TCC Expectativa de Vida

Sumary

Contexto

Embora tenha havido muitos estudos realizados no passado sobre fatores que afetam a expectativa de vida, considerando variáveis demográficas, composição de renda e taxas de mortalidade, verificou-se que o efeito da imunização e do índice de desenvolvimento humano não foi levado em consideração no passado. Além disso, algumas das pesquisas anteriores foram feitas considerando a regressão linear múltipla com base no conjunto de dados de um ano para todos os países. Portanto, isso dá motivação para resolver ambos os fatores declarados anteriormente, formulando um modelo de regressão baseado no modelo de efeitos mistos e na regressão linear múltipla, considerando os dados de um período de 2000 a 2015 para todos os países. Imunizações importantes como hepatite B, poliomielite e difteria também serão consideradas. Em suma, este estudo se concentrará em fatores de imunização, fatores de mortalidade, fatores econômicos, fatores sociais e outros fatores relacionados à saúde também. Como as observações desse conjunto de dados são baseadas em diferentes países, será mais fácil para um país determinar o fator de previsão que está contribuindo para diminuir o valor da expectativa de vida. Isso ajudará a sugerir a um país qual área deve receber importância para melhorar de forma eficiente a expectativa de vida de sua população.

Conteùdo

O projeto depende da precisão dos dados. O repositório de dados do Global Health Observatory (GHO) sob a Organização Mundial da Saúde (OMS) acompanha o estado de saúde, bem como muitos outros fatores relacionados para todos os países. Os conjuntos de dados são disponibilizados ao público para fins de análise de dados de saúde. O conjunto de dados relacionados à expectativa de vida e fatores de saúde para 193 países foi coletado do mesmo site do repositório de dados da OMS e seus dados econômicos correspondentes foram coletados do site das Nações Unidas. Entre todas as categorias de fatores relacionados à saúde, foram escolhidos apenas os fatores críticos que são mais representativos. Observou-se que, nos últimos 15 anos, houve um grande desenvolvimento no setor da saúde, resultando na melhoria das taxas de mortalidade humana, especialmente nas nações em desenvolvimento, em comparação com os últimos 30 anos. Portanto, neste projeto, consideramos os dados do ano 2000-2015 para 193 países para uma análise posterior. Os arquivos de dados individuais foram mesclados em um único conjunto de dados. Na inspeção visual inicial, os dados mostraram alguns valores ausentes. Como os conjuntos de dados eram da OMS, não encontramos erros evidentes. Os dados ausentes foram tratados no software R usando o comando Missmap. O resultado indicou que a maioria dos dados ausentes era para população, hepatite B e PIB. Os dados em falta eram de países menos conhecidos como Vanuatu, Tonga, Togo, Cabo Verde, etc. Encontrar todos os dados para estes países foi difícil e, portanto, decidiu-se excluir estes países do conjunto de dados do modelo final. O arquivo final mesclado (conjunto de dados final) consiste em 22 colunas e 2938 linhas, o que significa 20 variáveis de previsão. Todas as variáveis de previsão foram então divididas em várias categorias amplas:

Fatores relacionados à imunização, Fatores de mortalidade, Fatores econômicos e Fatores sociais.

Fonte dos dados

1 - Datos Expectativa de vida

Os dados foram coletados do site da OMS e das Nações Unidas com a ajuda de Deeksha Russell e Duan Wang. O repositório de dados do Observatório Global de Saúde (GHO) da Organização Mundial da Saúde (OMS) acompanha o estado de saúde, bem como muitos outros fatores relacionados para todos os países. Os conjuntos de dados são disponibilizados ao público para fins de análise de dados de saúde. O conjunto de dados relacionado à expectativa de vida e fatores de saúde para 193 países foi coletado do mesmo site do repositório de dados da OMS e seus dados econômicos correspondentes foram coletados do site das Nações Unidas. Entre todas as categorias de fatores relacionados à saúde, foram escolhidos apenas os fatores críticos que são mais representativos. Observou-se que, nos últimos 15 anos, houve um grande desenvolvimento no setor da saúde, resultando na melhoria das taxas de mortalidade humana, especialmente nas nações em desenvolvimento, em comparação com os últimos 30 anos. Portanto, neste projeto, consideramos os dados do ano 2000-2015 para 193 países para uma análise posterior. Os arquivos de dados individuais foram mesclados em um único conjunto de dados. Na inspeção visual inicial, os dados mostraram alguns valores ausentes. Como os conjuntos de dados eram da OMS, não encontramos erros evidentes. Os dados ausentes foram tratados no software R usando o comando Missmap. O resultado indicou que a maioria dos dados ausentes era para população, hepatite B e PIB. Os dados em falta eram de países menos conhecidos como Vanuatu, Tonga, Togo, Cabo Verde, etc. Encontrar todos os dados para estes países foi difícil e, portanto, decidiu-se excluir estes países do conjunto de dados do modelo final. O arquivo final mesclado (conjunto de dados final) consiste em 22 colunas e 2938 linhas, o que significa 20 variáveis de previsão. Todas as variáveis de previsão foram então divididas em várias categorias amplas: Fatores relacionados à imunização, Fatores de mortalidade, Fatores econômicos e Fatores sociais.

3 - Data CO_2 and Greenhouse Gas Emissions by Our World in Data

Data on CO_2 and Greenhouse Gas Emissions by Our World in Data Our complete CO_2 and Greenhouse Gas Emissions dataset is a collection of key metrics maintained by Our World in Data. It is updated regularly and includes data on CO_2 emissions (annual, per capita, cumulative and consumption-based), other greenhouse gases, energy mix, and other relevant metrics.

card_index_dividers Download our complete CO_2 and Greenhouse Gas Emissions dataset : CSV | XLSX | JSON We will continue to publish updated data on CO_2 and Greenhouse Gas Emissions as it becomes available. Most metrics are published on an annual basis.

Our data sources CO_2 emissions: this data is sourced from the Global Carbon Project. The Global Carbon Project typically releases a new update of CO_2 emissions annually. Greenhouse gas emissions (including methane, and nitrous oxide): this data is sourced from the CAIT Climate Data Explorer, and downloaded from the Climate Watch Portal. Energy (primary energy, energy mix and energy intensity): this data is sourced from a combination of two sources. The BP Statistical Review of World Energy is published annually, but it does not provide data on primary energy consumption for all countries. For countries absent from this dataset, we calculate primary energy by multiplying the World Bank, World Development Indicators metric Energy use per capita by total population figures. The World Bank sources this metric from the IEA. Other variables: this data is collected from a variety of sources (United Nations, World Bank, Gapminder, Maddison Project Database, etc.). More information is available in our codebook. The complete Our World in Data CO2 and Greenhouse Gas Emissions dataset Our complete CO2 and Greenhouse Gas Emissions dataset is available in CSV, XLSX, and JSON formats.

The CSV and XLSX files follow a format of 1 row per location and year. The JSON version is split by country, with an array of yearly records.

The variables represent all of our main data related to CO2 emissions, other greenhouse gas emissions, energy mix, as well as other variables of potential interest.

A full codebook is made available, with a description and source for each variable in the dataset.

Changelog On August 7, 2020, the first version of this dataset was made available. On February 8, 2021 we updated this dataset with the latest annual release from the Global Carbon Project. Data alterations We standardize names of countries and regions. Since the names of countries and regions are different in different data sources, we standardize all names to the Our World in Data standard entity names. We recalculate carbon emissions to CO2. The primary data sources on CO2 emissions—the Global Carbon Project, for example—typically report emissions in tonnes of carbon. We have recalculated these figures as tonnes of CO2 using a conversion factor of 3.664. We calculate per capita figures. All of our per capita figures are calculated from our metric Population, which is included in the complete dataset. These population figures are sourced from Gapminder and the UN World Population Prospects (UNWPP). License All visualizations, data, and code produced by Our World in Data are completely open access under the Creative Commons BY license. You have the permission to use, distribute, and reproduce these in any medium, provided the source and authors are credited.

The data produced by third parties and made available by Our World in Data is subject to the license terms from the original third-party authors. We will always indicate the original source of the data in our database, and you should always check the license of any such third-party data before use.

Authors This data has been collected, aggregated, and documented by Hannah Ritchie, Max Roser and Edouard Mathieu.

The mission of Our World in Data is to make data and research on the world's largest problems understandable and accessible. Read more about our mission.

About Data on CO2 and greenhouse gas emissions by Our World in Data

ourworldindata.org/co2-and-other-greenhouse-gas-emissions Topics environment energy greenhouse-gas-emissions co2-emissions Resources Readme Sponsor this project https://ourworldindata.org/donate Contributors 4 @edomt edomt Edouard Mathieu @HannahRitchie HannahRitchie Hannah Ritchie @bnjmacdonald bnjmacdonald Bobbie Macdonald @krueschan krueschan Christian Holz Languages Python 100.0%

- https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions

3 - Data Country names I use the ISO 3166-1 alpha-2 standard to encode the country names.

4 - Data geolocation API

5 - https://alvarezsolucoesdigitais.com/web-scraping/coletando-dados-de-tabelas-da-wikipedia-usando-beautifulsoup-e-python/

Hipoteses

O conjunto de dados visa responder às seguintes questões-chave:

- Os vários fatores de previsão escolhidos inicialmente realmente afetam a expectativa de vida?
- Quais são as variáveis de previsão que realmente afetam a expectativa de vida?
- Um país com expectativa de vida menor (<65) deve aumentar seus gastos com saúde para melhorar sua expectativa de vida média?
- Como as taxas de mortalidade de bebês e adultos afetam a expectativa de vida?
- A expectativa de vida tem correlação positiva ou negativa com hábitos alimentares, estilo de vida, exercícios, fumo, bebida alcoólica etc.
- Qual é o impacto da escolaridade na expectativa de vida dos humanos?
- A expectativa de vida tem uma relação positiva ou negativa com o consumo de álcool?
- Países densamente povoados tendem a ter menor expectativa de vida?
- Qual é o impacto da cobertura de imunização na expectativa de vida?

Import libraries

In [1]:
```
#!pip install inflection
#!pip install pandas
#!pip install sqlalchemy
#!pip install geopy
#!pip install pycountry-convert
#!pip install seaborn
#!pip install plotly
#!pip install pycountry-convert
#!pip install melt
#!pip install sklearn
```

In [426…
```
import pandas as pd
import numpy as np
import inflection
import sqlite3
from sqlalchemy import create_engine
from geopy.geocoders import Nominatim
import seaborn as sns
import folium
from folium.plugins import MarkerCluster
from matplotlib import pyplot as plt
from matplotlib import gridspec
import ipywidgets as widgets
from ipywidgets import fixed
import plotly.express as px
from IPython.core.display import HTML
import plotly.express as px
from pycountry_convert import country_alpha2_to_continent_code,country_name_t
```

```
from datetime import datetime
from sklearn import linear_model as lm
from sklearn import model_selection as ms
from sklearn import metrics as m

#import time #multi-processing
#from multiprocessing import Pool #multi-processing
#import defs #function create by me
```

In [427…

```
def jupyter_settings(): #normalization graphs
    %matplotlib inline
    %pylab inline

    plt.style.use( 'bmh' )
    plt.rcParams['figure.figsize'] = [15, 8]
    plt.rcParams['font.size'] = 20

    display( HTML( '<style>.container { width:100% !important; }</style>') )
    pd.options.display.max_columns = None
    pd.options.display.max_rows = None
    pd.set_option( 'display.expand_frame_repr', False )

    sns.set()
jupyter_settings()
```

```
Populating the interactive namespace from numpy and matplotlib
/home/alessandra/.pyenv/versions/3.8.0/envs/tcc/lib/python3.8/site-packages/I
Python/core/magics/pylab.py:159: UserWarning:

pylab import has clobbered these variables: ['datetime', 'style']
`%matplotlib` prevents importing * from pylab and numpy
```

Building Dataset

In [5]:

```
df_expec=pd.read_csv('Datasets/Life_Expectancy_Data.csv',parse_dates=[1])
df_emi=pd.read_csv('Datasets/emission data.csv')
```

In [6]:

```
df_expec.head()
```

Out[6]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B
0	Afghanistan	2015-01-01	Developing	65.0	263.0	62	0.01	71.279624	65.0
1	Afghanistan	2014-01-01	Developing	59.9	271.0	64	0.01	73.523582	62.0
2	Afghanistan	2013-01-01	Developing	59.9	268.0	66	0.01	73.219243	64.0
3	Afghanistan	2012-01-01	Developing	59.5	272.0	69	0.01	78.184215	67.0
4	Afghanistan	2011-01-01	Developing	59.2	275.0	71	0.01	7.097109	68.0

5 rows × 22 columns

In [5]:
```
df_emi.head()
```

Out[5]:

	Country	1751	1752	1753	1754	1755	1756	1757	1758	1759	...	2008	
0	Afghanistan	0	0	0	0	0	0	0	0	0	...	8.515264e+07	9.19129!
1	Africa	0	0	0	0	0	0	0	0	0	...	3.183077e+10	3.30190.
2	Albania	0	0	0	0	0	0	0	0	0	...	2.287948e+08	2.33169
3	Algeria	0	0	0	0	0	0	0	0	0	...	2.894820e+09	3.01500!
4	Americas (other)	0	0	0	0	0	0	0	0	0	...	7.746025e+10	7.96178

5 rows × 268 columns

In [6]:
```
# Supress Scientific Notation
np.set_printoptions(suppress=True)
pd.set_option('display.float_format', '{:.2f}'.format)
```

In [7]:
```
# Building DataFrame
df_emi=df_emi[['Country','2000','2001','2002','2003','2004','2005','2006','20
               '2009','2010', '2011', '2012', '2013', '2014', '2015']]

df_emi=df_emi.melt(id_vars=['Country'])  # empilha os dados com granularidade
df_emi.columns=['Country','Year','Emission']
```

In [8]:
```
df_emi.head()
```

Out[8]:

	Country	Year	Emission
0	Afghanistan	2000	71717793.00
1	Africa	2000	23640083267.00
2	Albania	2000	196932672.00
3	Algeria	2000	2118624684.00
4	Americas (other)	2000	60974588046.00

In [9]:
```
# Trasforming variable date

df_emi['Year'] = pd.to_datetime( df_emi['Year'] ).dt.strftime('%Y')
df_expec['Year'] = pd.to_datetime( df_expec['Year'] ).dt.strftime('%Y')
```

In [10]:
```
# Merge Dataframes

df=pd.merge(df_expec,df_emi,how='left',on=['Country','Year'])
```

In [11]:
```
df.head()
```

Out[11]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	M
0	Afghanistan	2015	Developing	65.00	263.00	62	0.01	71.28	65.00	
1	Afghanistan	2014	Developing	59.90	271.00	64	0.01	73.52	62.00	
2	Afghanistan	2013	Developing	59.90	268.00	66	0.01	73.22	64.00	
3	Afghanistan	2012	Developing	59.50	272.00	69	0.01	78.18	67.00	
4	Afghanistan	2011	Developing	59.20	275.00	71	0.01	7.10	68.00	

5 rows × 23 columns

Changing columns name

In [12]:

```
df.columns
```

Out[12]:

```
Index(['Country', 'Year', 'Status', 'Life expectancy ', 'Adult Mortality',
       'infant deaths', 'Alcohol', 'percentage expenditure', 'Hepatitis B',
       'Measles ', ' BMI ', 'under-five deaths ', 'Polio', 'Total expenditur
e',
       'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population',
       ' thinness  1-19 years', ' thinness 5-9 years',
       'Income composition of resources', 'Schooling', 'Emission'],
      dtype='object')
```

In [13]:

```
cols_original=['Country', 'Year', 'Status', 'Life expectancy ', 'Adult Mortal
       'infant deaths', 'Alcohol', 'percentage expenditure', 'Hepatitis B',
       'Measles ', ' BMI ', 'under-five deaths ', 'Polio', 'Total expenditure
       'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population',
       ' thinness  1-19 years', ' thinness 5-9 years',
       'Income composition of resources', 'Schooling', 'Emission']

snakecase = lambda x: inflection.underscore(x)



cols = list(map(snakecase,cols_original))
snakecase = lambda x: inflection.parameterize(x)
cols = list(map(snakecase,cols_original))


df.columns=cols
```

Populating Country code and continent code from function

In [14]:

```
#function to convert to alpah2 country codes


def get_code(col):
    try:
        cn_a2_code = country_name_to_country_alpha2(col)
    except:
        cn_a2_code = 'Unknown'
    try:
        cn_continent = country_alpha2_to_continent_code(cn_a2_code)
    except:
```

```
        cn_continent = 'Unknown'
    return (cn_a2_code)

#function to convert to alpah2 country continent

def get_continent(col):
    try:
        cn_a2_code = country_name_to_country_alpha2(col)
    except:
        cn_a2_code = 'Unknown'
    try:
        cn_continent = country_alpha2_to_continent_code(cn_a2_code)
    except:
        cn_continent = 'Unknown'
    return (cn_continent)
```

In [15]:

```
df['continent'],df['code']=df['country'].apply(lambda x: get_continent(x)),df[
```

DF2 API geolocation (cycle 2 usar Parallel process theory)

In [16]:

```
## Return latitude
def geolocate_lat(country):
    geolocator = Nominatim(user_agent='geoapiExercises')

    try:
        # Geolocate the center of the country
        loc = geolocator.geocode(country)
        # And return latitude
        return loc.latitude

    except:
        # Return missing value
        return np.nan

## Return longitude
def geolocate_long(country):
    geolocator = Nominatim(user_agent='geoapiExercises')

    try:
        # Geolocate the center of the country
        loc = geolocator.geocode(country)
        # And return longitude
        return loc.longitude

    except:
        # Return missing value
        return np.nan
```

In [17]:

```
#df['lat'],df['long']=df['country'].apply(lambda x: geolocate_lat(x)),df['cou
```

In [18]:

```
# I run one time tne upper code and write in the new df csv named "Life_Expec
# df1=df.copy()
# df1.to_csv('Datasets/Life_Expectancy_Data_geolocation.csv',index=False)
```

Data Description

In [356…
```
## EDA
```

In [357…
```
# ### Explicaçao das colunas

# Country Country

# Year Year

# Status Status in development or under development

# Life expectancy Life expectancy at age

# Adult Mortality Adult mortality rates for both sexes (probability of dying

# infant deaths Infant deaths per 1000 population

# Alcohol Accounting for alcohol consumption per capita (15+) (in liters of p

# percentage expenditure Health care expenditure as a percentage of gross dom

# Hepatitis B Immunization coverage against hepatitis B (HepB) among one year

# Measles Measles - the number of reported cases per 1000 population

# BMI Average body mass index of the entire population

# under-five deaths Deaths of children under five years of age per 1000 popul

# Polio Polio immunization coverage (Pol3) among one-year-old children (%)

# Total expenditure Total government spending on health as a percentage of to

# Diphtheria Immunization coverage against diphtheria and pertussis tetanus (

# HIV / AIDS Mortality per 1,000 live births HIV / AIDS (0-4 years)

# GDP Gross Domestic Product per capita (in US dollars)

# Population Population of the country

# thinness 1-19 years Prevalence of thinness among children and adolescents a

# thinness 5-9 years Prevalence of thinness among children aged 5 to 9 (%)

# Income composition of resources Human Development Index in terms of income

# Schooling Number of years of study (years)
```

In [358…
```
# - Country=Paises analisados
# -Year=anos
# -Status
# -Life expectancy ',
# -Adult Mortality',
# -infant deaths',
# -Alcohol',
# -percentage expenditure',
# -Hepatitis B',
# -Measles ',
# -BMI ',
# -under-five deaths ',
# -Polio',
# -Total expenditure',
```

```
# -Diphtheria ',
# -HIV/AIDS',
# -GDP',
# -Population',
# -thinness  1-19 years',
# -thinness 5-9 years',
# -Income composition of resources',
# -Schooling']'
```

In [355…

```
df1=pd.read_csv('Datasets/Life_Expectancy_Data_geolocation.csv')
df1.head(10)
```

Out[355…

	country	year	status	life-expectancy	adult-mortality	infant-deaths	alcohol	percentage-expenditure	hepatitis-b	r
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	Afghanistan	2010	Developing	58.8	279.0	74	0.01	79.679367	66.0	
6	Afghanistan	2009	Developing	58.6	281.0	77	0.01	56.762217	63.0	
7	Afghanistan	2008	Developing	58.1	287.0	80	0.03	25.873925	64.0	
8	Afghanistan	2007	Developing	57.5	295.0	82	0.02	10.910156	63.0	
9	Afghanistan	2006	Developing	57.3	295.0	84	0.03	17.171518	64.0	

Data dimensions

In [359…

```
print('numero de linhas:{}'.format(df1.shape[0]))
print('numero de colunas:{}'.format(df1.shape[1]))
```

```
numero de linhas:2938
numero de colunas:27
```

Data Types

In [360…

```
df1.dtypes
```

Out[360…

```
country                              object
year                                  int64
status                               object
life-expectancy                     float64
adult-mortality                     float64
infant-deaths                         int64
alcohol                             float64
percentage-expenditure              float64
hepatitis-b                         float64
measles                               int64
bmi                                 float64
under-five-deaths                     int64
polio                               float64
```

```
total-expenditure                  float64
diphtheria                         float64
hiv-aids                           float64
gdp                                float64
population                         float64
thinness-1-19-years                float64
thinness-5-9-years                 float64
income-composition-of-resources    float64
schooling                          float64
emission                           float64
continent                           object
code                                object
lat                                float64
long                               float64
dtype: object
```

Change data types

In [361…

```
df1['year']=pd.to_datetime( df1['year'], format='%Y')
```

Controle dos NA's

In [362…

```
df1.isna().sum()
```

Out[362…

```
country                              0
year                                 0
status                               0
life-expectancy                     10
adult-mortality                     10
infant-deaths                        0
alcohol                            194
percentage-expenditure               0
hepatitis-b                        553
measles                              0
bmi                                 34
under-five-deaths                    0
polio                               19
total-expenditure                  226
diphtheria                          19
hiv-aids                             0
gdp                                448
population                         652
thinness-1-19-years                 34
thinness-5-9-years                  34
income-composition-of-resources    167
schooling                          163
emission                           338
continent                          338
code                                16
lat                                 16
long                                16
dtype: int64
```

Eliminar Na's

In [363…

```
df_raw= df1.dropna(subset=['life-expectancy','alcohol','hepatitis-b','bmi','t
```

In [364…

```
df1.loc[df1['continent'].isna(), 'country'].unique()
```

Out[364…
```
array(['Antigua and Barbuda', 'Bahamas', 'Barbados', 'Belize', 'Canada',
       'Costa Rica', 'Cuba', 'Dominica', 'Dominican Republic',
       'El Salvador', 'Grenada', 'Guatemala', 'Haiti', 'Honduras',
       'Jamaica', 'Mexico', 'Nicaragua', 'Panama',
       'Saint Kitts and Nevis', 'Saint Lucia',
       'Saint Vincent and the Grenadines', 'Trinidad and Tobago',
       'United States of America'], dtype=object)
```

In [365…
```
df_raw['continent'].fillna('NA', inplace=True )
```

In [366…
```
df_raw.sample(50)
```

Out[366…

	index	country	year	status	life-expectancy	adult-mortality	infant-deaths	alcohol	percentage-expenditure
720	1295	Italy	2002-01-01	Developed	80.0	72.0	2	9.25	2883.334911
1045	1781	Myanmar	2014-01-01	Developing	66.4	21.0	40	0.01	45.337887
34	35	Algeria	2012-01-01	Developing	75.1	113.0	21	0.66	555.926083
39	40	Algeria	2007-01-01	Developing	73.8	129.0	20	0.44	320.323924
1198	2049	Poland	2005-01-01	Developed	75.0	144.0	2	9.50	79.415027
1510	2710	Turkmenistan	2002-01-01	Developing	63.3	229.0	7	2.33	130.378483
43	49	Angola	2014-01-01	Developing	51.7	348.0	67	8.33	23.965612
13	13	Afghanistan	2002-01-01	Developing	56.2	3.0	88	0.01	16.887351
740	1336	Jordan	2009-01-01	Developing	73.3	118.0	4	0.59	668.744733
250	388	Bulgaria	2011-01-01	Developed	73.7	144.0	1	10.67	875.149519
268	407	Burkina Faso	2008-01-01	Developing	56.1	288.0	45	4.50	107.798834
20	20	Albania	2011-01-01	Developing	76.6	88.0	0	5.37	437.062100
619	1096	Guinea-Bissau	2009-01-01	Developing	56.3	288.0	4	2.55	47.129693
724	1300	Jamaica	2013-01-01	Developing	75.6	136.0	1	3.79	5.457289
1212	2064	Portugal	2006-01-01	Developed	78.5	96.0	0	13.11	2884.020194
1193	2044	Poland	2010-01-01	Developed	76.3	13.0	2	10.59	220.491685
1396	2500	Swaziland	2004-01-01	Developing	45.6	69.0	3	5.78	37.438577
833	1485	Lesotho	2004-01-01	Developing	44.8	666.0	5	1.80	67.913618
1239	2157	Rwanda	2009-01-01	Developing	61.0	288.0	17	7.11	9.165615

	index	country	year	status	life-expectancy	adult-mortality	infant-deaths	alcohol	percentage-expenditure
89	123	Australia	2004-01-01	Developed	86.0	69.0	1	9.84	588.568371
129	201	Bangladesh	2006-01-01	Developing	68.2	152.0	164	0.01	42.330455
822	1473	Lebanon	2000-01-01	Developing	72.7	112.0	1	2.26	404.387943
414	688	Cyprus	2000-01-01	Developed	78.1	7.0	0	9.56	950.802793
839	1495	Liberia	2010-01-01	Developing	59.7	272.0	9	3.64	41.910524
457	823	El Salvador	2010-01-01	Developing	72.0	191.0	2	2.36	469.390419
1604	2935	Zimbabwe	2002-01-01	Developing	44.8	73.0	25	4.43	0.000000
617	1094	Guinea-Bissau	2011-01-01	Developing	57.1	289.0	4	3.57	40.453674
1355	2432	Spain	2008-01-01	Developed	81.3	7.0	2	10.24	5596.535203
69	102	Armenia	2009-01-01	Developing	73.3	137.0	1	3.96	201.185546
550	991	Georgia	2002-01-01	Developing	71.7	142.0	2	2.72	60.558183
316	516	Central African Republic	2011-01-01	Developing	49.8	443.0	16	1.66	58.529475
95	131	Austria	2012-01-01	Developed	88.0	7.0	0	12.26	7878.372355
346	572	China	2003-01-01	Developing	73.1	13.0	391	2.96	122.936535
1112	1946	Pakistan	2011-01-01	Developing	65.5	167.0	371	0.04	57.877363
876	1559	Madagascar	2010-01-01	Developing	63.3	248.0	32	1.03	76.604422
242	363	Brazil	2004-01-01	Developing	72.0	17.0	81	6.85	186.609049
1129	1968	Panama	2006-01-01	Developing	76.2	125.0	1	5.72	631.125171
492	885	Ethiopia	2012-01-01	Developing	63.3	241.0	150	1.84	86.825511
736	1332	Jordan	2013-01-01	Developing	73.9	114.0	4	0.40	546.623516
295	485	Cameroon	2010-01-01	Developing	55.3	37.0	53	6.15	100.898745
1261	2215	Samoa	2000-01-01	Developing	72.0	18.0	0	3.00	21.254300
975	1681	Mauritius	2001-01-01	Developing	71.5	177.0	0	4.38	70.155370
663	1204	Indonesia	2013-01-01	Developing	68.7	181.0	124	0.09	22.847831

	index	country	year	status	life-expectancy	adult-mortality	infant-deaths	alcohol	percentage-expenditure
60	90	Argentina	2005-01-01	Developing	74.9	127.0	11	7.53	96.166534
1214	2066	Portugal	2004-01-01	Developed	78.0	99.0	0	13.45	276.099980
277	423	Burundi	2008-01-01	Developing	55.3	35.0	23	4.33	15.994152
1531	2738	Ukraine	2007-01-01	Developing	67.5	277.0	5	8.86	46.196854
881	1564	Madagascar	2005-01-01	Developing	69.0	265.0	37	0.72	33.747862
950	1645	Malta	2004-01-01	Developed	78.7	69.0	0	6.53	203.315750
1205	2057	Portugal	2013-01-01	Developed	86.0	79.0	0	10.00	2698.018170

Descriptive Statistics

In [367…

```
num_attributes = df_raw.select_dtypes( include=['int64', 'float64'] )
cat_attributes = df_raw.select_dtypes( exclude=['int64', 'float64', 'datetime
```

Numerical Atributes

In [368…

```
# Central Tendency - mean, meadina
ct1 = pd.DataFrame( num_attributes.apply( np.mean ) ).T
ct2 = pd.DataFrame( num_attributes.apply( np.median ) ).T

# dispersion - std, min, max, range, skew, kurtosis
d1 = pd.DataFrame( num_attributes.apply( np.std ) ).T
d2 = pd.DataFrame( num_attributes.apply( min ) ).T
d3 = pd.DataFrame( num_attributes.apply( max ) ).T
d4 = pd.DataFrame( num_attributes.apply( lambda x: x.max() - x.min() ) ).T
d5 = pd.DataFrame( num_attributes.apply( lambda x: x.skew() ) ).T
d6 = pd.DataFrame( num_attributes.apply( lambda x: x.kurtosis() ) ).T

# concatenar
m = pd.concat( [d2, d3, d4, ct1, ct2, d1, d5, d6] ).T.reset_index()
m.columns = ['attributes', 'min', 'max', 'range', 'mean', 'median', 'std', 's
m
```

Out[368…

	attributes	min	max	range	mean	median	
0	index	0.000000	2.937000e+03	2.937000e+03	1.414283e+03	1.453000e+03	8.40175
1	life-expectancy	44.000000	8.900000e+01	4.500000e+01	6.928183e+01	7.170000e+01	8.89247
2	adult-mortality	1.000000	7.230000e+02	7.220000e+02	1.682912e+02	1.490000e+02	1.26223
3	infant-deaths	0.000000	1.600000e+03	1.600000e+03	3.320597e+01	3.000000e+00	1.22305
4	alcohol	0.010000	1.787000e+01	1.786000e+01	4.520479e+00	3.770000e+00	4.00944
5	percentage-expenditure	0.000000	1.896135e+04	1.896135e+04	7.119260e+02	1.455965e+02	1.77897

	attributes	min	max	range	mean	median	
6	hepatitis-b	2.000000	9.900000e+01	9.700000e+01	7.919726e+01	8.900000e+01	2.55784
7	measles	0.000000	1.314410e+05	1.314410e+05	2.267469e+03	1.400000e+01	1.02087
8	bmi	2.000000	7.710000e+01	7.510000e+01	3.813528e+01	4.360000e+01	1.98182
9	under-five-deaths	0.000000	2.100000e+03	2.100000e+03	4.514561e+01	4.000000e+00	1.64855
10	polio	3.000000	9.900000e+01	9.600000e+01	8.342564e+01	9.300000e+01	2.25872
11	total-expenditure	1.100000	1.439000e+01	1.329000e+01	5.991649e+00	5.880000e+00	2.29834
12	diphtheria	2.000000	9.900000e+01	9.700000e+01	8.418668e+01	9.200000e+01	2.14544
13	hiv-aids	0.100000	5.060000e+01	5.050000e+01	2.028189e+00	1.000000e-01	6.10241
14	gdp	1.681350	1.191727e+05	1.191711e+05	5.628125e+03	1.618493e+03	1.15934
15	population	34.000000	1.293859e+09	1.293859e+09	1.448118e+07	1.419631e+06	7.09668
16	thinness-1-19-years	0.100000	2.720000e+01	2.710000e+01	4.816055e+00	3.000000e+00	4.62383
17	thinness-5-9-years	0.100000	2.820000e+01	2.810000e+01	4.868326e+00	3.100000e+00	4.67885
18	income-composition-of-resources	0.000000	9.360000e-01	9.360000e-01	6.308202e-01	6.750000e-01	1.84935
19	schooling	4.200000	2.070000e+01	1.650000e+01	1.211089e+01	1.230000e+01	2.82262
20	emission	963632.000000	1.710000e+11	1.709990e+11	4.873045e+09	3.316716e+08	1.48466
21	lat	-34.996496	6.106669e+01	9.606319e+01	1.646906e+01	1.525724e+01	2.48887
22	long	-175.202642	1.790123e+02	3.542149e+02	1.708857e+01	2.375000e+01	6.81178

Categorical Atributes

In [369…

```
cat_attributes.apply( lambda x: x.unique().shape[0] )
```

Out[369…

```
country      129
status         2
continent      6
code         129
dtype: int64
```

In [425…

```
    map=df_raw[['continent','country','lat','long','population','life-expecta

    fig=px.scatter_mapbox(map,
                          hover_name='country',
                          hover_data=["life-expectancy", "population"],
                          lat='lat',
                          lon='long',
                          size='life-expectancy',
                          color='continent',
                          color_continuous_scale=px.colors.cyclical.IceFire_r
                          size_max=10,
                          zoom=1)

    fig.update_layout(mapbox_style='open-street-map')
    fig.update_layout(height=500, margin={'r':0,'l':0,'t':0,'b':0})
    fig.show()
```

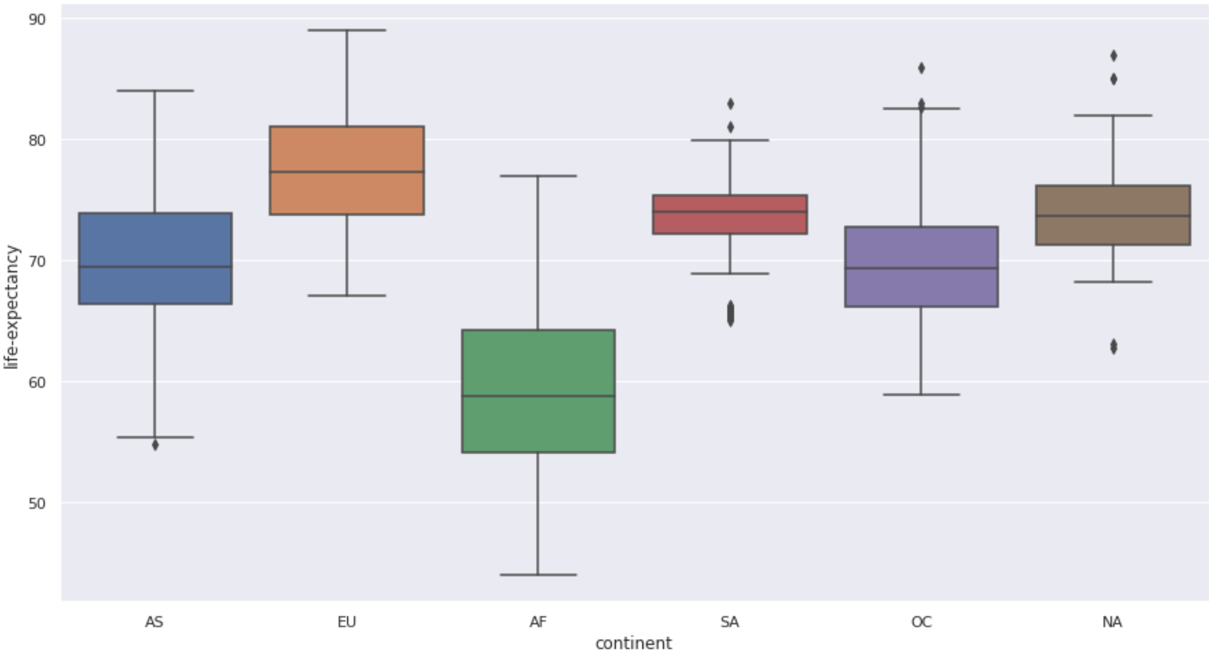
In [370…

```
aux = df_raw[['continent','life-expectancy']]

sns.boxplot( x='continent', y='life-expectancy', data=aux )
```

Out[370…

```
<AxesSubplot:xlabel='continent', ylabel='life-expectancy'>
```



Data Preparation

In [57]:
```
##features
X=df_raw.drop(['life-expectancy','year','country','status','continent','code'

# response variable
y=df_raw['life-expectancy'].copy()
```

In [58]:
```
x_train,x_test,y_train,y_test=ms.train_test_split(X,y,test_size=0.2,random_st
```

Model Training

In [60]:
```
#model description
model_lr=lm.LinearRegression()

# model training
model_lr.fit(x_train,y_train)
```

Out[60]:
```
LinearRegression()
```

In [61]:
```
# prediction_training
pred_train=model_lr.predict(x_train)

# prediction_test
pred_test=model_lr.predict(x_test)
```

Performance Metrics

In [67]:
```
# training - MAE, MAPE
mae_train=m.mean_absolute_error(y_train, pred_train)
mape_train=np.mean(np.abs((y_train - pred_train)/y_train))

# test - MAE, MAPE
mae_test=m.mean_absolute_error(y_test, pred_test)
mape_test=np.mean(np.abs((y_test - pred_test)/y_test))
```

In [71]:
```
data={'Dataframe':['training','test'],
      'MAE':[mae_train,mae_test],
      'MAPE':[mape_train,mape_test]
    }
pd.DataFrame(data)
```

Out[71]:

	Dataframe	MAE	MAPE
0	training	2.72	0.04
1	test	2.77	0.04

In [ ]:

In [ ]:

Map visualization

In [432…

```
# #installation
#!pip install folium
# # Create a world map to show distributions of users
import folium
from folium.plugins import MarkerCluster
#empty map
world_map= folium.Map(location=[33.77,66.24],
                      tiles="cartodbpositron",
                    zoom_start=1)

folium.Marker([33.77,66.24],'<i>Afghanistan</bin/i>').add_to(world_map)
world_map
# for each coordinate, create circlemarker of user percent
# for i in range(len(df_raw)):
#     lat = df_raw.iloc[i]['lat']
#     long = df_raw.iloc[i]['long']
```

Out[432… Make this Notebook Trusted to load map: File -> Trust Notebook

In [428…

```
# botoes interativos

continents_limit = widgets.Dropdown(
options = df_raw['continent'].unique().tolist(),
value = 'NA',
description = 'Continents',
disable=False
)

country_limit=widgets.Dropdown(
options = df_raw['country'].unique().tolist(),
value = 'Brazil',
description = 'Country',
disable=False

)

# year=widgets.SelectionSlider(
#     options = df_raw['year'].sort_values().unique().tolist(),
#     value=2015,
#     description='Expectancy year',
#     disable=False,
```

```
#     continuous_update=False,
#     orientation='vertical',
#     style={'description_width': 'initial'},
#     readout=True
# )
```

In [431…

```
def update_map( df,continents,countries):

    #map=df_raw[['continent','country','lat','long','population','life-expect
    map=df_raw[(df_raw['continent']==continents_limit)&(df_raw['country']==cc

#     houses = df[(df['price'] <= limit) &
# (df['is_waterfront'] == waterfront) &
# (df['sqft_living'] >= livingroom_limit) &
# (df['bathrooms'] >= bathroom_limit) &
# (df['sqft_basement'] >= basement_limit) &
# (df['condition'] >= condition_limit) &
# (df['yr_built'] >= year_limit )][['id', 'lat', 'long', 'price',␣
# , → 'level']]
    fig=px.scatter_mapbox(map,
                          hover_name='country',
                          hover_data=["life-expectancy", "population"],
                          lat='lat',
                          lon='long',
                          size='life-expectancy',
                          color='continent',
                          color_continuous_scale=px.colors.cyclical.IceFire_r
                          size_max=10,
                          zoom=1)

    fig.update_layout(mapbox_style='open-street-map')
    fig.update_layout(height=500, margin={'r':0,'l':0,'t':0,'b':0})
    fig.show()

widgets.interactive(update_map,df=fixed( df_raw),continents=continents_limit,
```

In [ ]:

In [ ]: