In [317…
```python
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 [19]:
```python
def world(x,y,z):  #defining the function for creating a datafram
    data = wb.data.DataFrame(x, y, mrv = z)
    data_t = data.T
    worlddata = wb.data.DataFrame(x, mrv = z)
    return data_t, worlddata
```

In [2]:
```python
country_codes = ["CAN","GBR","CHN"] #country codes
```

In [433…
```python
NO = {"EN.ATM.NOXE.ZG" : "NITROUS OXIDE(%)"}
CO2 = { "EN.ATM.CO2E.KT" : "C02 EMMISION (KT)"}
ARL = { "AG.LND.ARBL.ZS" : "ARABLE LAND(%)"}
FOR = {"AG.LND.FRST.ZS" : "FOREST AREA(%)"}
```

In [434…
```python
indicator_ids = {"EN.ATM.NOXE.ZG","EN.ATM.CO2E.KT","AG.LND.ARBL.ZS","AG.LND.FRST.ZS"}
```

In [435…
```python
wb.series.info(indicator_ids)
```

Out[435…

id	value
AG.LND.ARBL.ZS	Arable land (% of land area)
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)
	4 elements

In [125…
```python
NO_T, NO_world = world(NO.keys(),country_codes,30)
```

In [268…
```python
FOR_T, FOR_world = world(FOR.keys(),country_codes,30)
```

In [13]:
```python
ARL_T, ARL_world = world(ARL.keys(),country_codes,30)
```

In [14]:
```python
CO2_T, CO2_world = world(CO2.keys(),country_codes,30)
```

In [43]:
```python
C = CO2_world.mean()
C1 = pd.DataFrame(C)
C1.reset_index(level=0, inplace=True)
CO2W = C1.rename(columns ={"index": "year", 0: "mean"})
CO2W
```

Out[43]:

	year	mean
0	YR1990	106966.421053
1	YR1991	107401.099476
2	YR1992	107107.591623
3	YR1993	107791.518325
4	YR1994	108357.905759
5	YR1995	111451.361257
6	YR1996	113809.214660
7	YR1997	115640.366492
8	YR1998	116237.958115
9	YR1999	116785.759162
10	YR2000	120379.528796
11	YR2001	122421.623037
12	YR2002	124215.340314
13	YR2003	129841.413613
14	YR2004	135912.041885
15	YR2005	141116.073298
16	YR2006	145755.078534
17	YR2007	151377.329843
18	YR2008	152734.031414
19	YR2009	151075.445026
20	YR2010	160257.801047
21	YR2011	165278.691099
22	YR2012	167559.267016
23	YR2013	170707.853403
24	YR2014	170869.738220
25	YR2015	170137.591623
26	YR2016	170123.246073
27	YR2017	172245.654450
28	YR2018	175806.282723

In [ ]:

In [305…
```python
F = FOR_world.mean()
F1 = pd.DataFrame(F)
F1.reset_index(level=0, inplace=True)
FORW = F1.rename(columns ={"index": "year", 0: "mean"})
FORW
```

Out[305…

	year	mean
0	YR1991	33.219641
1	YR1992	33.178420
2	YR1993	33.116628
3	YR1994	33.106409
4	YR1995	33.058120
5	YR1996	33.009579
6	YR1997	32.960539
7	YR1998	32.935895
8	YR1999	32.887238
9	YR2000	32.455045
10	YR2001	32.813731
11	YR2002	32.777145
12	YR2003	32.736511
13	YR2004	32.691259
14	YR2005	32.652563
15	YR2006	32.655071
16	YR2007	32.624250
17	YR2008	32.592436
18	YR2009	32.559629
19	YR2010	32.130313
20	YR2011	32.045912
21	YR2012	32.010853
22	YR2013	31.967086
23	YR2014	31.925427
24	YR2015	31.885382
25	YR2016	31.810761
26	YR2017	31.764381
27	YR2018	31.720355
28	YR2019	32.140857
29	YR2020	32.089941

In [44]:
```python
N = NO_world.mean()
N1 = pd.DataFrame(N)
N1.reset_index(level=0, inplace=True)
NOW = N1.rename(columns ={"index": "year", 0: "mean"})
NOW
```

Out[44]:

	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 [31]:
```python
A = ARL_world.mean()
A1 = pd.DataFrame(A)
A1.reset_index(level=0, inplace=True)
ARLW = A1.rename(columns ={"index": "year", 0: "mean"})
ARLW
```

Out[31]:

	year	mean
0	YR1989	12.801786
1	YR1990	12.769271
2	YR1991	12.607600
3	YR1992	13.335661
4	YR1993	13.493982
5	YR1994	13.399176
6	YR1995	13.343575
7	YR1996	13.352387
8	YR1997	13.376995
9	YR1998	13.378886
10	YR1999	13.400522
11	YR2000	13.420427
12	YR2001	13.368009
13	YR2002	13.352498
14	YR2003	13.407669
15	YR2004	13.419622
16	YR2005	13.380016
17	YR2006	13.429336
18	YR2007	13.389845
19	YR2008	13.454144
20	YR2009	13.490118
21	YR2010	13.522882
22	YR2011	13.544884
23	YR2012	13.599025
24	YR2013	13.567235
25	YR2014	13.538021
26	YR2015	13.593621
27	YR2016	13.668183
28	YR2017	13.721972
29	YR2018	13.707560

In [58]:
```python
ARL_T
```

Out[58]:

economy	CAN	CHN	GBR
YR1989	4.619886	13.038759	27.702228
YR1990	4.613974	13.207996	27.363287
YR1991	4.608443	13.320891	27.144215
YR1992	4.602709	13.110486	27.090481
YR1993	4.596909	12.940082	25.176704
YR1994	4.591109	12.876101	24.432687
YR1995	4.585309	12.757302	24.536023
YR1996	4.579621	12.731638	25.205638
YR1997	4.577167	12.698985	26.321663
YR1998	4.574713	12.697844	25.846319
YR1999	4.572259	12.697446	24.457488
YR2000	4.569805	12.697062	24.288017
YR2001	4.567240	12.759859	23.357996
YR2002	4.530098	12.801435	23.916009
YR2003	4.492956	12.821155	23.391064
YR2004	4.455814	12.840873	24.052412
YR2005	4.418672	12.866959	23.680403
YR2006	4.381530	12.903017	25.189104
YR2007	4.337026	12.896634	25.151903
YR2008	4.292523	12.922614	24.821229
YR2009	4.248019	12.942968	25.003100
YR2010	4.203516	12.877841	24.676559
YR2011	4.158678	12.800417	25.056835
YR2012	4.186451	12.728202	25.676849
YR2013	4.214223	12.707193	25.897574
YR2014	4.241996	12.695893	25.765717
YR2015	4.269881	12.689357	24.846030
YR2016	4.297542	12.680817	24.908075
YR2017	4.295200	12.678503	25.143138
YR2018	4.315053	12.678246	24.956754

In [60]:
```python
CO2_T
```

Out[60]:

economy	CAN	CHN	GBR
YR1990	419120.0	2173360.0	556690.0
YR1991	412930.0	2302190.0	570340.0
YR1992	426000.0	2418180.0	552980.0
YR1993	422680.0	2643530.0	540520.0
YR1994	438820.0	2763900.0	535390.0
YR1995	449500.0	3084870.0	524290.0
YR1996	463160.0	3064880.0	544660.0
YR1997	478530.0	3128650.0	522330.0
YR1998	486760.0	3232120.0	525030.0
YR1999	495140.0	3149200.0	522360.0
YR2000	515000.0	3344090.0	530120.0
YR2001	506940.0	3526750.0	544480.0
YR2002	525470.0	3808330.0	530230.0
YR2003	545360.0	4413300.0	542080.0
YR2004	537250.0	5121830.0	542320.0
YR2005	549970.0	5819120.0	540600.0
YR2006	541070.0	6431910.0	541640.0
YR2007	571310.0	6988210.0	529160.0
YR2008	550200.0	7195010.0	514410.0
YR2009	520800.0	7715110.0	465120.0
YR2010	535030.0	8470570.0	481790.0
YR2011	547720.0	9278250.0	444420.0
YR2012	547140.0	9533210.0	467080.0
YR2013	556350.0	9936680.0	453060.0
YR2014	561830.0	9894940.0	414850.0
YR2015	558800.0	9830430.0	400370.0
YR2016	550480.0	9814310.0	378890.0
YR2017	562260.0	10017770.0	366380.0
YR2018	574400.0	10313460.0	358800.0

In [303…
```python
n=NO_T.rename(columns ={"index": "year", 0: "mean"})
no_t = a.rename_axis("year")
```

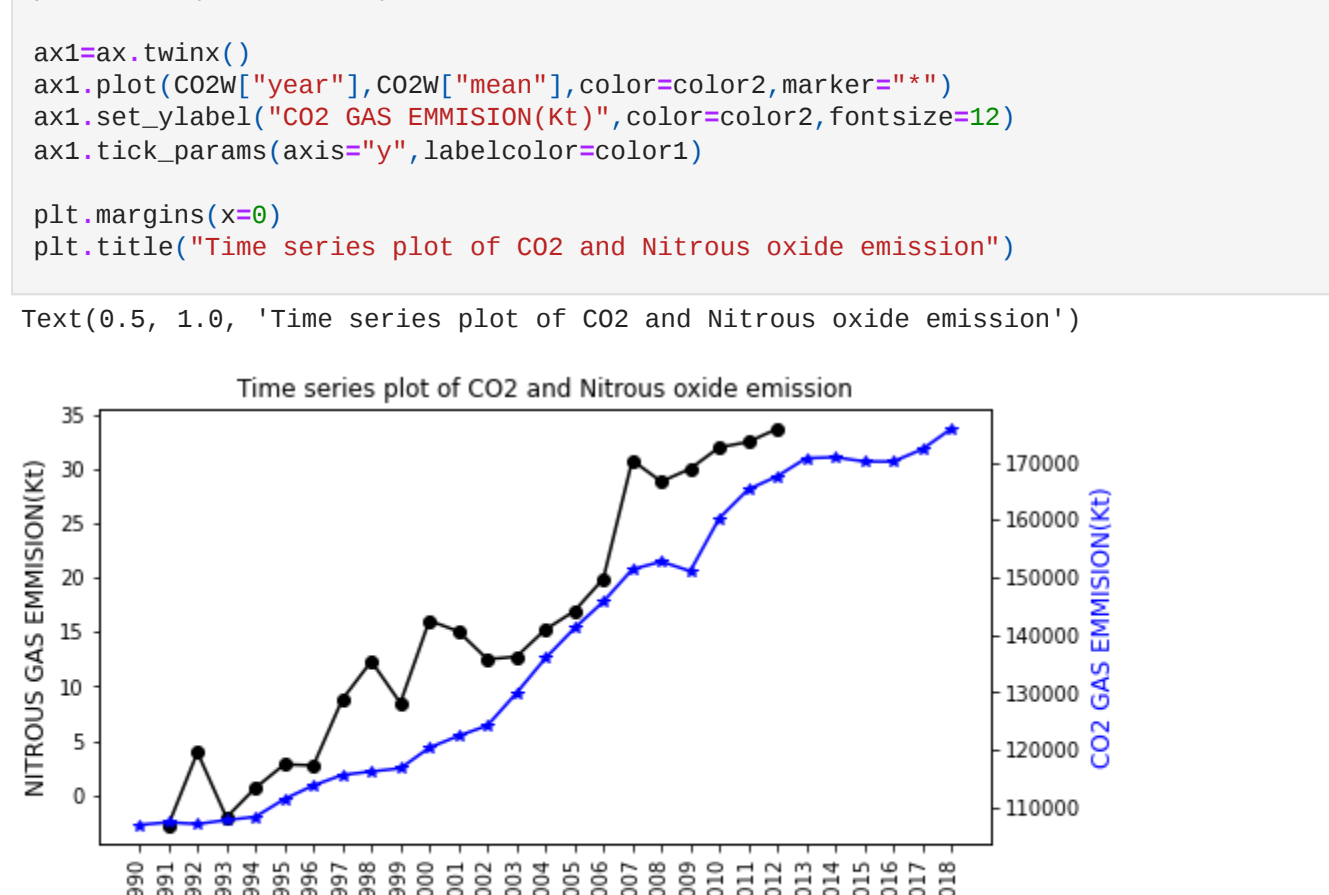
In [302…
```python
c=CO2_T.rename(columns ={"index": "year", 0: "mean"})
co2_t = c.rename_axis("year")
```

In [299…
```python
f=FOR_T.rename(columns ={"index": "year", 0: "mean"})
fo_t = f.rename_axis("year")
```

In [298…
```python
a=ARL_T.rename(columns ={"index": "year", 0: "mean"})
arl_t = a.rename_axis("year")
```

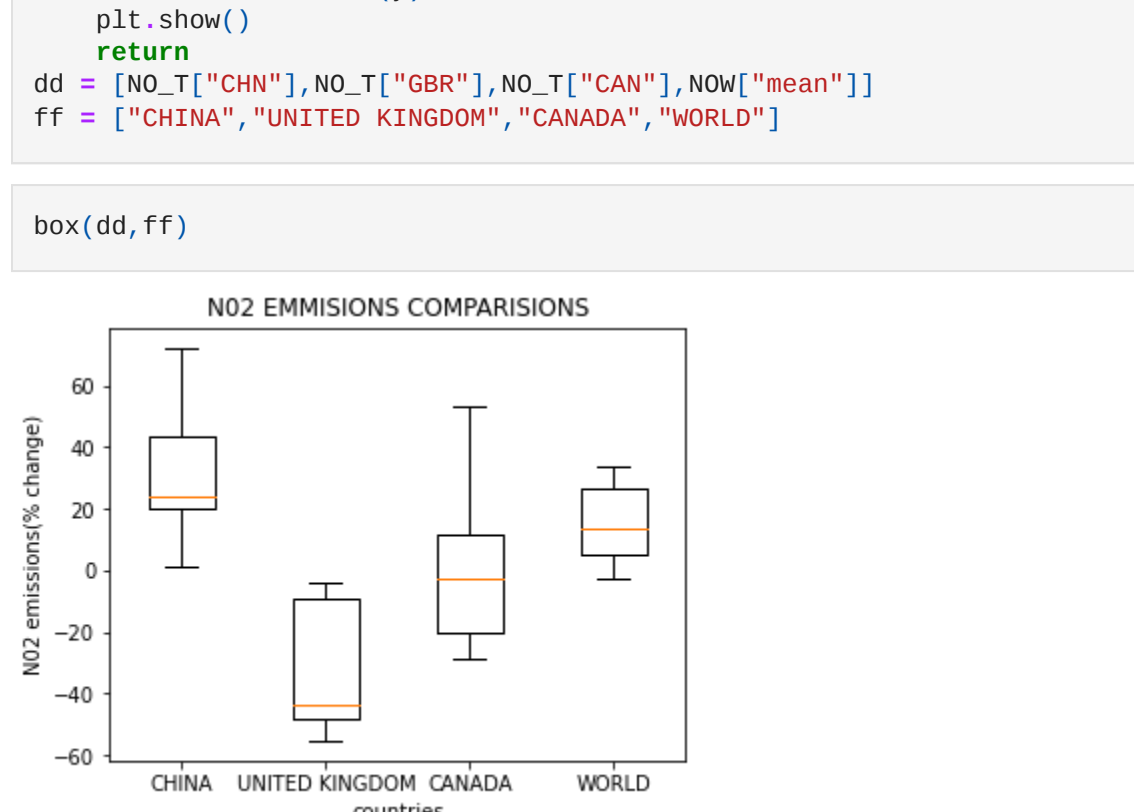
In [151…
```python
# Making plot between foreast cover and arable land for whole world
fig,ax=plt.subplots(figsize=[10,4])
color1="black"
color2="orange"
color3 = "red"
ax.plot(FORW["year"],FORW["mean"],marker="o",color=color1)
ax.set_ylabel("FOREST AREA(% of total land)",color=color1,fontsize=8)
ax.set_xlabel("Year",color=color1,fontsize=16)
ax.tick_params(axis="y",labelcolor=color1)
plt.xticks(rotation=90)

ax1=ax.twinx()
ax1.plot(ARLW["year"],ARLW["mean"],color=color2,marker="*")
ax1.set_ylabel("AREBLE AREA(% of total land)",color=color2,fontsize=8)
ax1.tick_params(axis="y",labelcolor=color1)

plt.margins(x=0)
plt.title("Time series plot of Forest area and arable land (% of total land)")
```

Out[151…
```
Text(0.5, 1.0, 'Time series plot of Forest area and arable land (% of total land)')
```



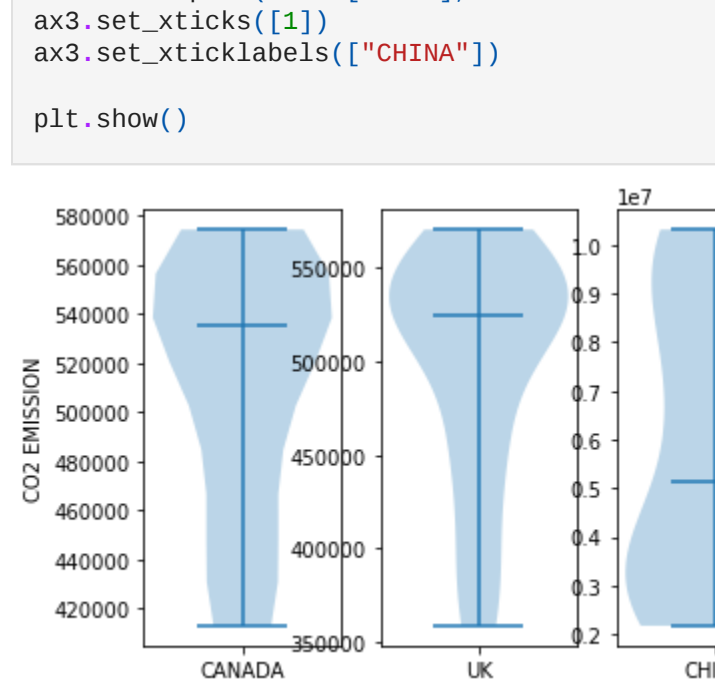
In [429…
```python
# Making plot between co2 and total GHG for whole world
fig,ax=plt.subplots(figsize=[8,4])
color1="black"
color2="blue"
ax.plot(NOW["year"],NOW["mean"],marker="o",color=color1)
ax.set_ylabel("NITROUS GAS EMMISION(Kt)",color=color1,fontsize=12)
ax.set_xlabel("Year",color=color1,fontsize=12)
ax.tick_params(axis="y",labelcolor=color1)
plt.xticks(rotation=90)

ax1=ax.twinx()
ax1.plot(CO2W["year"],CO2W["mean"],color=color2,marker="*")
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[429…
```
Text(0.5, 1.0, 'Time series plot of CO2 and Nitrous oxide emission')
```

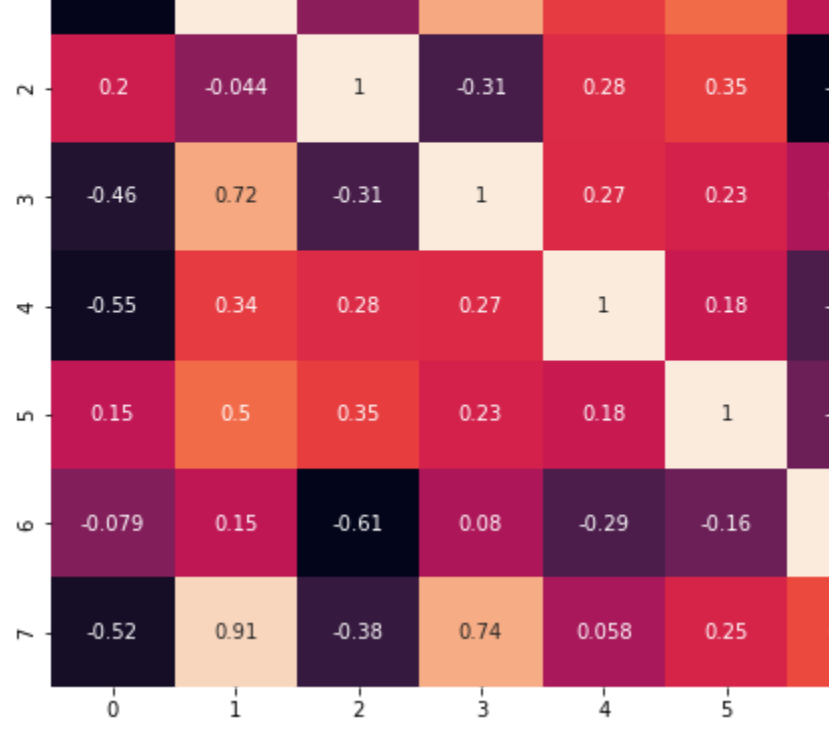


In [344…
```python
def box(x,y):
    fig = plt.figure(figsize = (4,3))
    ax = fig.add_axes([0,0,1,1])
    cc = ax.boxplot(x)
    ax.set_xlabel("countries")
    ax.set_ylabel("N02 emissions(% change)" )
    ax.set_title("N02 EMMISIONS COMPARISIONS")
    ax.set_xticks([1,2,3,4])
    ax.set_xticklabels(y)
    plt.show()
    return
dd = [NO_T["CHN"],NO_T["GBR"],NO_T["CAN"],NOW["mean"]]
ff = ["CHINA","UNITED KINGDOM","CANADA","WORLD"]
```

In [345…
```python
box(dd,ff)
```



In [427…
```python
fig, (ax1, ax2, ax3) = plt.subplots(nrows=1, ncols=3)
ax1.violinplot(CO2_T["CAN"], showmedians=True, points=10)
ax1.set_xticks([1])
ax1.set_ylabel("CO2 EMISSION")
ax1.set_xticklabels(["CANADA"])

ax2.violinplot(CO2_T["GBR"], showmedians=True, points=100)
ax2.set_xticks([1])
ax2.set_xticklabels(["UK"])

ax3.violinplot(CO2_T["CHN"], showmedians=True, points=500)
ax3.set_xticks([1])
ax3.set_xticklabels(["CHINA"])

plt.show()
```



In [414…
```python
rs = np.random.RandomState(0)
FORW = pd.DataFrame(rs.rand(8, 8))
corr = FORw.corr()
plt.figure(figsize=(11,8))
sns.heatmap(corr,annot=True)
plt.show()
```


In [ ]: