```
#Descriptive analysis of demographic data
#The International Data Base (IDB) of the U.S. Census Bureau contains various demographic
# data (currently from 1950 to 2100) on all states and regions of our world that are
# recognized by the US Department of State and have a population of 5000 or more. The
# sources of the database are information from state institutions, such as censuses, surveys
# or administrative records, as well as estimates and projections by the U.S. Census Bureau
# itself.
```

```
# Importing required libraries
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
import seaborn as sns
```

```
# Font, Fontsize
sns.set(rc={'figure.figsize':(15,8)}, font_scale = 1.5)
sns.set_style({'font.family':'serif', 'font.serif':'sans-serif'})
```

```
#read census2001_2021.csv file
census_df = pd.read_csv('census2001_2021.csv', encoding = 'latin-1')
```

```
# view first 5 rows of data
census_df.head()
```

	Country.Name	Subregion	Region	Year	Life.Expectancy..Both.Sexes	Life.Expectancy.
0	Afghanistan	South-Central Asia	Asia	2001	45.81	
1	Afghanistan	South-Central Asia	Asia	2021	53.25	
2	Albania	Southern Europe	Europe	2001	75.14	
3	Albania	Southern Europe	Europe	2021	79.23	
4	Algeria	Northern Africa	Africa	2001	72.19	

```
#Changing Column Names for better readability
census_df.columns = ["Country","Subregion","Region","Year","LifeExp_both","LifeExpMale","LifeExpFemale","In
```

```
# Get description of data
description = census_df.describe()
#Save in latex table
description.to_latex('data_desc.tex')
description
```

	Year	LifeExp_both	LifeExpMale	LifeExpFemale	InfantMortRate_both
count	454.000000	448.000000	448.000000	448.000000	448.000000
mean	2011.000000	71.443103	69.043192	73.968192	27.512612
std	10.011031	8.806907	8.495558	9.255673	27.986507
min	2001.000000	44.210000	43.060000	44.780000	1.530000
25%	2001.000000	67.612500	64.995000	69.565000	7.045000
50%	2011.000000	73.405000	70.985000	76.210000	16.300000
75%	2021.000000	77.767500	74.992500	80.742500	37.922500
max	2021.000000	89.400000	85.550000	93.400000	144.770000

```
census_df.skew(axis = 0, skipna = True)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance col
  """Entry point for launching an IPython kernel.
Year                    0.000000
LifeExp_both           -0.986360
LifeExpMale            -0.905063
LifeExpFemale          -1.030130
InfantMortRate_both     1.590706
dtype: float64
```

Descriptive Analysis

```
# Filter the dataset to the year 2021 for task 1 to 3
census_2021 = census_df[census_df['Year'] == 2021]
census_2021.head()
```

	Country	Subregion	Region	Year	LifeExp_both	LifeExpMale	LifeExpFemale	Infan
1	Afghanistan	South-Central Asia	Asia	2021	53.25	51.73	54.85	
3	Albania	Southern Europe	Europe	2021	79.23	76.55	82.12	
5	Algeria	Northern Africa	Africa	2021	77.79	76.32	79.33	

```
#new column to store the difference between male and female life expectancy
census_2021['LifeExpDif'] = census_2021['LifeExpMale'] - census_2021['LifeExpFemale']
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.
```

```
# Get description of data for year 2021
description = census_2021.describe()
#Save in latex table
description.to_latex('data2021_desc.tex')
description
```

	Year	LifeExp_both	LifeExpMale	LifeExpFemale	InfantMortRate_both	LifeExpDi
count	227.0	227.000000	227.000000	227.000000	227.000000	227.00000
mean	2021.0	74.276432	71.784802	76.891189	20.245683	-5.10638
std	0.0	6.912253	6.742388	7.208768	19.192837	1.74342
min	2021.0	53.250000	51.730000	54.850000	1.530000	-11.44000
25%	2021.0	69.730000	67.585000	72.290000	6.270000	-6.06500
50%	2021.0	75.560000	72.990000	78.360000	12.580000	-4.87000
75%	2021.0	79.425000	76.945000	82.340000	29.480000	-3.84000
max	2021.0	89.400000	85.550000	93.400000	106.750000	2.11000

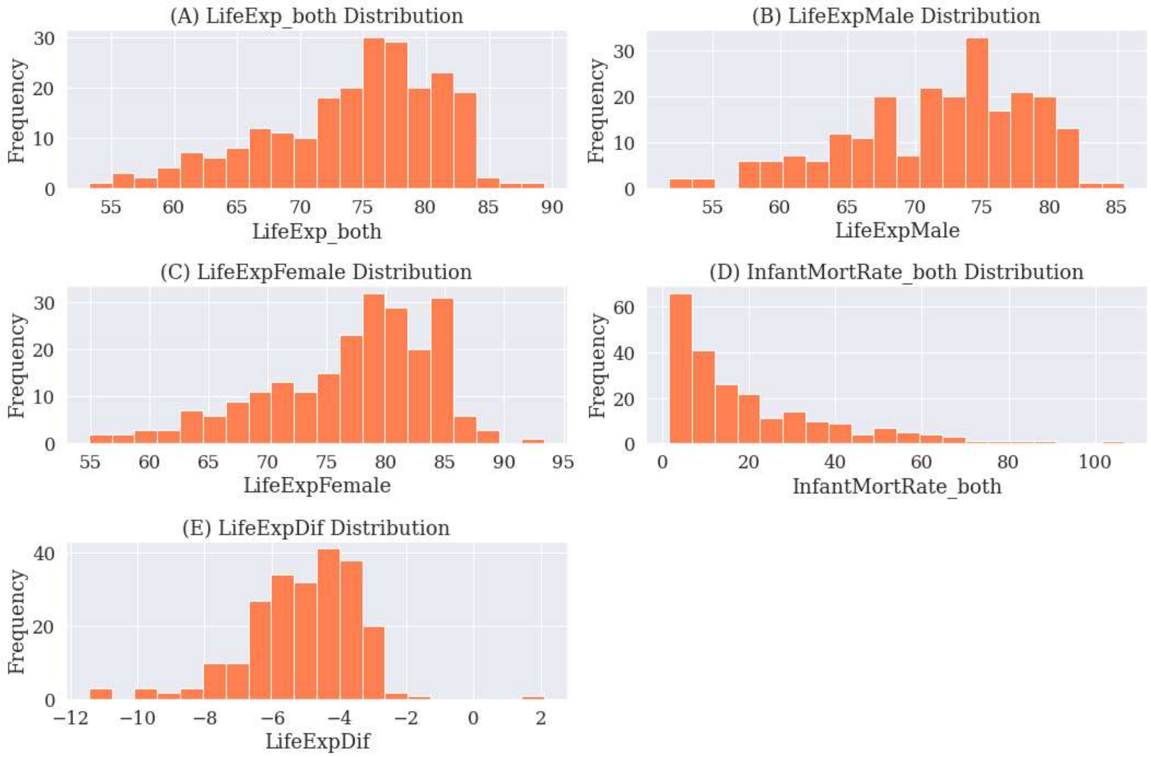
```
census_2021.skew(axis = 0, skipna = True)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance col
  """Entry point for launching an IPython kernel.
Year                    0.000000
LifeExp_both           -0.727938
LifeExpMale            -0.645764
LifeExpFemale          -0.775516
InfantMortRate_both     1.564223
LifeExpDif             -0.621411
dtype: float64
```

```
#filtering the columns for univariate analysis
#Columns[LifeExp_both,LifeExpMale,LifeExpFemale,InfantMortRate_both,LifeExpDif]
Uni_analysis = ['LifeExp_both','LifeExpMale','LifeExpFemale','InfantMortRate_both','LifeExpDif']
```

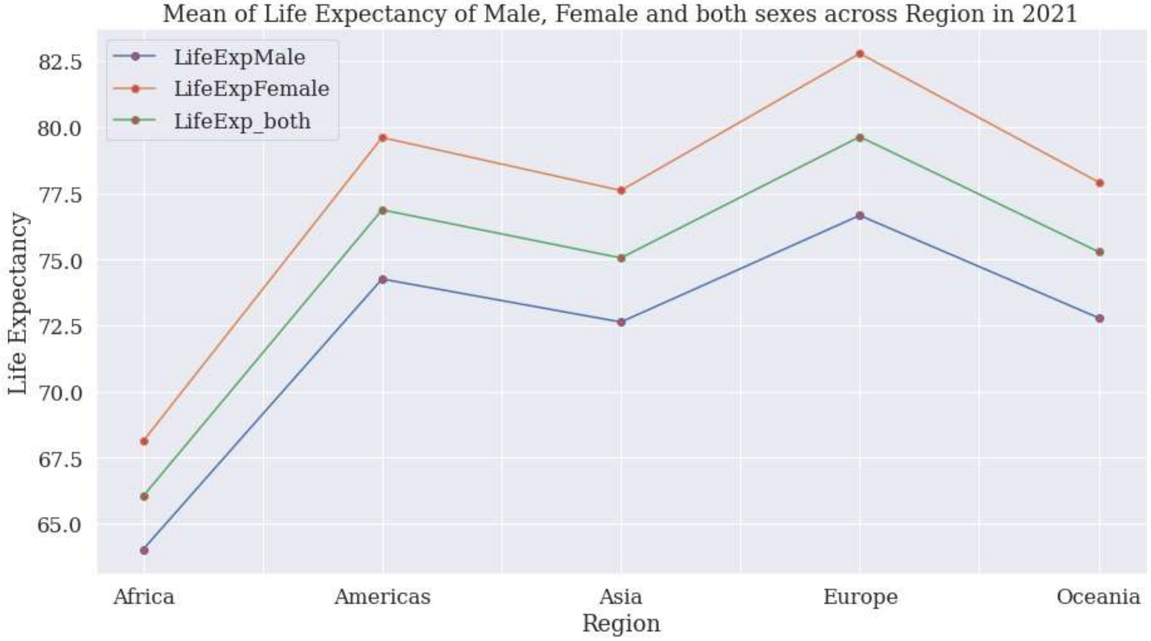
```
# Function to draw histograms for the Fertility and Life Expectancy values in a single figure.
#Histogram plots for - Columns[LifeExp_both,LifeExpMale,LifeExpFemale,InfantMortRate_both,LifeExpDif] -univ

def plot_histogram(datFra, var, rows, cols):
    fig = plt.figure()
    for i, var_name in enumerate(var):
        fig.set_figheight(10)
        ax = fig.add_subplot(rows,cols,i + 1)
        datFra[var_name].hist(bins=20,ax=ax,color='#FF7F50')
        ax.set_title('('+chr(i+65)+') '+var_name +" Distribution")
        ax.set_xlabel(var_name)
        ax.set_ylabel('Frequency')
    fig.tight_layout()
    #Saving the figure to a pdf file
    fig.savefig('histograms.pdf')
    plt.show()
```

```
histograms_df = census_2021[Uni_analysis]
plot_histogram(histograms_df, histograms_df.columns, 3, 2)
```

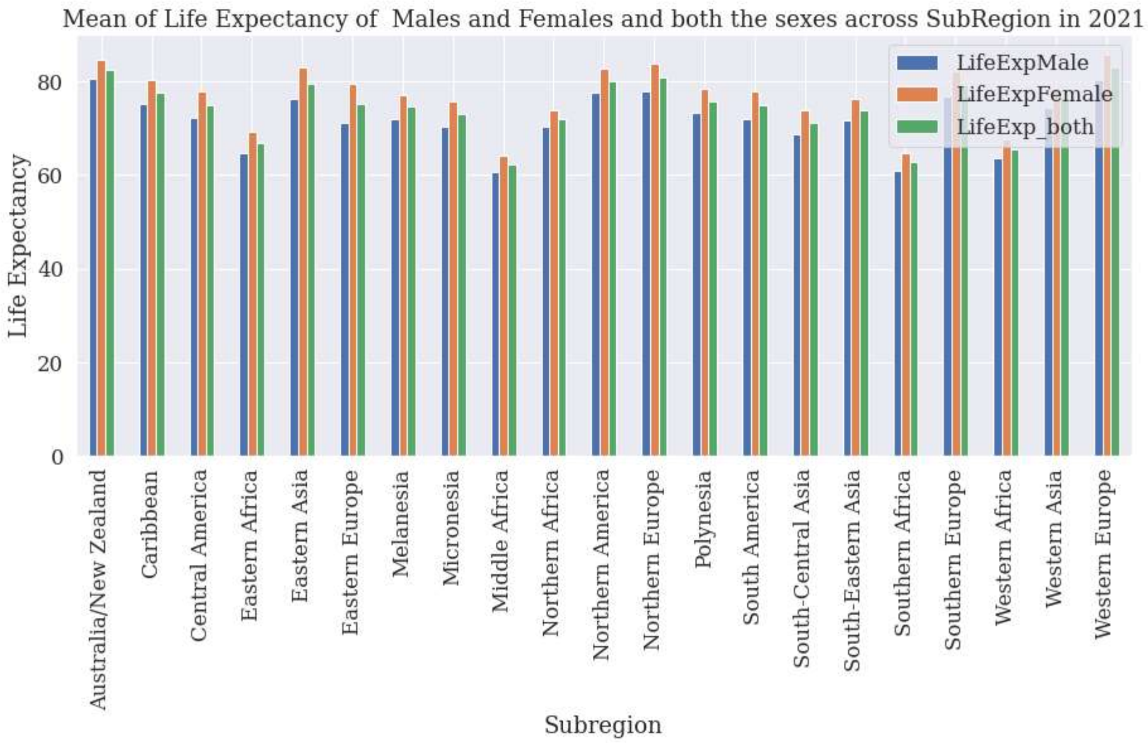


```
#plotting the Mean of Life expectancy for both, and male and female across given regions in 2021
census_2021.groupby(['Region'])[['LifeExpMale', 'LifeExpFemale','LifeExp_both']].mean().plot(marker = 'o',
plt.ylabel('Life Expectancy')
plt.savefig('LifeExpAcrossRegion.pdf')
plt.title("Mean of Life Expectancy of Male, Female and both sexes across Region in 2021")
```

```
Text(0.5, 1.0, 'Mean of Life Expectancy of Male, Female and both sexes across Region
in 2021')
```



```
#plotting the Mean of Life expectancy for both,and male and female across given Subregions in 2021
census_2021.groupby(['Subregion'])[['LifeExpMale', 'LifeExpFemale','LifeExp_both']].mean().plot(kind = 'bar
plt.ylabel('Life Expectancy')
plt.subplots_adjust(bottom=0.3)
plt.title("Mean of Life Expectancy of  Males and Females and both the sexes across SubRegion in 2021")
plt.savefig('LifeExpSubregion.pdf')
```



Correlation Analysis

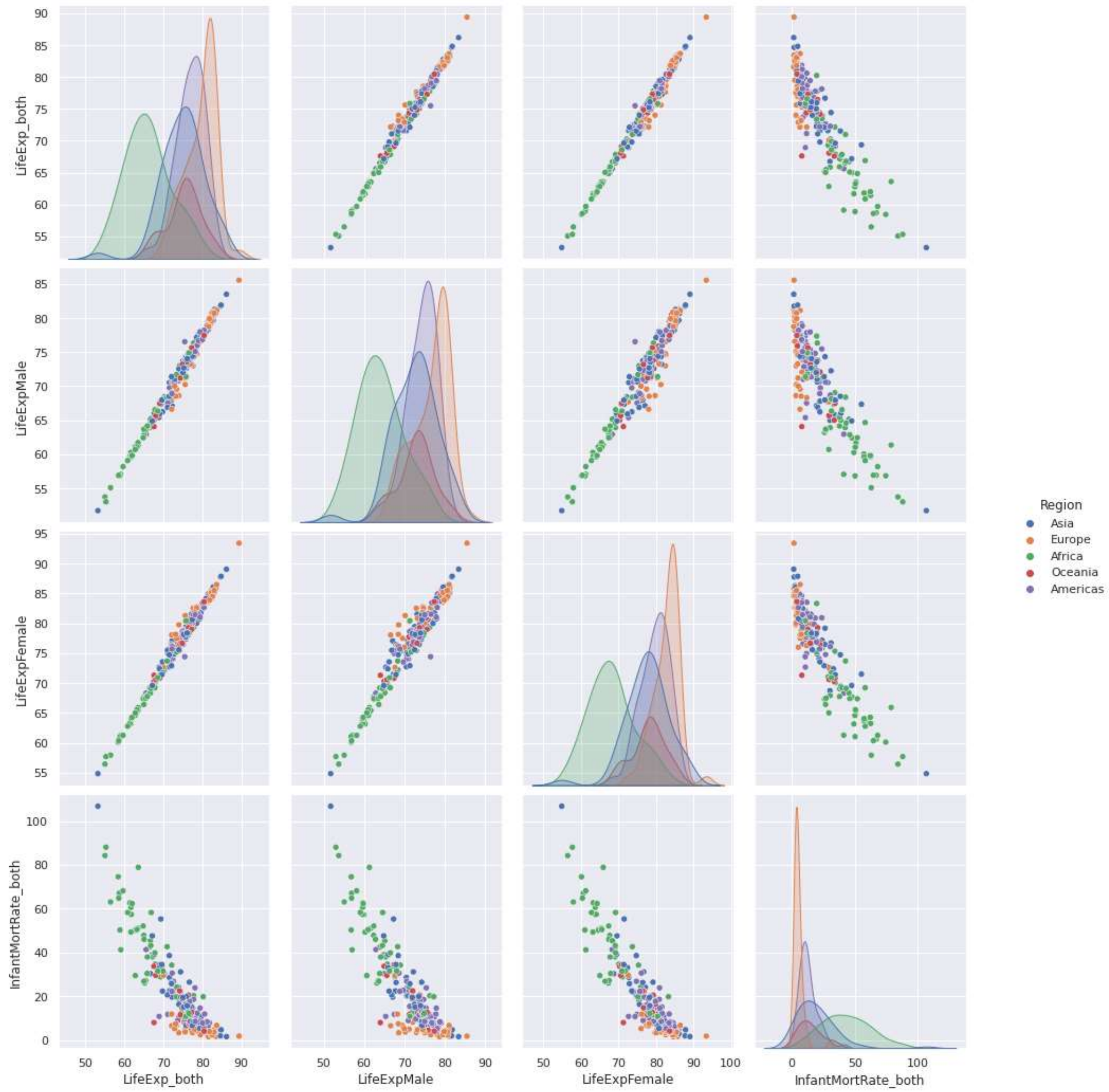
```
var = ['LifeExp_both','LifeExpMale','LifeExpFemale','InfantMortRate_both']
```

```
#correlation matrix for each pair of the four numeric variables
corr_matrix = census_2021[var].corr()
corr_matrix.to_latex('corr_matrix.tex')
corr_matrix
```

	LifeExp_both	LifeExpMale	LifeExpFemale	InfantMortRate_both
LifeExp_both	1.000000	0.992540	0.992870	-0.904929
LifeExpMale	0.992540	1.000000	0.970969	-0.883141
LifeExpFemale	0.992870	0.970969	1.000000	-0.913436
InfantMortRate_both	-0.904929	-0.883141	-0.913436	1.000000

```
# Drop the non-relavent columns
corr_var=census_2021
corr_var = corr_var.drop(columns = ['Year', 'Country','Subregion','LifeExpDif'])
```

```
sns.set(rc={'figure.figsize':(15,8)}, font_scale = 1.0)
pairplot=sns.pairplot(corr_var, hue = 'Region')
pairplot.fig.set_size_inches(15,15)
plt.savefig('pairplot.pdf')
```



```
sns.set(rc={'figure.figsize':(15,8)}, font_scale = 1.5)
```

```
a = census_2021['LifeExp_both'].corr(census_2021['LifeExpMale'])
b = census_2021['LifeExp_both'].corr(census_2021['LifeExpFemale'])
c = census_2021['LifeExp_both'].corr(census_2021['InfantMortRate_both'])
print(a,b,c)
```

```
0.9925398873900177 0.9928695953869655 -0.904928515580336
```

Variability Analysis

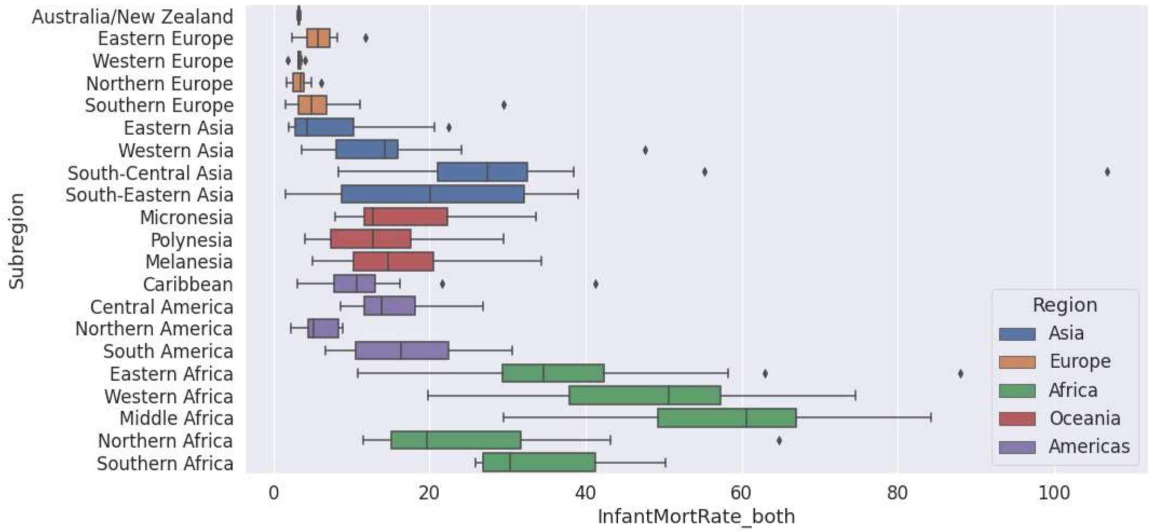
```
# Median of each variable grouped by Subregion
subregion_medians = census_2021.groupby(['Subregion'])[var].median()
subregion_medians.to_latex('subregion_medians.tex')
subregion_medians
```

	LifeExp_both	LifeExpMale	LifeExpFemale	InfantMortRate_both
Subregion				
Australia/New Zealand	82.610	80.650	84.680	3.275
Caribbean	78.310	75.960	81.090	10.700
Central America	75.005	71.940	77.910	13.885
Eastern Africa	67.070	64.980	69.220	34.620
Eastern Asia	81.865	78.795	85.100	4.360
Eastern Europe	74.655	70.820	79.235	5.705
Melanesia	74.870	73.180	76.820	14.690
Micronesia	74.380	72.060	76.760	12.790
Middle Africa	61.710	60.270	63.810	60.580
Northern Africa	74.180	72.990	75.450	19.680
Northern America	81.200	78.730	83.700	5.220
Northern Europe	81.685	79.705	83.885	3.495
Polynesia	76.890	74.050	78.990	12.730
South America	75.035	71.515	78.725	16.340
South-Central Asia	72.095	69.510	75.590	27.480
South-Eastern Asia	72.820	70.620	75.120	20.160
Southern Africa	65.040	63.210	66.420	30.380
Southern Europe	80.740	77.740	83.600	4.910
Western Africa	63.530	61.700	65.550	50.710
Western Asia	76.400	74.250	78.680	14.250
Western Europe	82.360	79.720	85.190	3.290

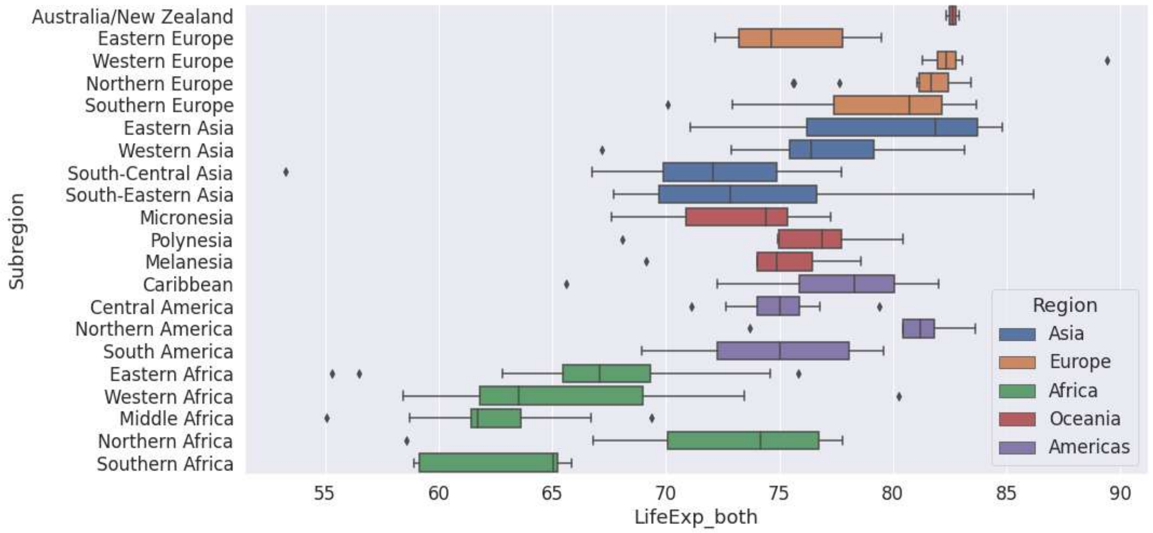
```
# Inter-quartile of each variable based on the sub-regions
grouper = census_2021.groupby(['Subregion'])[var]
q1, q3 = grouper.quantile(0.25), grouper.quantile(0.75)
subregions_iqr = q3 - q1
subregions_iqr.to_latex('subregions_iqr.tex')
subregions_iqr
```

	LifeExp_both	LifeExpMale	LifeExpFemale	InfantMortRate_both
Subregion				
Australia/New Zealand	0.2800	0.0800	0.4900	0.2250
Caribbean	4.1800	2.8700	4.9000	5.2100
Central America	1.8225	2.7550	2.6700	6.5125
Eastern Africa	3.8400	4.1000	3.3600	12.9800
Eastern Asia	7.5250	7.7675	7.1850	7.3900
Eastern Europe	4.5525	5.7050	3.4975	2.9100
Melanesia	2.4500	2.4600	2.5900	10.2500
Micronesia	4.4600	5.0750	4.4200	10.7000
Middle Africa	2.1700	1.6900	2.7600	17.7400
Northern Africa	6.6550	6.6400	6.6750	16.5350
Northern America	1.4000	0.6500	2.4400	3.9100
Northern Europe	1.2675	1.6275	1.0250	1.3900
Polynesia	2.7500	3.4650	2.4050	10.1850
South America	5.8275	5.4625	5.7575	11.8925
South-Central Asia	4.9675	5.0075	4.5050	11.4450
South-Eastern Asia	6.9300	6.3650	7.5550	23.3300
Southern Africa	6.1100	6.6300	6.0400	14.3800
Southern Europe	4.7650	4.9700	4.5125	3.5850
Western Africa	7.1900	7.5200	6.6900	19.4000
Western Asia	3.6900	4.3750	3.9000	7.8700
Western Europe	0.8300	1.0000	1.0900	0.2100

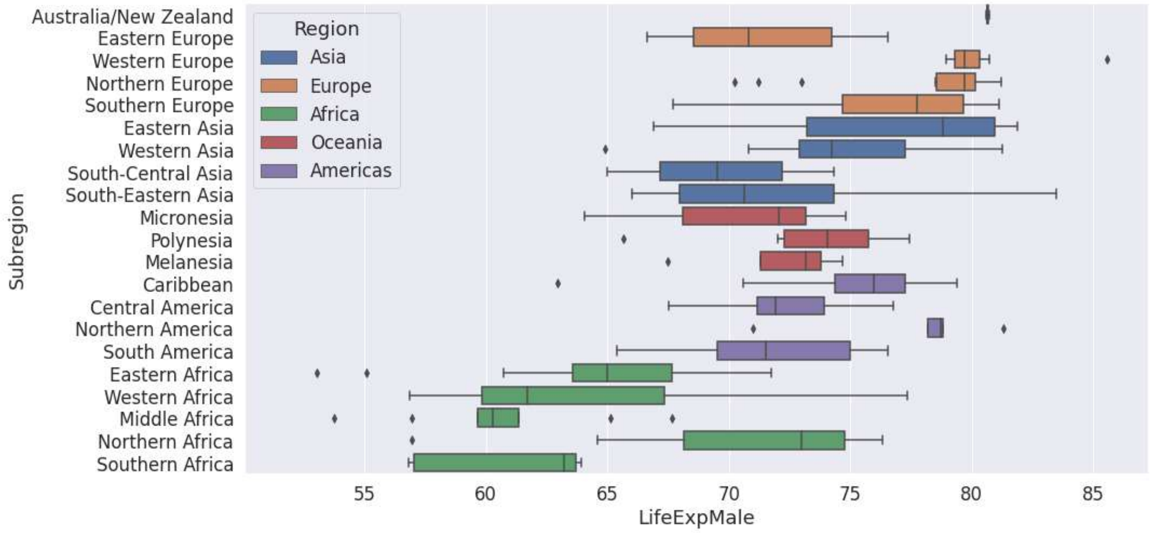
```
# Ordering the Subregions
orderList = ['Australia/New Zealand', 'Eastern Europe', 'Western Europe', 'Northern Europe',
                'Southern Europe','Eastern Asia', 'Western Asia', 'South-Central Asia',
                'South-Eastern Asia', 'Micronesia', 'Polynesia', 'Melanesia', 'Caribbean', 'Central America
                'Northern America','South America','Eastern Africa', 'Western Africa', 'Middle Africa', 'No
                'Southern Africa']


sns.boxplot(data=census_2021, x="InfantMortRate_both", y="Subregion", hue="Region", dodge=False,order=order
plt.savefig('InfantMortRate_both.pdf', format="pdf")
plt.show()
```



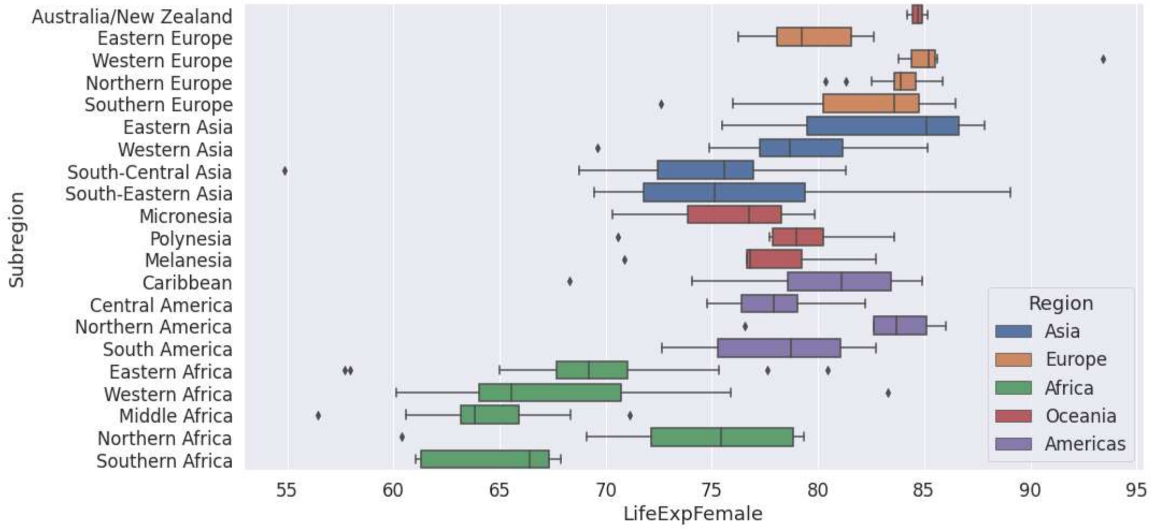
```
sns.boxplot(data=census_2021, x="LifeExp_both", y="Subregion", hue="Region", dodge=False,order=orderList)
plt.savefig('LifeExp_both.pdf', format="pdf")
plt.show()
```



```
sns.boxplot(data=census_2021, x="LifeExpMale", y="Subregion", hue="Region", dodge=False,order=orderList)
plt.savefig('LifeExpMale.pdf', format="pdf")
plt.show()
```



```
sns.boxplot(data=census_2021, x="LifeExpFemale", y="Subregion", hue="Region", dodge=False,order=orderList)
plt.savefig('LifeExpFemale.pdf', format="pdf")
plt.show()
```



Trend Analysis

```
# Filter the dataset to the year 2001
census_2001 = census_df[census_df['Year'] == 2001]
```

```
#plot ScatterPlot 2001 vs 2000
def plot_scatterplot(var):
    fig = plt.figure()
    fig = plt.figure()
    # The code below is used to place the plots in a grid-like figure
    for i, var_name in enumerate(var):
        fig.set_figheight(12)
        ax = fig.add_subplot(2, 2, i + 1)
        x = census_2001[var_name]
        y = census_2021[var_name]
        plt.scatter(x, y,color='#FF7F50')
        plt.plot(x, x, color = '#67A3D9', label='x=y')
```