Muhammad Sarmad Sajjad

Student ID: 21031175

In [1]:
```python
import pandas as pd
import matplotlib.pyplot as plt
import wbgapi as wb
from matplotlib.ticker import FuncFormatter
import plotly.graph_objs as go
import plotly.plotly as py
```

In [2]:
```python
def read_data(filename):
    '''
    Returns two dataframes one with country as columns and the other with years as columns
    '''
    df = pd.read_csv(filename,skiprows=3)
    yrs_df = df.drop(labels=['Country Code', 'Indicator Name','Indicator Code','Unnamed: 65'],axis=1)
    cou_df = yrs_df.transpose()
    return yrs_df, cou_df
```

In [3]:
```python
#reading emmissions data
em_df1, em_df2 = read_data('API_EN.ATM.CO2E.PC_DS2_en_csv_v2_3919482.csv')
```

In [4]:
```python
em_df1
```

Out[4]:

	Country Name	1960	1961	1962	1963	1964	1965	1966	1967	1968	...	2011	2012	2013	2014	2015	2016	2017	2018	2019
0	Aruba	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Africa Eastern and Southern	0.906060	0.922474	0.930816	0.940570	0.996033	1.047280	1.033908	1.052204	1.079727	...	1.005338	1.021646	1.031833	1.041145	0.987393	0.971016	0.959978	0.933541	NaN
2	Afghanistan	0.046057	0.053589	0.073721	0.074161	0.086174	0.101285	0.107399	0.123409	0.115142	...	0.407074	0.335351	0.263716	0.234037	0.232176	0.208857	0.203328	0.200151	NaN
3	Africa Western and Central	0.090880	0.095283	0.096612	0.112376	0.133258	0.184803	0.193676	0.189305	0.143989	...	0.497023	0.490867	0.504655	0.507671	0.480743	0.472959	0.476438	0.515544	NaN
4	Angola	0.100835	0.082204	0.210533	0.202739	0.213562	0.205891	0.268937	0.172096	0.289702	...	1.216317	1.204799	1.261542	1.285365	1.260921	1.227703	1.034317	0.887380	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
261	Kosovo	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
262	Yemen, Rep.	0.011038	0.013599	0.012729	0.014518	0.017550	0.017926	0.017023	0.017380	0.021384	...	0.869051	0.804146	1.047834	1.034330	0.536269	0.400468	0.361418	0.326682	NaN
263	South Africa	5.727223	5.832621	5.887168	5.961337	6.332343	6.616545	6.453858	6.560824	6.593620	...	7.867124	8.076633	8.137333	8.213158	7.671202	7.564451	7.632729	7.496645	NaN
264	Zambia	NaN	NaN	NaN	NaN	0.946606	1.096875	0.951122	1.261960	1.166853	...	0.218210	0.277909	0.284058	0.311693	0.319282	0.341615	0.414748	0.446065	NaN
265	Zimbabwe	NaN	NaN	NaN	NaN	1.034904	1.166242	1.307904	1.108579	1.291861	...	0.891090	0.923360	0.925816	0.894256	0.897598	0.783303	0.718570	0.849793	NaN

266 rows × 62 columns

In [5]:
```python
def y_fmt(x, pos):
    if x >= 1000000000000:
        return '%1.0fT' % (x*1e-12)
    elif x >= 1000000000:
        return '%1.0fB' % (x*1e-9)
    elif x >= 1000000:
        return '%1.0fM' % (x*1e-6)
    elif x >= 1000:
        return '%1.0fK' % (x*1e-3)
    else:
        return int(x)
```

Highest CO2 emission in some countries

In [6]:
```python
df_co2_emmision=em_df1
# df_co2_emmision.drop(columns='Indicator Name', inplace=True)
df_co2_emmision=df_co2_emmision.T.reset_index()
df_co2_emmision.columns = df_co2_emmision.iloc[0]
df_co2_emmision=df_co2_emmision.reindex(df_co2_emmision.index.drop(0))
df_co2_emmision.rename(columns={'Country Name': 'Years'}, inplace=True)
df_co2_emmision['Years']=pd.to_numeric(df_co2_emmision['Years'])
```

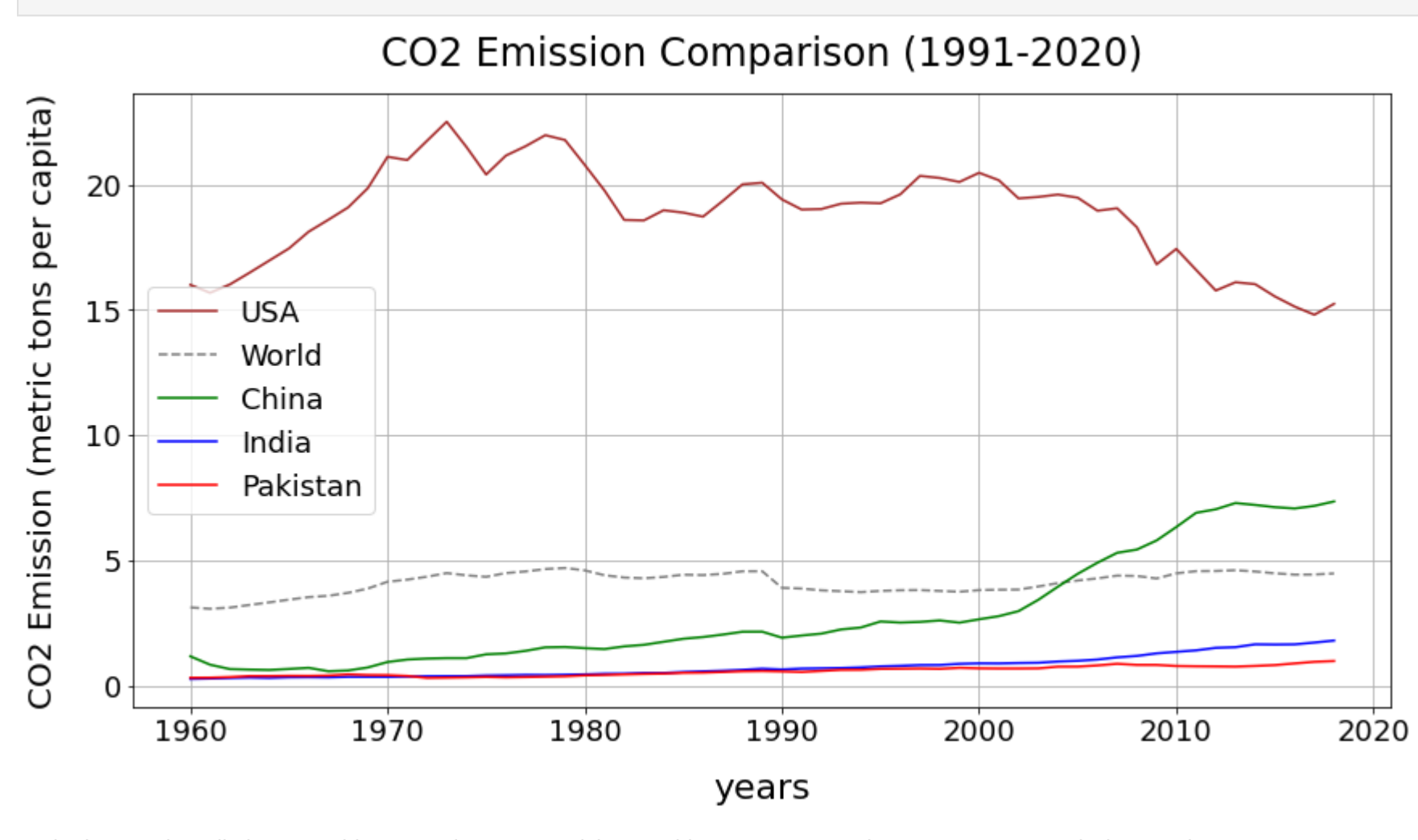
In [7]:
```python
df_co2_emmision
```

Out[7]:

	Years	Aruba	Africa Eastern and Southern	Afghanistan	Africa Western and Central	Angola	Albania	Andorra	Arab World	United Arab Emirates	...	Virgin Islands (U.S.)	Vietnam	Vanuatu	World	Samoa	Kosovo	Yemen, Rep.	South Africa	Zambia	Zi
1	1960	NaN	0.90606	0.046057	0.09088	0.100835	1.258195	NaN	0.609268	0.119037	...	NaN	0.181947	NaN	3.121158	0.135031	NaN	0.011038	5.727223	NaN	
2	1961	NaN	0.922474	0.053589	0.095283	0.082204	1.374186	NaN	0.662618	0.109136	...	NaN	0.183099	NaN	3.06809	0.163542	NaN	0.013599	5.832621	NaN	
3	1962	NaN	0.930816	0.073721	0.096612	0.210533	1.439956	NaN	0.727117	0.163542	...	NaN	0.217694	0.595002	3.114839	0.158377	NaN	0.012729	5.887168	NaN	
4	1963	NaN	0.94057	0.074161	0.112376	0.202739	1.181681	NaN	0.853116	0.175833	...	NaN	0.196997	0.471849	3.221195	0.184037	NaN	0.014518	5.961337	NaN	
5	1964	NaN	0.996033	0.086174	0.133258	0.213562	1.111742	NaN	0.972381	0.132815	...	NaN	0.20987	0.864427	3.324046	0.208106	NaN	0.01755	6.332343	0.946606	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
57	2016	NaN	0.971016	0.208857	0.472959	1.227703	1.714126	6.0806	4.570308	21.30761	...	NaN	2.390741	0.646724	4.431337	1.542099	NaN	0.400468	7.564451	0.341615	(
58	2017	NaN	0.959978	0.203328	0.476438	1.034317	1.948872	6.104134	4.528351	21.950614	...	NaN	2.348081	0.595449	4.435536	1.535642	NaN	0.361418	7.632729	0.414748	
59	2018	NaN	0.933541	0.200151	0.515544	0.88738	1.939732	5.973405	4.438716	20.797498	...	NaN	2.698806	0.615017	4.477639	1.631588	NaN	0.326682	7.496645	0.446065	(
60	2019	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
61	2020	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

61 rows × 267 columns

In [8]:
```python
formatter = FuncFormatter(y_fmt)
fig_dims=(14,7)
fig, ax = plt.subplots(figsize=fig_dims)
df_co2_emmision.set_index('Years')['United States'].plot(color="brown", linestyle="-")
df_co2_emmision.set_index('Years')['World'].plot(color="grey", linestyle="--")
df_co2_emmision.set_index('Years')['China'].plot(color="green", linestyle="-")
df_co2_emmision.set_index('Years')['India'].plot(color="blue", linestyle="-")
df_co2_emmision.set_index('Years')['Pakistan'].plot(color="red", linestyle="-")
plt.xlabel("years", fontsize=22, labelpad=16)
plt.ylabel("CO2 Emission (metric tons per capita)", fontsize=20, labelpad=16)
plt.title("CO2 Emission Comparison (1991-2020)",y=1.03, fontsize=24)
plt.tick_params(axis="both", labelsize=18)
plt.legend(['USA', 'World', 'China', 'India', 'Pakistan'],
           loc=6, fontsize=18, frameon=True);
ax.grid(b=True)
ax.yaxis.set_major_formatter(formatter);
```



-United States is well above World average in CO2 emmision, world average at 5% whereas USA CO2 emission rate is 20%.

-Pakistan and India is constantly below world average at around 0.9%

In [9]:
```python
#reading emmissions data
el_df1, el_df2 = read_data('API_EN.CO2.ETOT.ZS_DS2_en_csv_v2_3931061.csv')
emi_df1, emi_df2 = read_data('API_EN.CO2.MANF.ZS_DS2_en_csv_v2_3931063.csv')
ebld_df1, ebld_df2 = read_data('API_EN.CO2.BLDG.ZS_DS2_en_csv_v2_3931060.csv')
etr_df1, etr_df2 = read_data('API_EN.CO2.TRAN.ZS_DS2_en_csv_v2_3931065.csv')
```

In [10]:
```python
emi_df1
```

Out[10]:

	Country Name	1960	1961	1962	1963	1964	1965	1966	1967	1968	...	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
0	Aruba	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Africa Eastern and Southern	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	12.595571	12.270974	12.691466	13.032239	NaN	NaN	NaN	NaN	NaN	NaN
2	Afghanistan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	Africa Western and Central	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	12.467917	11.073301	11.622648	12.652729	NaN	NaN	NaN	NaN	NaN	NaN
4	Angola	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	15.174129	13.963682	9.070796	7.772021	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
261	Kosovo	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	7.888631	7.011070	6.738869	5.945946	NaN	NaN	NaN	NaN	NaN	NaN
262	Yemen, Rep.	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	13.423517	12.858783	14.112228	14.807873	NaN	NaN	NaN	NaN	NaN	NaN
263	South Africa	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	11.552401	11.383649	12.373618	12.582024	NaN	NaN	NaN	NaN	NaN	NaN
264	Zambia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	48.989899	55.147059	50.175439	46.708464	NaN	NaN	NaN	NaN	NaN	NaN
265	Zimbabwe	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	13.188799	16.695804	8.503401	8.964317	NaN	NaN	NaN	NaN	NaN	NaN

266 rows × 62 columns

In [11]:
```python
df_el_2014=el_df1.loc[:,['Country Name','2014']]
df_el_2014=pd.pivot_table(df_el_2014, values='2014', index='Country Name')
df_el_2014.reset_index(inplace=True)
df_el_2014=df_el_2014.reindex(df_el_2014.index.drop(3))
df_el_2014=df_el_2014[:10]
```

In [12]:
```python
df_emi_2014=emi_df1.loc[:,['Country Name','2014']]
df_emi_2014=pd.pivot_table(df_emi_2014, values='2014', index='Country Name')
df_emi_2014.reset_index(inplace=True)
df_emi_2014=df_emi_2014.reindex(df_emi_2014.index.drop(3))
df_emi_2014=df_emi_2014[:10]
```

In [13]:
```python
df_ebld_2014=ebld_df1.loc[:,['Country Name','2014']]
df_ebld_2014=pd.pivot_table(df_ebld_2014, values='2014', index='Country Name')
df_ebld_2014.reset_index(inplace=True)
df_ebld_2014=df_ebld_2014.reindex(df_ebld_2014.index.drop(3))
df_ebld_2014=df_ebld_2014[:10]
```

In [14]:
```python
df_etr_2014=etr_df1.loc[:,['Country Name','2014']]
df_etr_2014=pd.pivot_table(df_etr_2014, values='2014', index='Country Name')
df_etr_2014.reset_index(inplace=True)
df_etr_2014=df_etr_2014.reindex(df_etr_2014.index.drop(3))
df_etr_2014=df_etr_2014[:10]
```

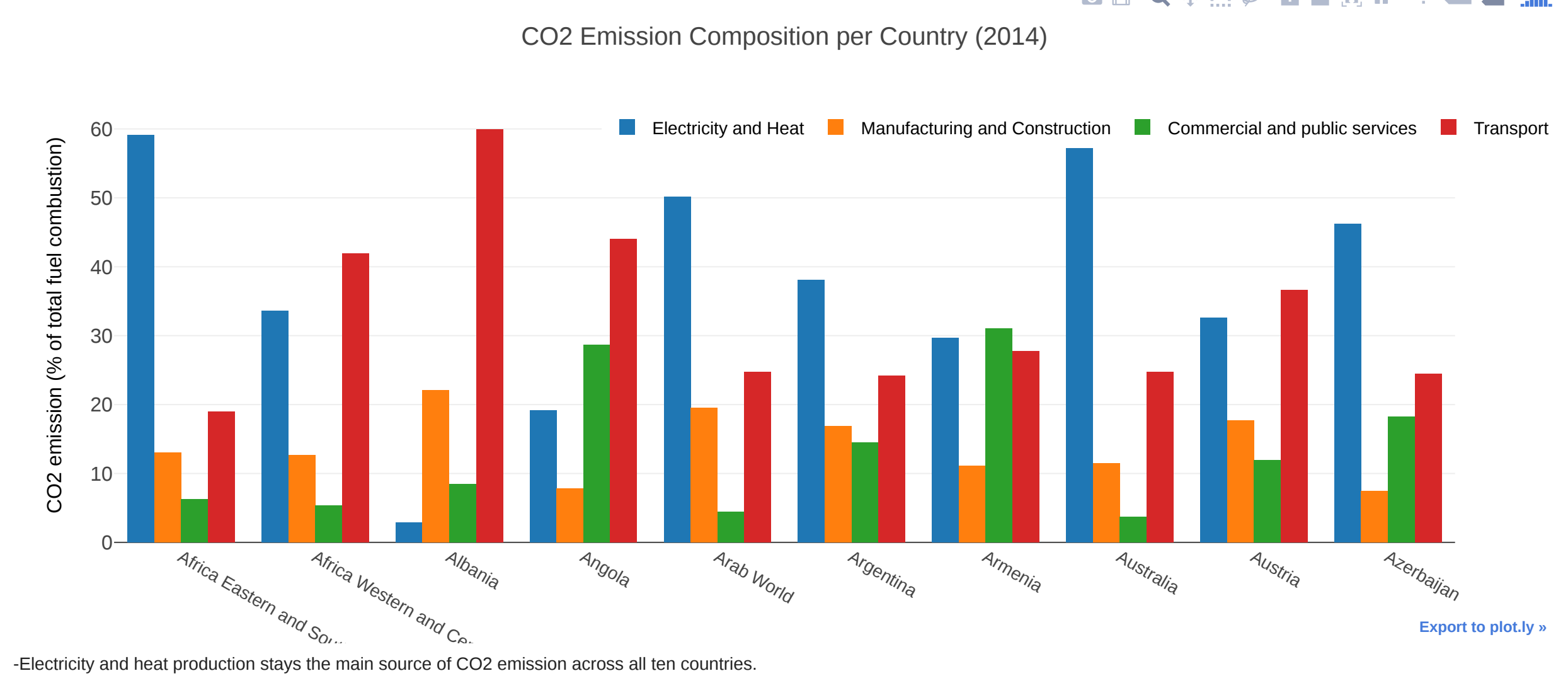
In [15]:
```python
df_el_2014
```

Out[15]:

	Country Name	2014
0	Africa Eastern and Southern	59.086377
1	Africa Western and Central	33.604851
2	Albania	2.912621
4	Angola	19.170984
5	Arab World	50.159275
6	Argentina	38.038563
7	Armenia	29.693487
8	Australia	58.357858
9	Austria	32.576505
10	Azerbaijan	46.183826

In [19]:
```python
from plotly.offline import iplot, init_notebook_mode
import plotly.graph_objs as go
init_notebook_mode()
trace1 = go.Bar(
    x=df_el_2014['Country Name'],
    y=df_el_2014['2014'],
    name='Electricity and Heat'
)
trace2 = go.Bar(
    x=df_emi_2014['Country Name'],
    y=df_emi_2014['2014'],
    name='Manufacturing and Construction'
)
trace3 = go.Bar(
    x=df_ebld_2014['Country Name'],
    y=df_ebld_2014['2014'],
    name='Commercial and public services'
)
trace4 = go.Bar(
    x=df_etr_2014['Country Name'],
    y=df_etr_2014['2014'],
    name='Transport'
)


data = [trace1, trace2, trace3, trace4]
layout = go.Layout(
    barmode='group',
    title="CO2 Emission Composition per Country (2014)",
                font=dict(size=14),
                yaxis=dict(title='CO2 emission (% of total fuel combustion)',
                       tickfont=dict(size=16),
                       titlefont=dict(
                        size=16,
                        color='black'
                        )),
               showlegend=True,
                legend=dict(x=48, y=1, font=dict(
            family='sans-serif',
            size=14,
            color='#000'
        ),orientation="h"),
)
fig = go.Figure(data=data, layout=layout)
iplot(fig)
```



-Electricity and heat production stays the main source of CO2 emission across all ten countries.

-In most of the countries, second contributing sector is transport where it is residential buildings. They can mitigate some of hazardous emissions by launching more policies focused on pollution from transport.

In [ ]:

In [103…
```python
rur_df1, rur_df2 = read_data('API_SP.RUR.TOTL.ZS_DS2_en_csv_v2_3932166.csv')
urb_df1, urb_df2 = read_data('API_SP.URB.TOTL.IN.ZS_DS2_en_csv_v2_3930188.csv')
```

In [114…
```python
df_rural=rur_df1
# df_co2_emmision.drop(columns='Indicator Name', inplace=True)
df_rural=df_rural.T.reset_index()
df_rural.columns = df_rural.iloc[0]
df_rural=df_rural.reindex(df_rural.index.drop(0))
df_rural.rename(columns={'Country Name': 'Years'}, inplace=True)
df_rural['Years']=pd.to_numeric(df_rural['Years'])
```

In [115…
```python
df_rural
```

Out[115…

	Years	Aruba	Africa Eastern and Southern	Afghanistan	Africa Western and Central	Angola	Albania	Andorra	Arab World	United Arab Emirates	...	Virgin Islands (U.S.)	Vietnam	Vanuatu	World	Samoa	Kosovo	Yemen, Rep.	South Africa	Zambia	Zimbabwe
1	1960	49.224	85.295312	91.599	85.329671	89.565	69.295	41.55	68.765858	26.5	...	43.523	85.3	89.596	66.382028	81.074	NaN	90.9	53.381	81.855	87.392
2	1961	49.239	85.055541	91.316	84.946423	89.202	69.057	39.017	68.020073	25.617	...	42.136	84.969	89.417	65.88451	81.014	NaN	90.541	53.207	81.049	87.179
3	1962	49.254	84.814392	91.024	84.550718	88.796	68.985	36.538	67.27088	24.752	...	40.759	84.631	89.235	65.442851	80.939	NaN	90.169	53.094	80.215	86.918
4	1963	49.27	84.555586	90.724	84.135979	88.376	68.914	34.128	66.4984	23.907	...	39.396	84.286	89.05	65.056266	80.859	NaN	89.784	52.98	79.288	86.422
5	1964	49.285	84.280924	90.414	83.705016	87.942	68.842	31.795	65.696398	23.081	...	38.048	83.936	88.862	64.663846	80.779	NaN	89.386	52.866	77.985	85.908
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
57	2016	56.808	65.134899	74.98	54.598191	35.851	41.579	11.752	41.654373	14.035	...	4.52	65.49	74.938	45.626367	81.319	NaN	64.606	34.659	57.562	67.704
58	2017	56.707	64.667627	74.75	53.980666	35.161	40.617	11.85	41.357678	13.752	...	4.397	64.787	74.837	45.172197	81.548	NaN	63.984	34.15	57.024	67.763
59	2018	56.589	64.19223	74.505	53.366862	34.486	39.681	11.938	41.073405	13.478	...	4.279	64.081	74.726	44.723309	81.757	NaN	63.358	33.645	56.479	67.791
60	2019	56.454	63.708678	74.246	52.757147	33.823	38.771	12.016	40.798354	13.211	...	4.168	63.372	74.606	44.279569	81.944	NaN	62.727	33.144	55.928	67.79
61	2020	56.303	63.216694	73.974	52.151375	33.175	37.888	12.084	40.528234	12.952	...	4.061	62.66	74.475	43.843651	82.111	NaN	62.092	32.646	55.371	67.758

61 rows × 267 columns

In [116…
```python
df_urban=urb_df1
df_urban=df_urban.T.reset_index()
df_urban.columns = df_urban.iloc[0]
df_urban=df_urban.reindex(df_urban.index.drop(0))
df_urban.rename(columns={'Country Name': 'Years'}, inplace=True)
df_urban['Years']=pd.to_numeric(df_urban['Years'])
```

In [117…
```python
df_urban
```

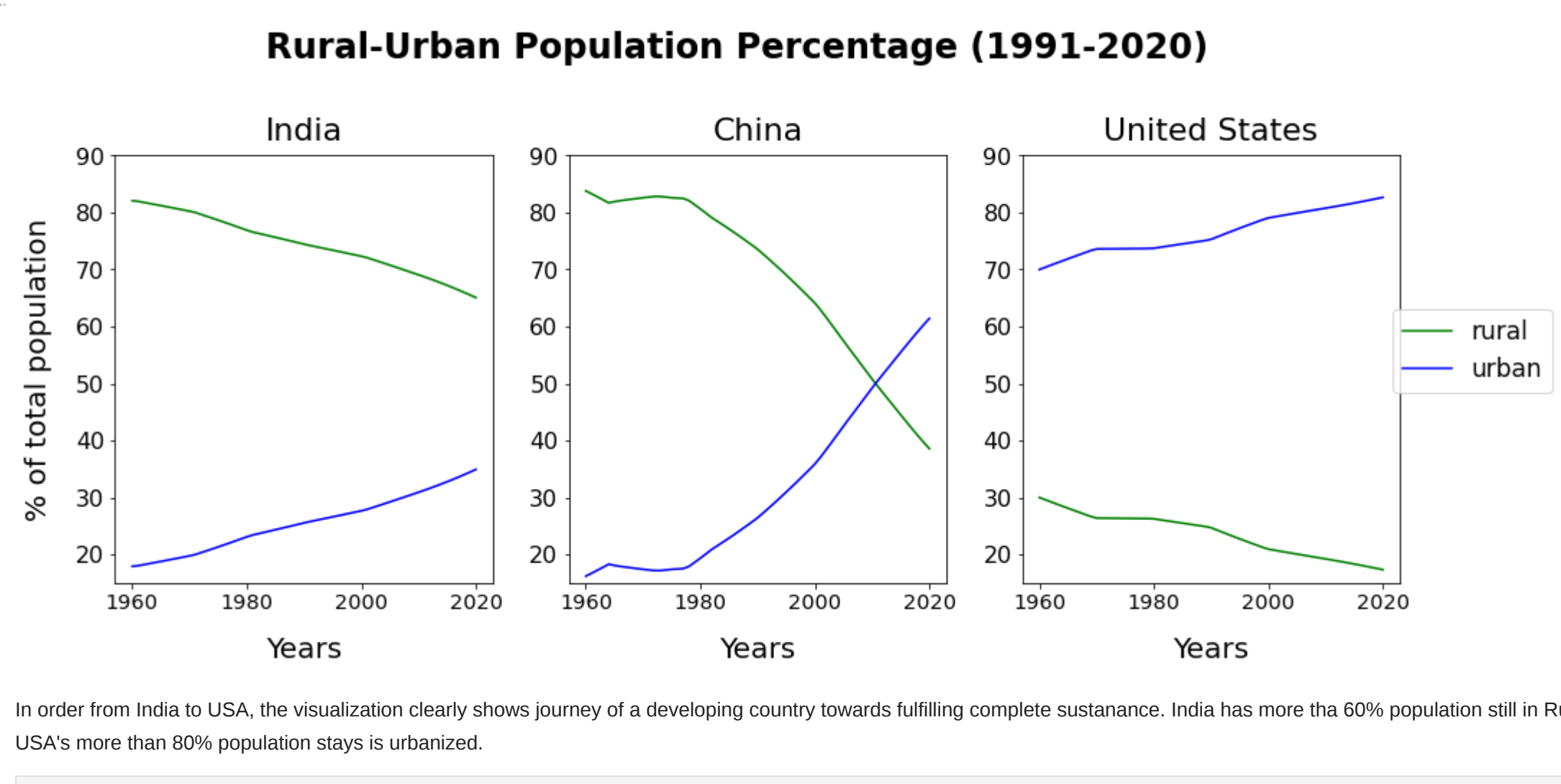
Out[117…

	Years	Aruba	Africa Eastern and Southern	Afghanistan	Africa Western and Central	Angola	Albania	Andorra	Arab World	United Arab Emirates	...	Virgin Islands (U.S.)	Vietnam	Vanuatu	World	Samoa	Kosovo	Yemen, Rep.	South Africa	Zambia	Zimbabwe
1	1960	50.776	14.704688	8.401	14.670329	10.435	30.705	58.45	31.234142	73.5	...	56.477	14.7	10.404	33.617972	18.926	NaN	9.1	46.619	18.145	12.608
2	1961	50.761	14.944459	8.684	15.053577	10.798	30.943	60.983	31.979927	74.383	...	57.864	15.031	10.583	34.11549	18.986	NaN	9.459	46.793	18.951	12.821
3	1962	50.746	15.185608	8.976	15.449282	11.204	31.015	63.462	32.72912	75.248	...	59.241	15.369	10.765	34.557149	19.061	NaN	9.831	46.906	19.785	13.082
4	1963	50.73	15.444414	9.276	15.864021	11.624	31.086	65.872	33.5016	76.093	...	60.604	15.714	10.95	34.943734	19.141	NaN	10.216	47.02	20.712	13.578
5	1964	50.715	15.719076	9.586	16.294984	12.058	31.158	68.205	34.303602	76.919	...	61.952	16.064	11.138	35.336154	19.221	NaN	10.614	47.134	22.015	14.092
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
57	2016	43.192	34.865101	25.02	45.401809	64.149	58.421	88.248	58.345627	85.965	...	95.48	34.51	25.062	54.373633	18.681	NaN	35.394	65.341	42.438	32.296
58	2017	43.293	35.332373	25.25	46.019334	64.839	59.383	88.15	58.642322	86.248	...	95.603	35.213	25.163	54.827803	18.452	NaN	36.016	65.85	42.976	32.237
59	2018	43.411	35.80777	25.495	46.633138	65.514	60.319	88.062	58.926595	86.522	...	95.721	35.919	25.274	55.276691	18.243	NaN	36.642	66.355	43.521	32.209
60	2019	43.546	36.291322	25.754	47.242853	66.177	61.229	87.984	59.201646	86.789	...	95.832	36.628	25.394	55.720431	18.056	NaN	37.273	66.856	44.072	32.21
61	2020	43.697	36.783306	26.026	47.848625	66.825	62.112	87.916	59.471766	87.048	...	95.939	37.34	25.525	56.156349	17.889	NaN	37.908	67.354	44.629	32.242

61 rows × 267 columns

In [128…
```python
formatter = FuncFormatter(y_fmt)
fig_dims=(16,12)
fig, ax = plt.subplots(figsize=fig_dims)
plt.suptitle("Rural-Urban Population Percentage (1991-2020)",y=.98, fontsize=24, fontweight='bold')
ax1=plt.subplot(2, 3, 1)
ax1.set_ylim(15,90)
ax1.set_ylabel("% of total population", labelpad=16, fontsize=20)
ax1.set_xlabel("Years", labelpad=16, fontsize=20)
ax1.set_title("India", y=1.015, fontsize=22)
ax1.xaxis.set_tick_params(labelsize=14.5)
ax1.yaxis.set_tick_params(labelsize=16)
l1=ax1.plot(df_rural['Years'], df_rural['India'], color="green", linestyle="-")[0]
l2=ax1.plot(df_urban['Years'], df_urban['India'], color="blue", linestyle="-")[0]
ax2=plt.subplot(2, 3, 2)
ax2.set_ylim(15,90)
ax2.set_xlabel("Years", labelpad=16, fontsize=20)
ax2.set_title("China", y=1.015, fontsize=22)
ax2.xaxis.set_tick_params(labelsize=14.5)
ax2.yaxis.set_tick_params(labelsize=16)
df_rural.set_index('Years')['China'].plot(color="green", linestyle="-")
df_urban.set_index('Years')['China'].plot(color="blue", linestyle="-")
ax3=plt.subplot(2, 3, 3)
ax3.set_ylim(15,90)
ax3.set_xlabel("Years", labelpad=16, fontsize=20)
ax3.set_title("United States", y=1.015, fontsize=22)
ax3.xaxis.set_tick_params(labelsize=14.5)
ax3.yaxis.set_tick_params(labelsize=16)
df_rural.set_index('Years')['United States'].plot(color="green", linestyle="-")
df_urban.set_index('Years')['United States'].plot(color="blue", linestyle="-");
plt.figlegend([l1, l2],["rural", "urban"], loc = "right", fancybox=True, fontsize=17)
```

Out[128…
```
<matplotlib.legend.Legend at 0x249fca08730>
```



In order from India to USA, the visualization clearly shows journey of a developing country towards fulfilling complete sustanance. India has more tha 60% population still in Rural areas, whereas USA's more than 80% population stays is urbanized.

In [21]:
```python
df_gdp, gdp_df2 = read_data('API_NY.GDP.MKTP.CD_DS2_en_csv_v2_3930485.csv')
```

In [22]:
```python
# df_gdp.drop(columns='Indicator Name', inplace=True)
df_gdp=df_gdp.T.reset_index()
df_gdp.columns = df_gdp.iloc[0]
df_gdp=df_gdp.reindex(df_gdp.index.drop(0))
df_gdp.rename(columns={'Country Name': 'Years'}, inplace=True)
df_gdp['Years']=pd.to_numeric(df_gdp['Years'])
```

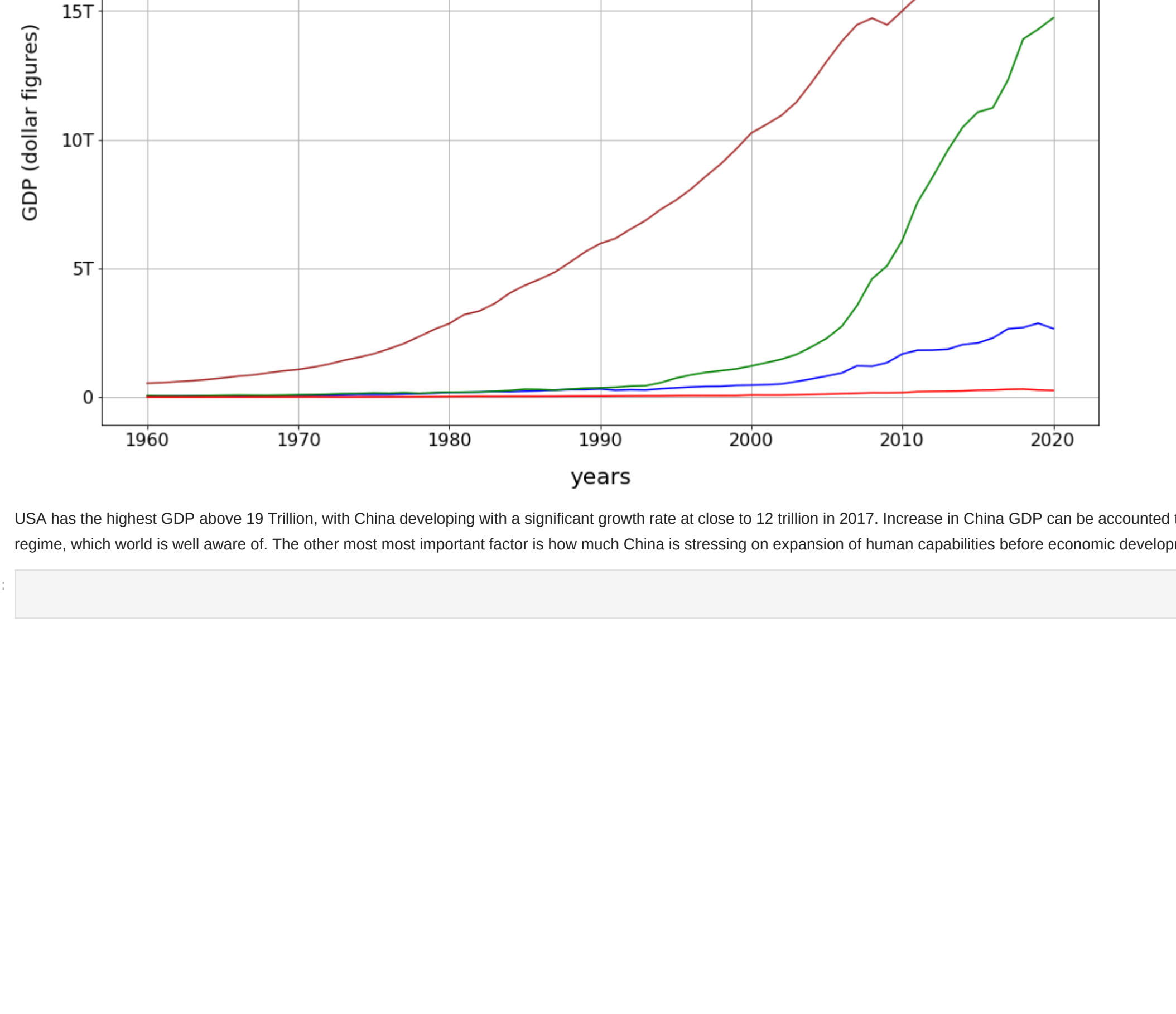
In [23]:
```python
df_gdp
```

Out[23]:

	Years	Aruba	Africa Eastern and Southern	Afghanistan	Africa Western and Central	Angola	Albania	Andorra	Arab World	United Arab Emirate
1	1960	NaN	19313106301.7896	537777811.111111	10404280784.153099	NaN	NaN	NaN	NaN	Nal
2	1961	NaN	19723488056.843899	548888895.555556	11128050589.282101	NaN	NaN	NaN	NaN	Nal
3	1962	NaN	21493920015.015099	546666677.777778	11943353288.4767	NaN	NaN	NaN	NaN	Nal
4	1963	NaN	25733212133.783699	751111191.111111	12676515453.6754	NaN	NaN	NaN	NaN	Nal
5	1964	NaN	23527443250.685001	800000044.444444	13838577015.184799	NaN	NaN	NaN	NaN	Nal
...	...	...	...	...	...	...	...	...	...	.
57	2016	2983636871.50838	873354876607.317017	18116562464.908798	690543020916.229004	101123851090.455002	11861199830.8396	2896679211.86628	2411981152232.620117	357045064669.84301
58	2017	3092430167.59777	985355704660.083008	18753469630.258598	683741574626.192017	122123822333.589996	13019689336.6919	3000180750.11297	2466442560038.419922	385605506848.19598
59	2018	3202188606.92893	1012853224905.790039	18053228578.887798	741691624372.787964	101353230784.593994	15156432309.897699	3218316013.22626	2730779681432.549805	422215043594.28198
60	2019	NaN	1009909855421.839966	18799450742.782299	794572474605.871948	89417190341.253296	15400242874.881201	3155065487.51819	2776468692322.959961	417215559513.36401
61	2020	NaN	920792331527.745972	20116137325.820599	784587603322.865967	58375976292.967796	14887629268.2927	NaN	2447584445275.52002	358868765174.92401

61 rows × 267 columns

In [24]:
```python
formatter = FuncFormatter(y_fmt)
fig_dims=(16,10)
fig, ax = plt.subplots(figsize=fig_dims)
#df_gdp.set_index('Years')['World'].plot(color="grey", linestyle="--")
df_gdp.set_index('Years')['United States'].plot(color="brown", linestyle="-")
df_gdp.set_index('Years')['India'].plot(color="blue", linestyle="-")
df_gdp.set_index('Years')['China'].plot(color="green", linestyle="-")
df_gdp.set_index('Years')['Pakistan'].plot(color="red", linestyle="-")
plt.xlabel("years", fontsize=20, labelpad=16)
plt.ylabel("GDP (dollar figures)", fontsize=18, labelpad=16)
plt.title("GDP Comparison (1991-2020)",y=1.03, fontsize=20)
plt.tick_params(axis="both", labelsize=16)
plt.legend(['USA', 'India', 'China','Pakistan'],
          loc=0, fontsize=16, frameon=True, title="Country Name", bbox_to_anchor=(0., 1., 1., 0.));
ax.grid(b=True)
ax.yaxis.set_major_formatter(formatter);
```



USA has the highest GDP above 19 Trillion, with China developing with a significant growth rate at close to 12 trillion in 2017. Increase in China GDP can be accounted towards its authoritarian regime, which world is well aware of. The other most most important factor is how much China is stressing on expansion of human capabilities before economic development.

In [ ]: