts_train = ts_train[['Streams','Listeners','EGR','IGReach','TWR','TWF']]
#ts_test = ts_test[['Streams','Listeners','EGR','IGReach','TWR','TWF']]
```

```
Apple Music_Train: 602
Apple Music_Test: 67
Apple Music_Forecast: 66
SoundCloud_Train: 602
SoundCloud_Test: 67
SoundCloud_Forecast: 66
Spotify_Train: 601
Spotify_Test: 67
Spotify_Forecast: 66
(1802, 7)
(201, 7)
Train.head():
                         S   L     EGR     CPP         IGR   TWR   TWF
Platform    Date
```

```
Apple Music 2020-03-04   7  7  379454  465864  186.250000    0  226
            2020-03-05  58  7  404875  487377  168.500000    0  226
            2020-03-06  59  8  407748  494572  164.666667    0  227
            2020-03-07  57  4  413262  498931  179.000000    0  227
            2020-03-08  19  5  411552  497810  180.666667    0  227

Train.tail():
                        S     L     EGR     CPP  IGR  TWR  TWF
Platform Date
Spotify  2021-10-21  2842  1886  155106  178234  0.0    0  367
         2021-10-22  2806  1912  159061  181421  0.0    0  367
         2021-10-23  2895  1858  154096  177380  0.0    0  368
         2021-10-24  2602  1735  154098  177339  0.0    0  368
         2021-10-25  2470  1784  169973  193002  0.0    0  367

Test.head():
                          S   L    EGR     CPP    IGR  TWR  TWF
Platform    Date
Apple Music 2021-10-26  126  96  90094  106612    0.0    0  366
            2021-10-27  127  92  91363  108366    0.0    0  366
            2021-10-28  147  96  87277  104554    0.0    0  367
            2021-10-29  161  93  86850  103855  243.5    0  367
            2021-10-30  113  84  76326   93772  226.0    0  367

Test.tail():
                        S     L     EGR     CPP    IGR  TWR  TWF
Platform Date
Spotify  2021-12-27  5430  4137  131006  160044  291.5    2  381
         2021-12-28  5917  4513  128797  157601    0.0    2  381
         2021-12-29  6479  4782  117195  143391    0.0    2  381
         2021-12-30  5854  4571  116277  142352    0.0    2  381
         2021-12-31  5476  4283  114926  140372  559.0    2  382
```

Performing Augmented Dickey-Fuller Tests on each time-series to check for unit roots, trend-stationarity

In [70]:

```python
def adfuller_test(series, signif=0.05, name='', verbose=False):
    """Perform ADFuller to test for Stationarity of given series and print report"""
    r = adfuller(series, autolag='AIC')
    output = {'test_statistic':round(r[0], 4), 'pvalue':round(r[1], 4), 'n_lags':round(r[2
    p_value = output['pvalue']
    def adjust(val, length= 6): return str(val).ljust(length)

    # Print Summary
    print(f'    Augmented Dickey-Fuller Test on "{name}"', "\n   ", '-'*47)
    print(f' Null Hypothesis: Data has unit root. Non-Stationary.')
    print(f' Significance Level    = {signif}')
    print(f' Test Statistic        = {output["test_statistic"]}')
    print(f' No. Lags Chosen       = {output["n_lags"]}')

    for key,val in r[4].items():
        print(f' Critical value {adjust(key)} = {round(val, 3)}')

    if p_value <= signif:
        print(f" => P-Value = {p_value}. Rejecting Null Hypothesis.")
        print(f" => Series is Stationary.")
    else:
        print(f" => P-Value = {p_value}. Weak evidence to reject the Null Hypothesis.")
        print(f" => Series is Non-Stationary.")
```

In [71]:

```python
print("Apple Music, Augmented Dickey-Fuller Test")
for name, column in ts_trainAM.iteritems():
```

```
    adfuller_test(column, name=column.name)
    print('\n')
```

```
Apple Music, Augmented Dickey-Fuller Test
    Augmented Dickey-Fuller Test on "S"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level    = 0.05
 Test Statistic        = -4.7232
 No. Lags Chosen       = 4
 Critical value 1%     = -3.441
 Critical value 5%     = -2.866
 Critical value 10%    = -2.569
 => P-Value = 0.0001. Rejecting Null Hypothesis.
 => Series is Stationary.


    Augmented Dickey-Fuller Test on "L"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level    = 0.05
 Test Statistic        = -3.5668
 No. Lags Chosen       = 2
 Critical value 1%     = -3.441
 Critical value 5%     = -2.866
 Critical value 10%    = -2.569
 => P-Value = 0.0064. Rejecting Null Hypothesis.
 => Series is Stationary.


    Augmented Dickey-Fuller Test on "EGR"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level    = 0.05
 Test Statistic        = -2.0469
 No. Lags Chosen       = 19
 Critical value 1%     = -3.442
 Critical value 5%     = -2.867
 Critical value 10%    = -2.569
 => P-Value = 0.2664. Weak evidence to reject the Null Hypothesis.
 => Series is Non-Stationary.


    Augmented Dickey-Fuller Test on "CPP"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level    = 0.05
 Test Statistic        = -2.083
 No. Lags Chosen       = 19
 Critical value 1%     = -3.442
 Critical value 5%     = -2.867
 Critical value 10%    = -2.569
 => P-Value = 0.2514. Weak evidence to reject the Null Hypothesis.
 => Series is Non-Stationary.


    Augmented Dickey-Fuller Test on "IGR"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level    = 0.05
 Test Statistic        = -3.1834
 No. Lags Chosen       = 17
 Critical value 1%     = -3.442
 Critical value 5%     = -2.867
 Critical value 10%    = -2.569
```

```
=> P-Value = 0.021. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "TWR"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -3.1575
No. Lags Chosen       = 14
Critical value 1%     = -3.442
Critical value 5%     = -2.866
Critical value 10%    = -2.569
=> P-Value = 0.0226. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "TWF"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -2.3205
No. Lags Chosen       = 0
Critical value 1%     = -3.441
Critical value 5%     = -2.866
Critical value 10%    = -2.569
=> P-Value = 0.1654. Weak evidence to reject the Null Hypothesis.
=> Series is Non-Stationary.
```

In [72]:

```python
print("SoundCloud, Augmented Dickey-Fuller Test")
for name, column in ts_trainSC.iteritems():
    adfuller_test(column, name=column.name)
    print('\n')
```

```
SoundCloud, Augmented Dickey-Fuller Test
    Augmented Dickey-Fuller Test on "S"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level    = 0.05
 Test Statistic        = -4.0453
 No. Lags Chosen       = 19
 Critical value 1%     = -3.442
 Critical value 5%     = -2.867
 Critical value 10%    = -2.569
 => P-Value = 0.0012. Rejecting Null Hypothesis.
 => Series is Stationary.


    Augmented Dickey-Fuller Test on "L"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level    = 0.05
 Test Statistic        = -2.2244
 No. Lags Chosen       = 15
 Critical value 1%     = -3.442
 Critical value 5%     = -2.866
 Critical value 10%    = -2.569
 => P-Value = 0.1975. Weak evidence to reject the Null Hypothesis.
 => Series is Non-Stationary.


    Augmented Dickey-Fuller Test on "EGR"
    -----------------------------------------------
```

```
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -2.0469
No. Lags Chosen       = 19
Critical value 1%     = -3.442
Critical value 5%     = -2.867
Critical value 10%    = -2.569
=> P-Value = 0.2664. Weak evidence to reject the Null Hypothesis.
=> Series is Non-Stationary.


   Augmented Dickey-Fuller Test on "CPP"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -2.083
No. Lags Chosen       = 19
Critical value 1%     = -3.442
Critical value 5%     = -2.867
Critical value 10%    = -2.569
=> P-Value = 0.2514. Weak evidence to reject the Null Hypothesis.
=> Series is Non-Stationary.


   Augmented Dickey-Fuller Test on "IGR"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -3.1834
No. Lags Chosen       = 17
Critical value 1%     = -3.442
Critical value 5%     = -2.867
Critical value 10%    = -2.569
=> P-Value = 0.021. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "TWR"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -3.1575
No. Lags Chosen       = 14
Critical value 1%     = -3.442
Critical value 5%     = -2.866
Critical value 10%    = -2.569
=> P-Value = 0.0226. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "TWF"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -2.3205
No. Lags Chosen       = 0
Critical value 1%     = -3.441
Critical value 5%     = -2.866
Critical value 10%    = -2.569
=> P-Value = 0.1654. Weak evidence to reject the Null Hypothesis.
=> Series is Non-Stationary.
```

In [73]:

```
print("Spotify, Augmented Dickey-Fuller Test")
```

```
for name, column in ts_trainSP.iteritems():
    adfuller_test(column, name=column.name)
    print('\n')
```

```
Spotify, Augmented Dickey-Fuller Test
    Augmented Dickey-Fuller Test on "S"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level     = 0.05
 Test Statistic         = -2.6058
 No. Lags Chosen        = 19
 Critical value 1%      = -3.442
 Critical value 5%      = -2.867
 Critical value 10%     = -2.569
 => P-Value = 0.0918. Weak evidence to reject the Null Hypothesis.
 => Series is Non-Stationary.


    Augmented Dickey-Fuller Test on "L"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level     = 0.05
 Test Statistic         = -2.2394
 No. Lags Chosen        = 15
 Critical value 1%      = -3.442
 Critical value 5%      = -2.867
 Critical value 10%     = -2.569
 => P-Value = 0.1923. Weak evidence to reject the Null Hypothesis.
 => Series is Non-Stationary.


    Augmented Dickey-Fuller Test on "EGR"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level     = 0.05
 Test Statistic         = -2.046
 No. Lags Chosen        = 19
 Critical value 1%      = -3.442
 Critical value 5%      = -2.867
 Critical value 10%     = -2.569
 => P-Value = 0.2668. Weak evidence to reject the Null Hypothesis.
 => Series is Non-Stationary.


    Augmented Dickey-Fuller Test on "CPP"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level     = 0.05
 Test Statistic         = -2.0816
 No. Lags Chosen        = 19
 Critical value 1%      = -3.442
 Critical value 5%      = -2.867
 Critical value 10%     = -2.569
 => P-Value = 0.252. Weak evidence to reject the Null Hypothesis.
 => Series is Non-Stationary.


    Augmented Dickey-Fuller Test on "IGR"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level     = 0.05
 Test Statistic         = -3.2211
 No. Lags Chosen        = 14
 Critical value 1%      = -3.442
 Critical value 5%      = -2.866
```

```
Critical value 10%     = -2.569
=> P-Value = 0.0188. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "TWR"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level     = 0.05
Test Statistic         = -3.1578
No. Lags Chosen        = 14
Critical value 1%      = -3.442
Critical value 5%      = -2.866
Critical value 10%     = -2.569
=> P-Value = 0.0226. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "TWF"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level     = 0.05
Test Statistic         = -2.3112
No. Lags Chosen        = 0
Critical value 1%      = -3.441
Critical value 5%      = -2.866
Critical value 10%     = -2.569
=> P-Value = 0.1684. Weak evidence to reject the Null Hypothesis.
=> Series is Non-Stationary.
```

Some of these Time-Series are not Stationary, let's take the difference of them and then check
ADF test results once more;

In [74]:

```
Data['lnD_St'] = np.log(Data['S']).diff(periods=1).bfill().ffill()
Data['lnD_Li'] = np.log(Data['L']).diff(periods=1).bfill().ffill()
Data['lnD_InR'] = np.log(Data['IGR']).diff(periods=1).bfill().ffill()
Data['lnD_TwF'] = np.log(Data['TWF']).diff(periods=1).bfill().ffill()
Data['lnD_ChC'] = np.log(Data['CPP']).diff(periods=1).bfill().ffill()
Data['lnD_ChE'] = np.log(Data['EGR']).diff(periods=1).bfill().ffill()


Data['D_St'] = Data['S'].diff(periods=1).bfill().ffill()
Data['D_Li'] = Data['L'].diff(periods=1).bfill().ffill()
Data['D_InR'] = Data['IGR'].diff(periods=1).bfill().ffill()
Data['D_TwF'] = Data['TWF'].diff(periods=1).bfill().ffill()
Data['D_ChC'] = Data['CPP'].diff(periods=1).bfill().ffill()
Data['D_ChE'] = Data['EGR'].diff(periods=1).bfill().ffill()

Data.replace([-np.inf, np.inf], np.nan, inplace=True)

ln_Dict = {'lnD_St': Data['lnD_St'],
           'lnD_Li': Data['lnD_Li'],
           'lnD_InR': Data['lnD_InR'],
           'lnD_TwF': Data['lnD_TwF'],
           'lnD_ChC': Data['lnD_ChC']}
ln_Data = pd.DataFrame(ln_Dict)
ln_Data.index = Data.index
ln_Data = ln_Data.bfill().ffill()
```

```
/opt/anaconda3/lib/python3.8/site-packages/pandas/core/arraylike.py:364: RuntimeWarning: d
```

```
ivide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

In [75]:

```
ln_Data
```

Out[75]:

Platform	Date	lnD_St	lnD_Li	lnD_InR	lnD_TwF	lnD_ChC
Apple Music	2020-03-04	2.114533	0.000000	-0.100154	0.000000	0.045144
	2020-03-05	2.114533	0.000000	-0.100154	0.000000	0.045144
	2020-03-06	0.017094	0.133531	-0.023013	0.004415	0.014655
	2020-03-07	-0.034486	-0.693147	0.083463	0.000000	0.008775
	2020-03-08	-1.098612	0.223144	0.009268	0.000000	-0.002249
...	...	...	...	...	...	...
Spotify	2021-12-27	0.197239	0.225745	-0.113847	0.000000	-0.011894
	2021-12-28	0.085890	0.086991	-0.113847	0.000000	-0.015382
	2021-12-29	0.090737	0.057897	-0.113847	0.000000	-0.094491
	2021-12-30	-0.101441	-0.045127	-0.113847	0.000000	-0.007272
	2021-12-31	-0.066750	-0.065078	-0.113847	0.002621	-0.014007

2003 rows × 5 columns

In [76]:

```
lnAM = ln_Data.query("Platform == 'Apple Music'")
lnAM = lnAM.bfill().ffill()
lnSC = ln_Data.query("Platform == 'SoundCloud'")
lnSC = lnSC.bfill().ffill()
lnSP = ln_Data.query("Platform == 'Spotify'")
lnSP = lnSP.bfill().ffill()
```

In [77]:

```
print("Apple Music, Augmented Dickey-Fuller Test")
for name, column in lnAM.iteritems():
    adfuller_test(column, name=column.name)
    print('\n')
```

```
Apple Music, Augmented Dickey-Fuller Test
    Augmented Dickey-Fuller Test on "lnD_St"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level    = 0.05
 Test Statistic        = -10.3603
 No. Lags Chosen       = 8
 Critical value 1%     = -3.44
 Critical value 5%     = -2.866
 Critical value 10%    = -2.569
 => P-Value = 0.0. Rejecting Null Hypothesis.
 => Series is Stationary.


    Augmented Dickey-Fuller Test on "lnD_Li"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level    = 0.05
 Test Statistic        = -9.0894
 No. Lags Chosen       = 11
```

```
Critical value 1%      = -3.44
Critical value 5%      = -2.866
Critical value 10%     = -2.569
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "lnD_InR"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level     = 0.05
Test Statistic         = -5.5832
No. Lags Chosen        = 3
Critical value 1%      = -3.44
Critical value 5%      = -2.866
Critical value 10%     = -2.569
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "lnD_TwF"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level     = 0.05
Test Statistic         = -4.2629
No. Lags Chosen        = 19
Critical value 1%      = -3.44
Critical value 5%      = -2.866
Critical value 10%     = -2.569
=> P-Value = 0.0005. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "lnD_ChC"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level     = 0.05
Test Statistic         = -8.4517
No. Lags Chosen        = 18
Critical value 1%      = -3.44
Critical value 5%      = -2.866
Critical value 10%     = -2.569
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.
```

In [78]:

```
print("SoundCloud, Augmented Dickey-Fuller Test")
for name, column in lnSC.iteritems():
    adfuller_test(column, name=column.name)
    print('\n')
```

```
SoundCloud, Augmented Dickey-Fuller Test
    Augmented Dickey-Fuller Test on "lnD_St"
    -----------------------------------------------
 Null Hypothesis: Data has unit root. Non-Stationary.
 Significance Level     = 0.05
 Test Statistic         = -15.5951
 No. Lags Chosen        = 3
 Critical value 1%      = -3.44
 Critical value 5%      = -2.866
 Critical value 10%     = -2.569
 => P-Value = 0.0. Rejecting Null Hypothesis.
 => Series is Stationary.
```

```
   Augmented Dickey-Fuller Test on "lnD_Li"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level     = 0.05
Test Statistic         = -5.9224
No. Lags Chosen        = 20
Critical value 1%      = -3.44
Critical value 5%      = -2.866
Critical value 10%     = -2.569
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "lnD_InR"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level     = 0.05
Test Statistic         = -5.6107
No. Lags Chosen        = 3
Critical value 1%      = -3.44
Critical value 5%      = -2.866
Critical value 10%     = -2.569
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "lnD_TwF"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level     = 0.05
Test Statistic         = -4.7729
No. Lags Chosen        = 19
Critical value 1%      = -3.44
Critical value 5%      = -2.866
Critical value 10%     = -2.569
=> P-Value = 0.0001. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "lnD_ChC"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level     = 0.05
Test Statistic         = -8.4138
No. Lags Chosen        = 18
Critical value 1%      = -3.44
Critical value 5%      = -2.866
Critical value 10%     = -2.569
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.
```

In [79]:

```
print("Spotify, Augmented Dickey-Fuller Test")
for name, column in lnSP.iteritems():
    adfuller_test(column, name=column.name)
    print('\n')
```

```
Spotify, Augmented Dickey-Fuller Test
   Augmented Dickey-Fuller Test on "lnD_St"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level     = 0.05
Test Statistic         = -12.5365
```

```
No. Lags Chosen       = 7
Critical value 1%     = -3.44
Critical value 5%     = -2.866
Critical value 10%    = -2.569
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "lnD_Li"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -6.2093
No. Lags Chosen       = 20
Critical value 1%     = -3.441
Critical value 5%     = -2.866
Critical value 10%    = -2.569
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "lnD_InR"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -9.645
No. Lags Chosen       = 1
Critical value 1%     = -3.44
Critical value 5%     = -2.866
Critical value 10%    = -2.569
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "lnD_TwF"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -4.7606
No. Lags Chosen       = 19
Critical value 1%     = -3.44
Critical value 5%     = -2.866
Critical value 10%    = -2.569
=> P-Value = 0.0001. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "lnD_ChC"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -8.4037
No. Lags Chosen       = 18
Critical value 1%     = -3.44
Critical value 5%     = -2.866
Critical value 10%    = -2.569
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.
```

```
# NOTE: The log-differenced data appears to be more stationary across the
board as compared to the earlier
Augmented Dickey-Fuller Results of the non-transformed data.
```

Apple Music, VAR Estimation:

In [80]:

```
# make a VAR model
# Vector Autoregression Model
# VAR(p) processes

# See Wooldridge p. 626 for a discussion of VAR models.
# It should be noted that much financial timeseries data are non-stationary.
# Before any analysis it is a good idea to plot the data, test for unit roots, and probab
# construct the log-differences in the data to obtain % return that are almost always sta
# In the code above, I have constructed the log-differenced data for the etf funds.

# We are interested in modeling a T×K
# multivariate time series Y, where T denotes the number of observations
# and K the number of variables. One way of estimating relationships
# between the time series and their lagged values is the vector autoregression process:
# Yt = ν + A1 Yt−1 + … + Ap Yt−p + ut  where ut∼Normal(0,Σu) and A is a K×K coefficient ma
# see https://www.statsmodels.org/dev/vector_ar.html

var_modelAM = VAR(lnAM)
resultsAM = var_modelAM.fit()
print(f'resultsAM.summary():\n{resultsAM.summary()}\n')

resultsAM.plot()
resultsAM.plot_acorr()
```

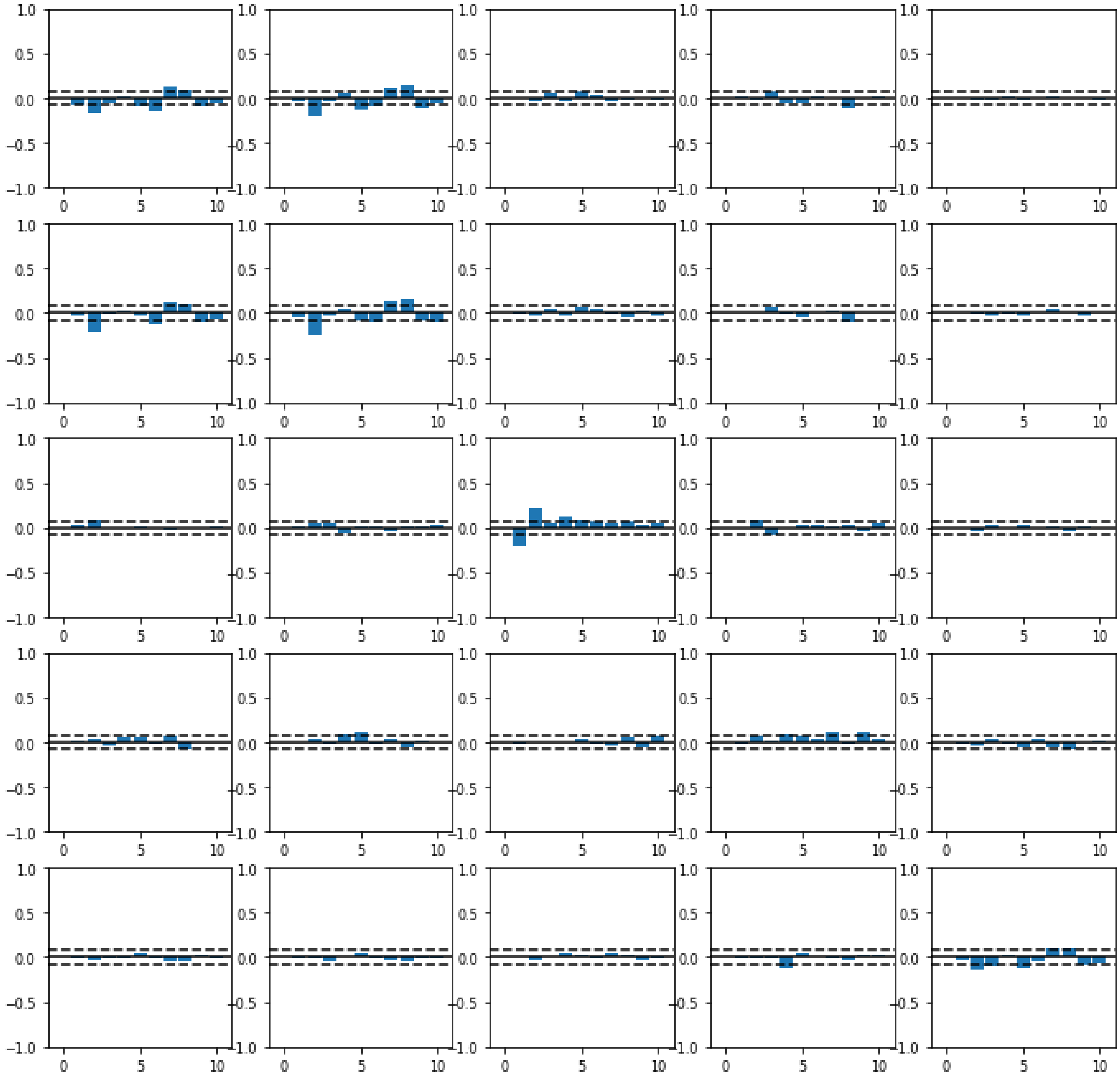
```
resultsAM.summary():
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Wed, 09, Feb, 2022
Time:                     14:46:59
--------------------------------------------------------------------
No. of Equations:         5.00000    BIC:                  -22.9775
Nobs:                     667.000    HQIC:                 -23.1016
Log likelihood:           3028.38    FPE:               8.57122e-11
AIC:                     -23.1800    Det(Omega_mle):    8.19590e-11
--------------------------------------------------------------------
Results for equation lnD_St
==============================================================================
                coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------
const              0.003497         0.015544            0.225           0.822
L1.lnD_St         -0.166957         0.059296           -2.816           0.005
L1.lnD_Li          0.018056         0.077844            0.232           0.817
L1.lnD_InR        -0.070031         0.053276           -1.314           0.189
L1.lnD_TwF        -0.115503         4.854209           -0.024           0.981
L1.lnD_ChC        -0.055748         0.066990           -0.832           0.405
==============================================================================

Results for equation lnD_Li
==============================================================================
                coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------
const              0.001406         0.011985            0.117           0.907
L1.lnD_St          0.035700         0.045720            0.781           0.435
L1.lnD_Li         -0.242497         0.060021           -4.040           0.000
L1.lnD_InR        -0.046572         0.041078           -1.134           0.257
L1.lnD_TwF        -0.360971         3.742806           -0.096           0.923
L1.lnD_ChC        -0.082395         0.051652           -1.595           0.111
==============================================================================

Results for equation lnD_InR
==============================================================================
```

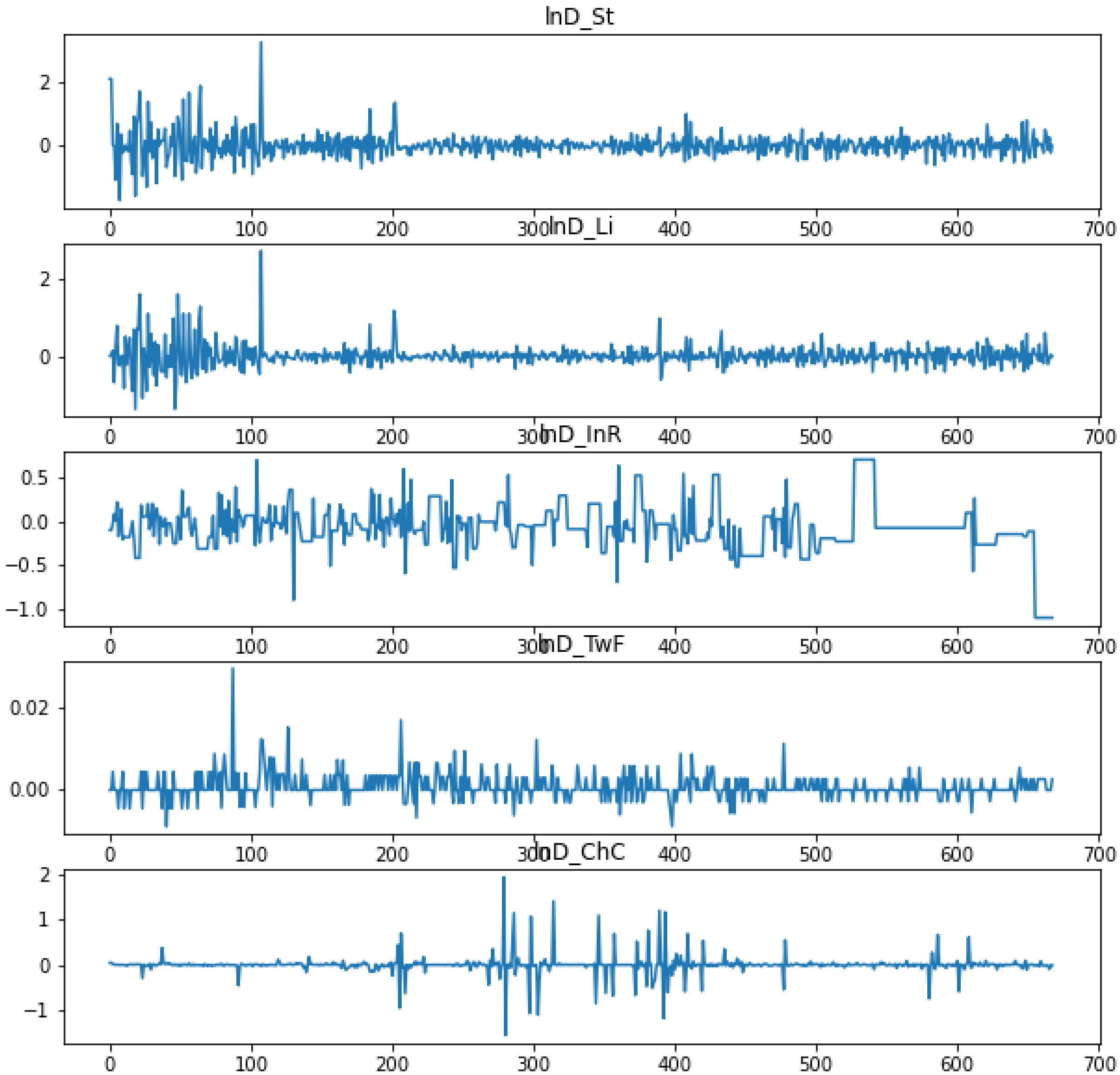
```
                 coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------
const              -0.030563         0.008976           -3.405           0.001
L1.lnD_St          -0.046902         0.034241           -1.370           0.171
L1.lnD_Li           0.039511         0.044952            0.879           0.379
L1.lnD_InR          0.625537         0.030765           20.333           0.000
L1.lnD_TwF          3.223425         2.803107            1.150           0.250
L1.lnD_ChC          0.031032         0.038684            0.802           0.422
==============================================================================

Results for equation lnD_TwF
==============================================================================
                 coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------
const               0.000715         0.000124            5.745           0.000
L1.lnD_St           0.000630         0.000475            1.329           0.184
L1.lnD_Li          -0.000215         0.000623           -0.345           0.730
L1.lnD_InR         -0.000476         0.000426           -1.116           0.265
L1.lnD_TwF          0.039666         0.038846            1.021           0.307
L1.lnD_ChC         -0.000898         0.000536           -1.675           0.094
==============================================================================

Results for equation lnD_ChC
==============================================================================
                 coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------
const              -0.003299         0.008763           -0.376           0.707
L1.lnD_St           0.006802         0.033430            0.203           0.839
L1.lnD_Li           0.005668         0.043887            0.129           0.897
L1.lnD_InR         -0.008013         0.030036           -0.267           0.790
L1.lnD_TwF          0.534147         2.736715            0.195           0.845
L1.lnD_ChC         -0.242485         0.037768           -6.420           0.000
==============================================================================

Correlation matrix of residuals
             lnD_St    lnD_Li   lnD_InR   lnD_TwF   lnD_ChC
lnD_St     1.000000  0.798351 -0.012173  0.098076  0.031564
lnD_Li     0.798351  1.000000  0.004503  0.097270  0.026902
lnD_InR   -0.012173  0.004503  1.000000  0.046996 -0.006647
lnD_TwF    0.098076  0.097270  0.046996  1.000000 -0.008008
lnD_ChC    0.031564  0.026902 -0.006647 -0.008008  1.000000
```

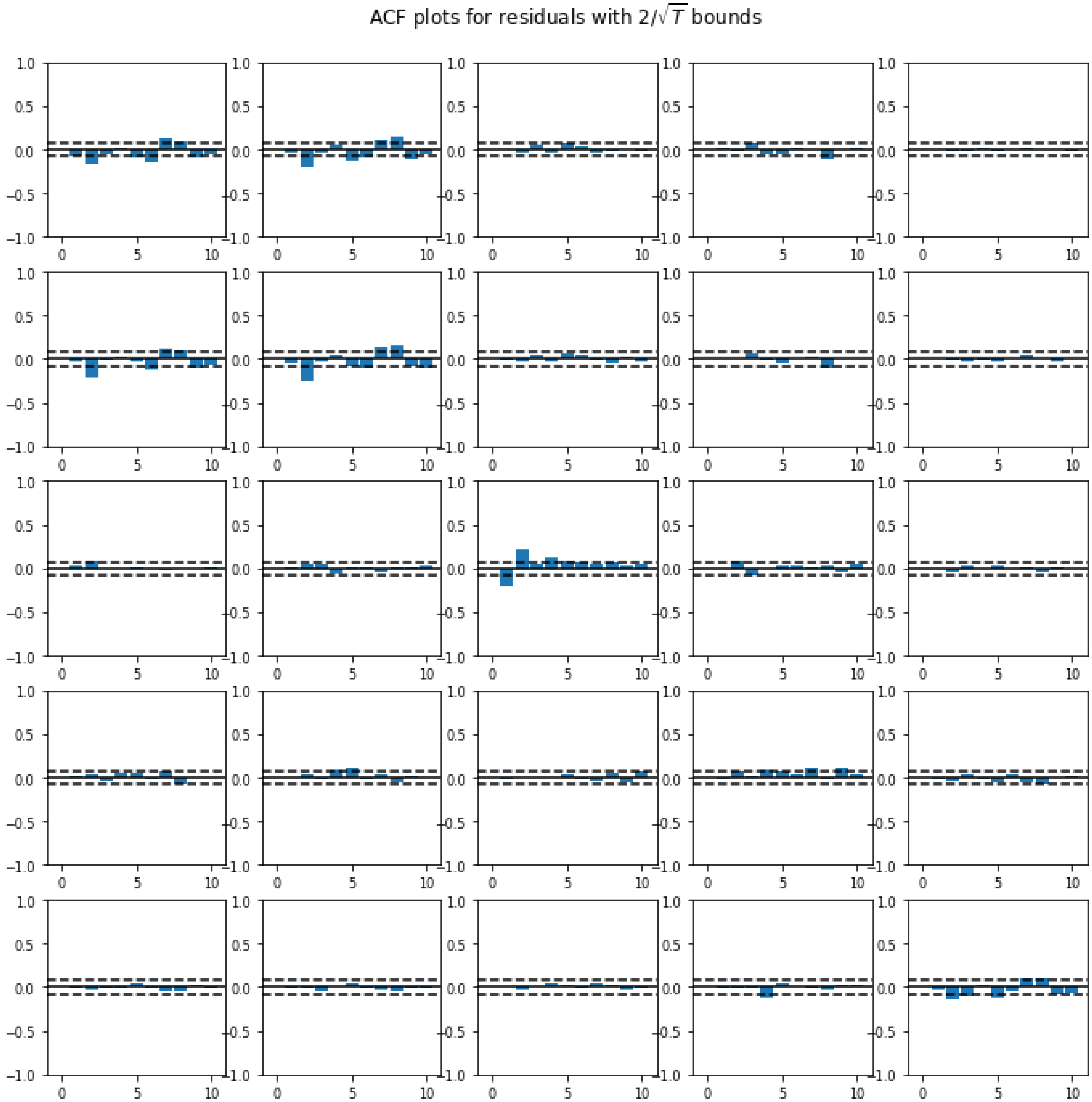
```
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/base/tsa_model.py:590: ValueWar
ning: An unsupported index was provided and will be ignored when e.g. forecasting.
  warnings.warn('An unsupported index was provided and will be'
```

Out[80]:

ACF plots for residuals with $2/\sqrt{T}$ bounds

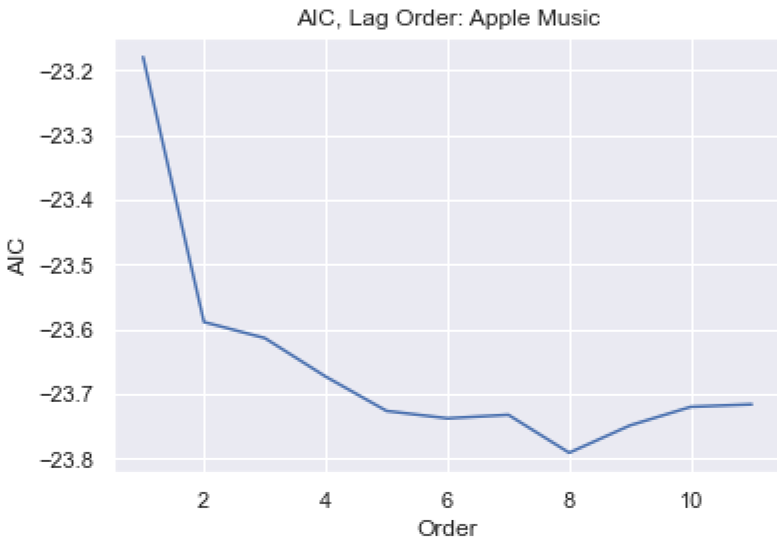



lnD_St
lnD_Li
lnD_InR
lnD_TwF
lnD_ChC



In [84]:

```
resultsAM_aic = []
for p in range(1,12):
    resultsAM = var_modelAM.fit(p)
    resultsAM_aic.append(resultsAM.aic)


import seaborn as sns
sns.set()
plt.title('AIC, Lag Order: Apple Music')
plt.plot(list(np.arange(1,12,1)), resultsAM_aic)
plt.xlabel("Order")
plt.ylabel("AIC")
plt.show()
```



Lowest AIC occurs at approx. (Order8, AIC−23.8)

In [85]:

```
# Lag order selection
# Choice of lag order can be a difficult problem.
# Standard analysis employs likelihood test or information criteria-based order selection.
# We have implemented the latter, accessible through the VAR class:

# When calling the fit function, one can pass a maximum number of lags
# and the order criterion to use for order selection:
var_modelAM.select_order(8)
resultsAMs = var_modelAM.fit(maxlags=8, ic='aic')
print(f'resultsAMs.summary():\n{resultsAMs.summary()}\n')
```

```
resultsAMs.summary():
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Wed, 09, Feb, 2022
Time:                     14:48:24
--------------------------------------------------------------------
No. of Equations:         5.00000    BIC:                   -22.8448
Nobs:                     663.000    HQIC:                  -23.3848
Log likelihood:           3291.54    FPE:                4.96376e-11
AIC:                     -23.7265    Det(Omega_mle):     4.09527e-11
--------------------------------------------------------------------
Results for equation lnD_St
==============================================================================
                coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------
const              0.002207         0.015930            0.139           0.890
L1.lnD_St         -0.333742         0.071592           -4.662           0.000
L1.lnD_Li          0.093105         0.090096            1.033           0.301
L1.lnD_InR        -0.164055         0.067968           -2.414           0.016
L1.lnD_TwF         4.500472         4.581397            0.982           0.326
L1.lnD_ChC        -0.071057         0.065664           -1.082           0.279
L2.lnD_St         -0.086975         0.076947           -1.130           0.258
L2.lnD_Li         -0.209766         0.097334           -2.155           0.031
L2.lnD_InR         0.002514         0.073343            0.034           0.973
L2.lnD_TwF         1.562049         4.558150            0.343           0.732
L2.lnD_ChC        -0.046135         0.068127           -0.677           0.498
L3.lnD_St         -0.004492         0.076646           -0.059           0.953
L3.lnD_Li         -0.178880         0.098269           -1.820           0.069
L3.lnD_InR         0.062961         0.075723            0.831           0.406
L3.lnD_TwF        11.803717         4.579774            2.577           0.010
```

L3.lnD_ChC	-0.071754	0.067955	-1.056	0.291
L4.lnD_St	-0.067689	0.067001	-1.010	0.312
L4.lnD_Li	0.035470	0.088317	0.402	0.688
L4.lnD_InR	-0.071479	0.072264	-0.989	0.323
L4.lnD_TwF	-5.580060	4.585804	-1.217	0.224
L4.lnD_ChC	0.001710	0.067858	0.025	0.980
L5.lnD_St	0.057861	0.057849	1.000	0.317
L5.lnD_Li	-0.219685	0.077895	-2.820	0.005
L5.lnD_InR	0.138306	0.068117	2.030	0.042
L5.lnD_TwF	-8.143492	4.611793	-1.766	0.077
L5.lnD_ChC	-0.034884	0.065308	-0.534	0.593

Results for equation lnD_Li

	coefficient	std. error	t-stat	prob
const	0.000471	0.012631	0.037	0.970
L1.lnD_St	0.136474	0.056766	2.404	0.016
L1.lnD_Li	-0.439898	0.071437	-6.158	0.000
L1.lnD_InR	-0.099410	0.053892	-1.845	0.065
L1.lnD_TwF	2.397368	3.632606	0.660	0.509
L1.lnD_ChC	-0.090818	0.052065	-1.744	0.081
L2.lnD_St	0.129453	0.061011	2.122	0.034
L2.lnD_Li	-0.450109	0.077176	-5.832	0.000
L2.lnD_InR	-0.014431	0.058154	-0.248	0.804
L2.lnD_TwF	2.618021	3.614173	0.724	0.469
L2.lnD_ChC	-0.039884	0.054018	-0.738	0.460
L3.lnD_St	0.117919	0.060773	1.940	0.052
L3.lnD_Li	-0.293311	0.077918	-3.764	0.000
L3.lnD_InR	0.019771	0.060041	0.329	0.742
L3.lnD_TwF	7.516511	3.631319	2.070	0.038
L3.lnD_ChC	-0.057304	0.053882	-1.064	0.288
L4.lnD_St	0.040298	0.053125	0.759	0.448
L4.lnD_Li	-0.108158	0.070027	-1.545	0.122
L4.lnD_InR	-0.030438	0.057299	-0.531	0.595
L4.lnD_TwF	-1.332743	3.636101	-0.367	0.714
L4.lnD_ChC	-0.028440	0.053805	-0.529	0.597
L5.lnD_St	0.149388	0.045869	3.257	0.001
L5.lnD_Li	-0.259054	0.061763	-4.194	0.000
L5.lnD_InR	0.111860	0.054010	2.071	0.038
L5.lnD_TwF	-4.212507	3.656707	-1.152	0.249
L5.lnD_ChC	-0.050251	0.051783	-0.970	0.332

Results for equation lnD_InR

	coefficient	std. error	t-stat	prob
const	-0.018119	0.009286	-1.951	0.051
L1.lnD_St	-0.012425	0.041730	-0.298	0.766
L1.lnD_Li	0.025246	0.052516	0.481	0.631
L1.lnD_InR	0.396740	0.039618	10.014	0.000
L1.lnD_TwF	3.609518	2.670440	1.352	0.176
L1.lnD_ChC	0.010000	0.038275	0.261	0.794
L2.lnD_St	0.106742	0.044851	2.380	0.017
L2.lnD_Li	-0.082794	0.056735	-1.459	0.144
L2.lnD_InR	0.303278	0.042751	7.094	0.000
L2.lnD_TwF	5.818522	2.656889	2.190	0.029
L2.lnD_ChC	-0.039724	0.039711	-1.000	0.317
L3.lnD_St	-0.011275	0.044676	-0.252	0.801
L3.lnD_Li	0.062792	0.057280	1.096	0.273
L3.lnD_InR	0.030037	0.044138	0.681	0.496
L3.lnD_TwF	-5.401066	2.669494	-2.023	0.043
L3.lnD_ChC	0.015569	0.039610	0.393	0.694
L4.lnD_St	0.032930	0.039054	0.843	0.399

```
L4.lnD_Li        -0.048070         0.051479         -0.934           0.350
L4.lnD_InR        0.059594         0.042122          1.415           0.157
L4.lnD_TwF       -2.689583         2.673009         -1.006           0.314
L4.lnD_ChC        0.017476         0.039554          0.442           0.659
L5.lnD_St         0.055518         0.033720          1.646           0.100
L5.lnD_Li        -0.055779         0.045404         -1.229           0.219
L5.lnD_InR        0.028861         0.039705          0.727           0.467
L5.lnD_TwF        0.676601         2.688157          0.252           0.801
L5.lnD_ChC        0.018827         0.038067          0.495           0.621
=========================================================================
```

```
Results for equation lnD_TwF
=========================================================================
               coefficient       std. error           t-stat            prob
-------------------------------------------------------------------------
const             0.000584         0.000137          4.266           0.000
L1.lnD_St         0.000555         0.000615          0.902           0.367
L1.lnD_Li        -0.000062         0.000774         -0.080           0.936
L1.lnD_InR       -0.000692         0.000584         -1.185           0.236
L1.lnD_TwF        0.016121         0.039381          0.409           0.682
L1.lnD_ChC       -0.000828         0.000564         -1.466           0.143
L2.lnD_St         0.000356         0.000661          0.538           0.591
L2.lnD_Li         0.000272         0.000837          0.325           0.745
L2.lnD_InR       -0.000148         0.000630         -0.235           0.814
L2.lnD_TwF        0.065212         0.039181          1.664           0.096
L2.lnD_ChC       -0.000426         0.000586         -0.727           0.467
L3.lnD_St        -0.000417         0.000659         -0.632           0.527
L3.lnD_Li         0.000734         0.000845          0.868           0.385
L3.lnD_InR        0.000188         0.000651          0.290           0.772
L3.lnD_TwF        0.016911         0.039367          0.430           0.668
L3.lnD_ChC        0.000323         0.000584          0.553           0.580
L4.lnD_St        -0.000665         0.000576         -1.154           0.248
L4.lnD_Li         0.001828         0.000759          2.408           0.016
L4.lnD_InR        0.000053         0.000621          0.085           0.932
L4.lnD_TwF        0.077761         0.039418          1.973           0.049
L4.lnD_ChC       -0.000007         0.000583         -0.012           0.990
L5.lnD_St        -0.000817         0.000497         -1.644           0.100
L5.lnD_Li         0.002264         0.000670          3.381           0.001
L5.lnD_InR        0.000506         0.000586          0.864           0.387
L5.lnD_TwF        0.048656         0.039642          1.227           0.220
L5.lnD_ChC       -0.000778         0.000561         -1.385           0.166
=========================================================================
```

```
Results for equation lnD_ChC
=========================================================================
               coefficient       std. error           t-stat            prob
-------------------------------------------------------------------------
const             0.003328         0.009510          0.350           0.726
L1.lnD_St        -0.000245         0.042741         -0.006           0.995
L1.lnD_Li         0.007530         0.053788          0.140           0.889
L1.lnD_InR       -0.052880         0.040577         -1.303           0.193
L1.lnD_TwF       -0.338182         2.735116         -0.124           0.902
L1.lnD_ChC       -0.287122         0.039202         -7.324           0.000
L2.lnD_St        -0.030760         0.045938         -0.670           0.503
L2.lnD_Li         0.028519         0.058109          0.491           0.624
L2.lnD_InR       -0.029952         0.043786         -0.684           0.494
L2.lnD_TwF        0.402669         2.721238          0.148           0.882
L2.lnD_ChC       -0.174014         0.040672         -4.278           0.000
L3.lnD_St         0.059495         0.045758          1.300           0.194
L3.lnD_Li        -0.080755         0.058667         -1.376           0.169
L3.lnD_InR        0.035243         0.045207          0.780           0.436
L3.lnD_TwF       -1.182716         2.734147         -0.433           0.665
L3.lnD_ChC       -0.160354         0.040570         -3.953           0.000
L4.lnD_St         0.015179         0.040000          0.379           0.704
L4.lnD_Li        -0.010626         0.052726         -0.202           0.840
L4.lnD_InR        0.064543         0.043142          1.496           0.135
```

```
L4.lnD_TwF        -8.471797         2.737747          -3.094           0.002
L4.lnD_ChC        -0.032020         0.040512          -0.790           0.429
L5.lnD_St          0.015374         0.034536           0.445           0.656
L5.lnD_Li          0.015497         0.046504           0.333           0.739
L5.lnD_InR         0.011747         0.040666           0.289           0.773
L5.lnD_TwF         2.981650         2.753263           1.083           0.279
L5.lnD_ChC        -0.131875         0.038989          -3.382           0.001
===============================================================================

Correlation matrix of residuals
            lnD_St    lnD_Li    lnD_InR   lnD_TwF   lnD_ChC
lnD_St     1.000000  0.838346 -0.001781  0.144712  0.016743
lnD_Li     0.838346  1.000000  0.015307  0.132774  0.016169
lnD_InR   -0.001781  0.015307  1.000000  0.040911 -0.015565
lnD_TwF    0.144712  0.132774  0.040911  1.000000 -0.014769
lnD_ChC    0.016743  0.016169 -0.015565 -0.014769  1.000000
```

In [86]:

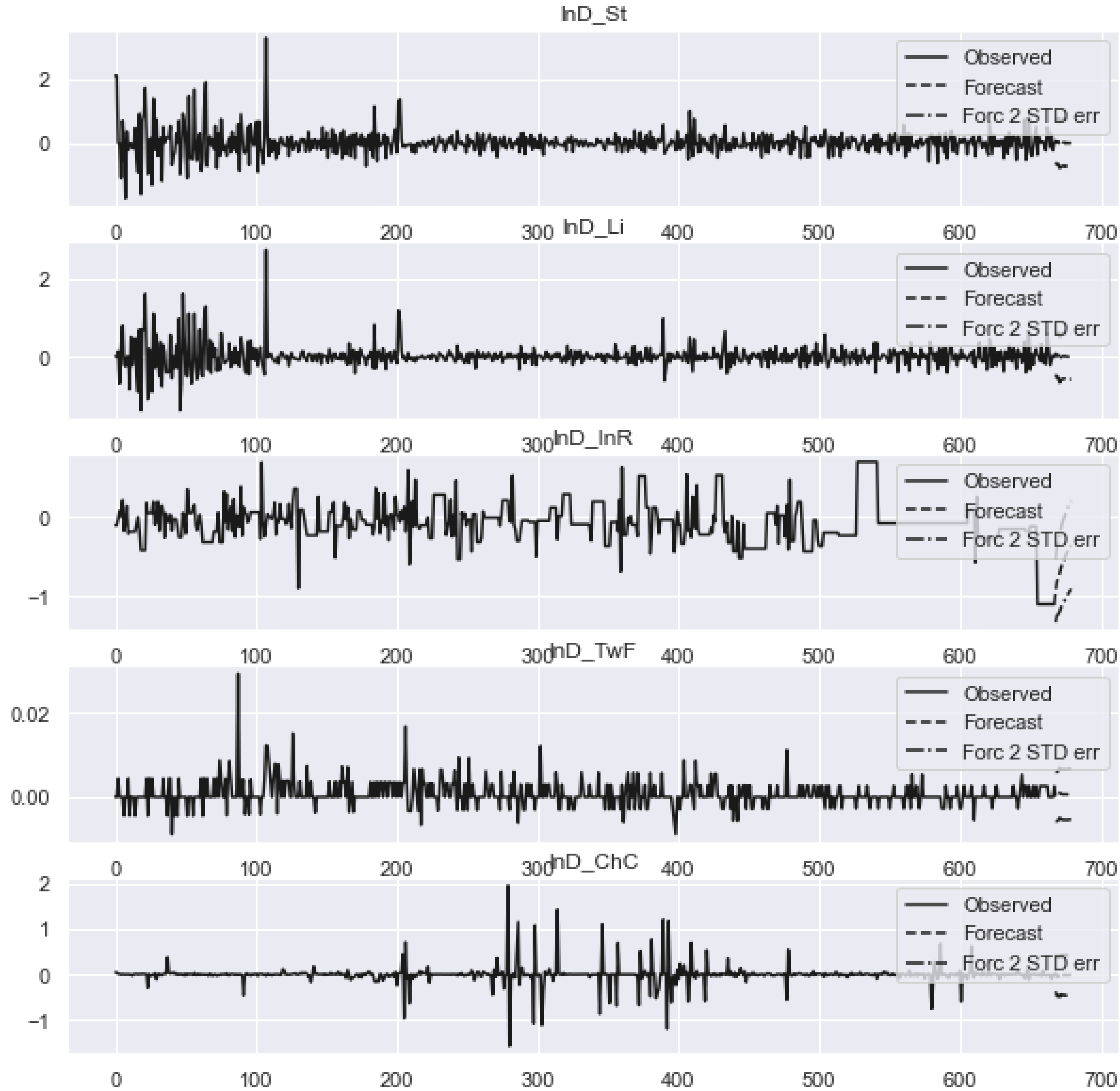
```
# Forecasting
# The linear predictor is the optimal h-step ahead forecast in terms of mean-squared error
# yt(h) = ν + A1 yt(h−1) + ⋯ + Ap yt(h−p)

# We can use the forecast function to produce this forecast.
# Note that we have to specify the “initial value” for the forecast:

lag_orderAM = resultsAM.k_ar
forecastAM = resultsAM.forecast(resultsAMs.params.to_numpy(), 12)
print(f'forecastAM:\n{forecastAM}\n')

print('Forecasts: Apple Music')
print('   lnD_St     lnD_Li      lnD_InR      lnD_TwF     lnD_ChC   ')
print(np.sum(forecastAM, axis=0))
```

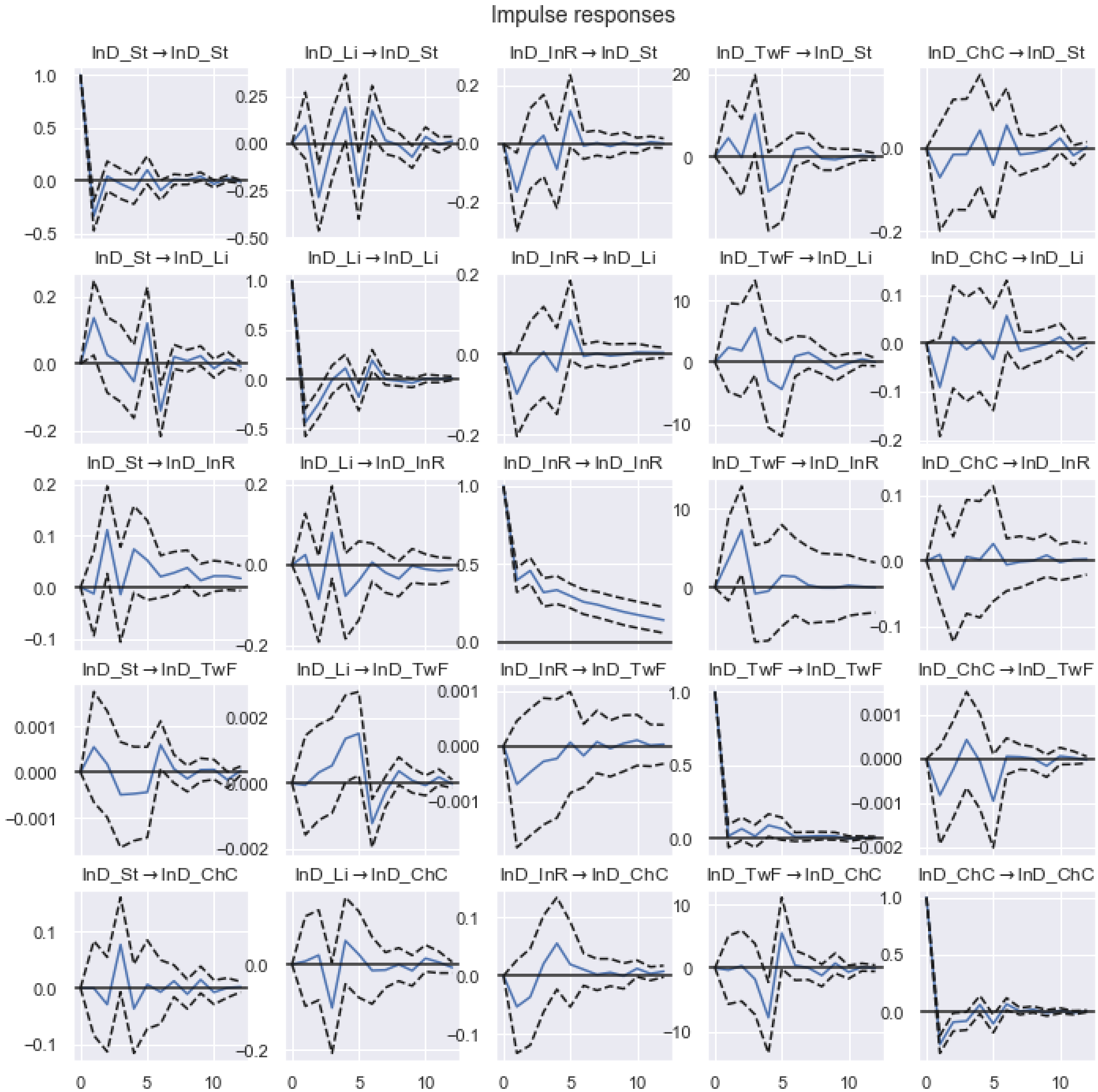
```
forecastAM:
[[ 1.20789227e+00  5.07864648e-01  2.75532426e-01  5.31786196e-03
  -5.42139478e-01]
 [ 2.50654165e-01 -3.03410134e-01  4.33508646e-01  1.40652575e-02
  -6.67119340e-01]
 [ 8.19293619e-01  1.26489439e+00  2.24920684e-01  1.61737193e-02
   4.82335160e-01]
 [ 2.51369292e-01 -3.57653290e-01  5.25715908e-01 -3.34866397e-03
   6.34082275e-01]
 [ 1.37411390e-01 -4.76994796e-03  3.93631536e-01  5.73328906e-03
   1.34233015e+00]
 [-1.00926924e+00 -5.00576721e-01  8.13789850e-01 -2.58118460e-03
  -5.89041494e-02]
 [-1.14195076e+00 -8.28709772e-01  4.71554954e-01  8.76854182e-04
   1.63273567e-01]
 [ 5.60944655e-01  4.90736412e-01  4.10267513e-01  5.53205183e-03
  -4.95800256e-01]
 [ 1.10809279e-01  1.08858864e-01  5.94301287e-01  5.24883896e-03
  -1.68674662e-01]
 [-3.90377116e-01 -2.81172965e-01  1.88133108e-01 -6.15582995e-03
  -1.44721708e-01]
 [ 2.87518446e-01  1.01324977e-01  2.27566877e-01 -2.20574091e-04
   1.01932604e-01]
 [ 2.66450076e-01  2.02735502e-01  3.11876007e-03  1.91965263e-03
   1.28731653e-01]]

Forecasts: Apple Music
   lnD_St     lnD_Li      lnD_InR      lnD_TwF     lnD_ChC
[1.35074608 0.40012197 4.56204155 0.04256127 0.77532582]
```

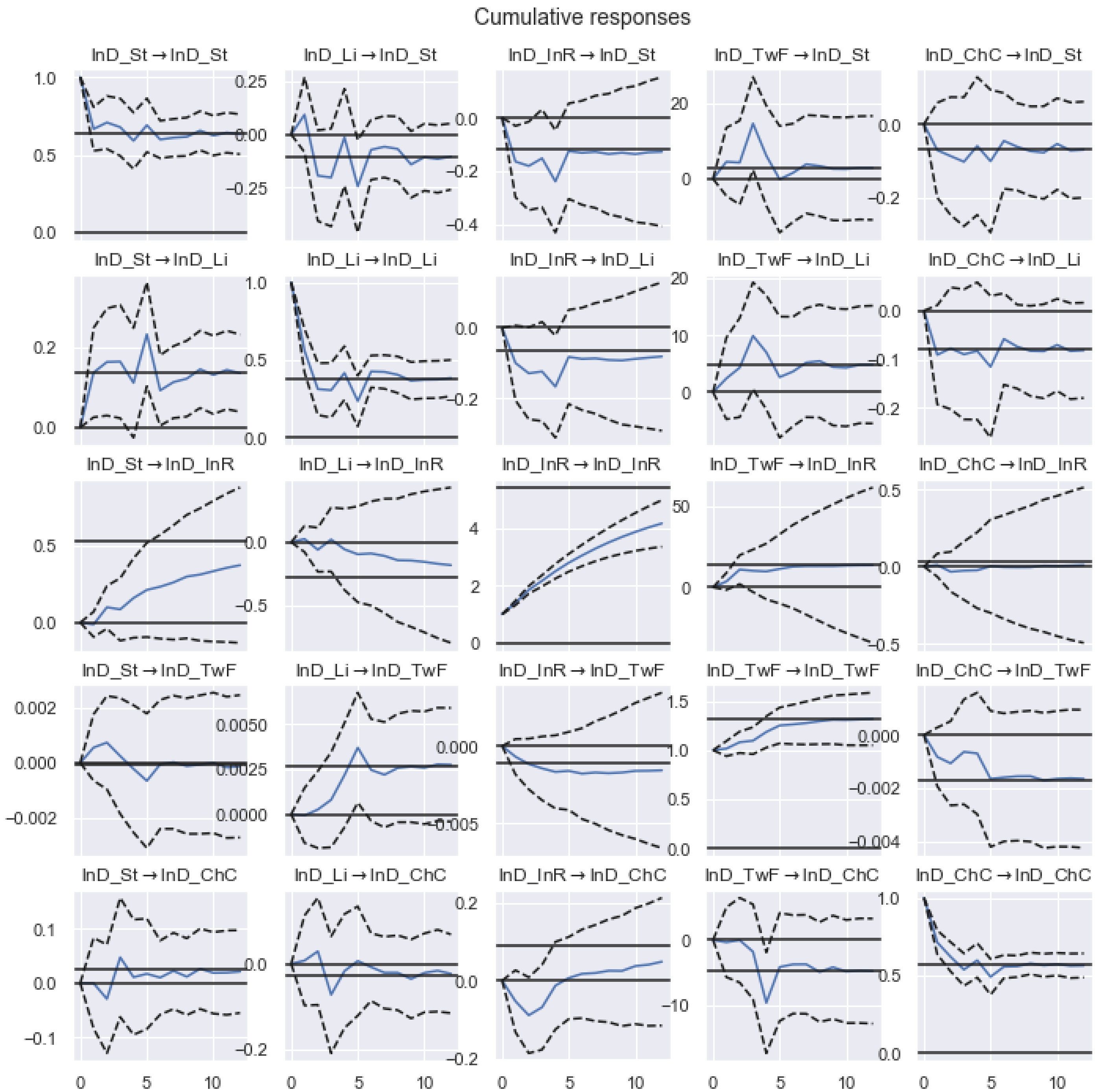
In [89]:

```
# The forecast_interval function will produce the above forecast along with asymptotic sta
# These can be visualized using the plot_forecast function:
resultsAMs.plot_forecast(12)

# Impulse Response Analysis
# Impulse responses are of interest in econometric studies:
#    they are the estimated responses to a unit impulse in one of the variables.
# They are computed in practice using the MA(∞) representation of the VAR(p) process:
# Yt = μ + ∑i=0 to ∞ Φi ut-i

# We can perform an impulse response analysis by calling the irf function on a VARResults
# These can be visualized using the plot function, in either orthogonalized or non-orthogo
# Asymptotic standard errors are plotted by default at the 95% significance level, which c
irf = resultsAMs.irf(12)
irf.plot(orth=False)

# The cumulative effects Ψn=∑ni=0Φi can be plotted with the long run effects as follows:
irf.plot_cum_effects(orth=False)

# Granger causality
# One is often interested in whether a variable or group of variables is "causal" for anot
# for some definition of "causal". In the context of VAR models,
# one can say that a set of variables are Granger-causal within one of the VAR equations.
# See Wooldridge p. 626 for a discussion of Granger causality.
# The VARResults object has the test_causality method
#  for performing either a Wald (χ2) test or an F-test.

print('+++++++++++++++++++++++++++++++++++')
print('Test for Granger causality for Streams\n')
print(resultsAMs.test_causality('lnD_St', ['lnD_Li', 'lnD_InR','lnD_TwF','lnD_ChC'], kind=

print('+++++++++++++++++++++++++++++++++++')
print('Test for Granger causality for IGReach\n')
print(resultsAMs.test_causality('lnD_InR', ['lnD_St', 'lnD_Li','lnD_TwF','lnD_ChC'], kind=

# Normality
# The white noise component ut is assumed to be normally distributed.
# While this assumption is not required for parameter estimates to be consistent or asympt
# results are generally more reliable in finite samples when residuals are Gaussian white
# To test whether this assumption is consistent with a data set, VARResults offers the tes
# Note: Stock market return data are frequently NOT normally distributed!
print('+++++++++++++++++++++++++++++++++++')
print(resultsAMs.test_normality())
```

```
+++++++++++++++++++++++++++++++++++
Test for Granger causality for Streams

<statsmodels.tsa.vector_ar.hypothesis_test_results.CausalityTestResults object. H_0: ['lnD
_Li', 'lnD_InR', 'lnD_TwF', 'lnD_ChC'] do not Granger-cause lnD_St: reject at 5% significa
nce level. Test statistic: 2.300, critical value: 1.574>, p-value: 0.001>
+++++++++++++++++++++++++++++++++++
Test for Granger causality for IGReach

<statsmodels.tsa.vector_ar.hypothesis_test_results.CausalityTestResults object. H_0: ['lnD
_St', 'lnD_Li', 'lnD_TwF', 'lnD_ChC'] do not Granger-cause lnD_InR: reject at 5% significa
nce level. Test statistic: 1.713, critical value: 1.574>, p-value: 0.025>
+++++++++++++++++++++++++++++++++++
<statsmodels.tsa.vector_ar.hypothesis_test_results.NormalityTestResults object. H_0: data
generated by normally-distributed process: reject at 5% significance level. Test statisti
c: 28663.622, critical value: 18.307>, p-value: 0.000>
```


lnD_St
Observed
Forecast
Forc 2 STD err
lnD_Li
lnD_InR
lnD_TwF
lnD_ChC


Impulse responses
lnD_St → lnD_St
lnD_Li → lnD_St
lnD_InR → lnD_St
lnD_TwF → lnD_St
lnD_ChC → lnD_St
lnD_St → lnD_Li
lnD_Li → lnD_Li
lnD_InR → lnD_Li
lnD_TwF → lnD_Li
lnD_ChC → lnD_Li
lnD_St → lnD_InR
lnD_Li → lnD_InR
lnD_InR → lnD_InR
lnD_TwF → lnD_InR
lnD_ChC → lnD_InR
lnD_St → lnD_TwF
lnD_Li → lnD_TwF
lnD_InR → lnD_TwF
lnD_TwF → lnD_TwF
lnD_ChC → lnD_TwF
lnD_St → lnD_ChC
lnD_Li → lnD_ChC
lnD_InR → lnD_ChC
lnD_TwF → lnD_ChC
lnD_ChC → lnD_ChC



SoundCloud, VAR Estimation:

In [90]:

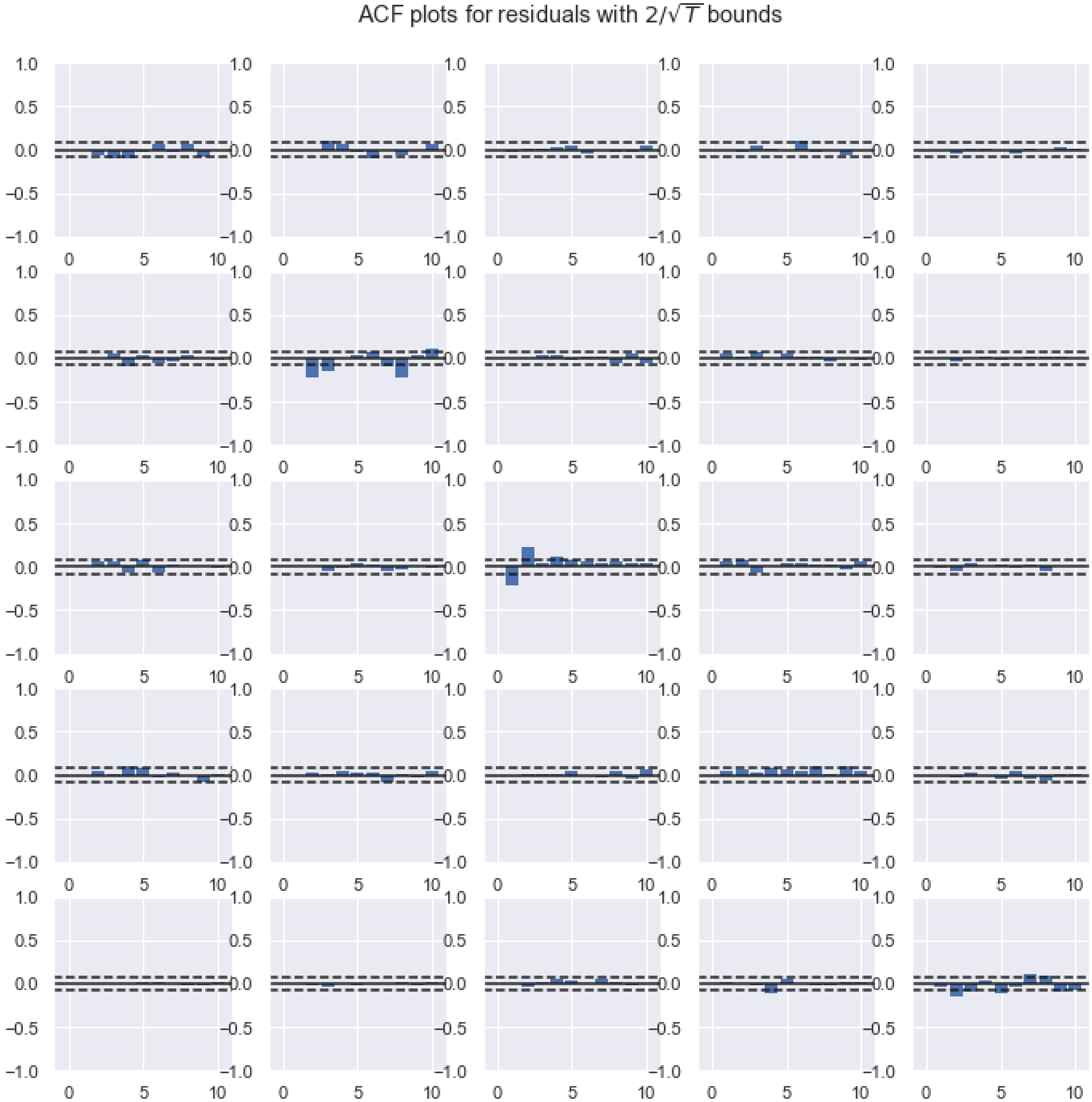
```
# make a VAR model
# Vector Autoregression Model
# VAR(p) processes

# See Wooldridge p. 626 for a discussion of VAR models.
# It should be noted that much financial timeseries data are non-stationary.
# Before any analysis it is a good idea to plot the data, test for unit roots, and probab
# construct the log-differences in the data to obtain % return that are almost always sta
# In the code above, I have constructed the log-differenced data for the etf funds.

# We are interested in modeling a T×K
# multivariate time series Y, where T denotes the number of observations
# and K the number of variables. One way of estimating relationships
# between the time series and their lagged values is the vector autoregression process:
# Yt = ν + A1 Yt−1 + … + Ap Yt−p + ut  where ut∼Normal(0,Σu) and A is a K×K coefficient ma
# see https://www.statsmodels.org/dev/vector_ar.html
```

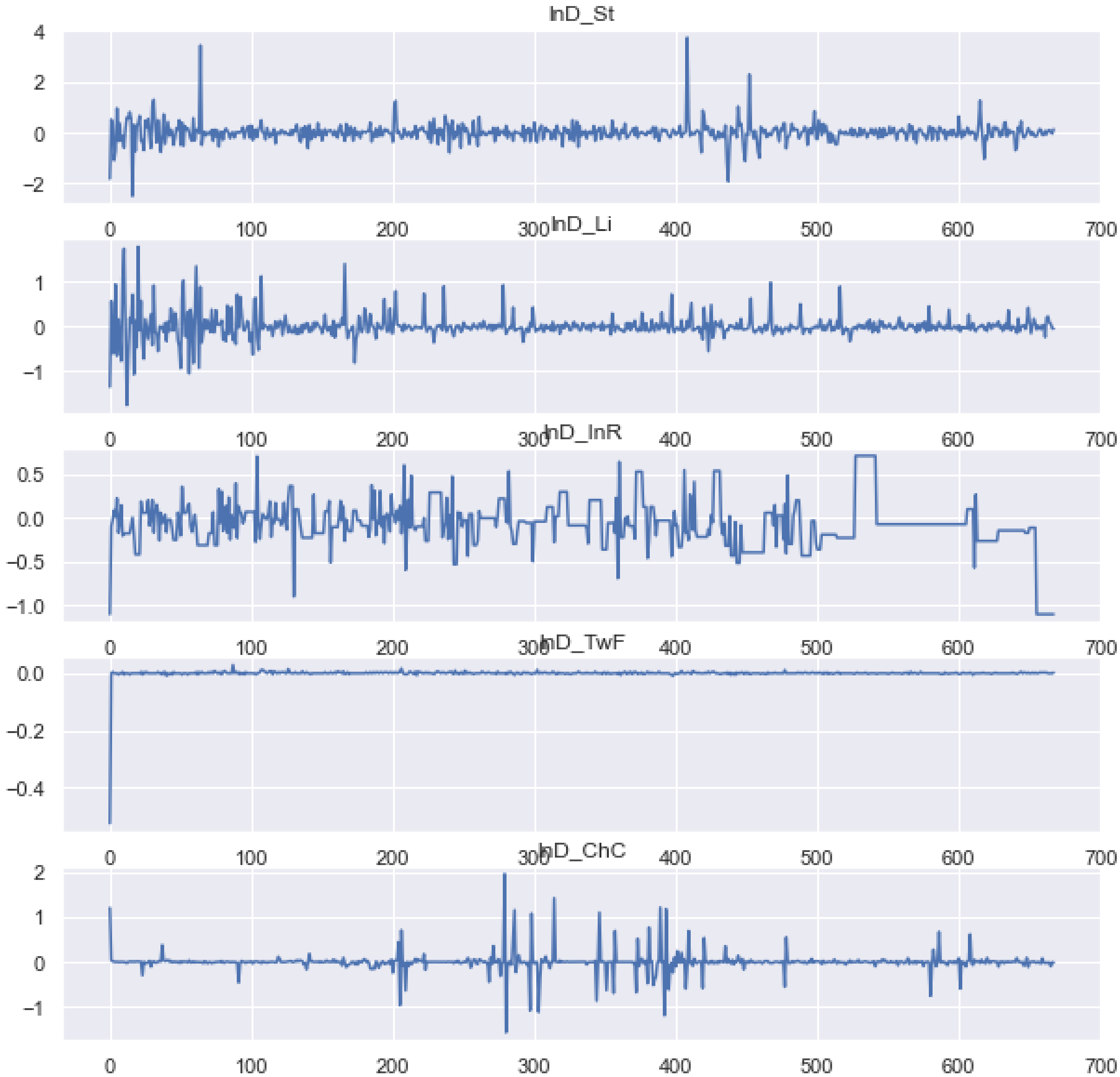
```python
var_modelSC = VAR(lnSC)
resultsSC = var_modelSC.fit()
print(f'resultsSC.summary():\n{resultsSC.summary()}\n')

resultsSC.plot()
resultsSC.plot_acorr()
```

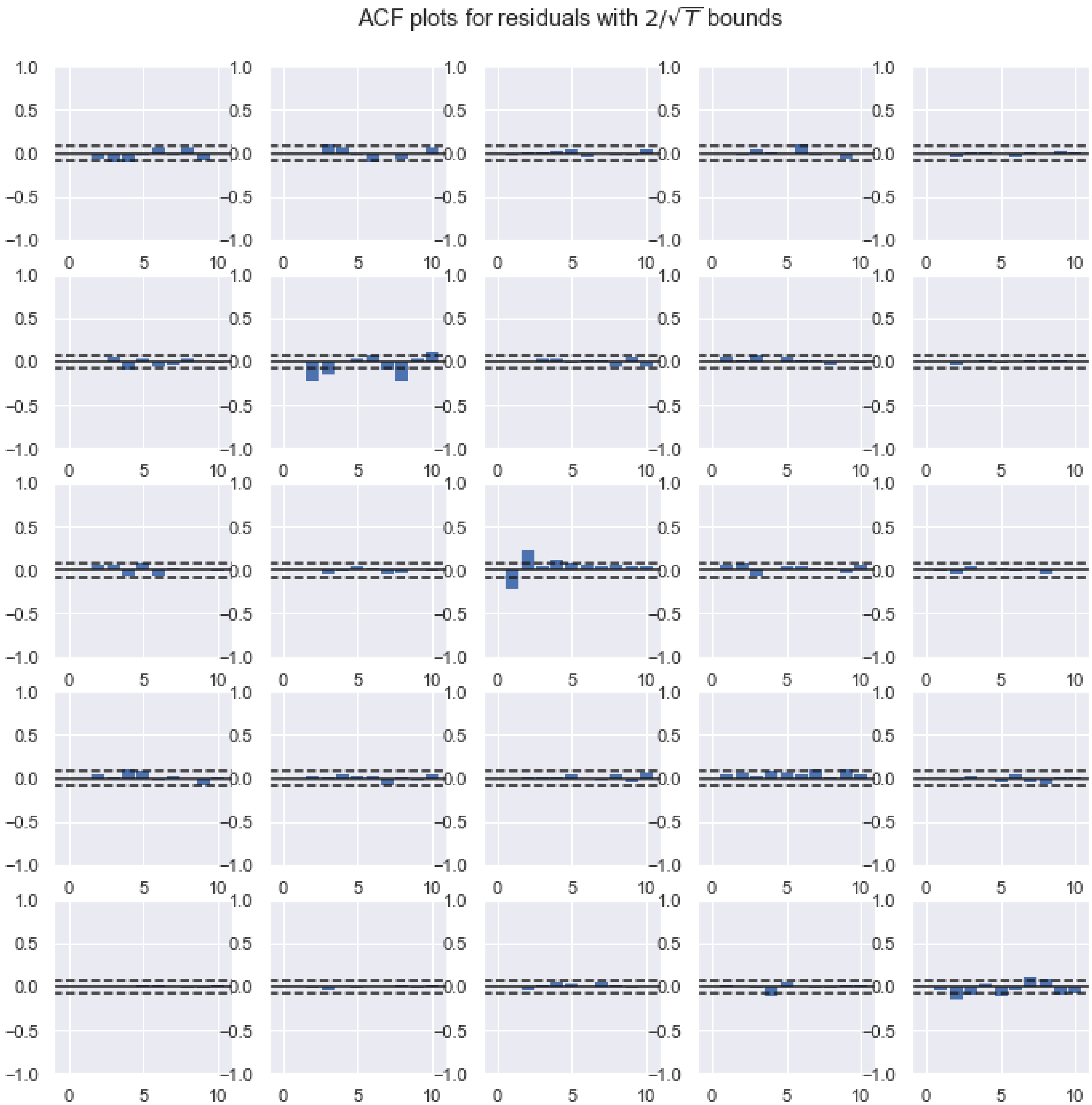
```
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/base/tsa_model.py:590: ValueWarning: An unsupported index was provided and will be ignored when e.g. forecasting.
  warnings.warn('An unsupported index was provided and will be'
resultsSC.summary():
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Wed, 09, Feb, 2022
Time:                     14:52:05
--------------------------------------------------------------------
No. of Equations:         5.00000    BIC:                 -22.1054
Nobs:                     667.000    HQIC:                -22.2295
Log likelihood:           2737.54    FPE:              2.05015e-10
AIC:                     -22.3079    Det(Omega_mle):   1.96037e-10
--------------------------------------------------------------------
Results for equation lnD_St
==============================================================================
                coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------
const              0.000653         0.014961            0.044           0.965
L1.lnD_St          0.008953         0.038811            0.231           0.818
L1.lnD_Li          0.001546         0.053252            0.029           0.977
L1.lnD_InR        -0.047559         0.052889           -0.899           0.369
L1.lnD_TwF        -1.090155         0.749148           -1.455           0.146
L1.lnD_ChC        -0.060268         0.066372           -0.908           0.364
==============================================================================

Results for equation lnD_Li
==============================================================================
                coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------
const              0.007355         0.010857            0.677           0.498
L1.lnD_St          0.047354         0.028163            1.681           0.093
L1.lnD_Li         -0.057146         0.038642           -1.479           0.139
L1.lnD_InR         0.000746         0.038379            0.019           0.984
L1.lnD_TwF        -0.943653         0.543618           -1.736           0.083
L1.lnD_ChC         0.016310         0.048163            0.339           0.735
==============================================================================

Results for equation lnD_InR
==============================================================================
                coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------
const             -0.027474         0.008729           -3.147           0.002
L1.lnD_St         -0.000081         0.022645           -0.004           0.997
L1.lnD_Li          0.033394         0.031071            1.075           0.282
L1.lnD_InR         0.627875         0.030859           20.347           0.000
L1.lnD_TwF        -1.095405         0.437103           -2.506           0.012
L1.lnD_ChC         0.031212         0.038726            0.806           0.420
==============================================================================

Results for equation lnD_TwF
==============================================================================
                coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------
const              0.000750         0.000121            6.199           0.000
L1.lnD_St          0.000294         0.000314            0.938           0.348
```

```
L1.lnD_Li            0.000216         0.000431            0.502           0.616
L1.lnD_InR          -0.000485         0.000428           -1.133           0.257
L1.lnD_TwF          -0.000060         0.006057           -0.010           0.992
L1.lnD_ChC          -0.000845         0.000537           -1.575           0.115
===============================================================================

Results for equation lnD_ChC
===============================================================================
                coefficient       std. error           t-stat            prob
-------------------------------------------------------------------------------
const               -0.002441         0.008498           -0.287           0.774
L1.lnD_St            0.016842         0.022045            0.764           0.445
L1.lnD_Li            0.023630         0.030247            0.781           0.435
L1.lnD_InR          -0.006857         0.030042           -0.228           0.819
L1.lnD_TwF          -0.714745         0.425523           -1.680           0.093
L1.lnD_ChC          -0.241659         0.037700           -6.410           0.000
===============================================================================

Correlation matrix of residuals
             lnD_St    lnD_Li   lnD_InR   lnD_TwF   lnD_ChC
lnD_St     1.000000  0.022592 -0.008506  0.002647 -0.017425
lnD_Li     0.022592  1.000000 -0.059154 -0.060409  0.033501
lnD_InR   -0.008506 -0.059154  1.000000  0.045228 -0.007489
lnD_TwF    0.002647 -0.060409  0.045228  1.000000 -0.007683
lnD_ChC   -0.017425  0.033501 -0.007489 -0.007683  1.000000
```

Out[90]:

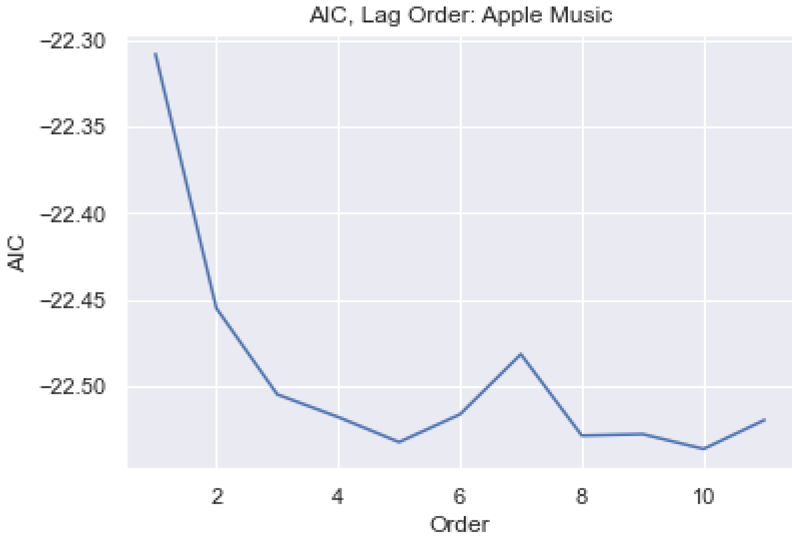

ACF plots for residuals with $2/\sqrt{T}$ bounds
1.0
0.5
0.0
−0.5
−1.0
0
5
10


lnD_St
lnD_Li
lnD_InR
lnD_TwF
lnD_ChC



In [91]:

```
resultsSC_aic = []
for p in range(1,12):
    resultsSC = var_modelSC.fit(p)
    resultsSC_aic.append(resultsSC.aic)


import seaborn as sns
sns.set()
plt.title('AIC, Lag Order: Apple Music')
plt.plot(list(np.arange(1,12,1)), resultsSC_aic)
plt.xlabel("Order")
plt.ylabel("AIC")
plt.show()
```



Lowest AIC appears to occur at approximately (Order10, AIC-22.50)

In [100…

```
# Lag order selection
# Choice of lag order can be a difficult problem.
# Standard analysis employs likelihood test or information criteria-based order selection.
# We have implemented the latter, accessible through the VAR class:

# When calling the fit function, one can pass a maximum number of lags
# and the order criterion to use for order selection:
var_modelSC.select_order(10)
resultsSCs = var_modelSC.fit(maxlags=10, ic='aic')
print(f'resultsSCs.summary():\n{resultsSCs.summary()}\n')
```

```
resultsSCs.summary():
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Wed, 09, Feb, 2022
Time:                     15:00:24
--------------------------------------------------------------------
No. of Equations:         5.00000    BIC:                  -21.8066
Nobs:                     664.000    HQIC:                 -22.2423
Log likelihood:           2870.09    FPE:               1.66195e-10
AIC:                     -22.5180    Det(Omega_mle):    1.42234e-10
--------------------------------------------------------------------
Results for equation lnD_St
==============================================================================
                coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------
const              0.002051         0.016135            0.127           0.899
L1.lnD_St         -0.021634         0.038986           -0.555           0.579
L1.lnD_Li          0.033627         0.055594            0.605           0.545
L1.lnD_InR        -0.076404         0.071034           -1.076           0.282
L1.lnD_TwF        -2.431470         4.768416           -0.510           0.610
L1.lnD_ChC        -0.062196         0.068599           -0.907           0.365
L2.lnD_St         -0.071144         0.038521           -1.847           0.065
L2.lnD_Li          0.000536         0.054713            0.010           0.992
L2.lnD_InR         0.025494         0.075619            0.337           0.736
L2.lnD_TwF        -2.282991         4.767552           -0.479           0.632
L2.lnD_ChC        -0.063400         0.070802           -0.895           0.371
L3.lnD_St         -0.097049         0.038527           -2.519           0.012
L3.lnD_Li          0.156864         0.054433            2.882           0.004
L3.lnD_InR         0.028806         0.075874            0.380           0.704
L3.lnD_TwF         5.676889         4.786808            1.186           0.236
```

L3.lnD_ChC	-0.029317	0.070941	-0.413	0.679
L4.lnD_St	-0.102680	0.038816	-2.645	0.008
L4.lnD_Li	0.124486	0.054770	2.273	0.023
L4.lnD_InR	0.020750	0.071531	0.290	0.772
L4.lnD_TwF	1.324049	0.759590	1.743	0.081
L4.lnD_ChC	-0.000605	0.068695	-0.009	0.993

Results for equation lnD_Li

	coefficient	std. error	t-stat	prob
const	0.004222	0.011302	0.374	0.709
L1.lnD_St	0.063431	0.027307	2.323	0.020
L1.lnD_Li	-0.095980	0.038939	-2.465	0.014
L1.lnD_InR	-0.035930	0.049754	-0.722	0.470
L1.lnD_TwF	4.445284	3.339924	1.331	0.183
L1.lnD_ChC	0.007010	0.048048	0.146	0.884
L2.lnD_St	-0.011243	0.026981	-0.417	0.677
L2.lnD_Li	-0.237696	0.038323	-6.203	0.000
L2.lnD_InR	-0.001470	0.052966	-0.028	0.978
L2.lnD_TwF	0.210201	3.339319	0.063	0.950
L2.lnD_ChC	-0.025883	0.049591	-0.522	0.602
L3.lnD_St	0.049976	0.026985	1.852	0.064
L3.lnD_Li	-0.160515	0.038126	-4.210	0.000
L3.lnD_InR	0.039736	0.053144	0.748	0.455
L3.lnD_TwF	6.372247	3.352807	1.901	0.057
L3.lnD_ChC	-0.005298	0.049689	-0.107	0.915
L4.lnD_St	-0.056511	0.027187	-2.079	0.038
L4.lnD_Li	-0.085151	0.038362	-2.220	0.026
L4.lnD_InR	0.019321	0.050102	0.386	0.700
L4.lnD_TwF	-1.246112	0.532037	-2.342	0.019
L4.lnD_ChC	0.019821	0.048116	0.412	0.680

Results for equation lnD_InR

	coefficient	std. error	t-stat	prob
const	-0.021078	0.008936	-2.359	0.018
L1.lnD_St	0.002079	0.021590	0.096	0.923
L1.lnD_Li	0.012384	0.030787	0.402	0.688
L1.lnD_InR	0.391736	0.039337	9.959	0.000
L1.lnD_TwF	4.124323	2.640649	1.562	0.118
L1.lnD_ChC	0.011133	0.037989	0.293	0.769
L2.lnD_St	0.029743	0.021332	1.394	0.163
L2.lnD_Li	0.024203	0.030299	0.799	0.424
L2.lnD_InR	0.312194	0.041876	7.455	0.000
L2.lnD_TwF	6.465456	2.640170	2.449	0.014
L2.lnD_ChC	-0.035852	0.039208	-0.914	0.361
L3.lnD_St	0.036965	0.021336	1.733	0.083
L3.lnD_Li	-0.045246	0.030144	-1.501	0.133
L3.lnD_InR	0.039676	0.042018	0.944	0.345
L3.lnD_TwF	-4.969857	2.650834	-1.875	0.061
L3.lnD_ChC	0.021752	0.039285	0.554	0.580
L4.lnD_St	-0.030140	0.021495	-1.402	0.161
L4.lnD_Li	-0.022427	0.030330	-0.739	0.460
L4.lnD_InR	0.060898	0.039612	1.537	0.124
L4.lnD_TwF	-0.007620	0.420645	-0.018	0.986
L4.lnD_ChC	0.012100	0.038042	0.318	0.750

Results for equation lnD_TwF

	coefficient	std. error	t-stat	prob

```
const               0.000648         0.000133            4.879           0.000
L1.lnD_St           0.000327         0.000321            1.019           0.308
L1.lnD_Li           0.000188         0.000457            0.411           0.681
L1.lnD_InR         -0.000940         0.000584           -1.608           0.108
L1.lnD_TwF          0.034510         0.039225            0.880           0.379
L1.lnD_ChC         -0.000940         0.000564           -1.666           0.096
L2.lnD_St           0.000531         0.000317            1.675           0.094
L2.lnD_Li           0.000550         0.000450            1.222           0.222
L2.lnD_InR          0.000112         0.000622            0.180           0.857
L2.lnD_TwF          0.071151         0.039218            1.814           0.070
L2.lnD_ChC         -0.000407         0.000582           -0.699           0.485
L3.lnD_St           0.000063         0.000317            0.198           0.843
L3.lnD_Li           0.000241         0.000448            0.538           0.591
L3.lnD_InR          0.000421         0.000624            0.674           0.500
L3.lnD_TwF          0.025598         0.039377            0.650           0.516
L3.lnD_ChC          0.000447         0.000584            0.766           0.443
L4.lnD_St           0.000791         0.000319            2.476           0.013
L4.lnD_Li           0.000636         0.000451            1.411           0.158
L4.lnD_InR          0.000187         0.000588            0.317           0.751
L4.lnD_TwF         -0.001250         0.006248           -0.200           0.841
L4.lnD_ChC          0.000318         0.000565            0.562           0.574
===============================================================================

Results for equation lnD_ChC
===============================================================================
                 coefficient       std. error           t-stat            prob
-------------------------------------------------------------------------------
const               0.000164         0.009256            0.018           0.986
L1.lnD_St           0.010791         0.022363            0.483           0.629
L1.lnD_Li           0.012737         0.031890            0.399           0.690
L1.lnD_InR         -0.037581         0.040747           -0.922           0.356
L1.lnD_TwF         -0.187661         2.735283           -0.069           0.945
L1.lnD_ChC         -0.293580         0.039350           -7.461           0.000
L2.lnD_St           0.003143         0.022097            0.142           0.887
L2.lnD_Li          -0.018107         0.031385           -0.577           0.564
L2.lnD_InR         -0.048345         0.043377           -1.115           0.265
L2.lnD_TwF         -0.599224         2.734787           -0.219           0.827
L2.lnD_ChC         -0.158314         0.040614           -3.898           0.000
L3.lnD_St           0.005585         0.022100            0.253           0.800
L3.lnD_Li          -0.027331         0.031224           -0.875           0.381
L3.lnD_InR          0.042748         0.043523            0.982           0.326
L3.lnD_TwF         -0.820204         2.745833           -0.299           0.765
L3.lnD_ChC         -0.140270         0.040693           -3.447           0.001
L4.lnD_St           0.007748         0.022266            0.348           0.728
L4.lnD_Li          -0.007229         0.031417           -0.230           0.818
L4.lnD_InR          0.070688         0.041032            1.723           0.085
L4.lnD_TwF         -0.365614         0.435720           -0.839           0.401
L4.lnD_ChC          0.005402         0.039405            0.137           0.891
===============================================================================

Correlation matrix of residuals
            lnD_St    lnD_Li   lnD_InR   lnD_TwF   lnD_ChC
lnD_St    1.000000  0.036389 -0.004696  0.005619 -0.018338
lnD_Li    0.036389  1.000000 -0.090208 -0.042498  0.013945
lnD_InR  -0.004696 -0.090208  1.000000  0.032745 -0.024368
lnD_TwF   0.005619 -0.042498  0.032745  1.000000 -0.010256
lnD_ChC  -0.018338  0.013945 -0.024368 -0.010256  1.000000
```

In [101…

```
# Forecasting
# The linear predictor is the optimal h-step ahead forecast in terms of mean-squared error
# yt(h) = ν + A1 yt(h-1) + ⋯ + Ap yt(h-p)
```

```
# We can use the forecast function to produce this forecast.
# Note that we have to specify the “initial value” for the forecast:

lag_orderSC = resultsSC.k_ar
forecastSC = resultsSC.forecast(resultsSCs.params.to_numpy(), 12)
print(f'forecastSC:\n{forecastSC}\n')

print('Forecasts: Apple Music')
print('   lnD_St      lnD_Li      lnD_InR      lnD_TwF     lnD_ChC  ')
print(np.sum(forecastSC, axis=0))
```

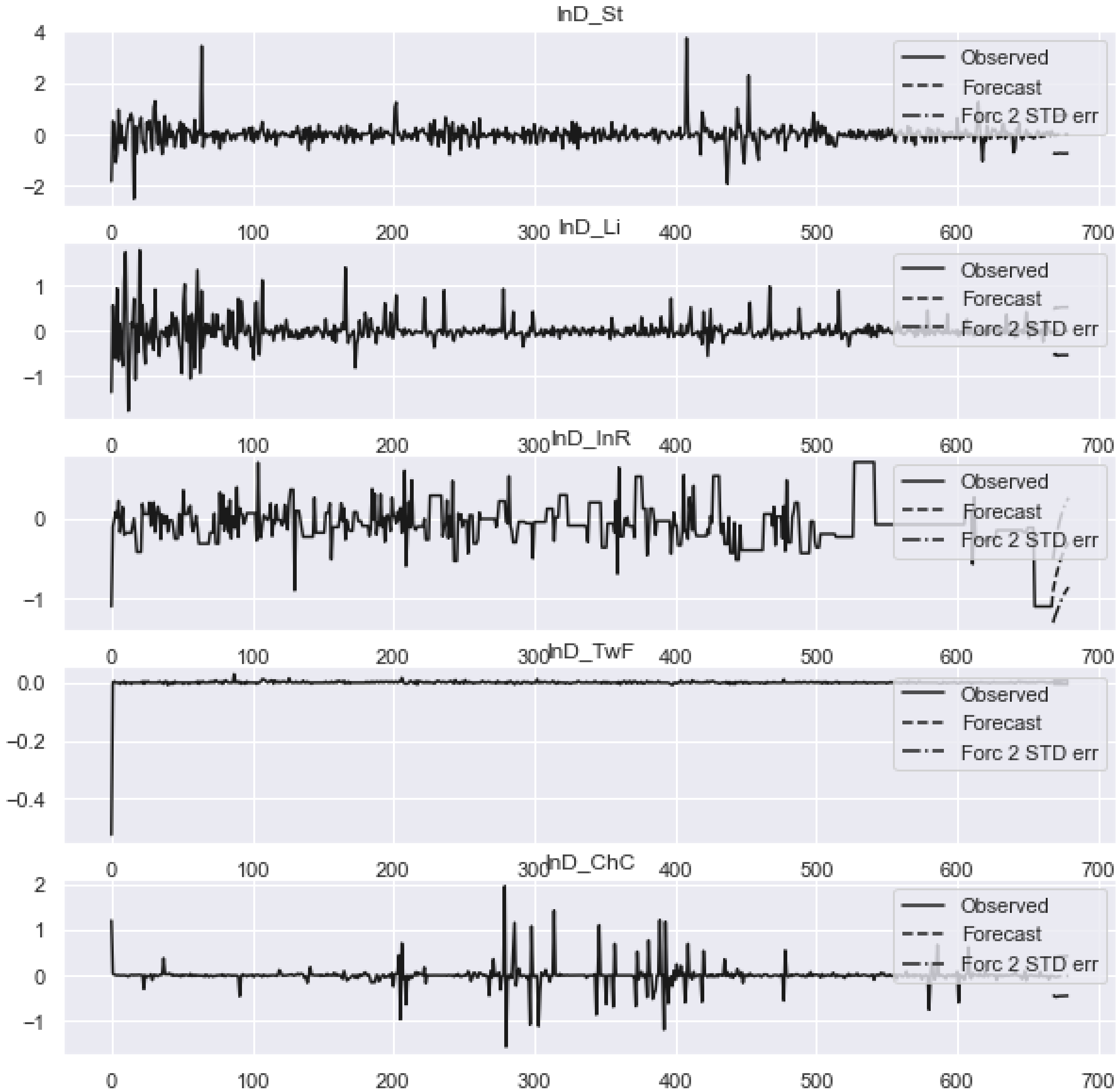
```
forecastSC:
[[-7.54672396e-01 -3.27320840e-01 -2.06287623e-01 -1.20922980e-04
  -4.17505785e-01]
 [-3.08800679e-01 -7.84717797e-01 -2.71585706e-01 -1.87481260e-03
   1.16755876e-01]
 [-1.02621849e+00 -6.39170906e-01 -3.47205451e-01 -9.18267165e-04
   2.88623770e-02]
 [-2.21379288e-01  7.21920284e-01 -2.39305769e-01 -9.96774191e-04
   3.70085144e-01]
 [ 1.52945297e+00  2.47229332e-01 -3.55003218e-01 -7.42571813e-04
   6.70551913e-02]
 [-1.69717797e-01  1.28541412e-01 -1.96724469e-01  1.89400937e-03
   1.16466953e-01]
 [ 2.31984689e-01  2.43695998e-01 -6.11218370e-02  5.93951133e-04
  -7.36269176e-02]
 [ 1.12232510e-01  4.12621197e-02 -1.11042989e-01  6.45108459e-04
   2.17671230e-03]
 [-1.29738890e-01 -1.12146717e-01 -9.45624232e-02  2.34379281e-03
  -1.24483918e-01]
 [-4.92382884e-02  2.76962951e-01  6.55530049e-02  4.56552537e-03
   7.27371515e-02]
 [ 8.06386215e-02  3.64611475e-02 -9.04549034e-02 -6.81936039e-04
   5.05650760e-02]
 [-1.72611945e-01 -2.30276661e-01 -6.10074582e-02  1.41158219e-03
   5.03278519e-02]]

Forecasts: Apple Music
   lnD_St      lnD_Li      lnD_InR      lnD_TwF     lnD_ChC  
[-0.87806898 -0.39755968 -1.96874884  0.00611868  0.25941571]
```

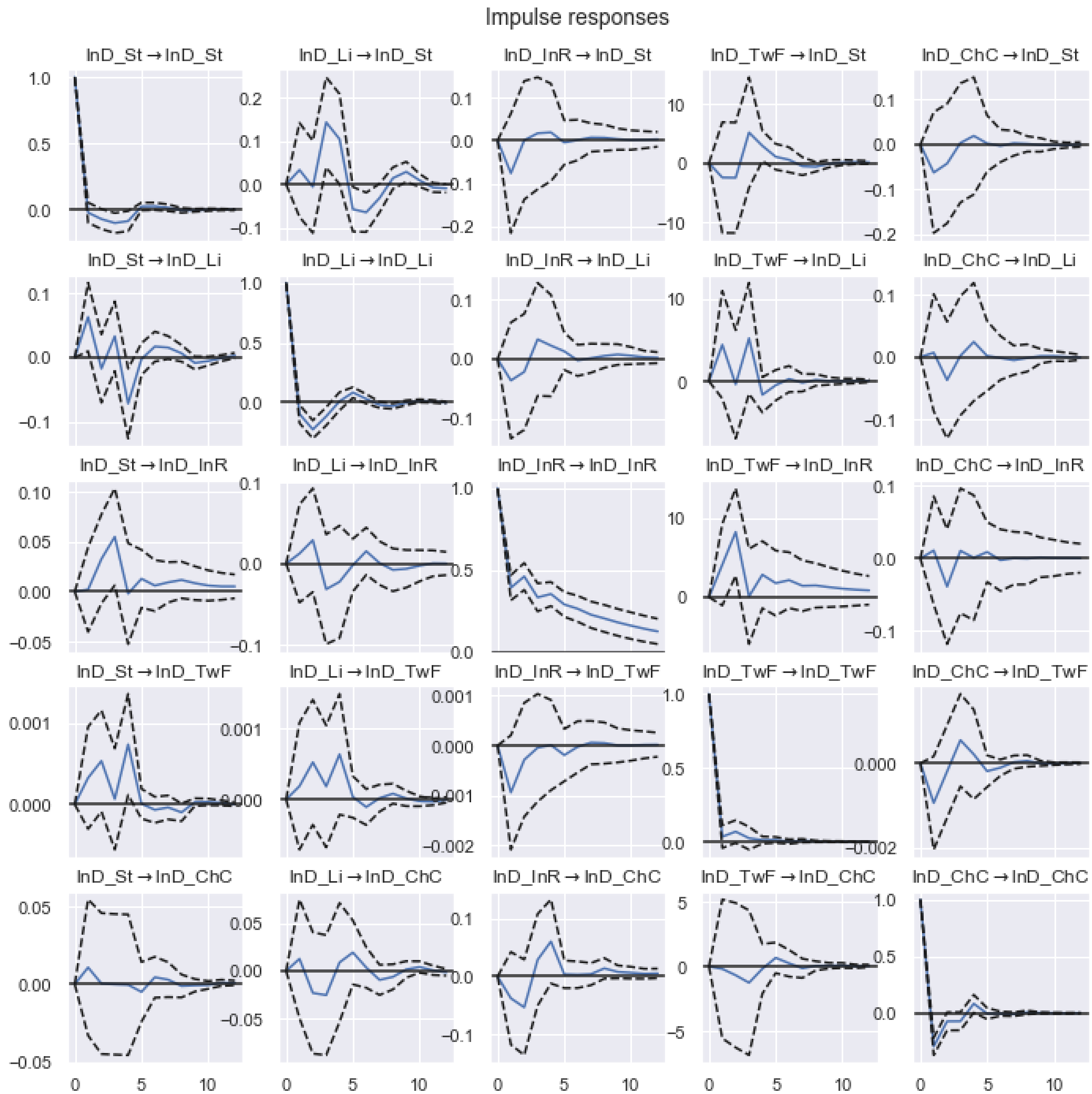
In [102…

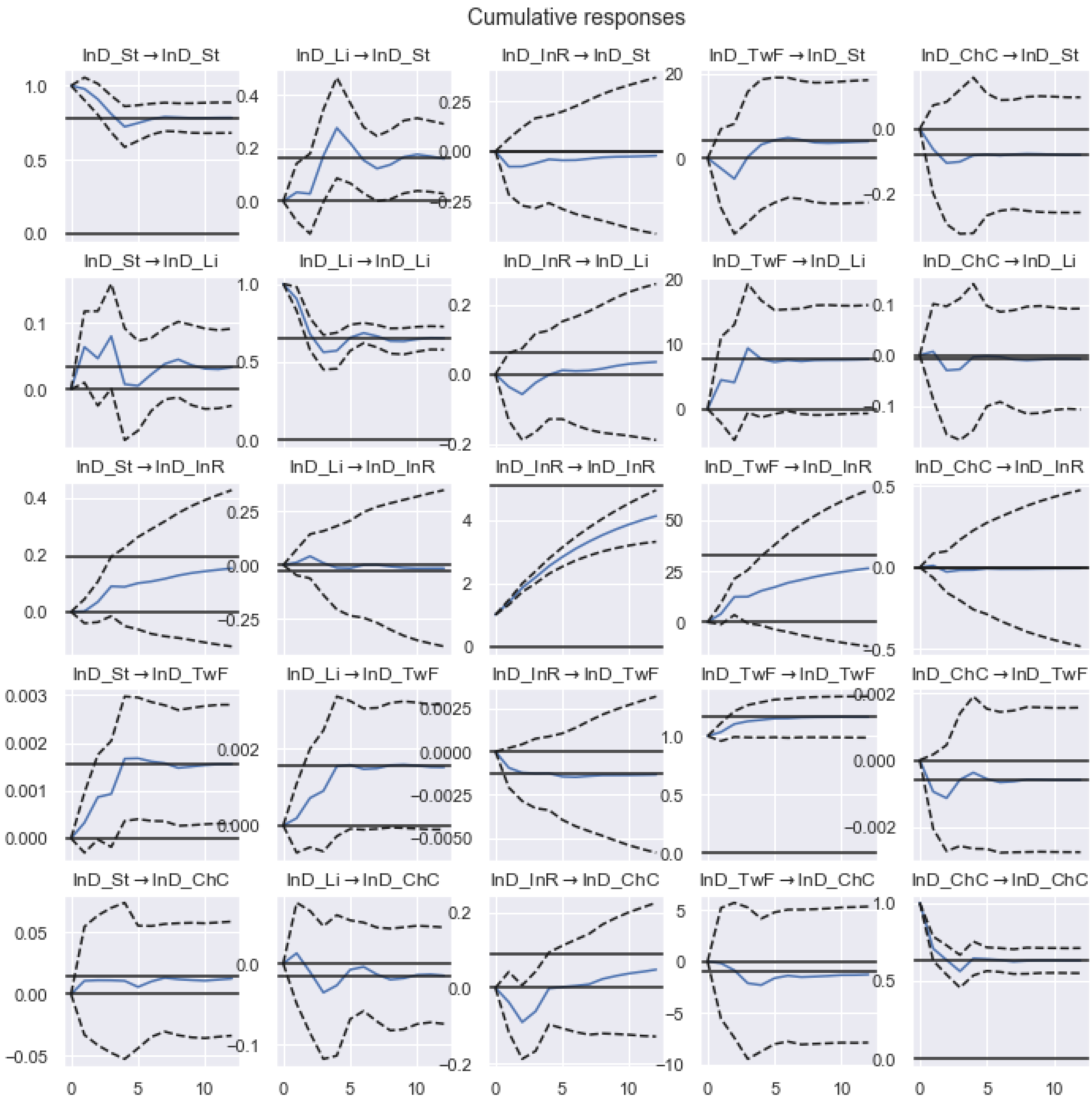
```
# The forecast_interval function will produce the above forecast along with asymptotic sta
# These can be visualized using the plot_forecast function:
resultsSCs.plot_forecast(12)

# Impulse Response Analysis
# Impulse responses are of interest in econometric studies:
#    they are the estimated responses to a unit impulse in one of the variables.
# They are computed in practice using the MA(∞) representation of the VAR(p) process:
# Yt = μ + ∑i=0 to ∞ Φi ut−i

# We can perform an impulse response analysis by calling the irf function on a VARResults
# These can be visualized using the plot function, in either orthogonalized or non-orthogo
# Asymptotic standard errors are plotted by default at the 95% significance level, which c
irf = resultsSCs.irf(12)
irf.plot(orth=False)

# The cumulative effects Ψn=∑ni=0Φi can be plotted with the long run effects as follows:
irf.plot_cum_effects(orth=False)

# Granger causality
# One is often interested in whether a variable or group of variables is “causal” for anot
# for some definition of “causal”. In the context of VAR models,
# one can say that a set of variables are Granger-causal within one of the VAR equations.
```

```
# See Wooldridge p. 626 for a discussion of Granger causality.
# The VARResults object has the test_causality method
#  for performing either a Wald (χ2) test or an F-test.

print('++++++++++++++++++++++++++++++++++++')
print('Test for Granger causality for Streams\n')
print(resultsSCs.test_causality('lnD_St', ['lnD_Li', 'lnD_InR','lnD_TwF','lnD_ChC'], kind=

print('++++++++++++++++++++++++++++++++++++')
print('Test for Granger causality for IGReach\n')
print(resultsSCs.test_causality('lnD_InR', ['lnD_St', 'lnD_Li','lnD_TwF','lnD_ChC'], kind=

# Normality
# The white noise component ut is assumed to be normally distributed.
# While this assumption is not required for parameter estimates to be consistent or asympt
# results are generally more reliable in finite samples when residuals are Gaussian white
# To test whether this assumption is consistent with a data set, VARResults offers the tes
# Note: Stock market return data are frequently NOT normally distributed!
print('++++++++++++++++++++++++++++++++++++')
print(resultsSCs.test_normality())
```

```
++++++++++++++++++++++++++++++++++++
Test for Granger causality for Streams

<statsmodels.tsa.vector_ar.hypothesis_test_results.CausalityTestResults object. H_0: ['lnD
_Li', 'lnD_InR', 'lnD_TwF', 'lnD_ChC'] do not Granger-cause lnD_St: fail to reject at 5% s
ignificance level. Test statistic: 1.359, critical value: 1.647>, p-value: 0.153>
++++++++++++++++++++++++++++++++++++
Test for Granger causality for IGReach

<statsmodels.tsa.vector_ar.hypothesis_test_results.CausalityTestResults object. H_0: ['lnD
_St', 'lnD_Li', 'lnD_TwF', 'lnD_ChC'] do not Granger-cause lnD_InR: fail to reject at 5% s
ignificance level. Test statistic: 1.609, critical value: 1.647>, p-value: 0.059>
++++++++++++++++++++++++++++++++++++
<statsmodels.tsa.vector_ar.hypothesis_test_results.NormalityTestResults object. H_0: data
generated by normally-distributed process: reject at 5% significance level. Test statisti
c: 46582.520, critical value: 18.307>, p-value: 0.000>
```


lnD_St
Observed
Forecast
Forc 2 STD err
lnD_Li
Observed
Forecast
Forc 2 STD err
lnD_InR
Observed
Forecast
Forc 2 STD err
lnD_TwF
Observed
Forecast
Forc 2 STD err
lnD_ChC
Observed
Forecast
Forc 2 STD err


Impulse responses
lnD_St → lnD_St
lnD_Li → lnD_St
lnD_InR → lnD_St
lnD_TwF → lnD_St
lnD_ChC → lnD_St
lnD_St → lnD_Li
lnD_Li → lnD_Li
lnD_InR → lnD_Li
lnD_TwF → lnD_Li
lnD_ChC → lnD_Li
lnD_St → lnD_InR
lnD_Li → lnD_InR
lnD_InR → lnD_InR
lnD_TwF → lnD_InR
lnD_ChC → lnD_InR
lnD_St → lnD_TwF
lnD_Li → lnD_TwF
lnD_InR → lnD_TwF
lnD_TwF → lnD_TwF
lnD_ChC → lnD_TwF
lnD_St → lnD_ChC
lnD_Li → lnD_ChC
lnD_InR → lnD_ChC
lnD_TwF → lnD_ChC
lnD_ChC → lnD_ChC



Spotify, VAR Estimation:

In [98]:

```
# make a VAR model
# Vector Autoregression Model
# VAR(p) processes

# See Wooldridge p. 626 for a discussion of VAR models.
# It should be noted that much financial timeseries data are non-stationary.
# Before any analysis it is a good idea to plot the data, test for unit roots, and probab
# construct the log-differences in the data to obtain % return that are almost always sta
# In the code above, I have constructed the log-differenced data for the etf funds.

# We are interested in modeling a T×K
# multivariate time series Y, where T denotes the number of observations
# and K the number of variables. One way of estimating relationships
# between the time series and their lagged values is the vector autoregression process:
# Yt = ν + A1 Yt−1 + … + Ap Yt−p + ut  where ut∼Normal(0,Σu) and A is a K×K coefficient ma
# see https://www.statsmodels.org/dev/vector_ar.html
```

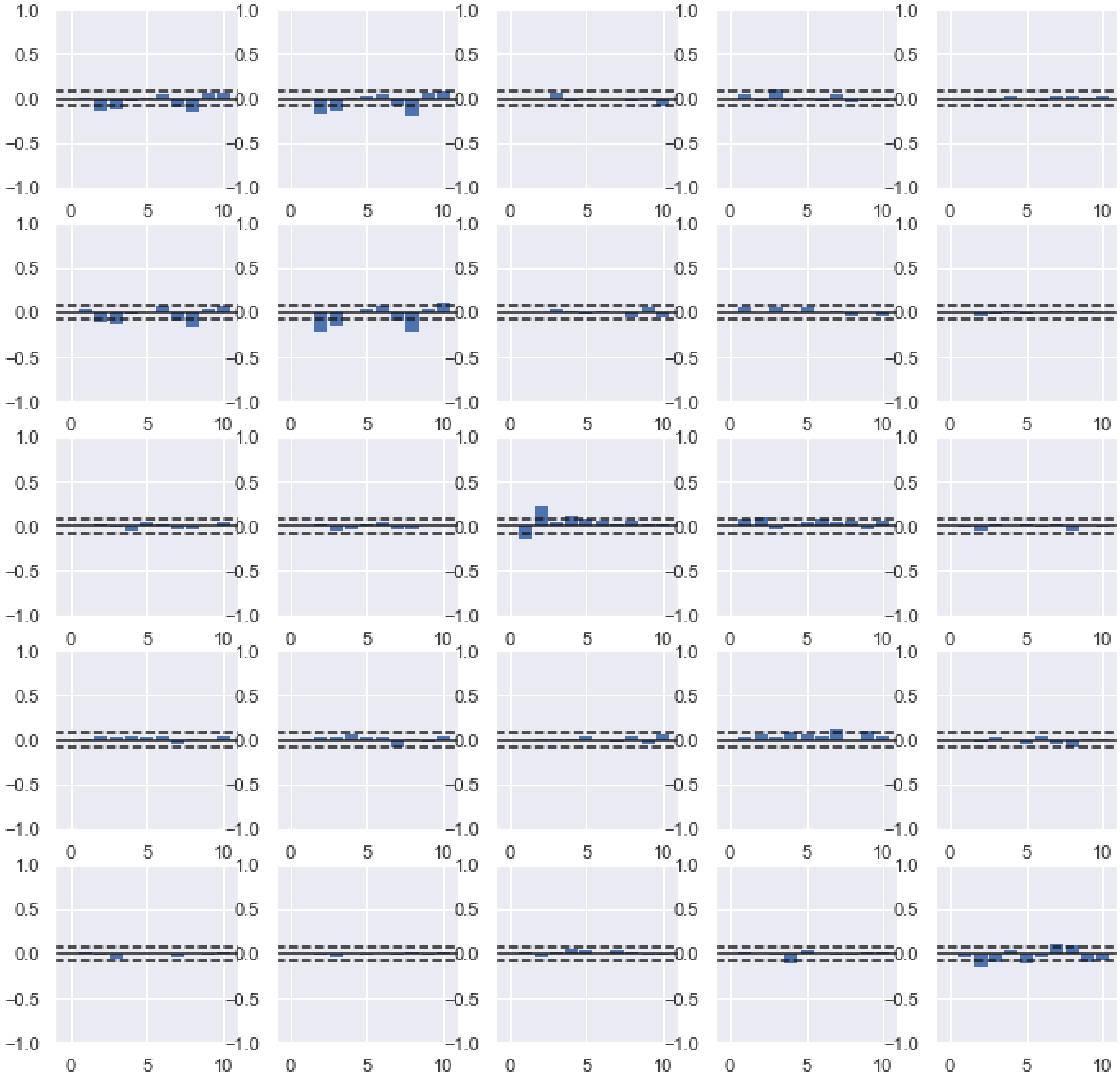
```python
var_modelSP = VAR(lnSP)
resultsSP = var_modelSP.fit()
print(f'resultsSP.summary():\n{resultsSP.summary()}\n')

resultsSP.plot()
resultsSP.plot_acorr()
```

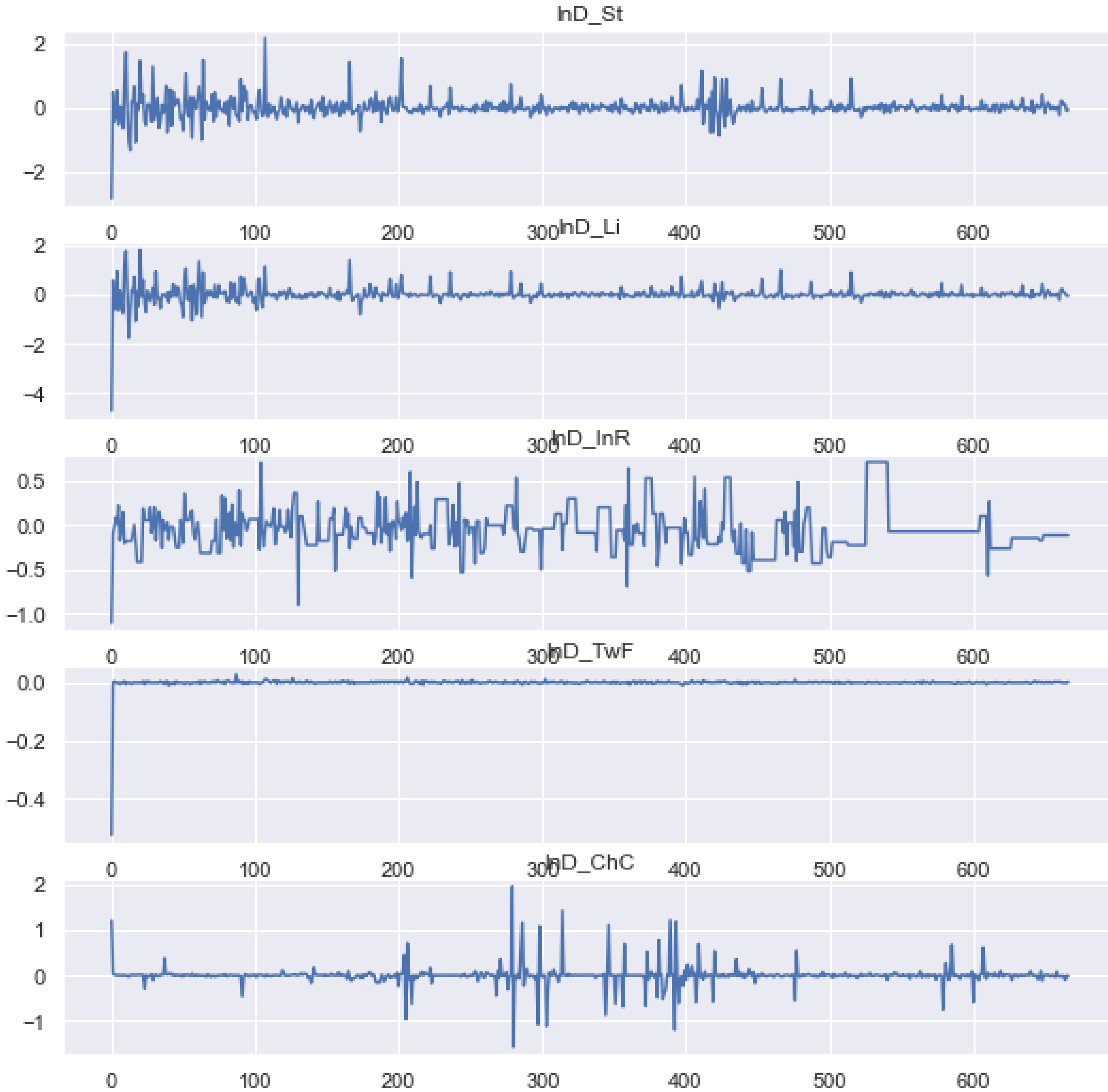
```
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/base/tsa_model.py:590: ValueWar
ning: An unsupported index was provided and will be ignored when e.g. forecasting.
  warnings.warn('An unsupported index was provided and will be'
resultsSP.summary():
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Wed, 09, Feb, 2022
Time:                     14:58:46
--------------------------------------------------------------------
No. of Equations:         5.00000    BIC:                  -23.9988
Nobs:                     666.000    HQIC:                 -24.1230
Log likelihood:           3364.05    FPE:               3.08608e-11
AIC:                     -24.2015    Det(Omega_mle):    2.95075e-11
--------------------------------------------------------------------
Results for equation lnD_St
=============================================================================
                coefficient       std. error           t-stat            prob
-----------------------------------------------------------------------------
const              0.008922         0.011316            0.788           0.430
L1.lnD_St         -0.455526         0.066067           -6.895           0.000
L1.lnD_Li          0.407337         0.070929            5.743           0.000
L1.lnD_InR         0.009000         0.046777            0.192           0.847
L1.lnD_TwF        -1.961263         0.678456           -2.891           0.004
L1.lnD_ChC         0.030918         0.050529            0.612           0.541
=============================================================================

Results for equation lnD_Li
=============================================================================
                coefficient       std. error           t-stat            prob
-----------------------------------------------------------------------------
const              0.007618         0.010802            0.705           0.481
L1.lnD_St         -0.082253         0.063067           -1.304           0.192
L1.lnD_Li          0.013424         0.067709            0.198           0.843
L1.lnD_InR         0.005799         0.044653            0.130           0.897
L1.lnD_TwF        -0.607308         0.647656           -0.938           0.348
L1.lnD_ChC         0.017583         0.048235            0.365           0.715
=============================================================================

Results for equation lnD_InR
=============================================================================
                coefficient       std. error           t-stat            prob
-----------------------------------------------------------------------------
const             -0.026452         0.008194           -3.228           0.001
L1.lnD_St         -0.031325         0.047841           -0.655           0.513
L1.lnD_Li          0.053371         0.051362            1.039           0.299
L1.lnD_InR         0.497939         0.033873           14.700           0.000
L1.lnD_TwF        -1.034490         0.491297           -2.106           0.035
L1.lnD_ChC         0.027309         0.036590            0.746           0.455
=============================================================================

Results for equation lnD_TwF
=============================================================================
                coefficient       std. error           t-stat            prob
-----------------------------------------------------------------------------
const              0.000759         0.000120            6.314           0.000
L1.lnD_St          0.001103         0.000702            1.572           0.116
```

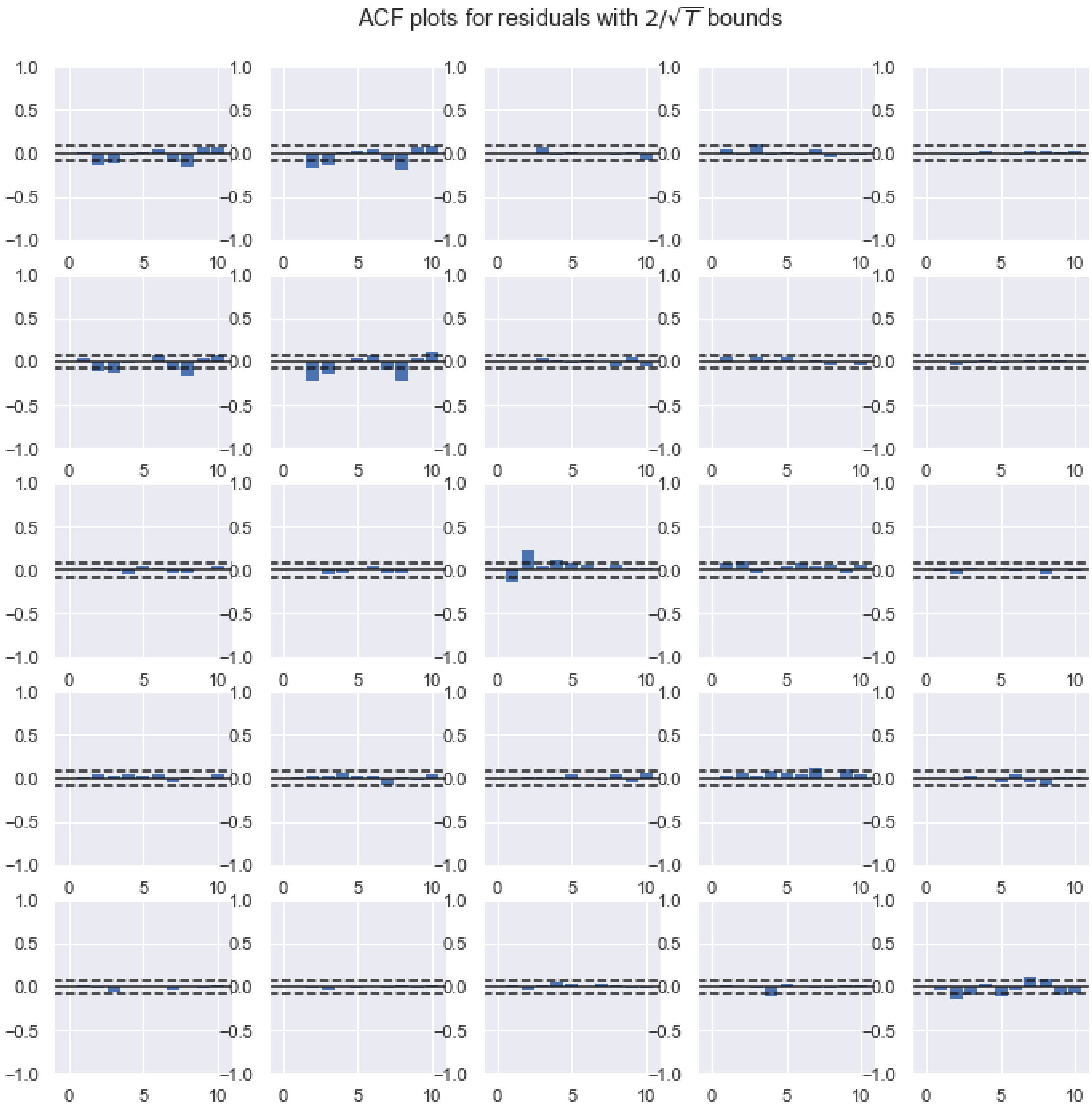
```
L1.lnD_Li         -0.000782          0.000753          -1.038           0.299
L1.lnD_InR        -0.000465          0.000497          -0.935           0.350
L1.lnD_TwF         0.002416          0.007207           0.335           0.737
L1.lnD_ChC        -0.000865          0.000537          -1.612           0.107
===============================================================================

Results for equation lnD_ChC
===============================================================================
                coefficient       std. error           t-stat            prob
-------------------------------------------------------------------------------
const             -0.002316          0.008443          -0.274           0.784
L1.lnD_St         -0.065227          0.049293          -1.323           0.186
L1.lnD_Li          0.080340          0.052921           1.518           0.129
L1.lnD_InR        -0.012589          0.034901          -0.361           0.718
L1.lnD_TwF        -0.943002          0.506209          -1.863           0.062
L1.lnD_ChC        -0.241039          0.037701          -6.393           0.000
===============================================================================

Correlation matrix of residuals
              lnD_St    lnD_Li   lnD_InR   lnD_TwF   lnD_ChC
lnD_St      1.000000  0.841388 -0.026631 -0.012648  0.026501
lnD_Li      0.841388  1.000000 -0.063575 -0.054145  0.033218
lnD_InR    -0.026631 -0.063575  1.000000  0.050737 -0.014276
lnD_TwF    -0.012648 -0.054145  0.050737  1.000000 -0.003551
lnD_ChC     0.026501  0.033218 -0.014276 -0.003551  1.000000
```

Out[98]:

ACF plots for residuals with $2/\sqrt{T}$ bounds

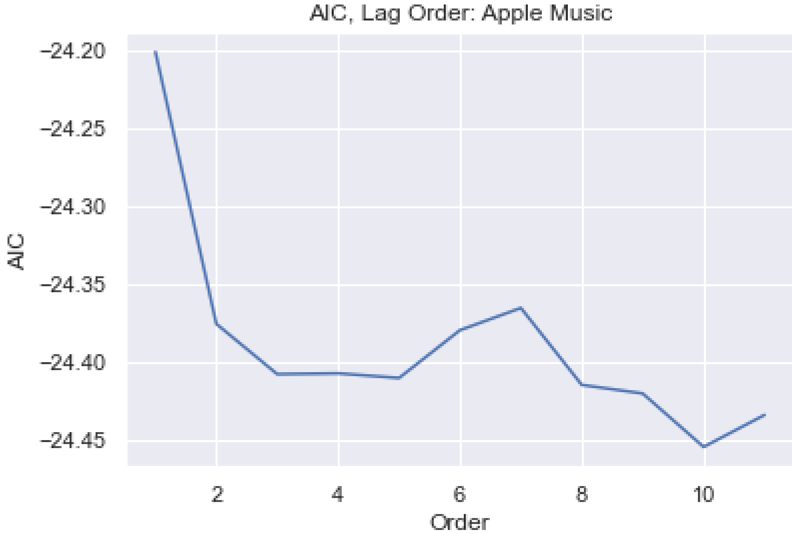



lnD_St
lnD_Li
lnD_lnR
lnD_TwF
lnD_ChC



In [99]:

```
resultsSP_aic = []
for p in range(1,12):
    resultsSP = var_modelSP.fit(p)
    resultsSP_aic.append(resultsSP.aic)


import seaborn as sns
sns.set()
plt.title('AIC, Lag Order: Apple Music')
plt.plot(list(np.arange(1,12,1)), resultsSP_aic)
plt.xlabel("Order")
plt.ylabel("AIC")
plt.show()
```




```
# Lowest AIC appears to occur @ (Order10, AIC-24.45)
```

In [105…

```python
# Lag order selection
# Choice of lag order can be a difficult problem.
# Standard analysis employs likelihood test or information criteria-based order selection.
# We have implemented the latter, accessible through the VAR class:

# When calling the fit function, one can pass a maximum number of lags
# and the order criterion to use for order selection:
var_modelSP.select_order(10)
resultsSPs = var_modelSP.fit(maxlags=10, ic='aic')
print(f'resultsSPs.summary():\n{resultsSPs.summary()}\n')
```

```
resultsSPs.summary():
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Wed, 09, Feb, 2022
Time:                     15:02:13
--------------------------------------------------------------------
No. of Equations:         5.00000    BIC:                   -23.8661
Nobs:                     664.000    HQIC:                  -24.1981
Log likelihood:           3472.61    FPE:                2.51028e-11
AIC:                     -24.4081    Det(Omega_mle):     2.22853e-11
--------------------------------------------------------------------
Results for equation lnD_St
==============================================================================
                coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------
const              0.011676         0.011863            0.984           0.325
L1.lnD_St         -0.423947         0.072287           -5.865           0.000
L1.lnD_Li          0.339963         0.077479            4.388           0.000
L1.lnD_InR        -0.013109         0.055536           -0.236           0.813
L1.lnD_TwF         2.573978         3.605078            0.714           0.475
L1.lnD_ChC         0.023728         0.051314            0.462           0.644
L2.lnD_St          0.038807         0.078530            0.494           0.621
L2.lnD_Li         -0.226892         0.081674           -2.778           0.005
L2.lnD_InR        -0.005501         0.056145           -0.098           0.922
L2.lnD_TwF        -3.228012         3.610026           -0.894           0.371
L2.lnD_ChC        -0.019940         0.052905           -0.377           0.706
L3.lnD_St          0.049092         0.072746            0.675           0.500
L3.lnD_Li         -0.204685         0.078402           -2.611           0.009
L3.lnD_InR         0.058303         0.055336            1.054           0.292
L3.lnD_TwF         1.908020         0.691945            2.757           0.006
```

L3.lnD_ChC	-0.035157	0.051410	-0.684	0.494

Results for equation lnD_Li

	coefficient	std. error	t-stat	prob
const	0.008217	0.011026	0.745	0.456
L1.lnD_St	0.058966	0.067187	0.878	0.380
L1.lnD_Li	-0.152594	0.072012	-2.119	0.034
L1.lnD_InR	-0.009618	0.051618	-0.186	0.852
L1.lnD_TwF	3.876442	3.350703	1.157	0.247
L1.lnD_ChC	0.018354	0.047693	0.385	0.700
L2.lnD_St	0.303226	0.072989	4.154	0.000
L2.lnD_Li	-0.497028	0.075911	-6.547	0.000
L2.lnD_InR	-0.018968	0.052183	-0.363	0.716
L2.lnD_TwF	-1.439495	3.355302	-0.429	0.668
L2.lnD_ChC	-0.026303	0.049172	-0.535	0.593
L3.lnD_St	0.093426	0.067613	1.382	0.167
L3.lnD_Li	-0.254724	0.072870	-3.496	0.000
L3.lnD_InR	0.051937	0.051432	1.010	0.313
L3.lnD_TwF	2.747385	0.643121	4.272	0.000
L3.lnD_ChC	-0.019810	0.047783	-0.415	0.678

Results for equation lnD_InR

	coefficient	std. error	t-stat	prob
const	-0.027718	0.008342	-3.323	0.001
L1.lnD_St	-0.031191	0.050831	-0.614	0.539
L1.lnD_Li	0.042049	0.054482	0.772	0.440
L1.lnD_InR	0.325681	0.039052	8.340	0.000
L1.lnD_TwF	4.721625	2.535014	1.863	0.063
L1.lnD_ChC	0.009938	0.036083	0.275	0.783
L2.lnD_St	-0.027133	0.055220	-0.491	0.623
L2.lnD_Li	0.064099	0.057432	1.116	0.264
L2.lnD_InR	0.294279	0.039480	7.454	0.000
L2.lnD_TwF	6.636291	2.538493	2.614	0.009
L2.lnD_ChC	-0.037402	0.037202	-1.005	0.315
L3.lnD_St	0.048530	0.051154	0.949	0.343
L3.lnD_Li	-0.071701	0.055130	-1.301	0.193
L3.lnD_InR	0.044204	0.038911	1.136	0.256
L3.lnD_TwF	-0.015995	0.486561	-0.033	0.974
L3.lnD_ChC	0.006762	0.036151	0.187	0.852

Results for equation lnD_TwF

	coefficient	std. error	t-stat	prob
const	0.000677	0.000129	5.243	0.000
L1.lnD_St	0.001572	0.000787	1.997	0.046
L1.lnD_Li	-0.001232	0.000844	-1.460	0.144
L1.lnD_InR	-0.000867	0.000605	-1.433	0.152
L1.lnD_TwF	0.034905	0.039268	0.889	0.374
L1.lnD_ChC	-0.000999	0.000559	-1.787	0.074
L2.lnD_St	0.001094	0.000855	1.279	0.201
L2.lnD_Li	-0.000577	0.000890	-0.649	0.516
L2.lnD_InR	0.000303	0.000612	0.496	0.620
L2.lnD_TwF	0.075368	0.039322	1.917	0.055
L2.lnD_ChC	-0.000483	0.000576	-0.839	0.402
L3.lnD_St	0.000947	0.000792	1.195	0.232
L3.lnD_Li	-0.000682	0.000854	-0.798	0.425
L3.lnD_InR	0.000410	0.000603	0.680	0.496
L3.lnD_TwF	0.003453	0.007537	0.458	0.647

```
L3.lnD_ChC          0.000357         0.000560            0.638           0.523
==============================================================================

Results for equation lnD_ChC
==============================================================================
                 coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------
const              -0.001704         0.008957           -0.190           0.849
L1.lnD_St          -0.058779         0.054582           -1.077           0.282
L1.lnD_Li           0.073861         0.058502            1.263           0.207
L1.lnD_InR         -0.032735         0.041934           -0.781           0.435
L1.lnD_TwF         -0.016577         2.722073           -0.006           0.995
L1.lnD_ChC         -0.285966         0.038745           -7.381           0.000
L2.lnD_St           0.020302         0.059295            0.342           0.732
L2.lnD_Li          -0.032683         0.061669           -0.530           0.596
L2.lnD_InR         -0.035921         0.042393           -0.847           0.397
L2.lnD_TwF         -0.595346         2.725808           -0.218           0.827
L2.lnD_ChC         -0.159228         0.039947           -3.986           0.000
L3.lnD_St          -0.052770         0.054928           -0.961           0.337
L3.lnD_Li           0.024438         0.059199            0.413           0.680
L3.lnD_InR          0.073007         0.041783            1.747           0.081
L3.lnD_TwF         -0.508729         0.522464           -0.974           0.330
L3.lnD_ChC         -0.141584         0.038818           -3.647           0.000
==============================================================================

Correlation matrix of residuals
             lnD_St    lnD_Li   lnD_InR   lnD_TwF   lnD_ChC
lnD_St     1.000000  0.839414 -0.035683 -0.004933  0.010160
lnD_Li     0.839414  1.000000 -0.072996 -0.057067  0.014727
lnD_InR   -0.035683 -0.072996  1.000000  0.028458 -0.020049
lnD_TwF   -0.004933 -0.057067  0.028458  1.000000 -0.004966
lnD_ChC    0.010160  0.014727 -0.020049 -0.004966  1.000000
```

In [107…

```python
# Forecasting
# The linear predictor is the optimal h-step ahead forecast in terms of mean-squared error
# yt(h) = ν + A1 yt(h−1) + ⋯ + Ap yt(h−p)

# We can use the forecast function to produce this forecast.
# Note that we have to specify the “initial value” for the forecast:

lag_orderSP = resultsSP.k_ar
forecastSP = resultsSP.forecast(resultsSPs.params.to_numpy(), 12)
print(f'forecastSP:\n{forecastSP}\n')

print('Forecasts: Spotify')
print('   lnD_St       lnD_Li      lnD_InR      lnD_TwF      lnD_ChC  ')
print(np.sum(forecastSP, axis=0))
```

```
forecastSP:
[[ 2.95411352e-01 -4.60089019e-01  4.24260073e-01  3.23069216e-03
   6.45902287e-01]
 [-9.15596486e-01 -9.94161026e-01  4.68782329e-01  2.44195734e-03
  -1.28555654e-01]
 [-3.49654484e-02  6.71716939e-01 -1.03642965e-01  1.93393877e-03
  -2.82891177e-01]
 [-4.50550908e-01 -7.15295557e-01 -1.06743534e-01  1.08508511e-02
   3.37459839e-01]
 [ 1.81074730e-01  1.60810299e-01  1.32015533e-01  7.48121485e-05
   1.91538698e-01]
 [-8.58876943e-02 -1.88883948e-01 -6.50467767e-02 -4.37790210e-03
  -2.03696923e-02]
```

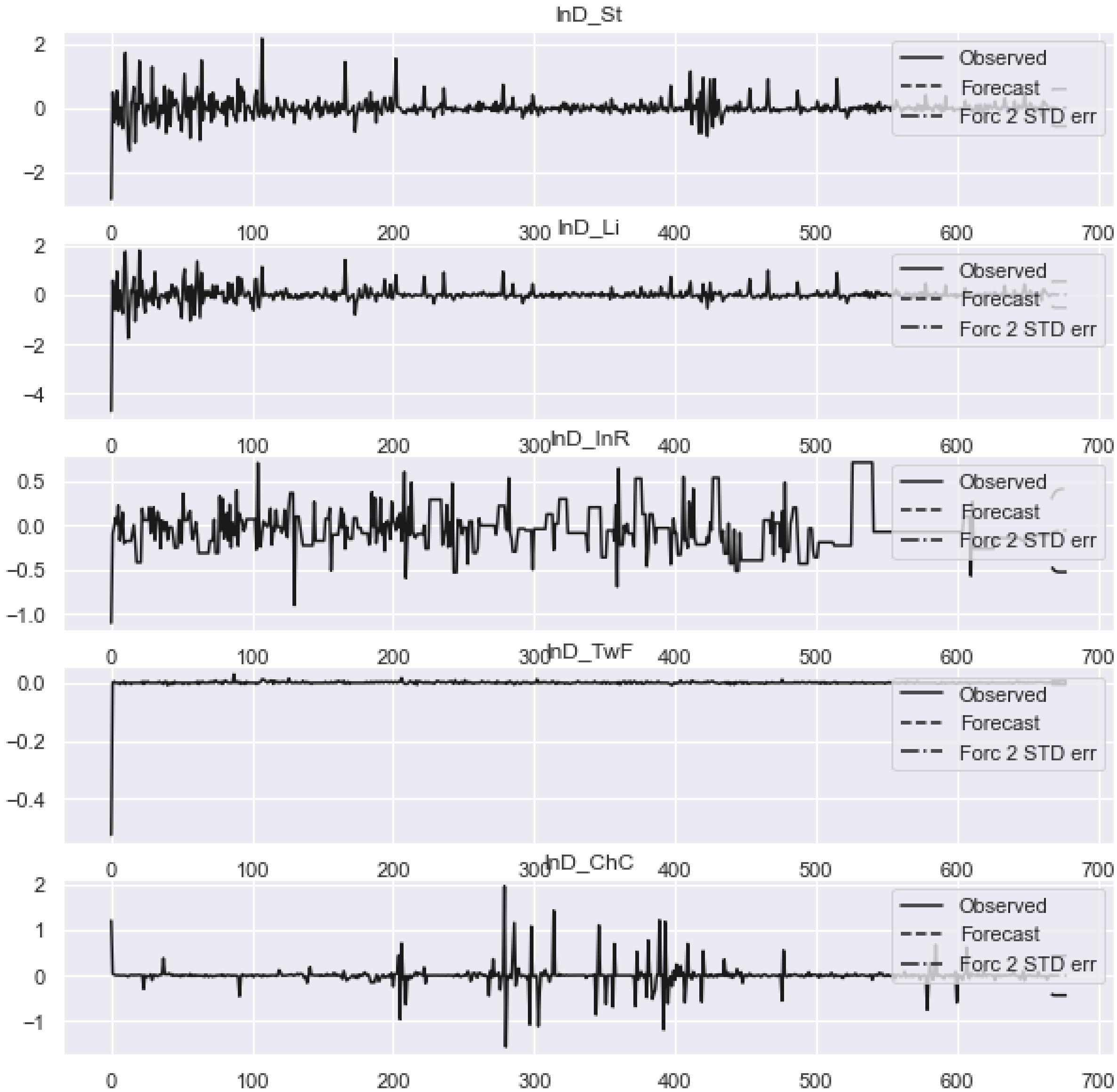
```
 [-2.74801299e-01 -3.72593491e-01 -1.45777060e-01  1.65183224e-03
  -9.87312231e-02]
 [-3.28261081e-02  4.67928905e-02 -1.09121191e-01 -6.73138559e-05
  -7.42529535e-02]
 [ 1.62393939e-01  3.19538589e-01 -1.24746178e-01  1.45624484e-03
   1.82449385e-01]
 [ 2.02361538e-01  2.91305901e-01 -1.04446365e-01 -3.23947720e-04
   9.95076406e-03]
 [-6.41755321e-02 -1.04780695e-01  5.00868322e-02  4.20137760e-03
  -3.89380657e-02]
 [ 2.98272360e-02 -2.37990162e-02 -6.45440337e-02 -1.11023405e-03
  -7.91119640e-02]]

Forecasts: Spotify
   lnD_St       lnD_Li     lnD_InR      lnD_TwF     lnD_ChC
[-0.98773468 -1.36943813  0.25107666  0.01996231  0.64445024]
```

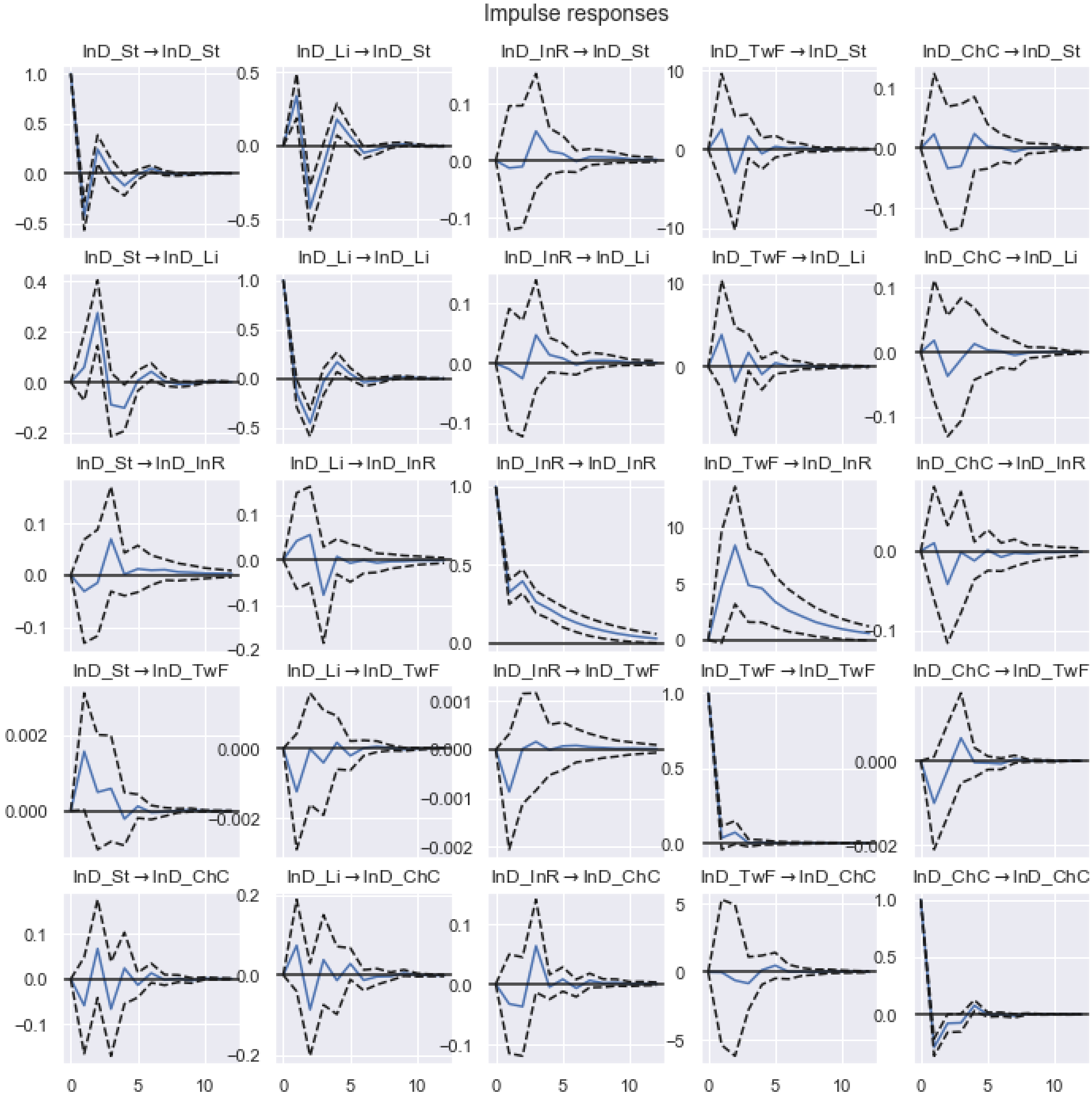
In [108…

```
# The forecast_interval function will produce the above forecast along with asymptotic sta
# These can be visualized using the plot_forecast function:
resultsSPs.plot_forecast(12)

# Impulse Response Analysis
# Impulse responses are of interest in econometric studies:
#    they are the estimated responses to a unit impulse in one of the variables.
# They are computed in practice using the MA(∞) representation of the VAR(p) process:
# Yt = μ + ∑i=0 to ∞ Φi ut−i

# We can perform an impulse response analysis by calling the irf function on a VARResults
# These can be visualized using the plot function, in either orthogonalized or non-orthogo
# Asymptotic standard errors are plotted by default at the 95% significance level, which c
irf = resultsSPs.irf(12)
irf.plot(orth=False)

# The cumulative effects Ψn=∑ni=0Φi can be plotted with the long run effects as follows:
irf.plot_cum_effects(orth=False)

# Granger causality
# One is often interested in whether a variable or group of variables is “causal” for ano
# for some definition of “causal”. In the context of VAR models,
# one can say that a set of variables are Granger-causal within one of the VAR equations.
# See Wooldridge p. 626 for a discussion of Granger causality.
# The VARResults object has the test_causality method
#  for performing either a Wald (χ2) test or an F-test.

print('++++++++++++++++++++++++++++++++++++')
print('Test for Granger causality for Streams\n')
print(resultsSPs.test_causality('lnD_St', ['lnD_Li', 'lnD_InR','lnD_TwF','lnD_ChC'], kind=

print('++++++++++++++++++++++++++++++++++++')
print('Test for Granger causality for IGReach\n')
print(resultsSPs.test_causality('lnD_InR', ['lnD_St', 'lnD_Li','lnD_TwF','lnD_ChC'], kind=

# Normality
# The white noise component ut is assumed to be normally distributed.
# While this assumption is not required for parameter estimates to be consistent or asympt
# results are generally more reliable in finite samples when residuals are Gaussian white
# To test whether this assumption is consistent with a data set, VARResults offers the tes
# Note: Stock market return data are frequently NOT normally distributed!
print('++++++++++++++++++++++++++++++++++++')
print(resultsSPs.test_normality())
```

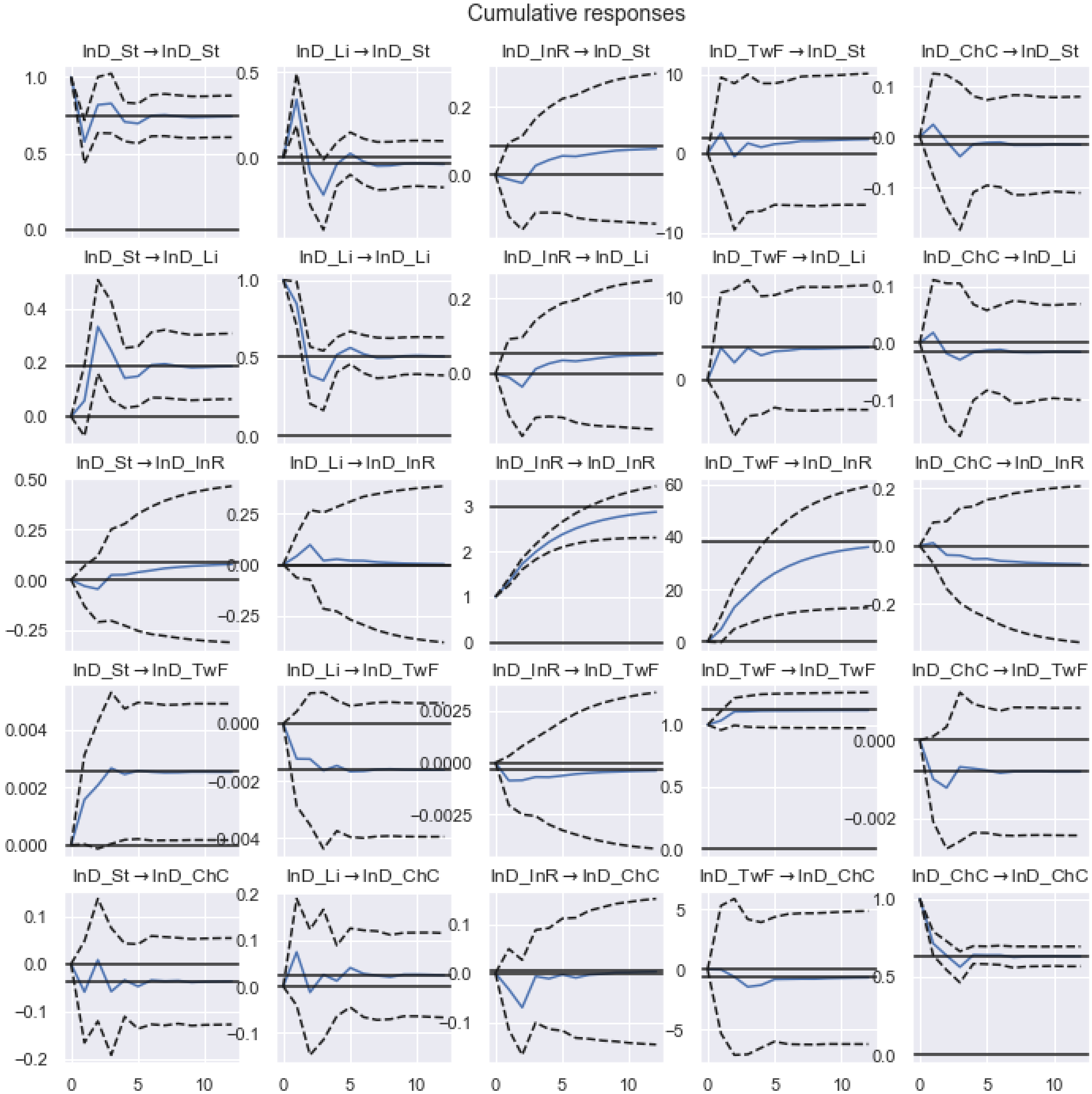
```
++++++++++++++++++++++++++++++++++++
Test for Granger causality for Streams

<statsmodels.tsa.vector_ar.hypothesis_test_results.CausalityTestResults object. H_0: ['lnD
```

```
_Li', 'lnD_InR', 'lnD_TwF', 'lnD_ChC'] do not Granger-cause lnD_St: reject at 5% significa
nce level. Test statistic: 5.013, critical value: 1.755>, p-value: 0.000>
++++++++++++++++++++++++++++++++++++
Test for Granger causality for IGReach

<statsmodels.tsa.vector_ar.hypothesis_test_results.CausalityTestResults object. H_0: ['lnD
_St', 'lnD_Li', 'lnD_TwF', 'lnD_ChC'] do not Granger-cause lnD_InR: fail to reject at 5% s
ignificance level. Test statistic: 1.654, critical value: 1.755>, p-value: 0.071>
++++++++++++++++++++++++++++++++++++
<statsmodels.tsa.vector_ar.hypothesis_test_results.NormalityTestResults object. H_0: data
generated by normally-distributed process: reject at 5% significance level. Test statisti
c: 26673.109, critical value: 18.307>, p-value: 0.000>
```





Impulse responses
lnD_St → lnD_St
lnD_Li → lnD_St
lnD_InR → lnD_St
lnD_TwF → lnD_St
lnD_ChC → lnD_St
lnD_St → lnD_Li
lnD_Li → lnD_Li
lnD_InR → lnD_Li
lnD_TwF → lnD_Li
lnD_ChC → lnD_Li
lnD_St → lnD_InR
lnD_Li → lnD_InR
lnD_InR → lnD_InR
lnD_TwF → lnD_InR
lnD_ChC → lnD_InR
lnD_St → lnD_TwF
lnD_Li → lnD_TwF
lnD_InR → lnD_TwF
lnD_TwF → lnD_TwF
lnD_ChC → lnD_TwF
lnD_St → lnD_ChC
lnD_Li → lnD_ChC
lnD_InR → lnD_ChC
lnD_TwF → lnD_ChC
lnD_ChC → lnD_ChC


Cumulative responses
lnD_St → lnD_St
lnD_Li → lnD_St
lnD_InR → lnD_St
lnD_TwF → lnD_St
lnD_ChC → lnD_St
lnD_St → lnD_Li
lnD_Li → lnD_Li
lnD_InR → lnD_Li
lnD_TwF → lnD_Li
lnD_ChC → lnD_Li
lnD_St → lnD_InR
lnD_Li → lnD_InR
lnD_InR → lnD_InR
lnD_TwF → lnD_InR
lnD_ChC → lnD_InR
lnD_St → lnD_TwF
lnD_Li → lnD_TwF
lnD_InR → lnD_TwF
lnD_TwF → lnD_TwF
lnD_ChC → lnD_TwF
lnD_St → lnD_ChC
lnD_Li → lnD_ChC
lnD_InR → lnD_ChC
lnD_TwF → lnD_ChC
lnD_ChC → lnD_ChC

```
In [ ]:
```