In [1]:
```
pip install wbgapi
```
```
Requirement already satisfied: wbgapi in c:\users\renusri\anaconda3\lib\site-packages (1.0.12)
Requirement already satisfied: PyYAML in c:\users\renusri\anaconda3\lib\site-packages (from wbgapi) (6.0)
Requirement already satisfied: tabulate in c:\users\renusri\anaconda3\lib\site-packages (from wbgapi) (0.8.10)
Requirement already satisfied: requests in c:\users\renusri\anaconda3\lib\site-packages (from wbgapi) (2.28.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\renusri\anaconda3\lib\site-packages (from requests->wbgapi) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\renusri\anaconda3\lib\site-packages (from requests->wbgapi) (1.26.11)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\renusri\anaconda3\lib\site-packages (from requests->wbgapi) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\renusri\anaconda3\lib\site-packages (from requests->wbgapi) (2022.9.14)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats
import wbgapi as wb
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [30]:
```
def world(a,b,c): #defining the function for creating a datafram
 data = wb.data.DataFrame(a, b, mrv = c)
 data_t = data.T
 worlddata = wb.data.DataFrame(a, mrv = c)
 return data_t, worlddata
```

In [4]:
```
country_codes = ["CAN","GBR","CHN"] #country codes
```

In [5]:
```
PG = {"SP.POP.GROW" : "POPULATION GROWTH(%)"}
CO2 = {"EN.ATM.CO2E.KT" : "CO2 EMISSION(KT)"}
NO = {"EN.ATM.NOXE.ZG" : "NITROUS OXIDE(%)"}
FOR = {"AG.LND.FRST.ZS" : "FOREST AREA(%)"}
UP = {"SP.URB.GROW" : "URBAN SP.URB.GROWPOPULATION"}
```

In [6]:
```
indicator_ids = {"SP.POP.GROW","EN.ATM.CO2E.KT","EN.ATM.NOXE.ZG","AG.LND.FRST.ZS","SP.URB.GROW"}
```

In [7]:
```
wb.series.info(indicator_ids)
```

Out[7]:

id	value
EN.ATM.CO2E.KT	CO2 emissions (kt)
AG.LND.FRST.ZS	Forest area (% of land area)
EN.ATM.NOXE.ZG	Nitrous oxide emissions (% change from 1990)
SP.POP.GROW	Population growth (annual %)
SP.URB.GROW	Urban population growth (annual %)
	5 elements

In [8]:
```
FOR_R, FOR_world = world(FOR.keys(),country_codes,30)
```

In [9]:
```
CO2_R, CO2_world = world(CO2.keys(),country_codes,30)
```

In [10]:
```
UP_R, UP_world = world(UP.keys(),country_codes,30)
```

In [11]:
```
PG_R, PG_world = world(PG.keys(),country_codes,30)
```

In [12]:
```
NO_R, NO_world = world(NO.keys(),country_codes,30)
```

In [13]:
```
N =  NO_world.mean()
NO1 = pd.DataFrame(N)
NO1.reset_index(level=0, inplace=True)
NOA = NO1.rename(columns = {"index": "year", 0: "mean"})
NOA
```

Out[13]:

	year	mean
0	YR1991	-2.713231
1	YR1992	3.942821
2	YR1993	-2.032096
3	YR1994	0.681969
4	YR1995	2.837167
5	YR1996	2.741599
6	YR1997	8.812441
7	YR1998	12.306721
8	YR1999	8.476841
9	YR2000	16.046840
10	YR2001	15.049856
11	YR2002	12.493209
12	YR2003	12.719952
13	YR2004	15.211419
14	YR2005	16.908735
15	YR2006	19.868958
16	YR2007	30.753311
17	YR2008	28.804922
18	YR2009	29.981987
19	YR2010	31.944920
20	YR2011	32.441431
21	YR2012	33.645745

In [14]:
```
C = CO2_world.mean()
C1 = pd.DataFrame(C)
C1.reset_index(level=0, inplace=True)
CO2A = C1.rename(columns ={"index": "year", 0: "mean"})
CO2A
```

Out[14]:

	year	mean
0	YR1990	7.596622e+05
1	YR1991	7.626341e+05
2	YR1992	7.496100e+05
3	YR1993	7.538010e+05
4	YR1994	7.551963e+05
5	YR1995	7.797061e+05
6	YR1996	7.944514e+05
7	YR1997	8.066165e+05
8	YR1998	8.153585e+05
9	YR1999	8.175166e+05
10	YR2000	8.411801e+05
11	YR2001	8.588245e+05
12	YR2002	8.715628e+05
13	YR2003	9.192566e+05
14	YR2004	9.690122e+05
15	YR2005	1.012921e+06
16	YR2006	1.054999e+06
17	YR2007	1.100060e+06
18	YR2008	1.115300e+06
19	YR2009	1.113084e+06
20	YR2010	1.184603e+06
21	YR2011	1.233433e+06
22	YR2012	1.255654e+06
23	YR2013	1.283613e+06
24	YR2014	1.289604e+06
25	YR2015	1.281995e+06
26	YR2016	1.285745e+06
27	YR2017	1.310145e+06
28	YR2018	1.344521e+06
29	YR2019	1.353171e+06

In [15]:
```
UP = UP_world.mean()
UP1 = pd.DataFrame(UP)
UP1.reset_index(level=0, inplace=True)
UPA = UP1.rename(columns ={"index": "year", 0: "mean"})
UPA
```

Out[15]:

	year	mean
0	YR1992	2.620064
1	YR1993	2.480620
2	YR1994	2.322222
3	YR1995	2.267111
4	YR1996	2.403954
5	YR1997	2.363944
6	YR1998	2.273382
7	YR1999	2.190812
8	YR2000	2.151387
9	YR2001	2.181543
10	YR2002	2.206183
11	YR2003	2.175917
12	YR2004	2.166844
13	YR2005	2.203761
14	YR2006	2.264110
15	YR2007	2.205984
16	YR2008	2.249005
17	YR2009	2.177430
18	YR2010	2.086551
19	YR2011	1.986108
20	YR2012	2.020726
21	YR2013	2.062696
22	YR2014	2.040090
23	YR2015	2.005127
24	YR2016	1.947165
25	YR2017	1.869566
26	YR2018	1.858017
27	YR2019	1.841975
28	YR2020	1.737996
29	YR2021	1.592221

In [16]:
```
PG = PG_world.mean()
PG1 = pd.DataFrame(PG)
PG1.reset_index(level=0, inplace=True)
PGA = PG1.rename(columns ={"index": "year", 0: "mean"})
PGA
```

Out[16]:

	year	mean
0	YR1992	1.743989
1	YR1993	1.650362
2	YR1994	1.541706
3	YR1995	1.519780
4	YR1996	1.665514
5	YR1997	1.624679
6	YR1998	1.535156
7	YR1999	1.455639
8	YR2000	1.432954
9	YR2001	1.404735
10	YR2002	1.440759
11	YR2003	1.431754
12	YR2004	1.431544
13	YR2005	1.461876
14	YR2006	1.543733
15	YR2007	1.542632
16	YR2008	1.520084
17	YR2009	1.460628
18	YR2010	1.379407
19	YR2011	1.297343
20	YR2012	1.349639
21	YR2013	1.401124
22	YR2014	1.375687
23	YR2015	1.323819
24	YR2016	1.271352
25	YR2017	1.198958
26	YR2018	1.179653
27	YR2019	1.156251
28	YR2020	1.054799
29	YR2021	0.905508

In [17]:
```
FOR = FOR_world.mean()
FOR1 = pd.DataFrame(FOR)
FOR1.reset_index(level=0, inplace=True)
FORA = FOR1.rename(columns ={"index": "year", 0: "mean"})
FORA
```

Out[17]:

	year	mean
0	YR1991	33.654898
1	YR1992	33.184709
2	YR1993	33.083255
3	YR1994	33.072996
4	YR1995	33.024235
5	YR1996	32.975222
6	YR1997	32.925691
7	YR1998	32.900866
8	YR1999	32.851716
9	YR2000	32.806406
10	YR2001	32.766187
11	YR2002	32.728906
12	YR2003	32.687015
13	YR2004	32.634522
14	YR2005	32.593613
15	YR2006	32.633317
16	YR2007	32.602219
17	YR2008	32.569917
18	YR2009	32.536616
19	YR2010	32.502827
20	YR2011	32.224482
21	YR2012	32.093941
22	YR2013	32.050349
23	YR2014	32.009783
24	YR2015	31.970608
25	YR2016	31.904915
26	YR2017	31.866522
27	YR2018	31.817643
28	YR2019	31.764707
29	YR2020	31.714485

In [18]:
```
PG_R
```

Out[18]:

economy	CAN	CHN	GBR
YR1992	1.183676	1.225536	0.270431
YR1993	1.098931	1.149619	0.239745
YR1994	1.095258	1.130261	0.254586
YR1995	1.034769	1.086509	0.264547
YR1996	1.045312	1.048142	0.254626
YR1997	0.993789	1.023450	0.257553
YR1998	0.829909	0.959550	0.291406
YR1999	0.812843	0.865851	0.333406
YR2000	0.931282	0.787957	0.357301
YR2001	1.086351	0.726381	0.384976
YR2002	1.087448	0.670000	0.423337
YR2003	0.901373	0.622861	0.465641
YR2004	0.933021	0.593933	0.568943
YR2005	0.944467	0.588125	0.686611
YR2006	1.010335	0.558374	0.735049
YR2007	0.971135	0.522272	0.778666
YR2008	1.082907	0.512387	0.787033
YR2009	1.141758	0.497381	0.756391
YR2010	1.111864	0.482960	0.783889
YR2011	0.978698	0.546458	0.781507
YR2012	1.085817	0.678345	0.695353
YR2013	1.056591	0.666073	0.669741
YR2014	1.005338	0.630326	0.736464
YR2015	0.746339	0.581456	0.792368
YR2016	1.132349	0.573051	0.757874
YR2017	1.199521	0.605245	0.679374
YR2018	1.412456	0.467672	0.605929
YR2019	1.436137	0.354741	0.564131
YR2020	1.152797	0.238041	0.365409
YR2021	0.547707	0.089252	0.365410

In [19]:
```
UP_R
```

Out[19]:

economy	CAN	CHN	GBR
YR1992	1.531545	4.425117	0.347214
YR1993	1.442998	4.301548	0.316469
YR1994	1.434272	4.245841	0.332530
YR1995	1.371352	4.159638	0.341154
YR1996	1.400009	4.086052	0.331174
YR1997	1.491581	4.008286	0.332766
YR1998	1.318880	3.908079	0.367839
YR1999	1.293117	3.770089	0.408506
YR2000	1.404226	3.649253	0.433616
YR2001	1.503209	4.059566	0.512040
YR2002	1.185132	4.198001	0.798501
YR2003	1.000212	4.078404	0.834360
YR2004	1.030512	3.975372	0.933797
YR2005	1.040619	3.882473	1.045134
YR2006	1.123847	3.674728	1.089795
YR2007	1.199019	3.511256	1.127190
YR2008	1.309030	3.433958	1.131869
YR2009	1.364894	3.338102	1.095105
YR2010	1.333269	3.255365	1.117771
YR2011	1.174953	3.123377	1.110599
YR2012	1.136362	3.130657	1.022146
YR2013	1.107110	3.048365	0.993032
YR2014	1.054601	2.953497	1.056285
YR2015	0.796808	2.842871	1.108748
YR2016	1.182792	2.775642	1.072052
YR2017	1.261001	2.739664	0.988959
YR2018	1.487416	2.503401	0.912161
YR2019	1.523308	2.290177	0.868231
YR2020	1.250930	2.078140	0.665012
YR2021	0.659219	1.838530	0.661742

In [20]:
```
p=PG_R.rename(columns ={"index": "year", 0: "mean"})
pg_r = p.rename_axis("year")
```

In [21]:
```
u=UP_R.rename(columns ={"index": "year", 0: "mean"})
up_r = u.rename_axis("year")
```

In [22]:
```
n=NO_R.rename(columns ={"index": "year", 0: "mean"})
no_r = n.rename_axis("year")
```

In [23]:
```
c=CO2_R.rename(columns ={"index": "year", 0: "mean"})
co2_r = c.rename_axis("year")
```

In [24]:
```
f=FOR_R.rename(columns ={"index": "year", 0: "mean"})
for_r = f.rename_axis("year")
```

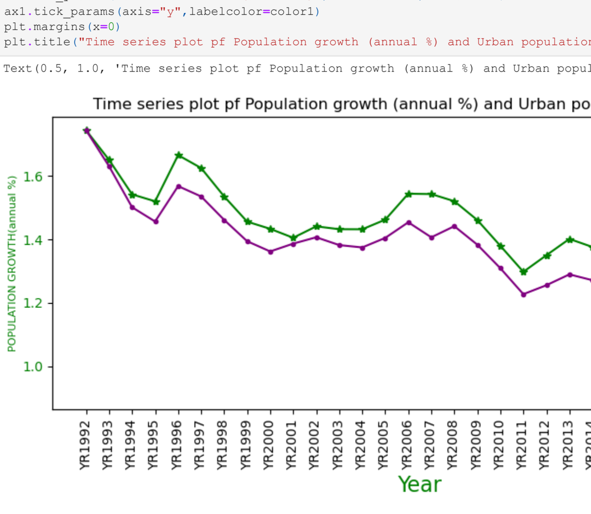
In [25]:
```
# Making plot between CO2 and NO2
fig,ax=plt.subplots(figsize=[8,4])
color1="red"
color2="blue"
ax.plot(NOA["year"],NOA["mean"],marker="*",color=color1)
ax.set_ylabel("NITROUS GAS EMMISION(Kt)",color=color1,fontsize=12)
ax.set_xlabel("Year",color=color1,fontsize=12)
ax.tick_params(axis="y",labelcolor=color1)
plt.xticks(rotation=90)
ax1=ax.twinx()
ax1.plot(CO2A["year"],CO2A["mean"],color=color2,marker="o")
ax1.set_ylabel("CO2 GAS EMMISION(Kt)",color=color2,fontsize=12)
ax1.tick_params(axis="y",labelcolor=color1)
plt.margins(x=0)
plt.title("Time series plot of CO2 and Nitrous Oxide Emission")
```

Out[25]:
```
Text(0.5, 1.0, 'Time series plot of CO2 and Nitrous Oxide Emission')
```

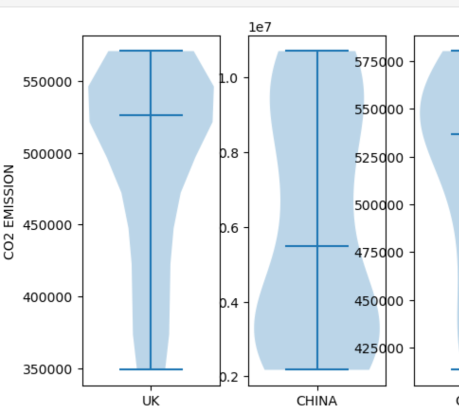
In [26]:
```
# Making plot between Population growth and Urban population
fig,ax=plt.subplots(figsize=[10,4])
color1="green"
color2="purple"
color3="red"
ax.plot(PGA["year"],PGA["mean"],marker="*",color=color1)
ax.set_ylabel("POPULATION GROWTH(annual %)",color=color1,fontsize=8)
ax.set_xlabel("Year",color=color1,fontsize=16)
ax.tick_params(axis="y",labelcolor=color1)
plt.xticks(rotation=90)
ax1=ax.twinx()
ax1.plot(UPA["year"],UPA["mean"],color=color2,marker=".")
ax1.set_ylabel("URBAN POPULATION(annual %)",color=color2,fontsize=8)
ax1.tick_params(axis="y",labelcolor=color1)
plt.margins(x=0)
plt.title("Time series plot pf Population growth (annual %) and Urban population (annual %)")
```

Out[26]:
```
Text(0.5, 1.0, 'Time series plot pf Population growth (annual %) and Urban population (annual %)')
```

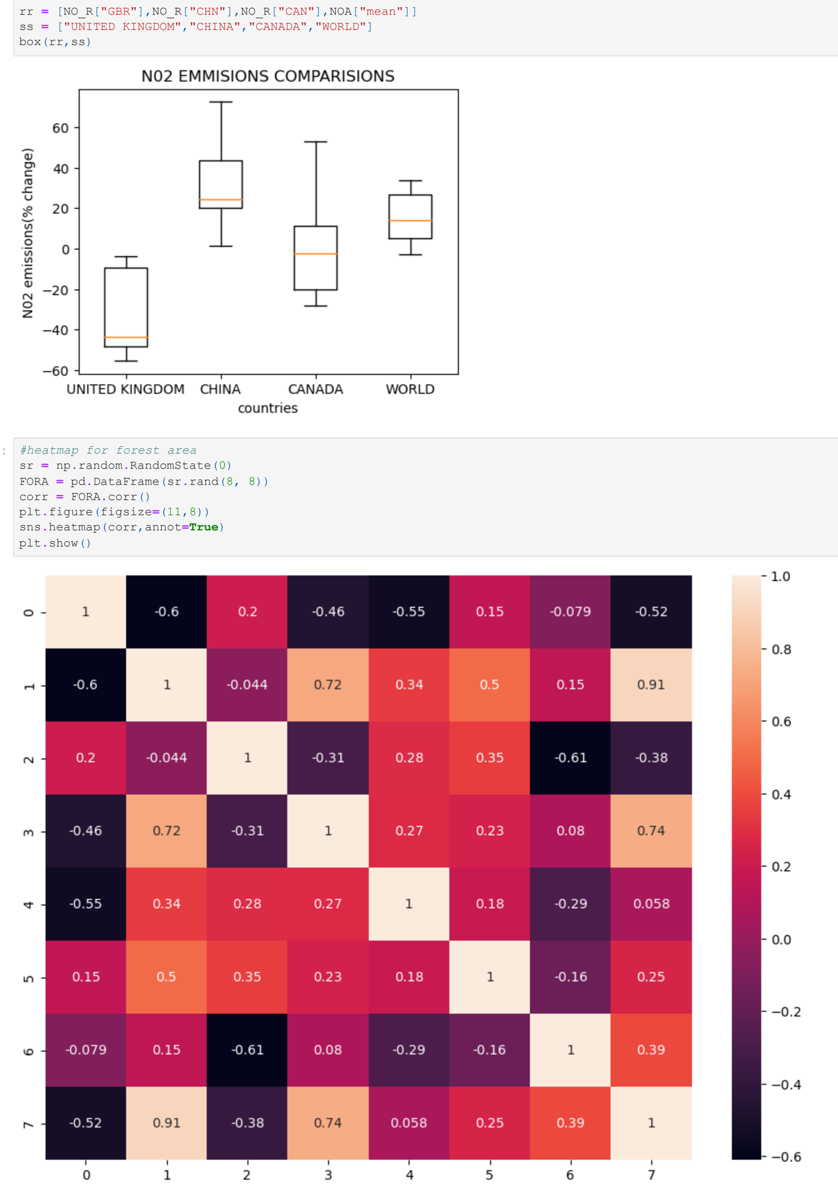
In [27]:
```
fig, (ax1, ax2, ax3) = plt.subplots(nrows=1, ncols=3)
ax1.violinplot(CO2_R["GBR"], showmedians=True, points=10)
ax1.set_xticks([1])
ax1.set_ylabel("CO2 EMISSION")
ax1.set_xticklabels(["UK"])
ax2.violinplot(CO2_R["CHN"], showmedians=True, points=100)
ax2.set_xticks([1])
ax2.set_xticklabels(["CHINA"])
ax3.violinplot(CO2_R["CAN"], showmedians=True, points=500)
ax3.set_xticks([1])
ax3.set_xticklabels(["CANADA"])
plt.show()
```



In [28]:
```
def box(x,y):
 fig = plt.figure(figsize = (4,3))
 ax = fig.add_axes([0,0,1,1])
 cc = ax.boxplot(x)
 ax.set_xlabel("countries")
 ax.set_ylabel("N02 emissions(% change)" )
 ax.set_title("N02 EMMISIONS COMPARISIONS")
 ax.set_xticks([1,2,3,4])
 ax.set_xticklabels(y)
 plt.show()
 return
rr = [NO_R["GBR"],NO_R["CHN"],NO_R["CAN"],NOA["mean"]]
ss = ["UNITED KINGDOM","CHINA","CANADA","WORLD"]
box(rr,ss)
```



In [29]:
```
#heatmap for forest area
sr = np.random.RandomState(0)
FORA = pd.DataFrame(sr.rand(8, 8))
corr = FORA.corr()
plt.figure(figsize=(11,8))
sns.heatmap(corr,annot=True)
plt.show()
```



In [ ]: