Problem Statement

```
Life Expectancy:
               The average time an organism is expected to live.

The problem statement motive is to determine the life expectancy considering various
factors like immunization factors,mortality factors,social factors,economic factors and
health related factors as well.

The main aim of the problem statement is Life Expectancy Prediction by taking many
factors into consideration.
It was found that affect of immunization and human development index was not taken into
account in the past. Also, some of the past research was done considering multiple linear
regression based on data set of one year for all the countries. Hence, this gives
motivation to resolve both the factors stated previously by formulating a regression
model based on mixed effects model and multiple linear regression while considering data
from a period of 2000 to 2015 for all the countries
```

Importing Necessary Libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Loading Data

In [2]:

```
data=pd.read_csv('Life Expectancy Data.csv')
```

In [3]:

```
data.head()
```

Out[3]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0

5 rows × 22 columns

Data Dictionary

Country -> Country
Year -> Year
Status -> Developed or Developing
Life expectancy -> Life Expectancy in age
Adult Mortality -> Adult mortality rates of both sexes(probability of dying between 15 and 60 years per 1000 population)
infant deaths -> no. of infant deaths per 1000 population
Alcohol -> alcohol recorded per capita(15+) consumption (in litres of pure alcohol) percentage expenditure -> expenditure on health as a percent of GDP per capita(%)
Hepatitis B -> Hepatitis B (HepB) immunization coverage among 1 year olds(%)
Measles -> Measles - no. of reported cases per 1000 population BMI -> average body mass index of entire population under-five deaths -> number of under-five deaths per 1000 population
Polio -> Polio(Pol3) immunization coverage among 1 year olds(%) Total expenditure -> general government expenditure on health as percent of total government expenditure(%) Diphtheria -> Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1 year old(%)
HIV/AIDS -> deaths per 1000 live births HIV/AIDS (0-4 years)
GDP -> gross domestic product per capita (in USD)
Population -> population of the country thinness 1-19 years -> prevalence of thinness among children and adolescents for age 10-19(%)
thinness 5-9 years -> prevalence of thinness among children for age 5 -9 (%) Income composition of resources -> Income composition of resources
Schooling -> number of years of schooling (years)

Exploratory Data Analysis

In [4]:

```
#checking the column wise info of the dataframe
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
Country                          2938 non-null object
Year                             2938 non-null int64
Status                           2938 non-null object
Life expectancy                  2928 non-null float64
Adult Mortality                  2928 non-null float64
infant deaths                    2938 non-null int64
Alcohol                          2744 non-null float64
percentage expenditure           2938 non-null float64
Hepatitis B                      2385 non-null float64
Measles                          2938 non-null int64
 BMI                             2904 non-null float64
under-five deaths                2938 non-null int64
Polio                            2919 non-null float64
Total expenditure                2712 non-null float64
Diphtheria                       2919 non-null float64
 HIV/AIDS                        2938 non-null float64
GDP                              2490 non-null float64
Population                       2286 non-null float64
 thinness  1-19 years            2904 non-null float64
 thinness 5-9 years              2904 non-null float64
Income composition of resources  2771 non-null float64
Schooling                        2775 non-null float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.0+ KB
```

In [5]:

```
# Get a summary of the dataframe using 'describe()'
data.describe()
```

Out[5]:

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepati
count	2938.000000	2928.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.00
mean	2007.518720	69.224932	164.796448	30.303948	4.602861	738.251295	80.94
std	4.613841	9.523867	124.292079	117.926501	4.052413	1987.914858	25.07
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.00
25%	2004.000000	63.100000	74.000000	0.000000	0.877500	4.685343	77.00
50%	2008.000000	72.100000	144.000000	3.000000	3.755000	64.912906	92.00
75%	2012.000000	75.700000	228.000000	22.000000	7.702500	441.534144	97.00
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.00

Observing Variables

In [6]:

```
# Categorical Variable 1 :: Country
distinct_countries = data['Country'].unique()
distinct_countries
```

Out[6]:

```
array(['Afghanistan', 'Albania', 'Algeria', 'Angola',
       'Antigua and Barbuda', 'Argentina', 'Armenia', 'Australia',
       'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain', 'Bangladesh',
       'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin', 'Bhutan',
       'Bolivia (Plurinational State of)', 'Bosnia and Herzegovina',
       'Botswana', 'Brazil', 'Brunei Darussalam', 'Bulgaria',
       'Burkina Faso', 'Burundi', "Côte d'Ivoire", 'Cabo Verde',
       'Cambodia', 'Cameroon', 'Canada', 'Central African Republic',
       'Chad', 'Chile', 'China', 'Colombia', 'Comoros', 'Congo',
       'Cook Islands', 'Costa Rica', 'Croatia', 'Cuba', 'Cyprus',
       'Czechia', "Democratic People's Republic of Korea",
       'Democratic Republic of the Congo', 'Denmark', 'Djibouti',
       'Dominica', 'Dominican Republic', 'Ecuador', 'Egypt',
       'El Salvador', 'Equatorial Guinea', 'Eritrea', 'Estonia',
       'Ethiopia', 'Fiji', 'Finland', 'France', 'Gabon', 'Gambia',
       'Georgia', 'Germany', 'Ghana', 'Greece', 'Grenada', 'Guatemala',
       'Guinea', 'Guinea-Bissau', 'Guyana', 'Haiti', 'Honduras',
       'Hungary', 'Iceland', 'India', 'Indonesia',
       'Iran (Islamic Republic of)', 'Iraq', 'Ireland', 'Israel', 'Italy',
       'Jamaica', 'Japan', 'Jordan', 'Kazakhstan', 'Kenya', 'Kiribati',
       'Kuwait', 'Kyrgyzstan', "Lao People's Democratic Republic",
       'Latvia', 'Lebanon', 'Lesotho', 'Liberia', 'Libya', 'Lithuania',
       'Luxembourg', 'Madagascar', 'Malawi', 'Malaysia', 'Maldives',
       'Mali', 'Malta', 'Marshall Islands', 'Mauritania', 'Mauritius',
       'Mexico', 'Micronesia (Federated States of)', 'Monaco', 'Mongolia',
       'Montenegro', 'Morocco', 'Mozambique', 'Myanmar', 'Namibia',
       'Nauru', 'Nepal', 'Netherlands', 'New Zealand', 'Nicaragua',
       'Niger', 'Nigeria', 'Niue', 'Norway', 'Oman', 'Pakistan', 'Palau',
       'Panama', 'Papua New Guinea', 'Paraguay', 'Peru', 'Philippines',
       'Poland', 'Portugal', 'Qatar', 'Republic of Korea',
       'Republic of Moldova', 'Romania', 'Russian Federation', 'Rwanda',
       'Saint Kitts and Nevis', 'Saint Lucia',
       'Saint Vincent and the Grenadines', 'Samoa', 'San Marino',
       'Sao Tome and Principe', 'Saudi Arabia', 'Senegal', 'Serbia',
       'Seychelles', 'Sierra Leone', 'Singapore', 'Slovakia', 'Slovenia',
       'Solomon Islands', 'Somalia', 'South Africa', 'South Sudan',
       'Spain', 'Sri Lanka', 'Sudan', 'Suriname', 'Swaziland', 'Sweden',
       'Switzerland', 'Syrian Arab Republic', 'Tajikistan', 'Thailand',
       'The former Yugoslav republic of Macedonia', 'Timor-Leste', 'Togo',
       'Tonga', 'Trinidad and Tobago', 'Tunisia', 'Turkey',
       'Turkmenistan', 'Tuvalu', 'Uganda', 'Ukraine',
       'United Arab Emirates',
       'United Kingdom of Great Britain and Northern Ireland',
       'United Republic of Tanzania', 'United States of America',
       'Uruguay', 'Uzbekistan', 'Vanuatu',
       'Venezuela (Bolivarian Republic of)', 'Viet Nam', 'Yemen',
       'Zambia', 'Zimbabwe'], dtype=object)
```

We have observed that for some country names there are whitespaces inside, for the sake of smooth learning going forward we will remove this white spaces within the values

In [7]:

```
def remove_whitespace(x):
    """
    Helper function to remove any blank space from a string
    x: a string
    """
    try:
        # Remove spaces inside of the string
        x = "".join(x.split())

    except:
        pass
    return x
```

In [8]:

```
data.Country = data.Country.apply(remove_whitespace)
```

In [9]:

```
distinct_countries = data['Country'].unique()
distinct_countries
```

Out[9]:

```
array(['Afghanistan', 'Albania', 'Algeria', 'Angola', 'AntiguaandBarbuda',
       'Argentina', 'Armenia', 'Australia', 'Austria', 'Azerbaijan',
       'Bahamas', 'Bahrain', 'Bangladesh', 'Barbados', 'Belarus',
       'Belgium', 'Belize', 'Benin', 'Bhutan',
       'Bolivia(PlurinationalStateof)', 'BosniaandHerzegovina',
       'Botswana', 'Brazil', 'BruneiDarussalam', 'Bulgaria',
       'BurkinaFaso', 'Burundi', "Côted'Ivoire", 'CaboVerde', 'Cambodia',
       'Cameroon', 'Canada', 'CentralAfricanRepublic', 'Chad', 'Chile',
       'China', 'Colombia', 'Comoros', 'Congo', 'CookIslands',
       'CostaRica', 'Croatia', 'Cuba', 'Cyprus', 'Czechia',
       "DemocraticPeople'sRepublicofKorea",
       'DemocraticRepublicoftheCongo', 'Denmark', 'Djibouti', 'Dominica',
       'DominicanRepublic', 'Ecuador', 'Egypt', 'ElSalvador',
       'EquatorialGuinea', 'Eritrea', 'Estonia', 'Ethiopia', 'Fiji',
       'Finland', 'France', 'Gabon', 'Gambia', 'Georgia', 'Germany',
       'Ghana', 'Greece', 'Grenada', 'Guatemala', 'Guinea',
       'Guinea-Bissau', 'Guyana', 'Haiti', 'Honduras', 'Hungary',
       'Iceland', 'India', 'Indonesia', 'Iran(IslamicRepublicof)', 'Iraq',
       'Ireland', 'Israel', 'Italy', 'Jamaica', 'Japan', 'Jordan',
       'Kazakhstan', 'Kenya', 'Kiribati', 'Kuwait', 'Kyrgyzstan',
       "LaoPeople'sDemocraticRepublic", 'Latvia', 'Lebanon', 'Lesotho',
       'Liberia', 'Libya', 'Lithuania', 'Luxembourg', 'Madagascar',
       'Malawi', 'Malaysia', 'Maldives', 'Mali', 'Malta',
       'MarshallIslands', 'Mauritania', 'Mauritius', 'Mexico',
       'Micronesia(FederatedStatesof)', 'Monaco', 'Mongolia',
       'Montenegro', 'Morocco', 'Mozambique', 'Myanmar', 'Namibia',
       'Nauru', 'Nepal', 'Netherlands', 'NewZealand', 'Nicaragua',
       'Niger', 'Nigeria', 'Niue', 'Norway', 'Oman', 'Pakistan', 'Palau',
       'Panama', 'PapuaNewGuinea', 'Paraguay', 'Peru', 'Philippines',
       'Poland', 'Portugal', 'Qatar', 'RepublicofKorea',
       'RepublicofMoldova', 'Romania', 'RussianFederation', 'Rwanda',
       'SaintKittsandNevis', 'SaintLucia', 'SaintVincentandtheGrenadines',
       'Samoa', 'SanMarino', 'SaoTomeandPrincipe', 'SaudiArabia',
       'Senegal', 'Serbia', 'Seychelles', 'SierraLeone', 'Singapore',
       'Slovakia', 'Slovenia', 'SolomonIslands', 'Somalia', 'SouthAfrica',
       'SouthSudan', 'Spain', 'SriLanka', 'Sudan', 'Suriname',
       'Swaziland', 'Sweden', 'Switzerland', 'SyrianArabRepublic',
       'Tajikistan', 'Thailand', 'TheformerYugoslavrepublicofMacedonia',
       'Timor-Leste', 'Togo', 'Tonga', 'TrinidadandTobago', 'Tunisia',
       'Turkey', 'Turkmenistan', 'Tuvalu', 'Uganda', 'Ukraine',
       'UnitedArabEmirates',
       'UnitedKingdomofGreatBritainandNorthernIreland',
       'UnitedRepublicofTanzania', 'UnitedStatesofAmerica', 'Uruguay',
       'Uzbekistan', 'Vanuatu', 'Venezuela(BolivarianRepublicof)',
       'VietNam', 'Yemen', 'Zambia', 'Zimbabwe'], dtype=object)
```

In [10]:

```
# Categorical Variable 2 :: Status
distinct_status = data['Status'].unique()
distinct_status
```

Out[10]:

```
array(['Developing', 'Developed'], dtype=object)
```

removing whitespaces within some of the rest of column names

In [11]:

```
orig_cols = list(data.columns)
new_cols = []
for col in orig_cols:
    new_cols.append(col.strip().replace('  ', ' ').replace(' ', '_').replace('-', '_').lowe

data.columns = new_cols
data.head(2)
```

Out[11]:

	country	year	status	life_expectancy	adult_mortality	infant_deaths	alcohol	percen
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	

2 rows × 22 columns

In [12]:

```
# Changing column names
data.rename(columns={'thinness_1-19_years':'thinness_10-19_years'}, inplace=True)
data.head(2)
```

Out[12]:

	country	year	status	life_expectancy	adult_mortality	infant_deaths	alcohol	percen
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	

2 rows × 22 columns

CHECKING CONSTRAINTS: Adult Mortality is never below 35 and over 600 in reality, Infant deaths and below 2 or more than 900, under 5 deaths cannot be below 3 or more than 800.

In [13]:

```
rtality = data.apply(lambda x: np.nan if (x.adult_mortality < 35 or x.adult_mortality > 600)
eaths = data.apply(lambda x: np.nan if (x.infant_deaths < 2 or x.infant_deaths > 900) else x
ve_deaths = data.apply(lambda x: np.nan if (x.under_five_deaths < 3 or x.under_five_deaths
```

Data Cleaning

In [14]:

```
#checking for duplicate values
duplicate = data[data.duplicated()]
duplicate
```

Out[14]:

country	year	status	life_expectancy	adult_mortality	infant_deaths	alcohol	percentage_exp

0 rows × 22 columns

There are no duplicates in our dataset

In [15]:

```
#checking null values
a=list(data.columns)
b=[]
for i in a:
    c=data[i].isnull().sum()
    b.append(c)
null_df=pd.DataFrame({'Feature name':a,'no. of Nan':b})
null_df
```

Out[15]:

	Feature name	no. of Nan
0	country	0
1	year	0
2	status	0
3	life_expectancy	10
4	adult_mortality	468
5	infant_deaths	1206
6	alcohol	194
7	percentage_expenditure	0
8	hepatitis_b	553
9	measles	0
10	bmi	34
11	under_five_deaths	1337
12	polio	19
13	total_expenditure	226
14	diphtheria	19
15	hiv/aids	0
16	gdp	448
17	population	652
18	thinness_1_19_years	34
19	thinness_5_9_years	34
20	income_composition_of_resources	167
21	schooling	163

There are no null values for categorical columns, so for the numerical ones we should fill them with the mean values.

In [16]:

```
data.fillna(data.mean(), inplace=True)
```

Data Visualization

visualization of all the numerical parameters against life expectancy

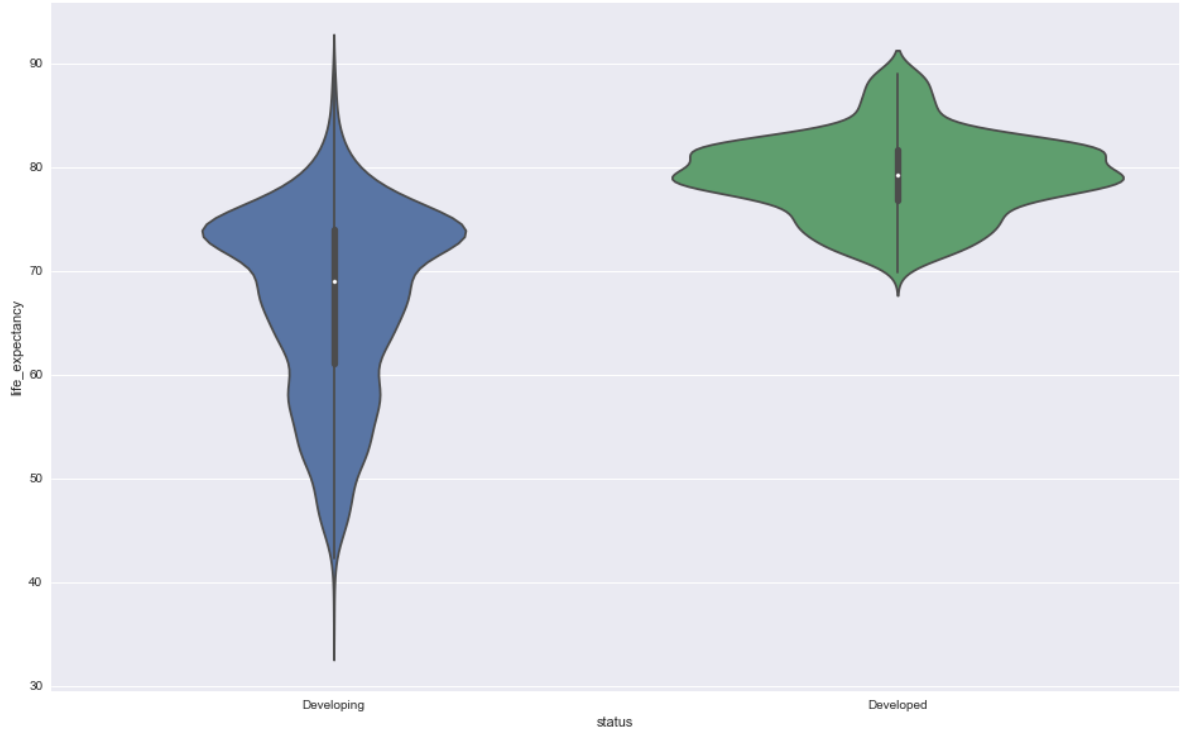
In [17]:

```
sns.set(rc={'figure.figsize':(16,10)})

sns.violinplot( data['status'],data['life_expectancy'])
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x27920619438>
```



The above diagram showcases the relationship between life expectancy and the country being developed or developing. As we can see the mean of the life expectancy is higher in developed countries. That can relate to multiple factors for example healthcare and income. The developed countries have a wider range if life expectancy in comparison.

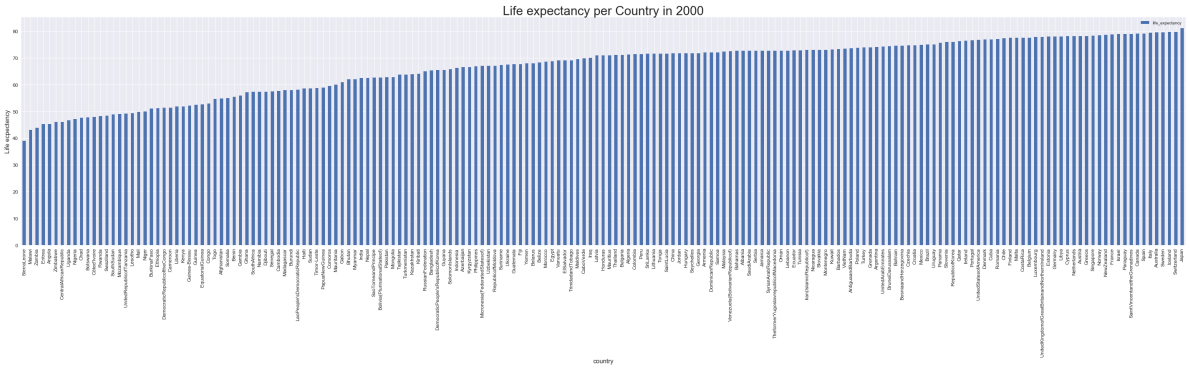
In [18]:

```
df_2000=(data[data.year==2000]
    .groupby("country")
    ["country", "life_expectancy"]
    .median()
    .sort_values(by="life_expectancy", ascending=True))

df_2000.plot(kind='bar', figsize=(50,10), fontsize=12)
plt.title("Life expectancy per Country in 2000",fontsize=30)
plt.xlabel("country",fontsize=15)
plt.ylabel("Life ecpectency",fontsize = 15)
```

Out[18]:

```
Text(0,0.5,'Life ecpectency')
```

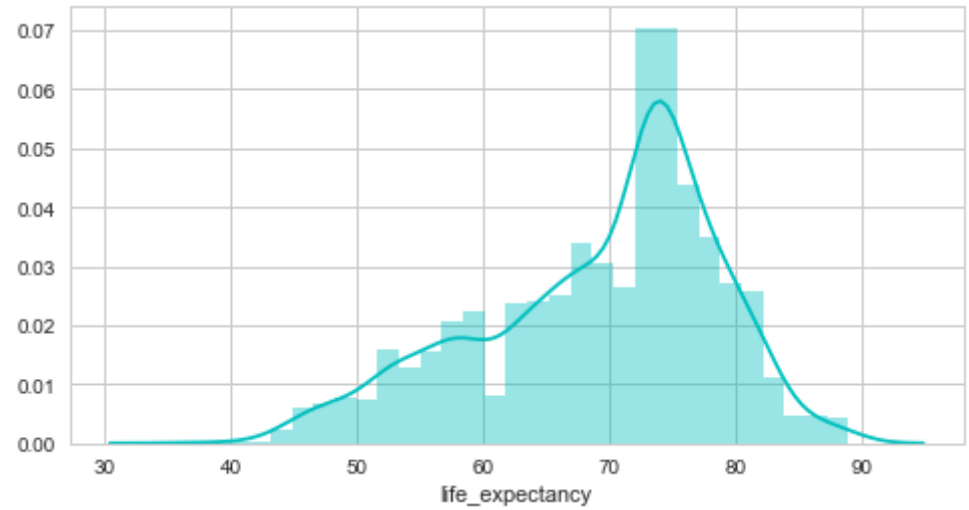


It displays that the mass majority of countries have good amount of worth of data. This is important to know mostly to make sure that certain countries are being overrepresented.

In [20]:

```
sns.set(style='whitegrid')
f,ax = plt.subplots(1,1,figsize = (8,4))
ax = sns.distplot(data['life_expectancy'], kde = True, color = 'c')
```

```
C:\Users\TARANYA SIMHADRI\anaconda\lib\site-packages\matplotlib\axes\_axes.p
y:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced
by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



In [19]:

```
# columns not required will be stored in this list
feature_not_required = []
```

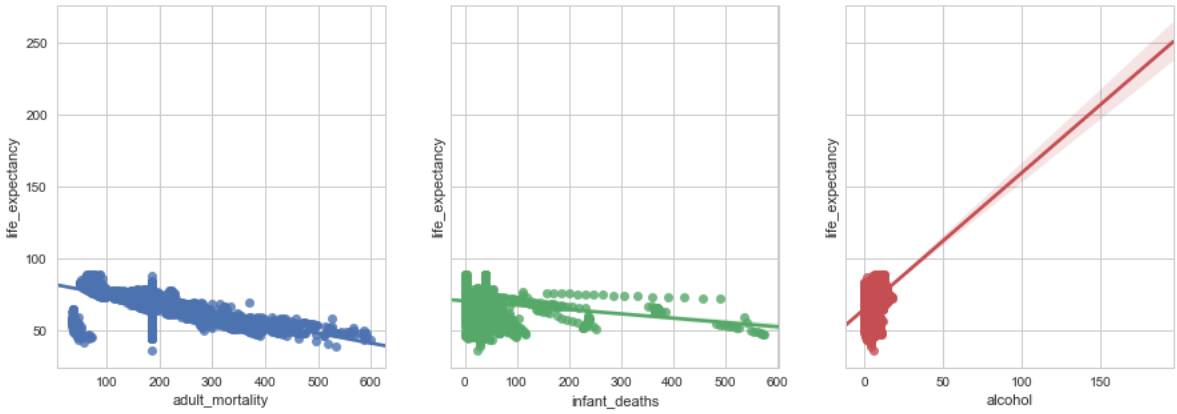
Life Expectancy is normally distributed and is thus fit for prediction using linear regression

Visualizing Life Expectancy distribution against adult_mortality,infant_deaths and alcohol

In [21]:

```
fig, (ax1, ax2, ax3) = plt.subplots(ncols=3, sharey=True, figsize=(15,5))
sns.regplot(x=data['adult_mortality'], y=data['life_expectancy'], ax=ax1)
sns.regplot(x=data['infant_deaths'], y=data['life_expectancy'], ax=ax2)
sns.regplot(x=data['alcohol'], y=data['life_expectancy'], ax=ax3)
print("Correlation of Adult Mortality to Life Expectancy:",np.corrcoef(data['adult_mortalit
print("Correlation of Infant Deaths to Life Expectancy:",np.corrcoef(data['infant_deaths'],
print("Correlation of Alcohol to Life Expectancy:",np.corrcoef(data['alcohol'], data['life_
```

```
Correlation of Adult Mortality to Life Expectancy: -0.7133009505677026
Correlation of Infant Deaths to Life Expectancy: -0.18082770202042894
Correlation of Alcohol to Life Expectancy: 0.39159833938428923
```



Observations: Adult mortality is showing a negative linear relationship. Infant death is showing no significant linear relationship, thus not required. Alcohol is showing no significant linear relationship, thus not required.

In [22]:

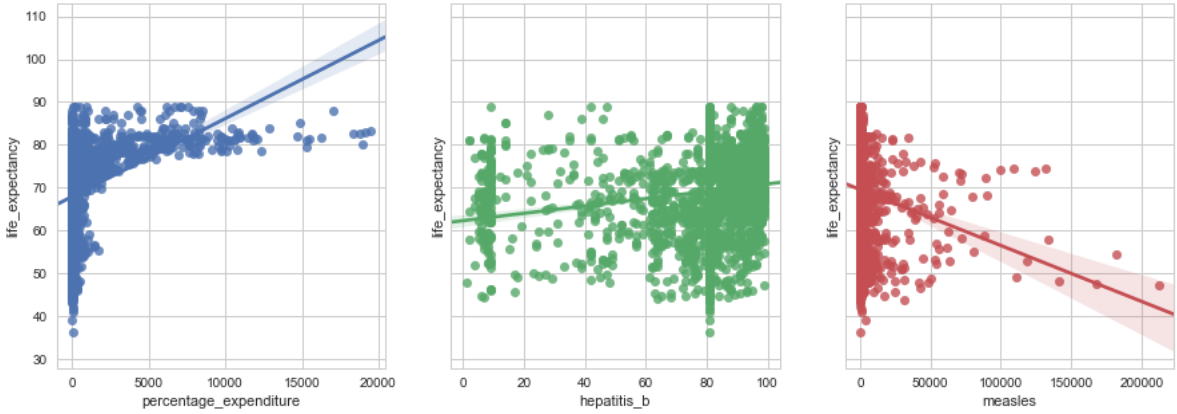
```
feature_not_required.append('infant_deaths')
feature_not_required.append('alcohol')
```

Visualizing Life Expectancy distribution against percentage_expenditure,hepatitis_b and measles

In [23]:

```
fig, (ax1, ax2, ax3) = plt.subplots(ncols=3, sharey=True, figsize=(15,5))
sns.regplot(x=data['percentage_expenditure'], y=data['life_expectancy'], ax=ax1)
sns.regplot(x=data['hepatitis_b'], y=data['life_expectancy'], ax=ax2)
sns.regplot(x=data['measles'], y=data['life_expectancy'], ax=ax3)
print("Correlation of Percentage Expenditure to Life Expectancy:",np.corrcoef(data['percent
print("Correlation of Hepatitis_B to Life Expectancy:",np.corrcoef(data['hepatitis_b'], dat
print("Correlation of Measles to Life Expectancy:",np.corrcoef(data['measles'], data['life_
```

```
Correlation of Percentage Expenditure to Life Expectancy: 0.3817911732064307
5
Correlation of Hepatitis_B to Life Expectancy: 0.20377143740026765
Correlation of Measles to Life Expectancy: -0.15757381859716962
```



Observations: Percentage_expenditure is showing no significant linear relationship, thus not required. Hepatitis_b is showing no significant linear relationship, thus not required. Measles is showing little significant linear relationship, thus required.

In [24]:

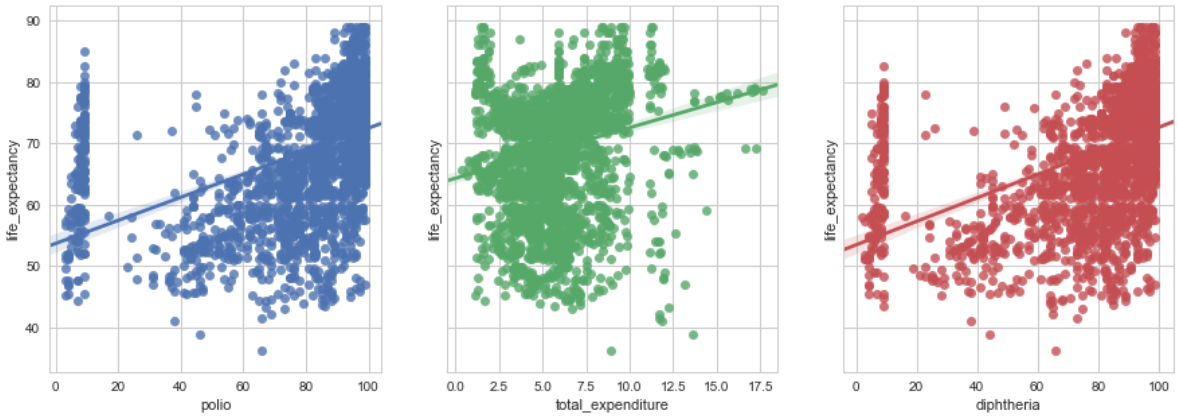
```
feature_not_required.append('percentage_expenditure')
feature_not_required.append('hepatitis_b')
```

Visualizing Life Expectancy distribution against polio,total_expenditure and diphtheria

In [25]:

```
fig, (ax1, ax2, ax3) = plt.subplots(ncols=3, sharey=True, figsize=(15,5))
sns.regplot(x=data['polio'], y=data['life_expectancy'], ax=ax1)
sns.regplot(x=data['total_expenditure'], y=data['life_expectancy'], ax=ax2)
sns.regplot(x=data['diphtheria'], y=data['life_expectancy'], ax=ax3)
print("Correlation of  polio  to Life Expectancy:",np.corrcoef(data['polio'], data['life_ex
print("Correlation of  total_expenditure to Life Expectancy:",np.corrcoef(data['total_exper
print("Correlation of  diphtheria to Life Expectancy:",np.corrcoef(data['diphtheria'], data
```

```
Correlation of  polio  to Life Expectancy: 0.46157377544579004
Correlation of  total_expenditure to Life Expectancy: 0.207980624518678
Correlation of  diphtheria to Life Expectancy: 0.4754183849366064
```



Observations: Polio is showing no significant linear relationship, thus not required. Total_expenditure is showing no significant linear relationship, thus not required. Diphtheria is showing no significant linear relationship, thus not required.

In [26]:

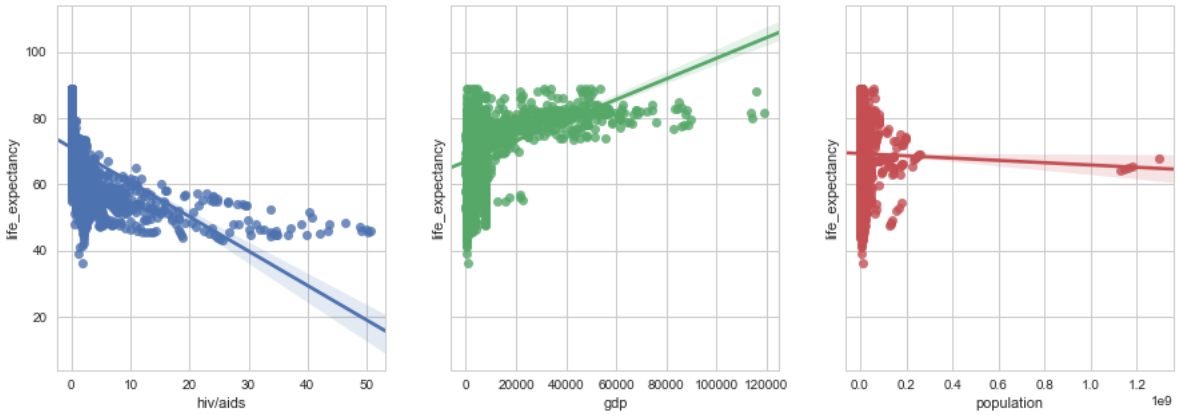
```
feature_not_required.append('polio')
feature_not_required.append('total_expenditure')
feature_not_required.append('diphtheria')
```

Visualizing Life Expectancy distribution against hiv/aids,gdp and population

In [27]:

```
fig, (ax1, ax2, ax3) = plt.subplots(ncols=3, sharey=True, figsize=(15,5))
sns.regplot(x=data['hiv/aids'], y=data['life_expectancy'], ax=ax1)
sns.regplot(x=data['gdp'], y=data['life_expectancy'], ax=ax2)
sns.regplot(x=data['population'], y=data['life_expectancy'], ax=ax3)
print("Correlation of  hiv/aids  to Life Expectancy:",np.corrcoef(data['hiv/aids'], data['l
print("Correlation of  gdp to Life Expectancy:",np.corrcoef(data['gdp'], data['life_expecta
print("Correlation of  population to Life Expectancy:",np.corrcoef(data['population'], data
```

```
Correlation of  hiv/aids  to Life Expectancy: -0.5564568165997136
Correlation of  gdp to Life Expectancy: 0.43049301854946426
Correlation of  population to Life Expectancy: -0.019637701509419594
```



Observations: HIV/aids is showing significant linear relationship, thus required. GDP is showing no significant linear relationship, thus not required. Population is showing no significant linear relationship, thus not required.

In [28]:

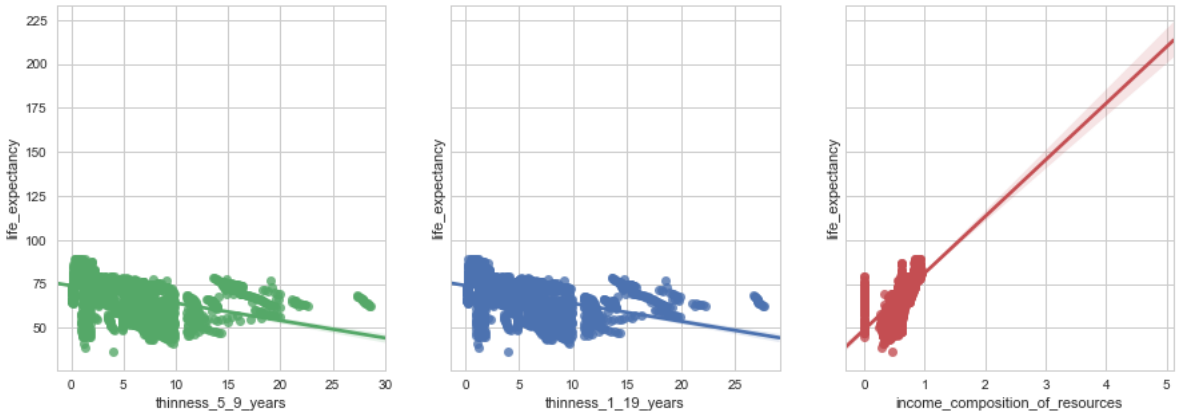
```
feature_not_required.append('gdp')
feature_not_required.append('population')
```

Visualizing Life Expectancy distribution against thinness_1_19_years,thinness_5_9_years and income_composition_of_resources

In [29]:

```
fig, (ax1, ax2, ax3) = plt.subplots(ncols=3, sharey=True, figsize=(15,5))
sns.regplot(x=data['thinness_1_19_years'], y=data['life_expectancy'], ax=ax2)
sns.regplot(x=data['thinness_5_9_years'], y=data['life_expectancy'], ax=ax1)
sns.regplot(x=data['income_composition_of_resources'], y=data['life_expectancy'], ax=ax3)
print("Correlation of  thinness_1_19_years to Life Expectancy:",np.corrcoef(data['thinness_
print("Correlation of  thinness_5_9_years to Life Expectancy:",np.corrcoef(data['thinness_5
print("Correlation of  income_composition_of_resources to Life Expectancy:",np.corrcoef(dat
```

```
Correlation of  thinness_1_19_years to Life Expectancy: -0.4721618794367624
Correlation of  thinness_5_9_years to Life Expectancy: -0.46662920814430126
Correlation of  income_composition_of_resources to Life Expectancy: 0.692482
8049608567
```



Observations: Income_composition is showing significant linear relationship, thus required. thinness_1-19 is showing no significant linear relationship, thus not required. thinness_5-9 is showing no significant linear relationship, thus not required.

In [30]:

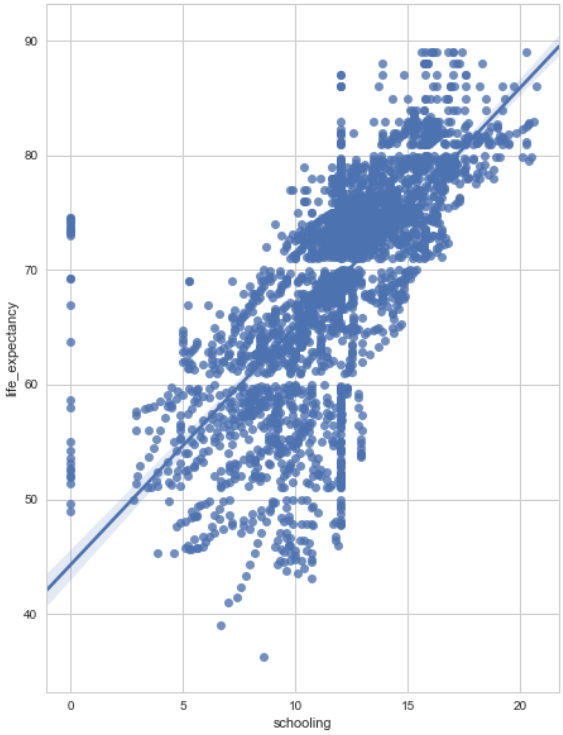
```
feature_not_required.append('thinness_1_19_years')
feature_not_required.append('thinness_5_9_years')
```

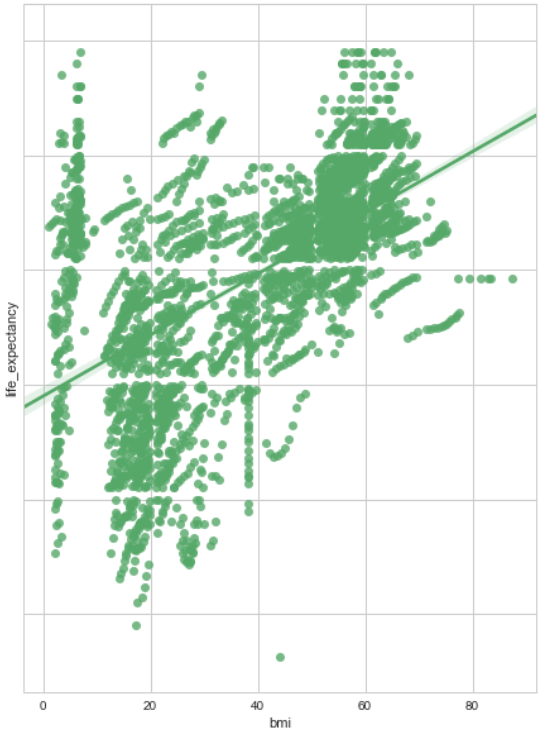
Visualizing Life Expectancy distribution against schooling and bmi

In [31]:

```
fig, (ax1,ax2) = plt.subplots(ncols=2, sharey=True)
sns.regplot(x=data['schooling'], y=data['life_expectancy'], ax=ax1)
sns.regplot(x=data['bmi'], y=data['life_expectancy'], ax=ax2)
print("Correlation of  schooling to Life Expectancy:",np.corrcoef(data['schooling'], data['
print("Correlation of  bmi to Life Expectancy:",np.corrcoef(data['bmi'], data['life_expecta
```

```
Correlation of  schooling to Life Expectancy: 0.7150663398620061
Correlation of  bmi to Life Expectancy: 0.5592553046406494
```





Observations : Schooling is showing significant linear relationship, thus required. BMI-19 is showing no significant linear relationship, thus not required.

In [32]:

```
feature_not_required.append('bmi')
```

In [45]:

```
print(feature_not_required)
```

```
['infant_deaths', 'alcohol', 'percentage_expenditure', 'hepatitis_b', 'poli
o', 'total_expenditure', 'diphtheria', 'gdp', 'population', 'thinness_1_19_y
ears', 'thinness_5_9_years', 'bmi']
```

Droping features which are not required

In [33]:

```
df_a = data.drop(feature_not_required,axis=1)
```

In [34]:

```
df_a.head()
```

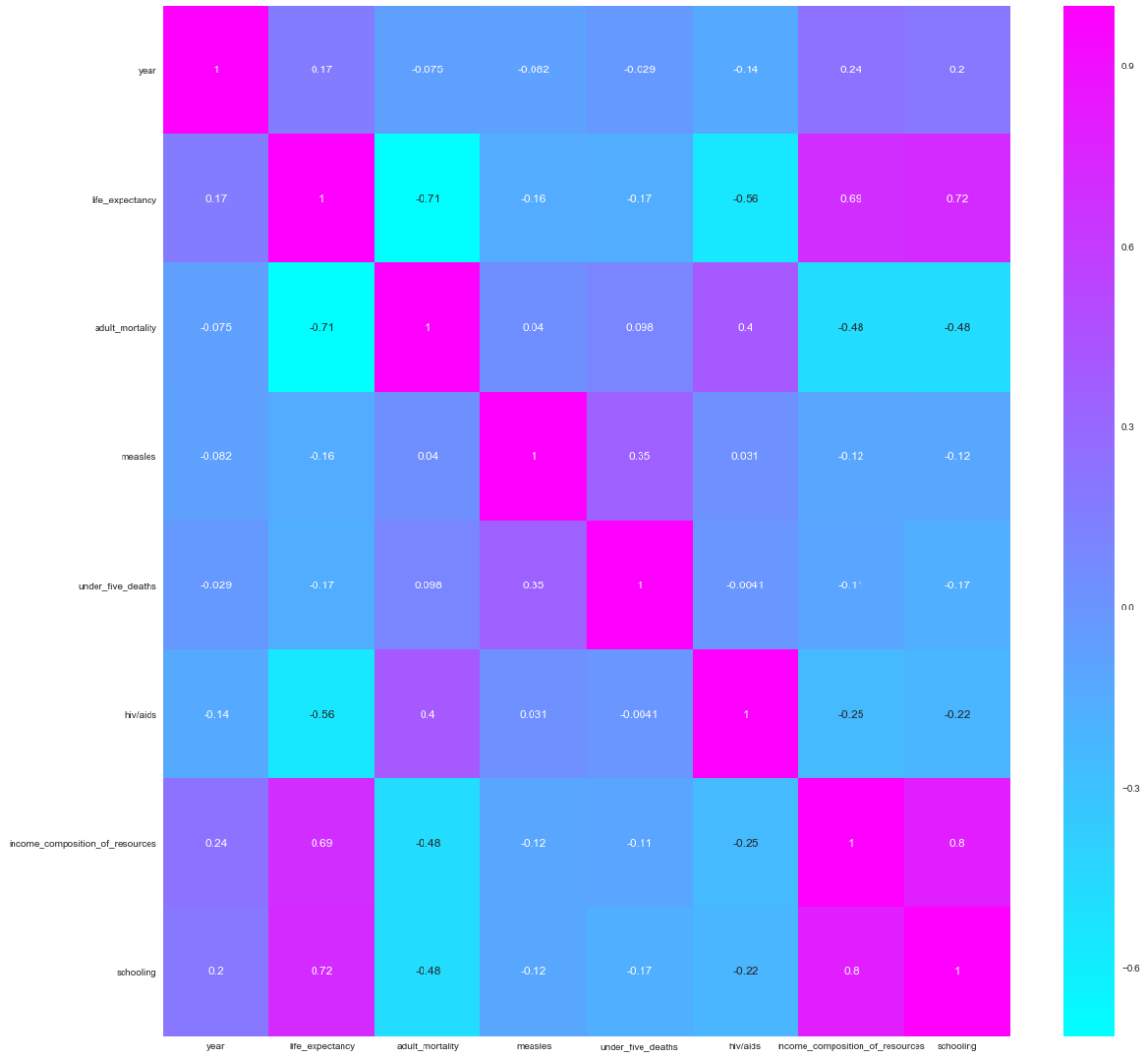
Out[34]:

	country	year	status	life_expectancy	adult_mortality	measles	under_five_deaths	h
0	Afghanistan	2015	Developing	65.0	263.0	1154	83.0	
1	Afghanistan	2014	Developing	59.9	271.0	492	86.0	
2	Afghanistan	2013	Developing	59.9	268.0	430	89.0	
3	Afghanistan	2012	Developing	59.5	272.0	2787	93.0	
4	Afghanistan	2011	Developing	59.2	275.0	3013	97.0	

Checking Correlation among the input features

In [35]:

```
f, ax = plt.subplots(1, 1, figsize=(20, 20))
ax = sns.heatmap(df_a.corr(), annot=True, cmap='cool')
```



Since Infant deaths have been removed as a feature, lets remove under_5_death as well since most of it represents the infant deaths only.

In [36]:

```
#dropping under_five_deaths
df_a.drop(['under_five_deaths'],axis=1)
```

Out[36]:

	country	year	status	life_expectancy	adult_mortality	measles	hiv/aids	income_
0	Afghanistan	2015	Developing	65.0	263.000000	1154	0.1	
1	Afghanistan	2014	Developing	59.9	271.000000	492	0.1	
2	Afghanistan	2013	Developing	59.9	268.000000	430	0.1	
3	Afghanistan	2012	Developing	59.5	272.000000	2787	0.1	
4	Afghanistan	2011	Developing	59.2	275.000000	3013	0.1	
5	Afghanistan	2010	Developing	58.8	279.000000	1989	0.1	
6	Afghanistan	2009	Developing	58.6	281.000000	2861	0.1	
7	Afghanistan	2008	Developing	58.1	287.000000	1599	0.1	
8	Afghanistan	2007	Developing	57.5	295.000000	1141	0.1	
9	Afghanistan	2006	Developing	57.3	295.000000	1990	0.1	
10	Afghanistan	2005	Developing	57.3	291.000000	1296	0.1	
11	Afghanistan	2004	Developing	57.0	293.000000	466	0.1	
12	Afghanistan	2003	Developing	56.7	295.000000	798	0.1	
13	Afghanistan	2002	Developing	56.2	186.581377	2486	0.1	
14	Afghanistan	2001	Developing	55.3	316.000000	8762	0.1	
15	Afghanistan	2000	Developing	54.8	321.000000	6532	0.1	
16	Albania	2015	Developing	77.8	74.000000	0	0.1	
17	Albania	2014	Developing	77.5	186.581377	0	0.1	
18	Albania	2013	Developing	77.2	84.000000	0	0.1	
19	Albania	2012	Developing	76.9	86.000000	9	0.1	
20	Albania	2011	Developing	76.6	88.000000	28	0.1	
21	Albania	2010	Developing	76.2	91.000000	10	0.1	
22	Albania	2009	Developing	76.1	91.000000	0	0.1	
23	Albania	2008	Developing	75.3	186.581377	0	0.1	
24	Albania	2007	Developing	75.9	186.581377	22	0.1	
25	Albania	2006	Developing	74.2	99.000000	68	0.1	
26	Albania	2005	Developing	73.5	186.581377	6	0.1	
27	Albania	2004	Developing	73.0	186.581377	7	0.1	
28	Albania	2003	Developing	72.8	186.581377	8	0.1	
29	Albania	2002	Developing	73.3	186.581377	16	0.1	
...	...	...	...	...	...	...	...	
2908	Zambia	2013	Developing	63.0	328.000000	35	4.8	
2909	Zambia	2012	Developing	59.2	349.000000	896	5.6	
2910	Zambia	2011	Developing	58.2	366.000000	13234	6.3	

	country	year	status	life_expectancy	adult_mortality	measles	hiv/aids	income_
2911	Zambia	2010	Developing	58.0	363.000000	15754	6.8	
2912	Zambia	2009	Developing	57.4	368.000000	26	9.1	
2913	Zambia	2008	Developing	55.7	45.000000	140	11.9	
2914	Zambia	2007	Developing	52.6	487.000000	535	13.6	
2915	Zambia	2006	Developing	58.0	526.000000	459	15.9	
2916	Zambia	2005	Developing	49.3	554.000000	45	17.0	
2917	Zambia	2004	Developing	47.9	578.000000	35	17.6	
2918	Zambia	2003	Developing	46.4	64.000000	881	18.2	
2919	Zambia	2002	Developing	45.5	69.000000	25036	18.4	
2920	Zambia	2001	Developing	44.6	186.581377	16997	18.6	
2921	Zambia	2000	Developing	43.8	186.581377	30930	18.7	
2922	Zimbabwe	2015	Developing	67.0	336.000000	0	6.2	
2923	Zimbabwe	2014	Developing	59.2	371.000000	0	6.3	
2924	Zimbabwe	2013	Developing	58.0	399.000000	0	6.8	
2925	Zimbabwe	2012	Developing	56.6	429.000000	0	8.8	
2926	Zimbabwe	2011	Developing	54.9	464.000000	0	13.3	
2927	Zimbabwe	2010	Developing	52.4	527.000000	9696	15.7	
2928	Zimbabwe	2009	Developing	50.0	587.000000	853	18.1	
2929	Zimbabwe	2008	Developing	48.2	186.581377	0	20.5	
2930	Zimbabwe	2007	Developing	46.6	67.000000	242	23.7	
2931	Zimbabwe	2006	Developing	45.4	186.581377	212	26.8	
2932	Zimbabwe	2005	Developing	44.6	186.581377	420	30.3	
2933	Zimbabwe	2004	Developing	44.3	186.581377	31	33.6	
2934	Zimbabwe	2003	Developing	44.5	186.581377	998	36.7	
2935	Zimbabwe	2002	Developing	44.8	73.000000	304	39.8	
2936	Zimbabwe	2001	Developing	45.3	186.581377	529	42.1	
2937	Zimbabwe	2000	Developing	46.0	186.581377	1483	43.5	

2938 rows × 9 columns

MODELLING

Splitting Exploratory and Response Variables

In [37]:

```
y=df_a['life_expectancy']
X=df_a.drop('life_expectancy',axis=1)
```

In [38]:

```
y.head()
```

Out[38]:

```
0    65.0
1    59.9
2    59.9
3    59.5
4    59.2
Name: life_expectancy, dtype: float64
```

In [39]:

```
X.head()
```

Out[39]:

	country	year	status	adult_mortality	measles	under_five_deaths	hiv/aids	income_
0	Afghanistan	2015	Developing	263.0	1154	83.0	0.1	
1	Afghanistan	2014	Developing	271.0	492	86.0	0.1	
2	Afghanistan	2013	Developing	268.0	430	89.0	0.1	
3	Afghanistan	2012	Developing	272.0	2787	93.0	0.1	
4	Afghanistan	2011	Developing	275.0	3013	97.0	0.1	

Encoding Categorical Variables

In [40]:

```
c_d=pd.get_dummies(X['country'])
s_d=pd.get_dummies(X['status'])
X.drop(['country','status'],inplace=True,axis=1)
X=pd.concat([X,c_d,s_d],axis=1)
```

In [41]:

```
X.head(20)
```

Out[41]:

oling	Afghanistan	Albania	Algeria	...	Uruguay	Uzbekistan	Vanuatu	Venezuela(BolivarianRepu
10.1	1	0	0	...	0	0	0	
10.0	1	0	0	...	0	0	0	
9.9	1	0	0	...	0	0	0	
9.8	1	0	0	...	0	0	0	
9.5	1	0	0	...	0	0	0	
9.2	1	0	0	...	0	0	0	
8.9	1	0	0	...	0	0	0	
8.7	1	0	0	...	0	0	0	
8.4	1	0	0	...	0	0	0	
8.1	1	0	0	...	0	0	0	
7.9	1	0	0	...	0	0	0	
6.8	1	0	0	...	0	0	0	
6.5	1	0	0	...	0	0	0	
6.2	1	0	0	...	0	0	0	
5.9	1	0	0	...	0	0	0	
5.5	1	0	0	...	0	0	0	
14.2	0	1	0	...	0	0	0	
14.2	0	1	0	...	0	0	0	
14.2	0	1	0	...	0	0	0	
14.2	0	1	0	...	0	0	0	

In [42]:

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)

from sklearn.linear_model import LinearRegression
Linear_model= LinearRegression()

Linear_model.fit(X_train,y_train)
y_prediction = Linear_model.predict(X_test)
mse=mean_squared_error(y_test,y_prediction)
print(mse)
rmse=np.sqrt(mse)
print(rmse)
r2=r2_score(y_test,y_prediction)
print(r2)
#print ('Coefficients: ', Linear_model.coef_)
```

```
3.6880825157246386
1.9204381051532586
0.9609023273898715
```

ACCURACY

In [43]:

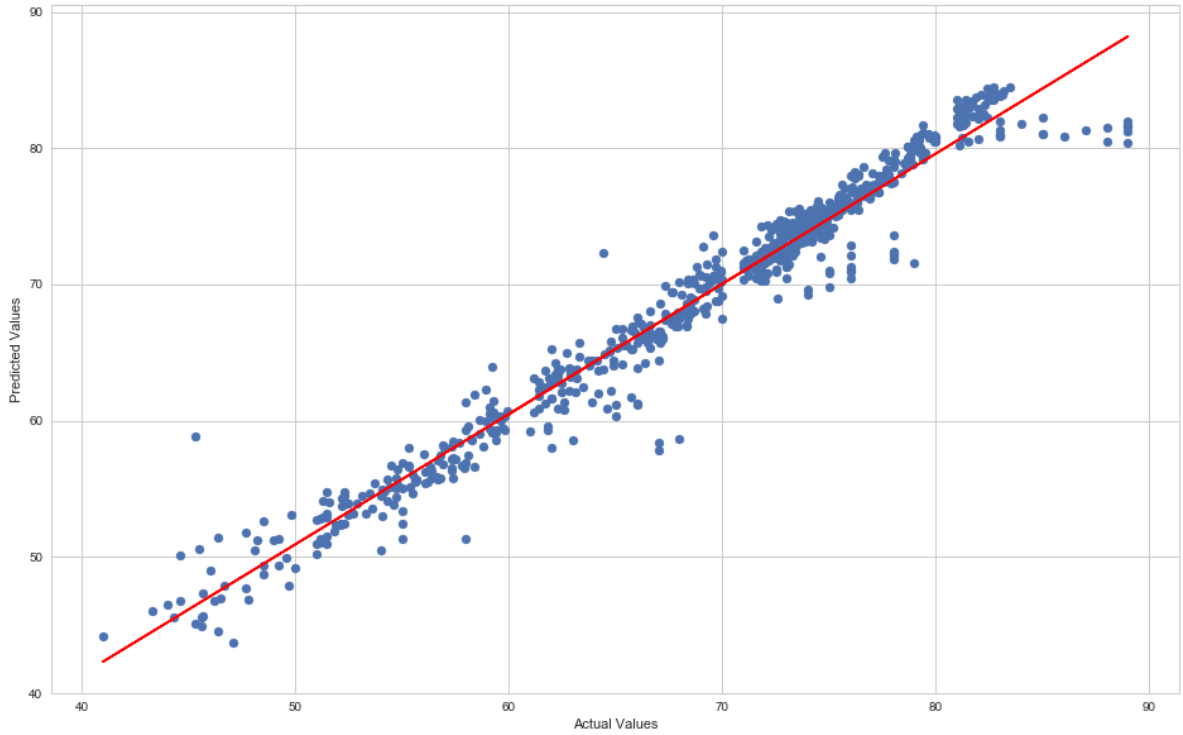
```
# variance score: 1 is perfect prediction
Linear_model.score(X_train,y_train)
```

Out[43]:

```
0.9618325855288178
```

In [44]:

```
plt.scatter(y_test, y_prediction)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
z = np.polyfit(y_test, y_prediction, 1)
p = np.poly1d(z)
plt.plot(y_test,p(y_test), color='red')
plt.show()
```



Conclusion

From the graphs above, we can see geographic correlations regarding the differences in Life Expectancy, Developing or Developed nations, Income Composition of Resources, and Highest Average Age of Schooling. It further supports our earlier findings that Developed nations on average have a higher life expectancy than those in Developing nations, and it shows that Income Composition of Resources and Highest Average Age of Schooling are also correlated to Life Expectancy.

Keeping the all mentioned factors which shows the good correlations were been taken into consideration and thus final output was given out to be 96%.