In [63]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Problem1

I got 6 out of 13.

Yes, i'm suprised that overall living conditions for human has improved a lot in the past 30-50 years.

I'd learn more about question 1:In the last 20 years the proportion of people living in extreme poverty worldwide, has almost halved ¶

I'll restate it as: How much improvement for poverty has been made in the past 30 years.

From the plot we see poverty rate has drastically decreased from ~60% to %~20 in the past 30 years worldwide

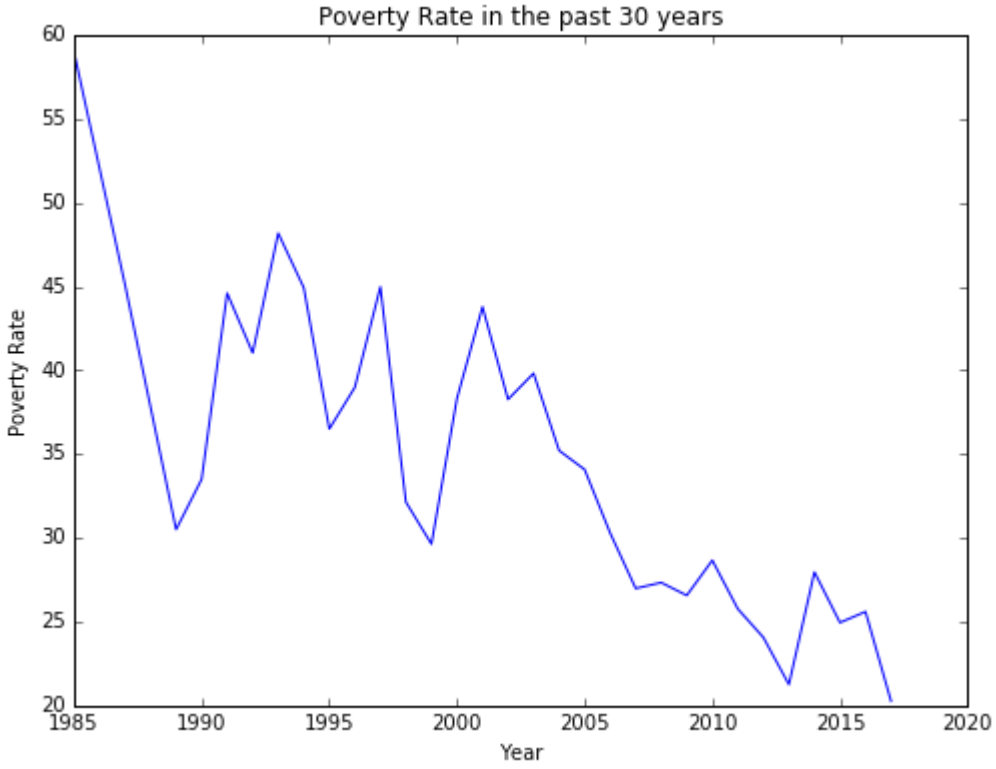
In [9]:
```
poverty_percent = pd.read_csv('C:\\Users\\hudie\\Downloads\\ddf--datapoints--alternative_poverty_percent_below_nationally_defined_poverty--by--geo--time.csv')
```

In [10]:
```
poverty_percent.alternative_poverty_percent_below_nationally_defined_poverty.describe()
```

Out[10]:
```
count    803.000000
mean      29.390286
std       17.066214
min        0.400000
25%       16.450000
50%       25.800000
75%       40.400000
max       83.300000
Name: alternative_poverty_percent_below_nationally_defined_poverty, dtype: float64
```

In [14]:
```
poverty_percent = poverty_percent.groupby('time').mean().reset_index()
poverty_percent.shape
plt.figure(figsize=(8,6))
plt.plot(poverty_percent['time'], poverty_percent['alternative_poverty_percent
_below_nationally_defined_poverty'].values)
plt.xlabel('Year')
plt.ylabel('Poverty Rate')
plt.title('Poverty Rate in the past 30 years')
```

Out[14]: <matplotlib.text.Text at 0x140cb15ea58>



Problem2

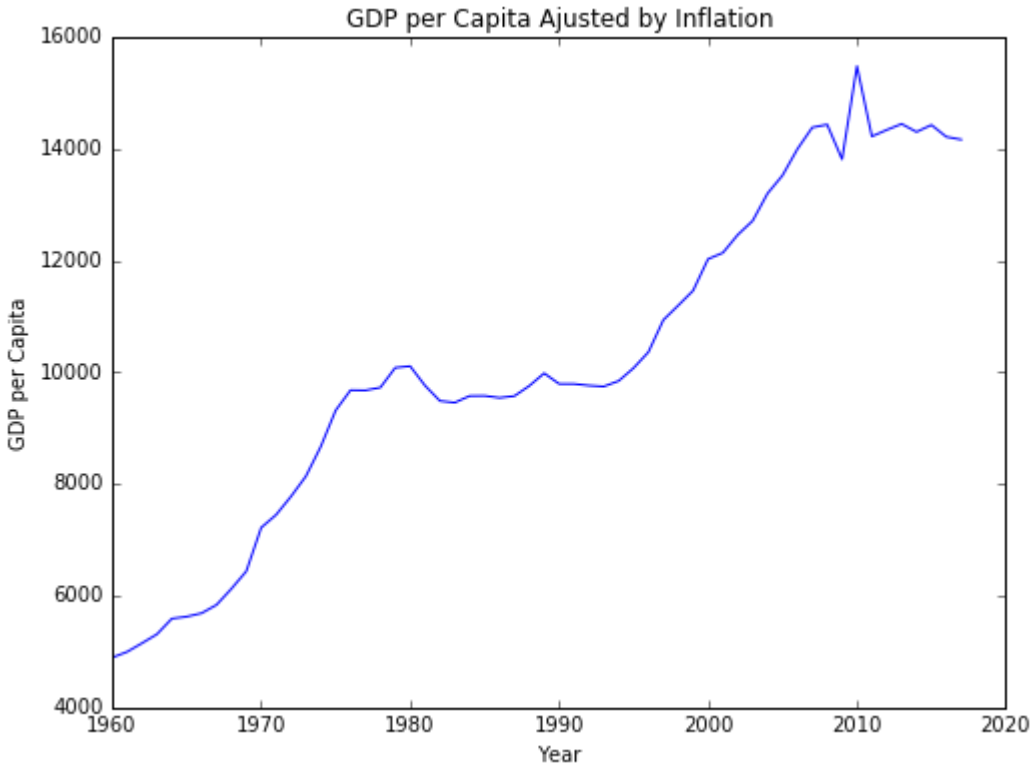
my data is wordwide GDP/Capita adjusted by inflation for this part, from 1960-2020

from the trend plot we can see there is a high momentum of income increasing from1960-1980, then the growth seems paused untill 1990 and restarted since then untill 1990s recession and 2018's recession. The trend make sense to me as we can imagine after world war2 people have good global environment for a period of time, and slow down roughly every 10 years when there is a recession. Caveat is economic is overall growing very fast in the past 50 years, however we see it's in cycle, and we might want to prepare for next recession in the near future.

In [23]:
```
gdp_capita = pd.read_csv('C:\\Users\\hudie\\Downloads\\ddf--datapoints--gdpper
capita_us_inflation_adjusted--by--geo--time.csv')
```

In [24]:
```
gdp_capita.gdppercapita_us_inflation_adjusted.describe()
gdp_capita = gdp_capita.groupby('time').mean().reset_index()
plt.figure(figsize=(8,6))
plt.plot(gdp_capita['time'], gdp_capita['gdppercapita_us_inflation_adjusted'].
values)
plt.xlabel('Year')
plt.ylabel('GDP per Capita')
plt.title('GDP per Capita Ajusted by Inflation')
```

Out[24]: <matplotlib.text.Text at 0x140cc2d0e48>



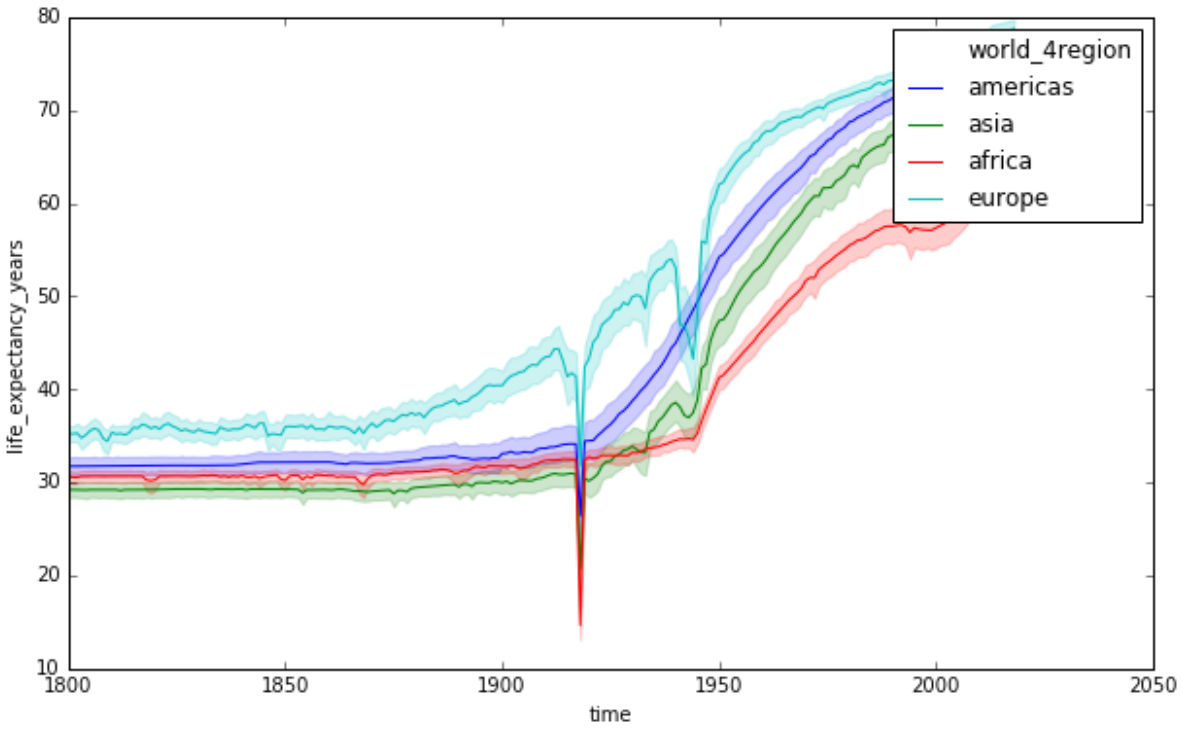
Problem3

In [75]:
```
life_expect=pd.read_csv("C:\\Users\\hudie\\Downloads\\ddf--datapoints--life_expectancy_years--by--geo--time.csv")
continent_map = pd.read_csv('C:\\Users\\hudie\\Downloads\\ddf--entities--geo--country.csv')
continent_map = continent_map[['country', 'world_4region']]
plot_data = pd.merge(life_expect, continent_map, left_on = 'geo', right_on='country')


plt.figure(figsize=(10, 6))
sns.lineplot(y='life_expectancy_years', x= 'time', hue='world_4region', data=plot_data )
```

```
C:\Users\hudie\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[75]: <matplotlib.axes._subplots.AxesSubplot at 0x140dffdb908>

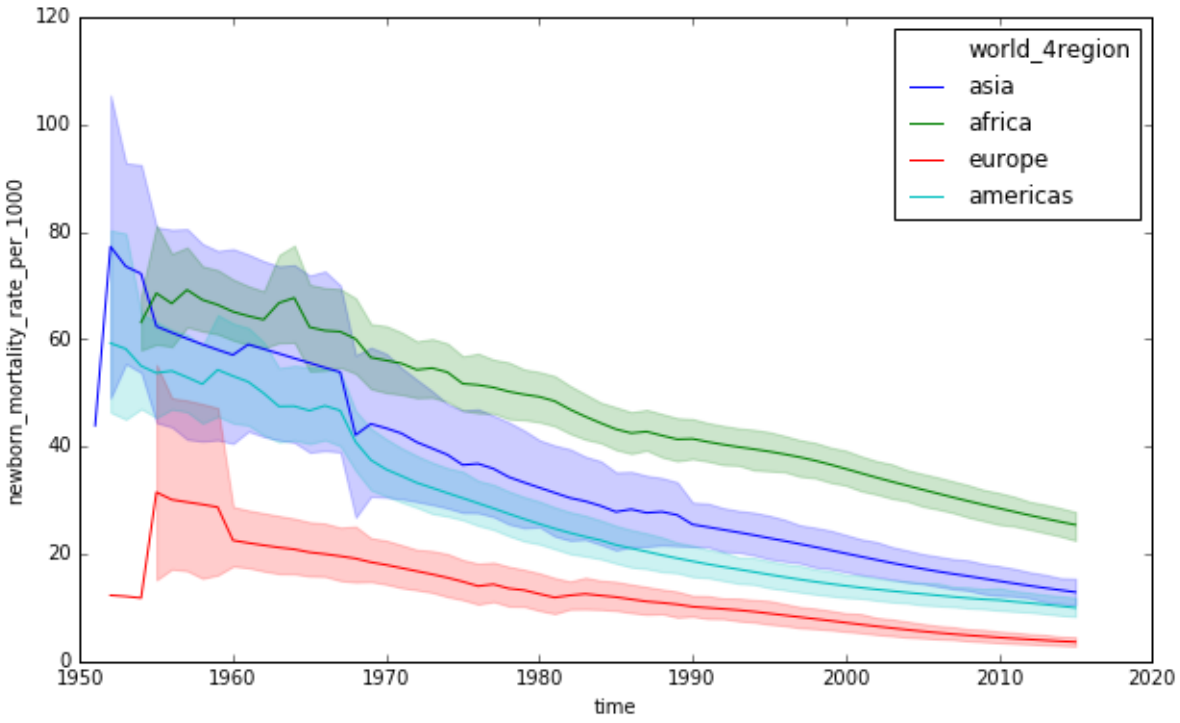


In [76]:
```
mortality=pd.read_csv("C:\\Users\\hudie\\Downloads\\ddf--datapoints--newborn_mortality_rate_per_1000--by--geo--time.csv")
continent_map = pd.read_csv('C:\\Users\\hudie\\Downloads\\ddf--entities--geo--country.csv')
continent_map = continent_map[['country', 'world_4region']]
plot_data = pd.merge(mortality, continent_map, left_on = 'geo', right_on='country')

plt.figure(figsize=(10, 6))
sns.lineplot(y='newborn_mortality_rate_per_1000', x= 'time', hue='world_4region', data=plot_data )
```

```
C:\Users\hudie\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[76]: <matplotlib.axes._subplots.AxesSubplot at 0x140e15055f8>

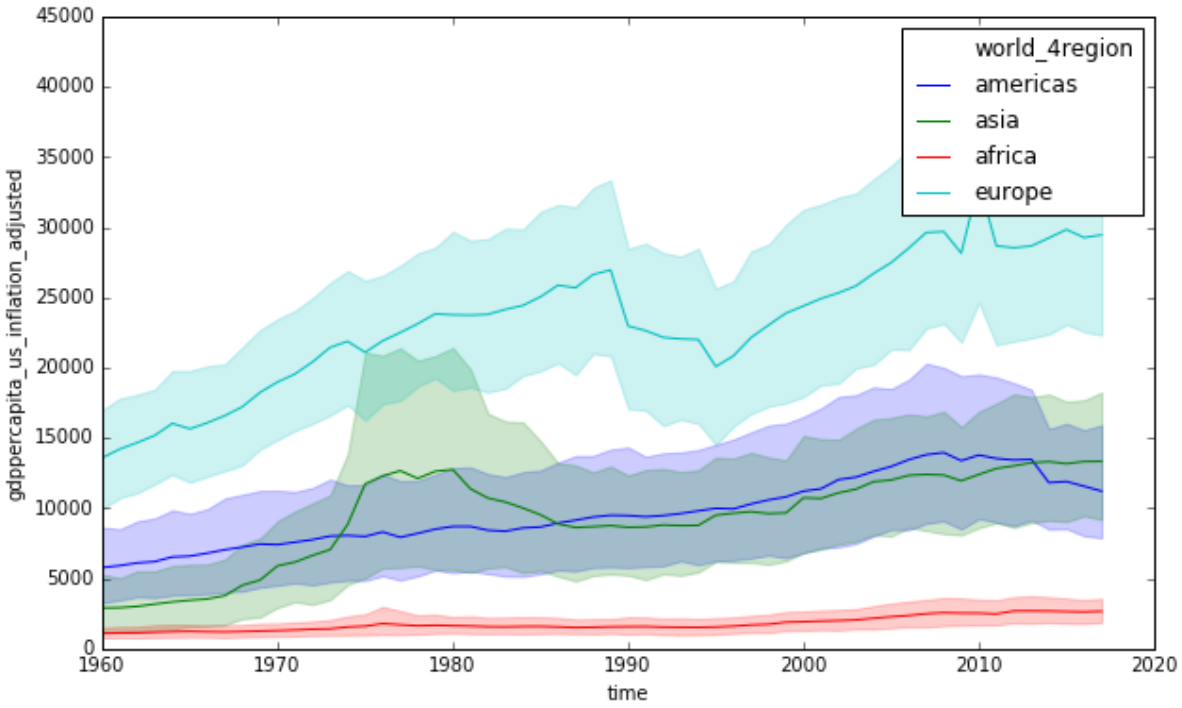


In [77]:
```
gdp_capita = pd.read_csv('C:\\Users\\hudie\\Downloads\\ddf--datapoints--gdpper
capita_us_inflation_adjusted--by--geo--time.csv')
continent_map = pd.read_csv('C:\\Users\\hudie\\Downloads\\ddf--entities--geo--
country.csv')
continent_map = continent_map[['country', 'world_4region']]
plot_data = pd.merge(gdp_capita, continent_map, left_on = 'geo', right_on='cou
ntry')

plt.figure(figsize=(10, 6))
sns.lineplot(y='gdppercapita_us_inflation_adjusted', x= 'time', hue='world_4re
gion', data=plot_data )
```

```
C:\Users\hudie\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureW
arning: Using a non-tuple sequence for multidimensional indexing is deprecate
d; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be in
terpreted as an array index, `arr[np.array(seq)]`, which will result either i
n an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[77]: <matplotlib.axes._subplots.AxesSubplot at 0x140e1505128>



From the plot we can see, overall, income and life expectancy are growing while children mortality is decreasing. We can notice there is some huge outlier in the life expectanct plot, which is caused by missing values.

Being specific to geo location, we see africa's income is really lagged behand, if we adjust gdp/capita by inflation, there is no much income growth in the past 50 years.On the other hand europe is growing the fastest as most of european become developed contryies in the past 50 years. Americas and Asia are in the middle range as there are both developed countries (U.S., Japan, etc) and lots of other developing countries in these area.

The plots make sense to me, as productivity increase by time, life expectancy is growing proportianlly and children mortality rate is decreasing proportionally. There are similar trend in different continents.

In [ ]:

Problem4

I'm looking at CO2 emission per capita and Coal consumption per capita, in different continents in past 50 years

Before making plots i assume they would have strong positive relations, because the more coal consumed the more co2 emission should we have. And after making the plots we can confirm our assumption. For regions, we see europe consume most coals and have the most co2 emission, while afirca has the least.

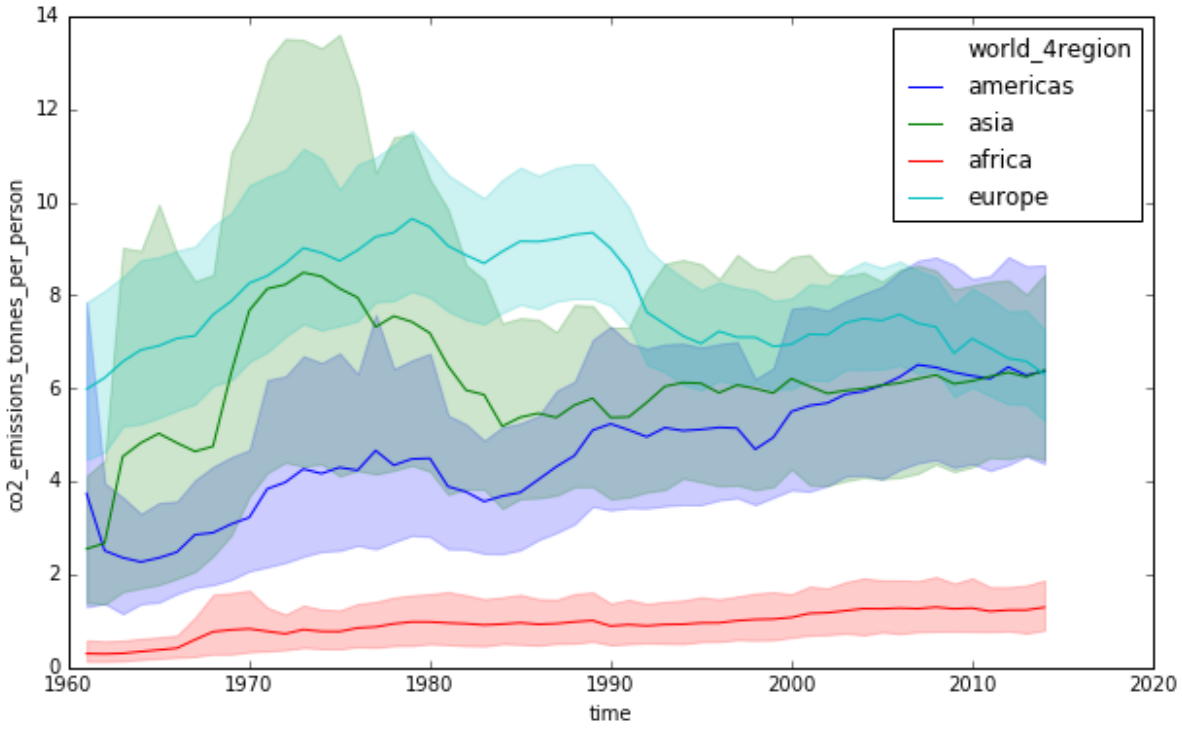
Looking at the trend we see both co2 emission and coal consumption are growing in 1960s to 1970s, then stop grow and (coal consumption even decreased a bit) after words. This might comes from adjustment by capita(we have more population), environmental protections, or swift to other more clean energy. (Say oil is more clean than coal)

In [88]:
```
co2_emission = pd.read_csv('C:\\Users\\hudie\\Downloads\\ddf--datapoints--co2_
emissions_tonnes_per_person--by--geo--time.csv')
co2_emission = co2_emission[co2_emission['time']>1960]
plot_data = pd.merge(co2_emission, continent_map, left_on = 'geo', right_on='c
ountry')

plt.figure(figsize=(10, 6))
sns.lineplot(y='co2_emissions_tonnes_per_person', x= 'time', hue='world_4regio
n', data=plot_data )
```

```
C:\Users\hudie\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureW
arning: Using a non-tuple sequence for multidimensional indexing is deprecate
d; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be in
terpreted as an array index, `arr[np.array(seq)]`, which will result either i
n an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[88]: <matplotlib.axes._subplots.AxesSubplot at 0x140e15dc390>



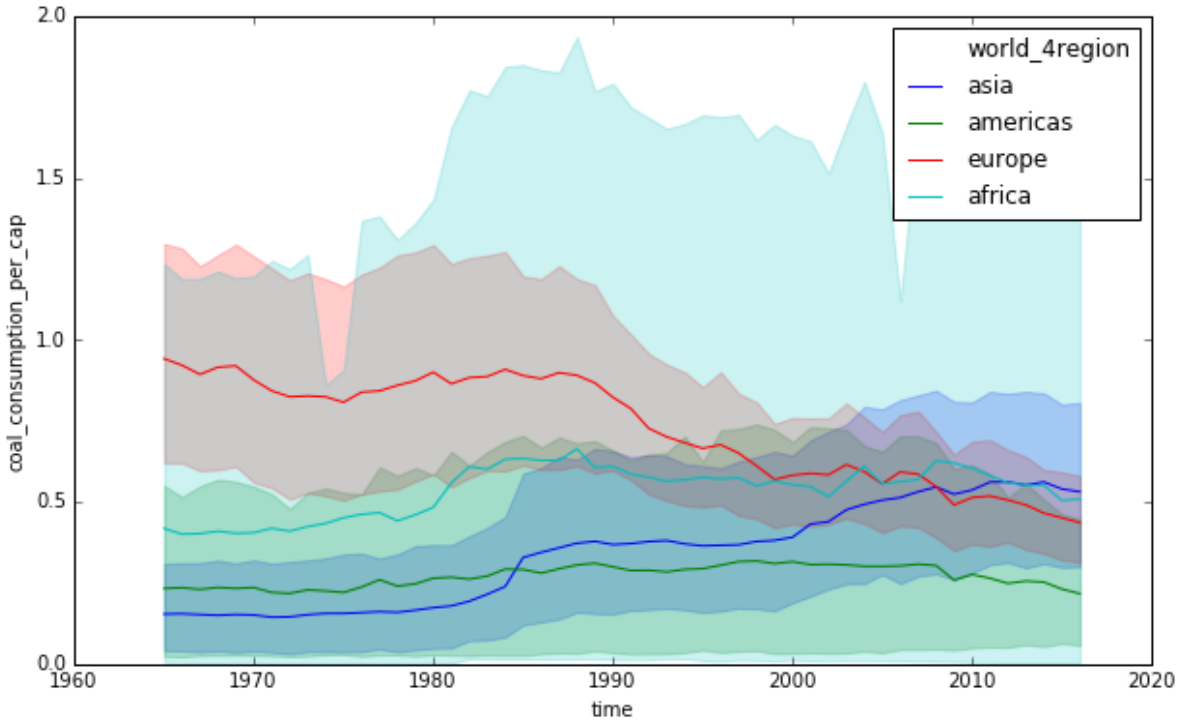
In [85]:
```
coal_consumption = pd.read_csv('C:\\Users\\hudie\\Downloads\\ddf--datapoints--coal_consumption_per_cap--by--geo--time.csv')

plot_data = pd.merge(coal_consumption, continent_map, left_on = 'geo', right_on='country')

plt.figure(figsize=(10, 6))
sns.lineplot(y='coal_consumption_per_cap', x= 'time', hue='world_4region', data=plot_data )
```

```
C:\Users\hudie\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[85]: <matplotlib.axes._subplots.AxesSubplot at 0x140e15df208>



Problem5

Did you use static or interactive plots to answer the previous problems?

Explore the data using the interactive visualization tools at https://www.gapminder.org/tools (https://www.gapminder.org/tools), and watch the TED talk "The best stats you've ever seen" at https://www.youtube.com/watch?v=hVimVzgtD6w (https://www.youtube.com/watch?v=hVimVzgtD6w). Discuss the advantages, disadvantages, and relative usefulness of using interactive/dynamic visualizations versus static visualizations.

I use static plots.

Advantages of Static Plot:

1. Require less resource, for example we can use static plots in books and other paper media but cannot really use dynamic ones.

2. Allow user to pause at anytime for discussions/thinking.

3. Less effort to make simpler plots

Advantages of Static Plot :

1. Great to showcase, and is great to visualize trend

2. Much easier and has better usage to show multi-dimensional data.

Overall i think static plots require less effort and is good when data/plots are in low-dimension and relatively simple; while dynamic visualization are stronger to show trends and more complex