Week6 - Mini Project Week

November 2, 2020

[2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats
from sklearn import datasets, linear_model
```

[3]:
```
data = pd.read_csv(r"C:
↪\Users\Anna\Desktop\EDX_DataScience\Week3\Week5-Visualization\Indicators.
↪csv")
```

0.0.1 How does the income affect CO2 emissions over time?

[4]:
```
list_income = ["low income", "lower middle income", "middle income", "upper␣
↪middle income", "high income"]
data_income = data[data.CountryName.str.lower().isin(list_income) & data.
↪IndicatorName.str.contains("CO2 emissions \(metric tons per capita\)")]
```

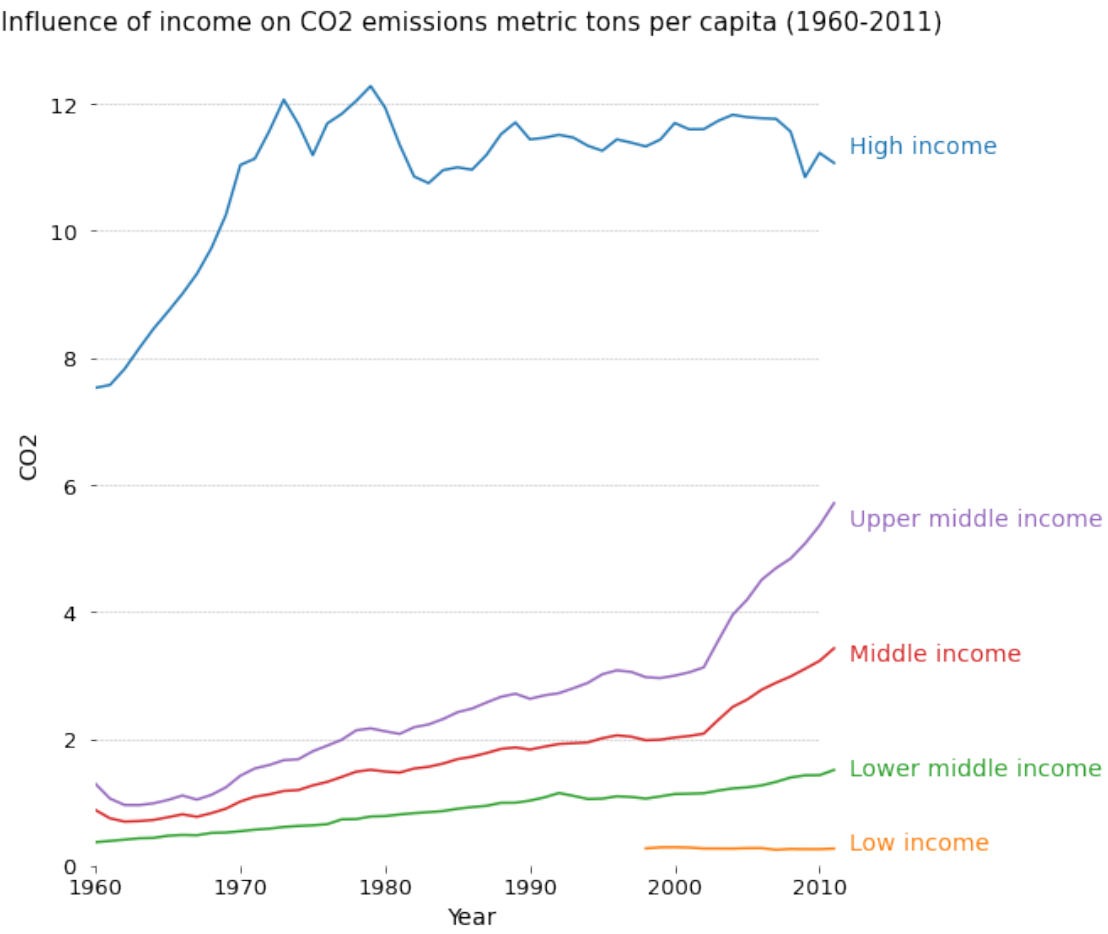
[5]:
```
data_income_ind = data_income.set_index(["Year", "CountryName"], drop = True)
```

[6]:
```
data_plt = data_income_ind["Value"].unstack(level=1)
```

[7]:
```
plt.figure(figsize=(8, 9))

ax = plt.subplot(111)
ax.spines["top"].set_visible(False)
ax.spines["bottom"].set_visible(False)
ax.spines["right"].set_visible(False)
ax.spines["left"].set_visible(False)

ax.get_xaxis().tick_bottom()
ax.get_yaxis().tick_left()

plt.ylim(0, 13)
plt.xlim(1960, 2012)

plt.ylabel("CO2", fontsize=14)
plt.xlabel("Year", fontsize=14)
```

```
plt.yticks(range(0, 13, 2), [str(x) + " " for x in range(0, 13, 2)],␣
 ↪fontsize=13)
plt.xticks(fontsize=13)

for y in range(0, 13, 2):
    plt.plot(range(1960, 2011), [y] * len(range(1960, 2011)), "--", lw=0.5,␣
 ↪color="black", alpha=0.3)

plt.plot(data_plt)

ax.annotate('Low income', xy=(2012, data_plt["Low income"].iloc[-2]),color =␣
 ↪"tab:orange", fontsize=14)
ax.annotate('Lower middle income', xy=(2012, data_plt["Lower middle income"].
 ↪iloc[-2]), color = "tab:green", fontsize=14)
ax.annotate('Middle income', xy=(2012, data_plt["Middle income"].iloc[-2]),␣
 ↪color = "tab:red", fontsize=14)
ax.annotate('Upper middle income', xy=(2012, data_plt["Upper middle income"].
 ↪iloc[-2]), color = "tab:purple", fontsize=14)
ax.annotate('High income', xy=(2012, data_plt["High income"].iloc[-2]), color =␣
 ↪"tab:blue", fontsize=14)

plt.title("Influence of income on CO2 emissions metric tons per capita␣
 ↪(1960-2011)", fontsize=15, ha = "center")
```

[7]: Text(0.5, 1.0, 'Influence of income on CO2 emissions metric tons per capita (1960-2011)')



0.0.2 Are CO2 emissions correlated with the income levels?

[7]:
```
not_countries = ['Arab World', 'Caribbean small states',
       'Central Europe and the Baltics',
       'East Asia & Pacific (all income levels)',
       'East Asia & Pacific (developing only)', 'Euro area',
       'European Union', 'Fragile and conflict affected situations',
       'Heavily indebted poor countries (HIPC)', 'High income',
       'High income: OECD',
       'Latin America & Caribbean (all income levels)',
       'Latin America & Caribbean (developing only)',
       'Least developed countries: UN classification',
       'Low & middle income', 'Lower middle income',
       'Middle East & North Africa (all income levels)',
       'Middle East & North Africa (developing only)', 'Middle income',
       'North America', 'OECD members', 'Pacific island small states',
       'Small states', 'South Asia',
       'Sub-Saharan Africa (all income levels)',
```

```
      'Sub-Saharan Africa (developing only)', 'Upper middle income', 'Europe &␣
↪Central Asia (all income levels)',
      'Europe & Central Asia (developing only)', 'High income: nonOECD','Low␣
↪income', 'Other small states', 'World']
```

[8]:
```
data_c_co2 = data[~data.CountryName.isin(not_countries) & data.IndicatorName.
↪str.contains("CO2 emissions \(metric tons per capita\)")]
data_gni = data[(~data.CountryName.isin(not_countries)) & (data.IndicatorName␣
↪== "GNI per capita, Atlas method (current US$)")]
```

[9]:
```
def df_func(year, df_co2, df_gni):
    df_co2_drop = df_co2.drop(["CountryCode", "IndicatorName",␣
↪"IndicatorCode"], axis = 1)
    df_gni_drop = df_gni.drop(["CountryCode", "IndicatorName",␣
↪"IndicatorCode"], axis = 1)

    df_co2_year = df_co2_drop[df_co2_drop.Year == year]
    df_gni_year = df_gni_drop[df_gni_drop.Year == year]

    df_co2_year.rename(columns = {'Value':'CO2_emissions'}, inplace = True)
    df_gni_year.rename(columns = {'Value':'GNI'}, inplace = True)

    df_co2_year_plt = df_co2_year.drop("Year", axis = 1)
    df_gni_year_plt = df_gni_year.drop("Year", axis = 1)

    df_all = df_co2_year_plt.merge(df_gni_year_plt)
    df_all_plt = df_all.set_index("CountryName")

    return df_all_plt
```

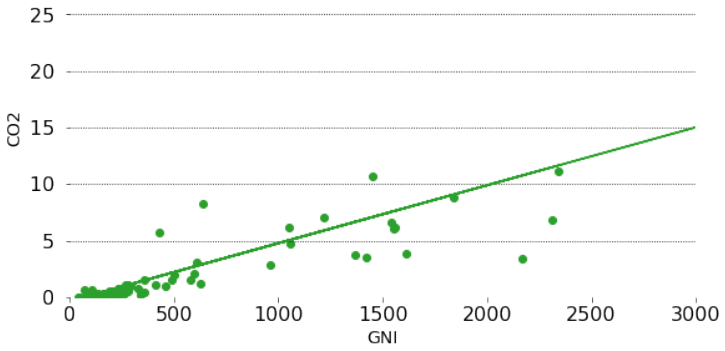
[10]:
```
df_1962 = df_func(1962, data_c_co2, data_gni).dropna()
df_1980 = df_func(1980, data_c_co2, data_gni).dropna()
df_2000 = df_func(2000, data_c_co2, data_gni).dropna()
df_2011 = df_func(2011, data_c_co2, data_gni).dropna()
```

[11]:
```
def plot_function(year_GNI, year_CO2, color, year, xlen):
    plt.figure(figsize=(9, 5))
    ax = plt.subplot(111)

    plt.ylim(0, 30)
    plt.xlim(0, xlen)

    plt.ylabel("CO2", fontsize=14)
    plt.xlabel("GNI", fontsize=14)

    plt.title("CO2 emissions (metric tons per capita) and GNI per capita, Atlas␣
↪method (current US$), Year " + str(year), fontsize=18, ha = "center")
```

```
    ax.spines["top"].set_visible(False)
    ax.spines["bottom"].set_visible(False)
    ax.spines["right"].set_visible(False)
    ax.spines["left"].set_visible(False)

    ax.get_xaxis().tick_bottom()
    ax.get_yaxis().tick_left()

    plt.yticks(range(0, 30, 5), [str(x) + " " for x in range(0, 30, 5)],␣
↪fontsize=16)
    plt.xticks(fontsize=16)

    plt.grid(axis = "y", linestyle = "dotted", color = "black")

    st_val = stats.linregress(year_GNI, year_CO2)

    plt.plot(year_GNI, st_val.intercept + st_val.slope*year_GNI, c = color)

    plt.scatter(year_GNI,year_CO2, color = color)

    return plt.show()
```

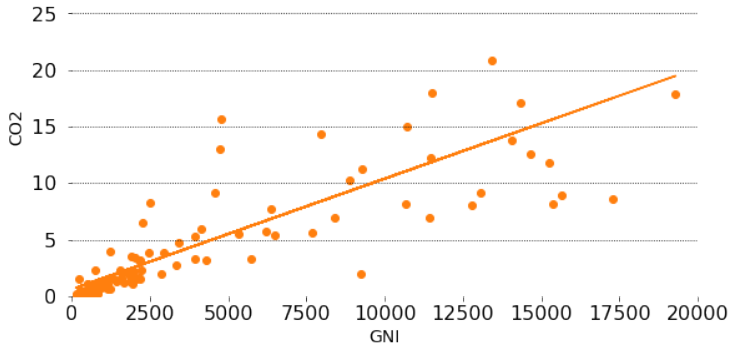
[12]:
```
plot_function(df_1962.GNI,  df_1962.CO2_emissions, "tab:green",  1962, 3000)
```

CO2 emissions (metric tons per capita) and GNI per capita, Atlas method (current US$), Year 1962



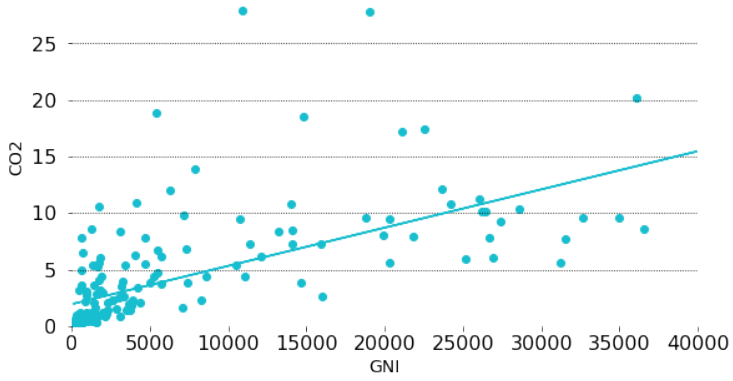
[13]:
```
plot_function(df_1980.GNI,  df_1980.CO2_emissions, "tab:orange",  1980, 20000)
```

CO2 emissions (metric tons per capita) and GNI per capita, Atlas method (current US$), Year 1980



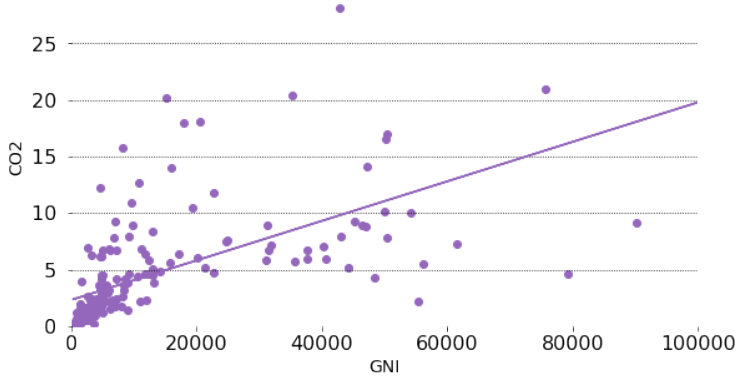
[14]:
```
plot_function(df_2000.GNI,  df_2000.CO2_emissions, "tab:cyan",  2000, 40000)
```

CO2 emissions (metric tons per capita) and GNI per capita, Atlas method (current US$), Year 2000



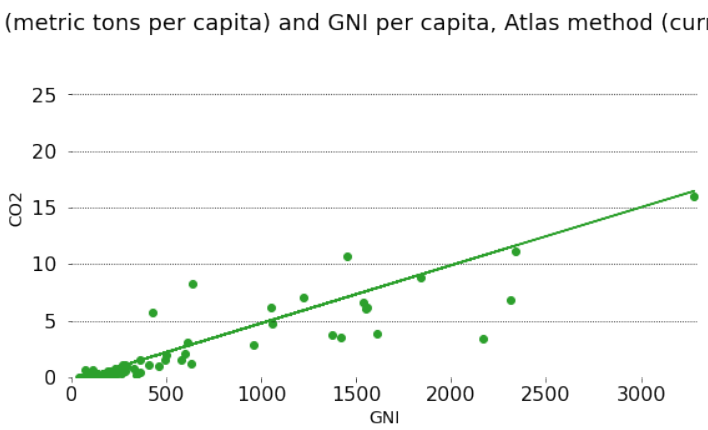
[15]:
```
plot_function(df_2011.GNI,  df_2011.CO2_emissions, "tab:purple",  2011, 100000)
```

CO2 emissions (metric tons per capita) and GNI per capita, Atlas method (current US$), Year 2011



[16]:
```
corrrelation1962 = df_1962.corr(method="pearson")
corrrelation1980 = df_1980.corr(method="pearson")
corrrelation2000 = df_2000.corr(method="pearson")
corrrelation2011 = df_2011.corr(method="pearson")
```

[17]:
```
plt.figure(figsize=(9, 5))
ax = plt.subplot(111)

plt.ylim(0, 30)
plt.xlim(0, 3300)

plt.ylabel("CO2", fontsize=14)
plt.xlabel("GNI", fontsize=14)

plt.title("CO2 emissions (metric tons per capita) and GNI per capita, Atlas␣
 ↪method (current US$), Year 1962" , fontsize=18, ha = "center")

ax.spines["top"].set_visible(False)
ax.spines["bottom"].set_visible(False)
ax.spines["right"].set_visible(False)
ax.spines["left"].set_visible(False)

ax.get_xaxis().tick_bottom()
ax.get_yaxis().tick_left()

plt.yticks(range(0, 30, 5), [str(x) + " " for x in range(0, 30, 5)],␣
 ↪fontsize=16)
plt.xticks(fontsize=16)

plt.grid(axis = "y", linestyle = "dotted", color = "black")

st_val = stats.linregress(df_1962.GNI, df_1962.CO2_emissions)

plt.plot(df_1962.GNI, st_val.intercept + st_val.slope*df_1962.GNI, c = "tab:
 ↪green")
plt.scatter(df_1962.GNI,df_1962.CO2_emissions, color = "tab:green")
plt.show()
```



[18]:
```
corrrelation1962
```

[18]:
```
                CO2_emissions       GNI
CO2_emissions        1.000000  0.685654
GNI                  0.685654  1.000000
```

[19]:
```
corrrelation1980
```

[19]:
```
                CO2_emissions       GNI
CO2_emissions        1.000000  0.748708
GNI                  0.748708  1.000000
```

[20]:
```
corrrelation2000
```

[20]:
```
                CO2_emissions      GNI
CO2_emissions         1.00000  0.64927
GNI                   0.64927  1.00000
```

[21]:
```
corrrelation2011
```

[21]:
```
                CO2_emissions       GNI
CO2_emissions        1.000000  0.545036
GNI                  0.545036  1.000000
```

0.0.3 How differently did the income levels combust fuel 2011?

[22]:
```
List_lowIncome = data[(data.CountryName == "Low income") & (data.IndicatorName.
 ↪str.contains("of total fuel combustion")) & (data.Year == 2012)]
List_LowMidIncome = data[(data.CountryName == "Lower middle income") & (data.
 ↪IndicatorName.str.contains("of total fuel combustion")) & (data.Year ==␣
 ↪2012)]
```

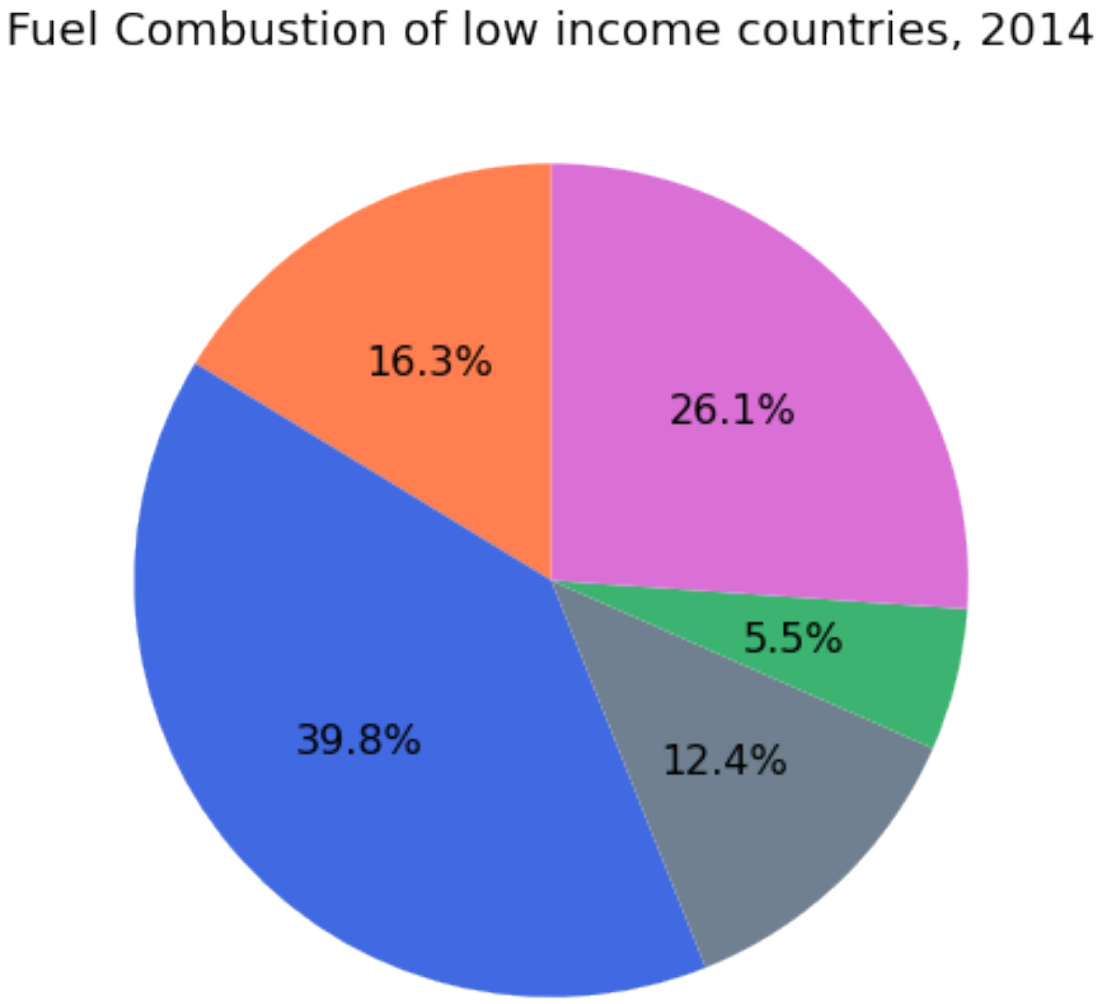
```
List_MidIncome = data[(data.CountryName == "Middle income") & (data.
 ↪IndicatorName.str.contains("of total fuel combustion")) & (data.Year ==␣
 ↪2012)]
List_UpMidIncome = data[(data.CountryName == "Upper middle income") & (data.
 ↪IndicatorName.str.contains("of total fuel combustion")) & (data.Year ==␣
 ↪2012)]
List_HiIncome = data[(data.CountryName == "High income") & (data.IndicatorName.
 ↪str.contains("of total fuel combustion")) & (data.Year == 2012)]
```

[23]:
```
list_locwincome_pop = List_lowIncome.drop(["CountryName", "CountryCode",␣
 ↪"IndicatorCode", "Year"], axis = 1)
list_LowMidIncome_pop = List_LowMidIncome.drop(["CountryName", "CountryCode",␣
 ↪"IndicatorCode", "Year"], axis = 1)
list_MidIncome_pop = List_MidIncome.drop(["CountryName", "CountryCode",␣
 ↪"IndicatorCode", "Year"], axis = 1)
list_UpMidIncome_pop = List_UpMidIncome.drop(["CountryName", "CountryCode",␣
 ↪"IndicatorCode", "Year"], axis = 1)
list_HiIncome_pop = List_HiIncome.drop(["CountryName", "CountryCode",␣
 ↪"IndicatorCode", "Year"], axis = 1)
```

[24]:
```
list_locwincome_pop.IndicatorName = ["electricity and heat production",␣
 ↪"manufacturing industries and construction", "other sectors", "residential␣
 ↪buildings and commercial and public services", "transport"]
list_LowMidIncome_pop.IndicatorName = ["electricity and heat production",␣
 ↪"manufacturing industries and construction", "other sectors", "residential␣
 ↪buildings and commercial and public services", "transport"]
list_MidIncome_pop.IndicatorName = ["electricity and heat production",␣
 ↪"manufacturing industries and construction", "other sectors", "residential␣
 ↪buildings and commercial and public services", "transport"]
list_UpMidIncome_pop.IndicatorName = ["electricity and heat production",␣
 ↪"manufacturing industries and construction", "other sectors", "residential␣
 ↪buildings and commercial and public services", "transport"]
list_HiIncome_pop.IndicatorName = ["electricity and heat production",␣
 ↪"manufacturing industries and construction", "other sectors", "residential␣
 ↪buildings and commercial and public services", "transport"]
```

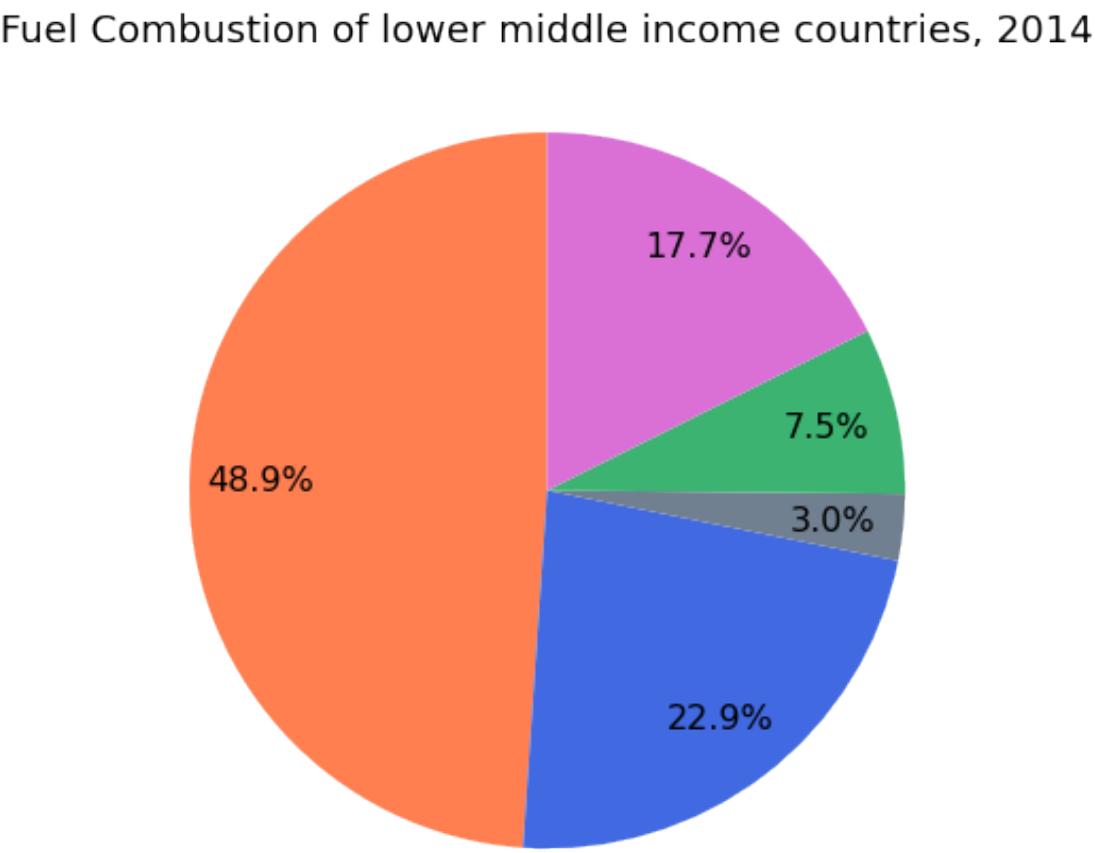
[25]:
```
fig, ax= plt.subplots(figsize=(6,6))
colors = ['coral','royalblue','slategrey','mediumseagreen', 'orchid']
plt.pie(list_locwincome_pop.Value, autopct='%1.1f%%', colors = colors, ␣
 ↪textprops={'fontsize': 16}, startangle=90)
plt.tight_layout()
plt.title("Fuel Combustion of low income countries, 2014", fontsize=18)
```

[25]:
```
Text(0.5, 1.0, 'Fuel Combustion of low income countries, 2014')
```



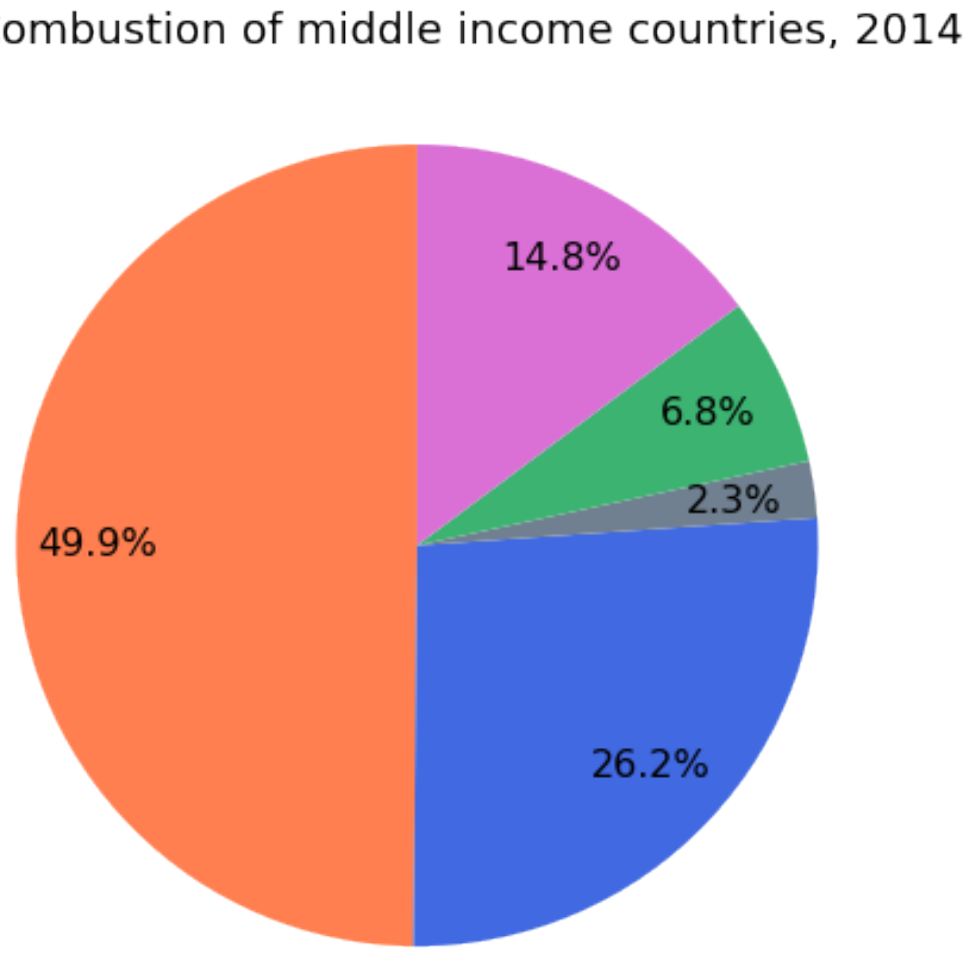
[26]:
```
fig, ax= plt.subplots(figsize=(6,6))
colors = ['coral','royalblue','slategrey','mediumseagreen', 'orchid']
ax.axis('equal')
plt.pie(list_LowMidIncome_pop.Value, autopct='%1.1f%%', colors = colors, ␣
↪textprops={'fontsize': 16}, pctdistance= 0.8, startangle= 90)
plt.tight_layout()
plt.title("Fuel Combustion of lower middle income countries, 2014", fontsize=18)
```

[26]:
```
Text(0.5, 1.0, 'Fuel Combustion of lower middle income countries, 2014')
```



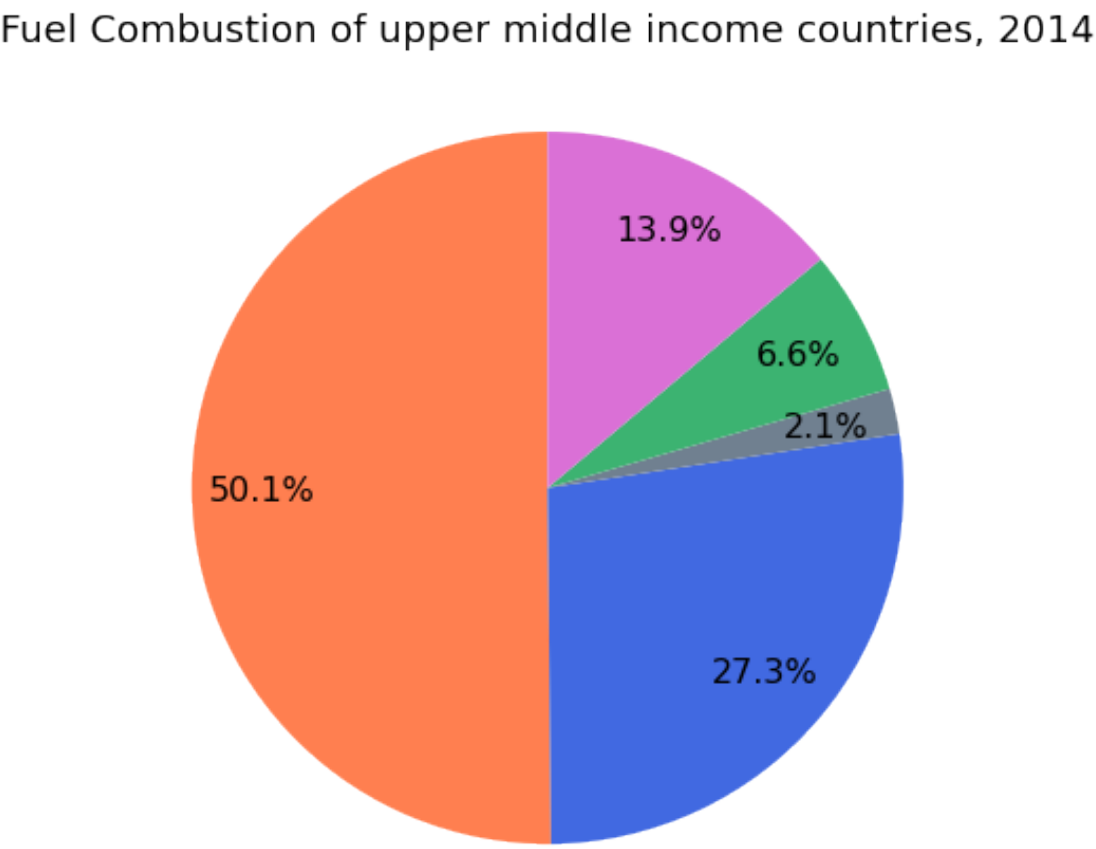
[27]:
```
fig, ax= plt.subplots(figsize=(6,6))
colors = ['coral','royalblue','slategrey','mediumseagreen', 'orchid']
ax.axis('equal')
plt.pie(list_MidIncome_pop.Value, autopct='%1.1f%%', colors = colors, ␣
↪textprops={'fontsize': 16}, pctdistance= 0.8, startangle= 90)
plt.tight_layout()
plt.title("Fuel Combustion of middle income countries, 2014", fontsize=18)
```

[27]:
```
Text(0.5, 1.0, 'Fuel Combustion of middle income countries, 2014')
```



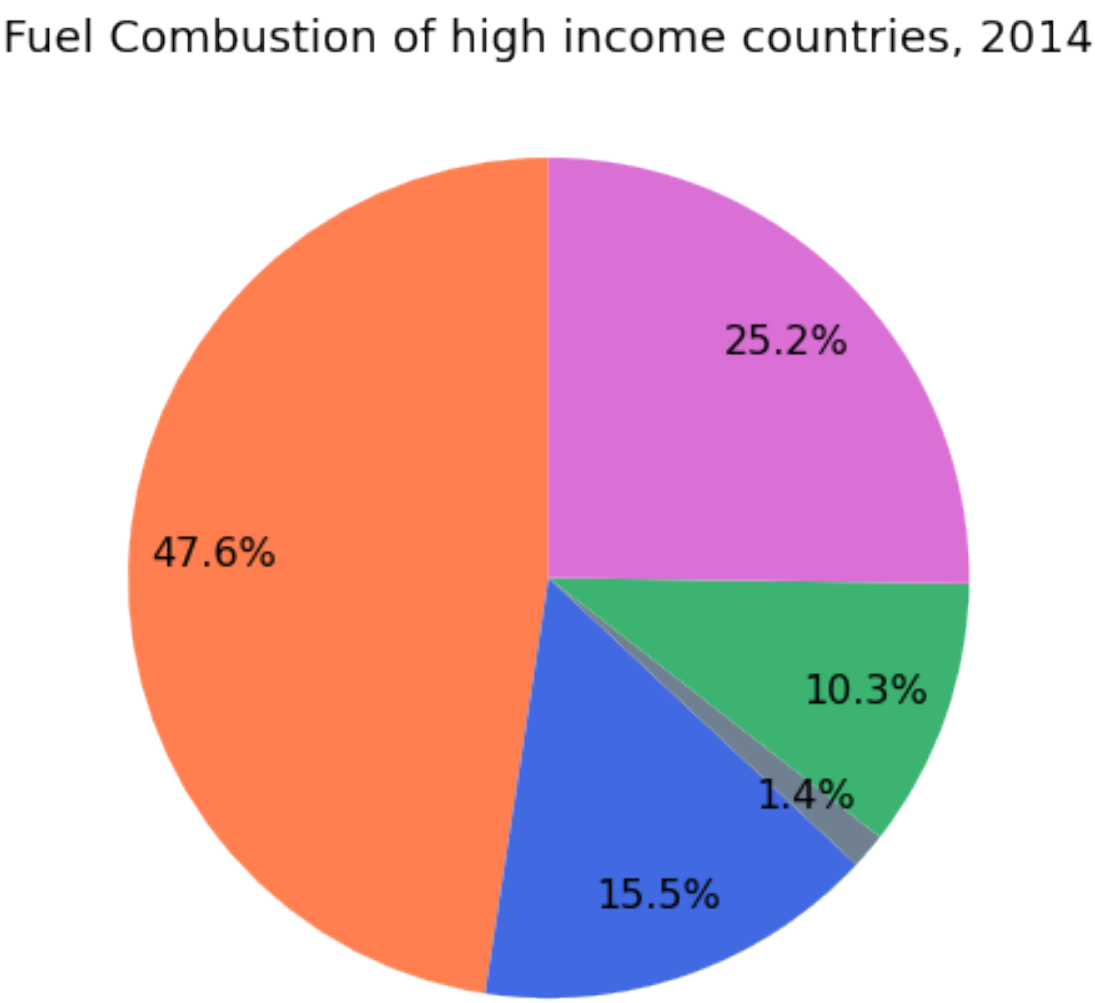
[28]:
```
fig, ax= plt.subplots(figsize=(6,6))
colors = ['coral','royalblue','slategrey','mediumseagreen', 'orchid']
ax.axis('equal')
plt.pie(list_UpMidIncome_pop.Value, autopct='%1.1f%%', colors = colors, ␣
↪textprops={'fontsize': 16}, pctdistance= 0.8, startangle= 90)
plt.tight_layout()
plt.title("Fuel Combustion of upper middle income countries, 2014", fontsize=18)
```

[28]:
```
Text(0.5, 1.0, 'Fuel Combustion of upper middle income countries, 2014')
```



[29]:
```
fig, ax= plt.subplots(figsize=(6,6))
colors = ['coral','royalblue','slategrey','mediumseagreen', 'orchid']
ax.axis('equal')
plt.pie(list_HiIncome_pop.Value, autopct='%1.1f%%', colors = colors, ␣
 ↪textprops={'fontsize': 16}, pctdistance= 0.8, startangle= 90)
plt.tight_layout()
plt.title("Fuel Combustion of high income countries, 2014", fontsize=18)
```

[29]:
```
Text(0.5, 1.0, 'Fuel Combustion of high income countries, 2014')
```


Fuel Combustion of high income countries, 2014
25.2%
47.6%
10.3%
1.4%
15.5%