Life Expectancy around the World

Author: *Patrick Fox*
Module: *BDL03 1 NoSQL Lab with Python & MongoDB*
***Date**: 7 April 2022*

Table of Contents

1 Overview

This report focuses on the estimated life expectancy of a country's residents. The analysis focuses on the life expectancy estimates for the year 2020 and the general trend experienced between 1950-2020. The explicit questions addressed and observations are listed in the Analysis section.

This report uses data from the following sources:

- International Database: World Population Estimates and Projections (https://www.census.gov/programs-surveys/international-programs/about/idb.html). This data is maintained by United States Census Bureau. It consists of population estimates and projections for various demographic measures of over 200 countries and areas of the world with populations of 5,000 or more. The primary data used from this API is the estimated life expectancy.
- restcountries.com (https://restcountries.com/v3.1/all). This is a JSON/Rest API that provides addition information on a country, such as region and geographical area etc.
- data.worldbank.org (https://data.worldbank.org/) is used to make a correction to a small number of individual data points from the Census Bureau data - see clean census data

The report uses an Extract Load Transform approach. The process is summarised as:

- Extract data from the JSON/API sources
- Load data into the specific "raw" collection
- Transform data within MongoDB to the "clean" collection and finally a "combined" collection.

An overview of the Data flow is shown in the graphic below:

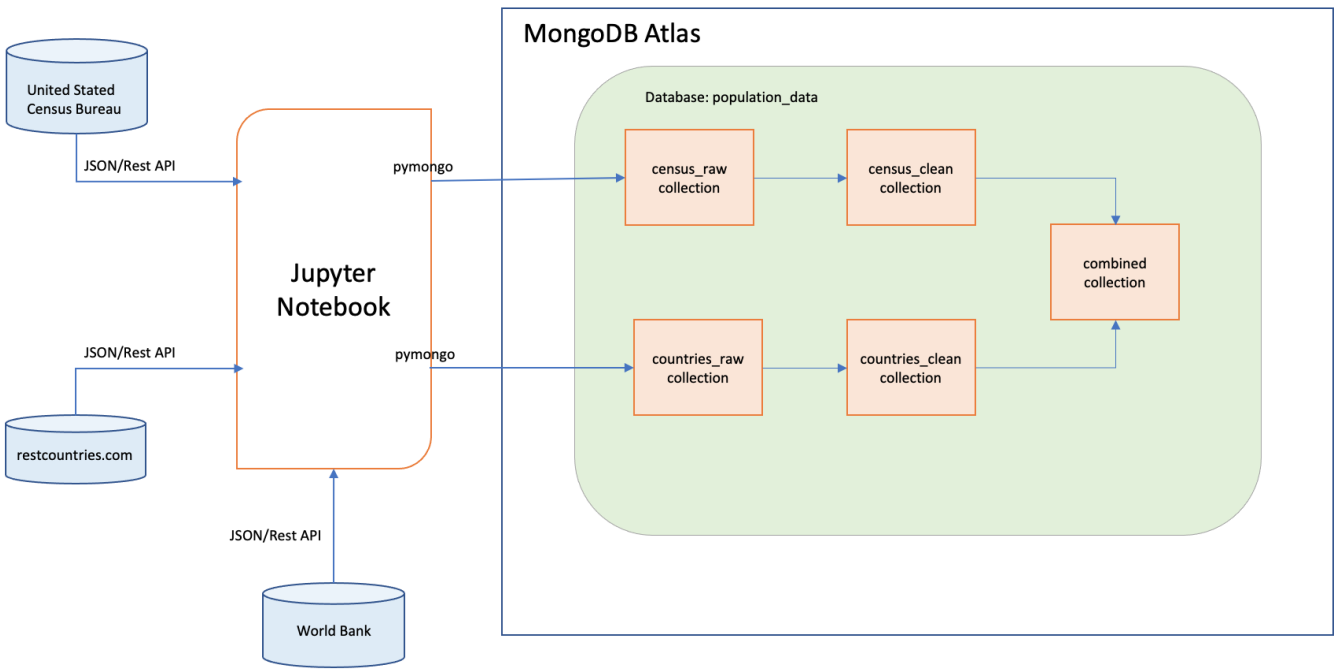


Figure 1. Data flow and transformation

The final "combined" collection held in MongoDB Atlas has the following structure:

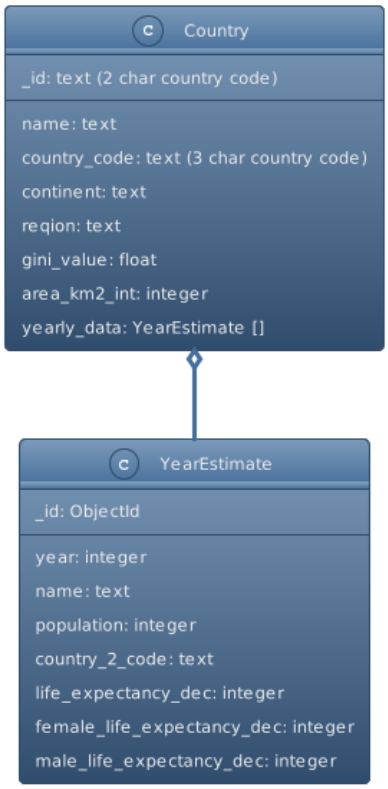


Figure 2. Class Diagram of cleaned/combined data.

A single document example for both "raw" collections and the "combined" collection can be seen in the Appendix

2 Environment Configuration

```
%matplotlib inline
import pymongo
import pprint as pp
import pandas as pd
import requests
import matplotlib.pyplot as plt
import folium
import numpy as np
from scipy.stats import pearsonr
```

The versions of packages installed in the python environment used to generate this report:

```
#pip3 command ↔
```

```
folium                          0.12.1.post1
geopandas                       0.10.2
matplotlib                      3.5.1
matplotlib-inline               0.1.3
numpy                           1.22.3
pandas                          1.4.1
pymongo                         4.0.2
requests                        2.27.1
scipy                           1.8.0
```

The following code cell contains helper functions defintions that are used for creating *choropleth* maps and *boxplots* throughout the Jupyter Notebook.

> **NOTE**: In order to avoid extensive clutter in the generated PDF, the plotting/graphing code cells have been folded away or in some cases hidden. However, all code is available for viewing/running in the accompanying Jupyter Notebook.

```
# function to create choropleth map
def create_choropleth_map(data_values,↔

# function to create a boxplot plot
def create_sub_boxplot(subplot, data, title, ylabel="",xlabel=""):↔
```

3 Extract Load Transform

3.1 Create connections to MongoDB

Define constants used throughout the report.

```
# username/password used for accessing the MongoDB Atlas instance.
MONGO_USERNAME = "pfox"
MONGO_PASSWORD = "Project1z3"
MONGO_CXN_STR = (
    f"mongodb+srv://{MONGO_USERNAME}:{MONGO_PASSWORD}@"
    "cluster0.9lwnm.mongodb.net")

CENSUS_URL = ("https://api.census.gov/data/timeseries/idb/5year?"
              "get=NAME,POP,E0,E0_F,E0_M,FPOP,MPOP,AREA_KM2"
              "&GENC=*&time=from+1950+to+2020")
COUNTRIES_URL = "https://restcountries.com/v3.1/all"

# MongoDB database name and collection names
DB_NAME = "population_data"
RAW_CENSUS_COLL_NAME = "census_raw"
CLEAN_CENSUS_COLL_NAME = "census_clean"
RAW_COUNTRIES_COLL_NAME = "countries_raw"
CLEAN_COUNTRIES_COLL_NAME = "countries_clean"
COMBINED_COLLECTION = "combined"
```

Create a connection to the MongoDB database and collections.

```
#create DB connection
client = pymongo.MongoClient(MONGO_CXN_STR)
the_db = client[DB_NAME]

#census data
raw_census_coll = the_db[RAW_CENSUS_COLL_NAME]
clean_census_coll = the_db[CLEAN_CENSUS_COLL_NAME]

# create countries collections
raw_countries_coll = the_db[RAW_COUNTRIES_COLL_NAME]
clean_countries_coll = the_db[CLEAN_COUNTRIES_COLL_NAME]

#combined data
combined_coll = the_db[COMBINED_COLLECTION]
```

3.2 Remove all Existing Documents

Remove any existing documents from MongoDB Atlas.

```
raw_census_coll.drop()
raw_countries_coll.drop()
clean_census_coll.drop()
clean_countries_coll.drop()
combined_coll.drop()
```

3.3 Fetch Data from JSON API's

Fetch data from the Rest API 'https://api.census.gov/data/timeseries/idb/5year', convert the response into a standard JSON format.

```
# fetch data from Census API
census_response = requests.get(CENSUS_URL)
census_raw_data = census_response.json()
# have to convert it to the standard data format.
census_raw_df = pd.DataFrame(census_raw_data[1:], columns=census_raw_data[0])
census_raw_json = census_raw_df.to_dict("records")
```

After formatting, the response contains 16117 documents.

Fetch data from the Rest API https://restcountries.com/v3.1/all (https://restcountries.com/v3.1/all) .

```
# Fetch data from Countries API
countries_response = requests.get(COUNTRIES_URL)
countries_raw_json = countries_response.json()
```

The formatted JSON contains 250 documents.

3.4 Insert into MongoDB

Load the extracted data from restcountries.com into 'countries_raw' collection and the data from Census Bureau API into the 'census_raw' collection.

```
%%capture
raw_census_coll.insert_many(census_raw_json)
raw_countries_coll.insert_many(countries_raw_json)
```

```
This inserted 16117 documents into the 'census_raw' collection and 250
documents into the 'countries_raw' collection.
```

3.5 Transform Collections

3.5.1 Transform Country data

Reshape the data in the 'countries_raw' collection and create the 'countries_clean' collection.

```
%%capture
#Transform country data
raw_countries_coll.aggregate([
    {"$project":
         {"_id": 0,
          "cca2": 1,
          "cca3": 1,
          "region": 1,
          "subregion": 1,
          "continents": 1,
          "gini": 1,
          "name": 1
          }
     },
    {"$set": {"_id": "$cca2"}},
    {"$set":
         {"gini_obj":
              {"$first": {"$objectToArray": "$gini"}}
          }
     },
    {"$set": {"gini_value": "$gini_obj.v"}},
    {"$out": CLEAN_COUNTRIES_COLL_NAME}
])
```

> The **$objectToArray** is used above as the *key* for the actual numeric gini value keeps changing (they appear to use a year value for the key).

3.5.2 Clean Country Data

Validate that each country document in 'countries_clean' has only one continent associated.

```
# check country has only one continent
countries_with_more_continents = clean_countries_coll.aggregate([
    {"$match":
          {"$or":
            [{"continents.0": {"$exists": False}},
             {"continents.1": {"$exists": True}}
            ]
          }
     }
])
message = (f"There are {len(list(countries_with_more_continents))} countries"
       " with zero continents or more than two continents assigned.")
print(message)
```

```
There are 0 countries with zero continents or more than two continents
assigned.
```

Visually validate the number of countries per continent are approximately correct.

```
# group by continent and count number of countries
country_count_per_region = clean_countries_coll.aggregate([
    {"$group": {"_id": {"$arrayElemAt": ["$continents", 0]},
                "number_countries": {"$sum": 1}
               }
    },
    {"$project":
         {"_id": 0,
          "continent": "$_id",
          "number_of_countries": "$number_countries"
          }
     },
    {"$sort": {"number_of_countries": -1}},
    {"$limit": 10}
])
country_count_per_region_df = pd.DataFrame(
    country_count_per_region).set_index("continent")
```

```
#Plot Table 1↔
```

Table 1: number of countries per continent.

continent	number_of_countries
Africa	58
Europe	53
Asia	52
North America	41
Oceania	27
South America	14
Antarctica	5

3.5.3 Transform Census Data

Reshape the data contained in 'census_raw' collection and place it in the 'cencus_clean' collection. As part of this transformation step, specific fields that are used in later calculation are converted to decimal types.

```
%%capture
# change to more understandable names.
raw_census_coll.update_many({}, {"$rename": {
    "E0": "life_expectancy",
    "E0_F": "female_life_expectancy",
    "E0_M": "male_life_expectancy", }})

# reshape data and convert types
raw_census_coll.aggregate([
    #filter out documents that contain a 'Null' life expectancy
    {"$match": {"life_expectancy": {"$ne": None}}},
    {"$addFields": {
        "year": {"$toInt": "$time"},
        "life_expectancy_dec": {"$toDecimal": "$life_expectancy"},
        "female_life_expectancy_dec": {"$toDecimal": "$female_life_expectancy"},
        "male_life_expectancy_dec": {"$toDecimal": "$male_life_expectancy"},
        "population": {"$toInt": "$POP"},
    }},
    #remove unneeded fields
    {"$project": {
        "life_expectancy": 0,
        "female_life_expectancy": 0,
        "male_life_expectancy": 0,
        "time": 0
    }},
    #rename fields in cleaned data
    {"$project": {
        "country_2_code": "$GENC",
        "area_km2": "$AREA_KM2",
        "name": "$NAME",
        "population": 1,
        "life_expectancy_dec": 1,
        "female_life_expectancy_dec": 1,
        "male_life_expectancy_dec": 1,
        "year": 1
    }},
    {"$out": CLEAN_CENSUS_COLL_NAME}
])
```

3.5.4 Clean Census Data

3.5.4.1 Extreme Life Expectancy Values

Search for any values that are very extreme - they would suggest data may not be correct.

```
#to do check for outliers in values for expected life expectancy
census_value_check = clean_census_coll.aggregate([
    {"$match":
        {"$or": [
            {"life_expectancy_dec": {"$not": {"$gte": 15, "$lt": 95}}},
            {"female_life_expectancy_dec": {"$not": {"$gte": 15, "$lt": 95}}},
            {"male_life_expectancy_dec": {"$not": {"$gte": 15, "$lt": 95}}},
        ]}},
    {"$project": {
        "_id": 0,
        "year": "$year",
        "name": "$name",
        "life_expectancy": "$life_expectancy_dec",
        "female_life_expectancy": "$female_life_expectancy_dec",
        "male_life_expectancy": "$male_life_expectancy_dec"
    }}
])
```

```
#Table 2↔
```

Table 2: Countries with extreme values for life expectancy

	year	name	life_expectancy	female_life_expectancy	male_life_expectancy
0	1975	Cambodia	16.20	19.64	12.91
1	1976	Cambodia	16.39	19.92	13.00
2	1977	Cambodia	16.57	20.20	13.10
3	1978	Cambodia	16.76	20.49	13.20
4	1994	Rwanda	5.43	5.52	5.33

The numbers for Rwanda seem extremely small for 1994 - we will look at Cambodia in section 4.6.

Extracted below is the life expectancy for Rwanda in the years surrounding 1994.

```
#search for rwanda during the 1990 - 1999
rwanda_query_1990_1999 = [
    {"$match": {"name": "Rwanda",
                "year": {"$in": list(range(1990, 2000))}
                }},
    {"$project": {"_id": 0,
                  "year": "$year",
                  "name": "$name",
                  "life_expectancy": "$life_expectancy_dec",
                  "female_life_expectancy": "$female_life_expectancy_dec",
                  "male_life_expectancy": "$female_life_expectancy_dec"
                  }
     },
    {"$sort": {"year": 1}},
    {"$limit": 12}
]
rwanda_1990_1999 = clean_census_coll.aggregate(rwanda_query_1990_1999)
```

```
#Table 3 ↔
```

Table 3: Rwanda's life expectancy in the nineties.

	year	name	life_expectancy	female_life_expectancy	male_life_expectancy
0	1990	Rwanda	52.91	54.98	54.98
1	1991	Rwanda	52.75	54.77	54.77
2	1992	Rwanda	52.56	54.54	54.54
3	1993	Rwanda	52.35	54.28	54.28
4	1994	Rwanda	5.43	5.52	5.52
5	1995	Rwanda	49.61	51.28	51.28
6	1996	Rwanda	49.80	51.42	51.42
7	1997	Rwanda	49.97	51.55	51.55
8	1998	Rwanda	50.13	51.67	51.67
9	1999	Rwanda	50.28	51.77	51.77

Data from the World Bank (https://data.worldbank.org/indicator/SP.DYN.LE00.IN?locations=RW) (for Rwanda during this timeperiod) shows that there was a dramatic drop in life expectancy. But this decrease in life expectancy occurred over approx 10 years and the lowest value for life expectancy is age 26. I decided to update the Census data for Rwanda ,for this specific time period, using the data from the World Bank - see below.

The data is fetched from the World Bank API and updated in the census_clean collection.

```
RWANDA_WORLD_BANK=("https://api.worldbank.org/v2/country/"
                   "rwa/indicator/SP.DYN.LE00.IN"
                   ";SP.DYN.LE00.FE.IN;SP.DYN.LE00.MA.IN?"
                   "format=json&date=1987:2000&source=2")
#retrieve json response from API and convert to DataFrame
rwanda_response = requests.get(RWANDA_WORLD_BANK)
rwanda_json = rwanda_response.json()
rwanda_df = pd.json_normalize(rwanda_json[1])

for the_year in range(1990,2000):
    #retrieve the values from json response
    life_expectancy_wbank = rwanda_df.loc[
        ((rwanda_df["indicator.id"]=="SP.DYN.LE00.IN") &
           (rwanda_df["date"]==str(the_year))),"value"].item()
    female_life_expectancy_wbank = rwanda_df.loc[
        ((rwanda_df["indicator.id"]=="SP.DYN.LE00.FE.IN") &
           (rwanda_df["date"]==str(the_year))),"value"].item()
    male_life_expectancy_wbank = rwanda_df.loc[(
        (rwanda_df["indicator.id"]=="SP.DYN.LE00.MA.IN") &
          (rwanda_df["date"]==str(the_year))),"value"].item()

    # update each year
    clean_census_coll.update_one(
        {"name": "Rwanda", "year": the_year},
        {"$set":
           {"life_expectancy_dec": float(life_expectancy_wbank),
            "female_life_expectancy_dec": float(female_life_expectancy_wbank),
            "male_life_expectancy_dec": float(male_life_expectancy_wbank)
            }
        }
    )
```

Rerun the query against MongoDB to ensure the update was successful.

```
rwanda_years_corrected = clean_census_coll.aggregate(rwanda_query_1990_1999)
```

```
#display table 4↔
```

Table 4: AFTER CORRECTION Rwanda's life expectancy in the nineties.

	year	name	life_expectancy	female_life_expectancy	male_life_expectancy
0	1990	Rwanda	33.413000	34.941000	34.941000
1	1991	Rwanda	29.248000	30.761000	30.761000
2	1992	Rwanda	26.691000	28.161000	28.161000
3	1993	Rwanda	26.172000	27.571000	27.571000
4	1994	Rwanda	27.738000	29.042000	29.042000
5	1995	Rwanda	31.037000	32.232000	32.232000
6	1996	Rwanda	35.380000	36.463000	36.463000
7	1997	Rwanda	39.838000	40.821000	40.821000
8	1998	Rwanda	43.686000	44.593000	44.593000
9	1999	Rwanda	46.639000	47.499000	47.499000

3.5.4.2 Validate Correct Range of Years

Check to ensure the correct range of years are returned from the Census Bureau API.

```
# check the year value is in the expected range
census_value_check = clean_census_coll.aggregate([
    {"$match": {"year": {"$not": {"$gte": 1950, "$lte": 2020}}},
     }
])
message=(f"There are {len(list(census_value_check))} countries "
         "with years outside the expected range (1950 - 2020).")
print(message)
```

```
There are 0 countries with years outside the expected range (1950 - 20
20).
```

3.5.5 Create Combined Collection

A 'combined' collection is created using the 'census_clean' collection and the 'countries_clean' collection.

The 'census_clean' data is the primary data and the 'countries_clean' data is used to supplement.

```
#create parent-child documents and write out as combined collection
clean_census_coll.aggregate([
    #group by country
    {"$group": {"_id": "$name",
                "area_km2": {"$last": "$area_km2"},
                "country_2_code": {"$last": "$country_2_code"}
               }
    },
    {"$addFields": {"area_km2_int": {"$toInt": "$area_km2"}}},
    {"$addFields": {"name": "$_id"}},
    #add orginal documents as children to new group
    {"$lookup":
         {"from": CLEAN_CENSUS_COLL_NAME,
          "localField": "name",
          "foreignField": "name",
          "as": "yearly_data"}},
    #add country details as a child document
    {"$lookup":
         {"from": CLEAN_COUNTRIES_COLL_NAME,
          "localField": "country_2_code",
          "foreignField": "_id",
          "as": "country_data"}},
    #flatten country child document into parent document
    {"$unwind": "$country_data"},
    {"$addFields": {"continent":
                        {"$arrayElemAt": ["$country_data.continents", 0]}}},
    {"$project": {"_id": "$country_2_code",
                  "name": "$name",
                  "country_code": "$country_data.cca3",
                  "area_km2_int": 1,
                  "region": "$country_data.region",
                  "subregion": "$country_data.subregion",
                  "gini_value": "$country_data.gini_value",
                  "yearly_data": 1,
                  "continent": 1,

                  }
     },
    {"$out": COMBINED_COLLECTION}])

#read in from mongodb, to confirm number of documents inserted
country_list = combined_coll.aggregate([
    {"$project": { "_id":0,"country": "$name"}}
])
message=(f"The 'combined' collection contains {len(list(country_list))} "
         "country documents in total.")
print(message)
```

```
The 'combined' collection contains 225 country documents in total.
```

Many countries do not have estimates of Life Expectancy attached for every year during the timeperiod 1950 - 2020. Many countries appeared to only have estimates for the later years. **Figure 3** (below) shows how many countries have estimates attached for each year.

```
# find number of life expectancy records/docs per year
expected_life_records_by_year = combined_coll.aggregate([
    {"$unwind": "$yearly_data"},
    {"$group": {"_id": "$yearly_data.year", "number_yearly_docs": {"$sum": 1}}},
    {"$set": {"year": "$_id"}},
    {"$project": {"_id": 0}},
    {"$sort": {"year": 1}},
    {"$limit": 100}
])
records_by_year_df = pd.DataFrame(expected_life_records_by_year)
```

```
#Plot the number of yearly docs↔
```

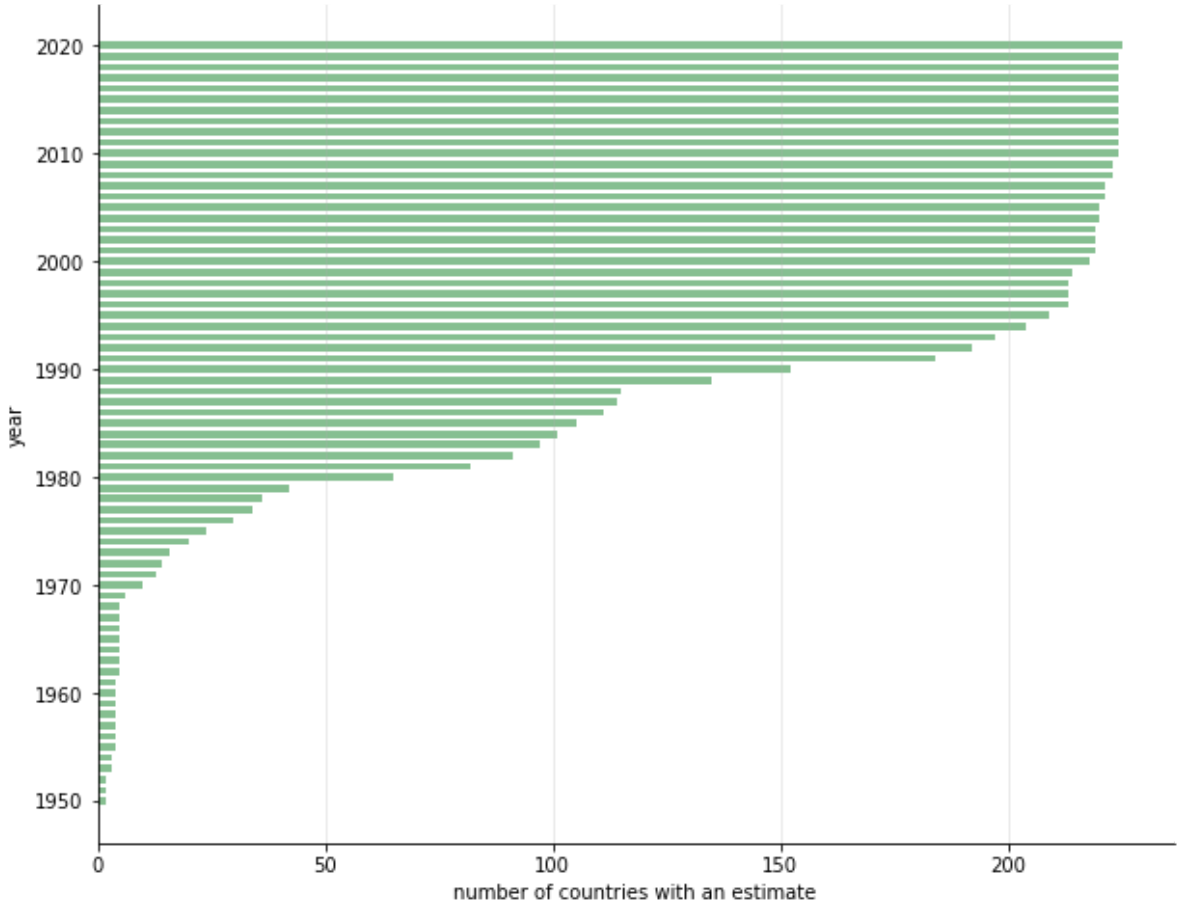


Figure 3: How many countries have a Life Expectancy estimate recorded for that specific year.

Show the top 5 countries with the most estimates attached.

```
# how many yearly documents each country has attached
amount_yearly_data = combined_coll.aggregate([
    {"$project": {"_id": 0,
                  "country": "$name",
                  "number_yearly_estimates": {"$size": "$yearly_data"}
                 }
    },
    {"$sort": {"number_yearly_estimates": -1}}])

amount_yearly_data_df = pd.DataFrame(amount_yearly_data)
```

```
#display Table 5 ↔
```

Table 5: Countries with most yearly estimates.

country	number_yearly_estimates
Bhutan	71
Djibouti	71
Nigeria	68
Guinea	66
Cambodia	59

4 Analysis

4.1 Worldwide in 2020, which countries have the highest and lowest life expectancy?

Fetch the life expectancy per country for the year 2020.

```
life_expt_2020 = combined_coll.aggregate([
    {"$unwind": "$yearly_data"},
    {"$match": {"yearly_data.year": 2020}},
    {"$sort": {"yearly_data.life_expectancy_dec": -1}},
    {"$project":
       {"_id": 0,
        "country": "$name",
        "country_code": 1,
        "area_km2": "$area_km2_int",
        "gini_value": 1,
        "country_population": "$yearly_data.population",
        "life_expectancy": "$yearly_data.life_expectancy_dec",
        "female_life_expectancy": "$yearly_data.female_life_expectancy_dec",
        "male_life_expectancy": "$yearly_data.male_life_expectancy_dec"
        }
     },
    {"$limit": 300}
])

life_expt_2020_df = pd.DataFrame(life_expt_2020)
# convert from type Decimal to float (needed for plotting)
life_expt_2020_df["life_expectancy"] = life_expt_2020_df[
    "life_expectancy"].astype(str).astype(float)
life_expt_2020_df["male_life_expectancy"] = life_expt_2020_df[
    "male_life_expectancy"].astype(str).astype(float)
life_expt_2020_df["female_life_expectancy"] = life_expt_2020_df[
    "female_life_expectancy"].astype(str).astype(float)
```

```
#display Table 6 ↔
```

Table 6: The 5 countries world wide with highest life expectancy in 2020.

country	life_expectancy	area_km2	country_population
Monaco	89.270000	2	31066
Singapore	86.030000	709	5810285
Macau	84.630000	28	625295
Japan	84.470000	364485	125135727
San Marino	83.490000	61	34247

```
#display Table 7 ↔
```

Table 7: The 5 countries worldwide with lowest life expectancy in 2020.

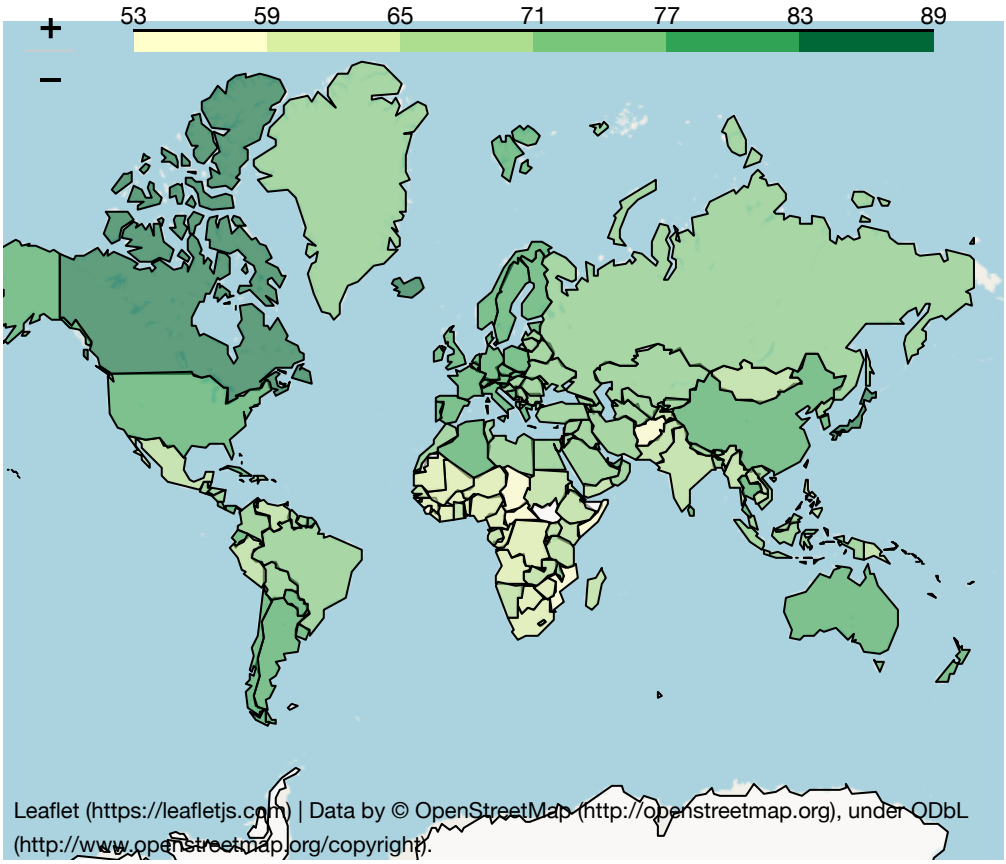
country	life_expectancy	area_km2	country_population
Mozambique	55.870000	786380	30097734
South Sudan	55.540000	644329	10560984
Somalia	54.920000	627337	11818529
Central African Republic	54.620000	622984	5262642
Afghanistan	52.840000	652230	36594776

Plot the average life expectancy on a choropleth map. The darker green signifies the longer life expectancy.

Observation:

- Monaco has the highest life expectancy (age 89).
- Afghanistan has the lowest life expectancy (age 52) - (at least from the Census Bureau data).
- From the map (below) it appears that life expectancy is generally high in Europe, North America and parts of Asia.
- From the map (below) it seems that central region of Africa has a lower life expectancy.

```
#plot map - Figure 4
```



4.2 Worldwide in 2020, do females or males have the longest life expectancy?

Using the life expectancy reported per country (worldwide) in 2020; compare the median female life expectancy against the median male life expectancy.

```
#Plot Figure 5↔
```

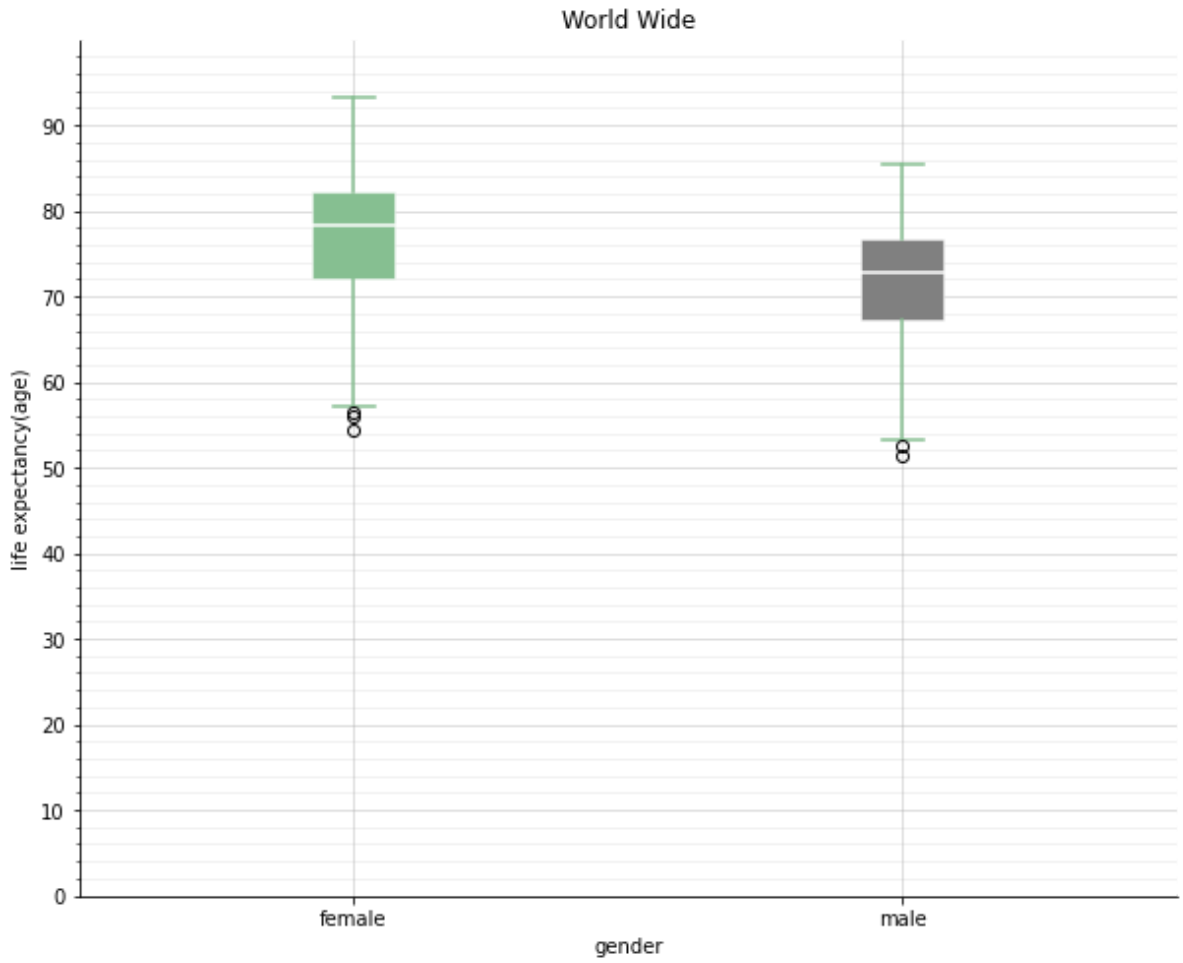


Figure 5: 2020 worldwide, Boxplot of Life Expectancy reported by country.

Observations:

- It seems the median female life expectancy reported from countries world wide is approx 78 years while male is approx 73 years
- The distribution of both females and males life expectancy(per country worldwide) seems reasonably semetric (with a few lower outliers)

4.3 In 2020, how does female and male life expectancy break down per continent?

Group the countries by continent and average the countries life expectancy.

In the grouping below, all the actual female and male life expectancy values per country are retained in the lists **female_life_expectancy_list** and **male_life_expectancy_list**. These where used in the subsequent boxplots in Figure 4

```
# group by continent
group_by_continent_2020 = combined_coll.aggregate([
    #flatten out
    {"$unwind": "$yearly_data"},
    #match only 2020
    {"$match": {"yearly_data.year": 2020}},
    #group by continent
    {"$group":
      {"_id": "$continent",
       "area_km2": {"$sum": "$area_km2_int"},
       "life_expectancy": {"$avg": "$yearly_data.life_expectancy_dec"},
       "female_life_expectancy":
           {"$avg": "$yearly_data.female_life_expectancy_dec"},
        "male_life_expectancy":
            {"$avg": "$yearly_data.male_life_expectancy_dec"},
        #retain original values - used by boxplot
        "female_life_expectancy_list":
            {"$push": "$yearly_data.female_life_expectancy_dec"},
        "male_life_expectancy_list":
            {"$push": "$yearly_data.male_life_expectancy_dec"},
        "number_of_countries": {"$sum": 1},
        }
     },
    {"$project":
     {"_id": 0,
      "continent": "$_id",
      "area_km2(millions)": {"$round": [{"$divide": ["$area_km2", 1000000]}, 2]},
      "avg_life_expectancy": {"$round": ["$life_expectancy", 2]},
      "female_life_expectancy": {"$round": ["$female_life_expectancy", 2]},
      "male_life_expectancy": {"$round": ["$male_life_expectancy", 2]},
      "female_life_expectancy_list": 1,
      "male_life_expectancy_list": 1,
      "number_of_countries": 1
        }
     },
    {"$sort": {"avg_life_expectancy": -1}},
    {"$limit": 10}
])
group_by_continent_2020_df = pd.DataFrame(group_by_continent_2020)
```

```
#display Table 8. ↔
```

Table 8: Average country life expectancy per continent in 2020.

continent	avg_life_expectancy	number_of_countries	area_km2(millions)
Europe	79.48	50	22.180000
North America	77.26	38	23.090000
Asia	74.92	48	30.850000
South America	74.85	12	17.320000
Oceania	74.76	22	8.500000
Africa	65.62	55	29.870000

Create a boxplot subplot for each continent, that shows the median male and female life expectancy for each country.

```
#plot boxplots per continent Figure 6↔
```

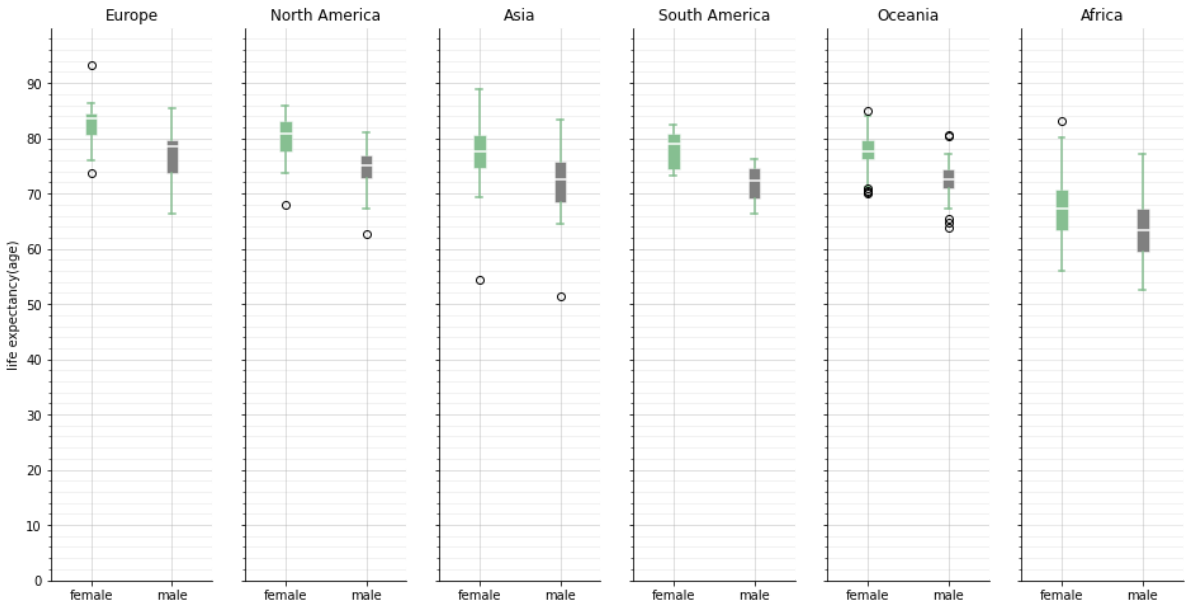


Figure 4. Boxplot of life expectancy reported by country, grouped by continent.

Observations:

- In all continents it seems the median for life expectancy per country is consisently higher for females than males.
- Europe seems to have the highest medians for both female and male.
- Africa has the lowest medians and the widest distributions of life expectancies of any of the continents.

4.4 In 2020, which european countries have the highest/lowest life expectancy?

```
europe_diff = combined_coll.aggregate([
    {"$unwind": "$yearly_data"},
    {"$match": {"yearly_data.year": 2020}},
    {"$match": {"continent": "Europe"}},
    {"$project":
     {"_id": 0,
      "country": "$name",
      "country_code": 1,
      "life_expectancy": {"$round": ["$yearly_data.life_expectancy_dec", 2]},
      "female_life_expectancy":
          {"$round": ["$yearly_data.female_life_expectancy_dec", 2]},
      "male_life_expectancy":
          {"$round": ["$yearly_data.male_life_expectancy_dec", 2]},
      "female_male_difference":
          {"$round":
           [{ "$subtract":
              ["$yearly_data.female_life_expectancy_dec",
               "$yearly_data.male_life_expectancy_dec"]
            }, 2]
          },
        }
     },
    {"$sort": {"life_expectancy": -1}},
    {"$limit": 100}])

europe_df = pd.DataFrame(europe_diff)
europe_df["life_expectancy"] = europe_df["life_expectancy"].astype(str).astype(flo
europe_df["female_male_difference"] = europe_df[
    "female_male_difference"].astype(str).astype(float)
```

```
#display Table 9 ↔
```

Table 9: The top 6 countries in Europe by life expectancy in 2020.

	country	life_expectancy
0	Monaco	89.270000
1	San Marino	83.490000
2	Iceland	83.260000
3	Andorra	83.030000
4	Guernsey	82.840000
5	Switzerland	82.830000

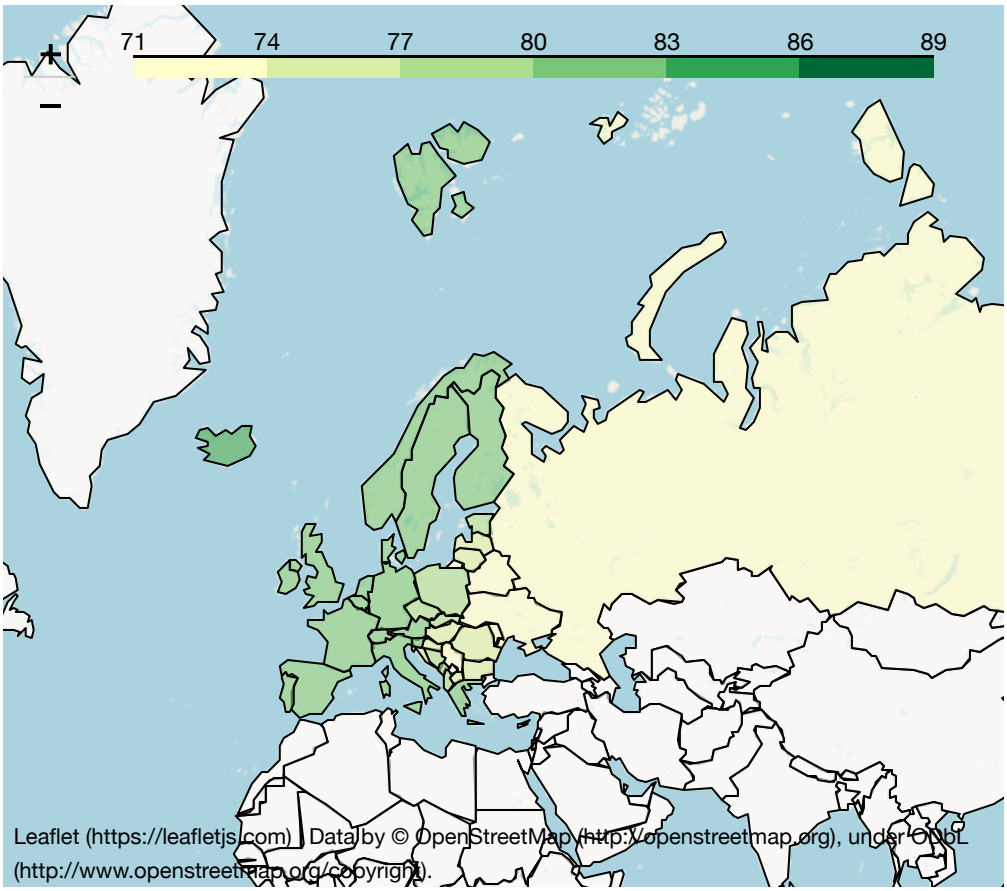
```
#Table 10↔
```

Table 10: The lowest 6 countries in Europe by life expectancy in 2020.

	country	life_expectancy
44	Serbia	73.890000
45	Belarus	73.740000
46	Ukraine	72.900000
47	Moldova	71.880000
48	Russia	71.880000
49	Kosovo	71.050000

Map life expectency on the map of Europe for 2020. Darker green shows longer life expectancy - legend in years.

```
#Figure 7
create_choropleth_map(europe_df,"life_expectancy",the_focus=[65,30],the_zoom=2.0)
```



Observations:

- Monaco, although small, has the highest life expectancy in Europe.
- Kosovo has the lowest life expectancy in Europe.
- Life expectancy seems lowest in the east of Europe. It appears to increase in the west and north of Europe

4.5 In 2020 within Europe, where is largest divergence between female and male life expectancy?

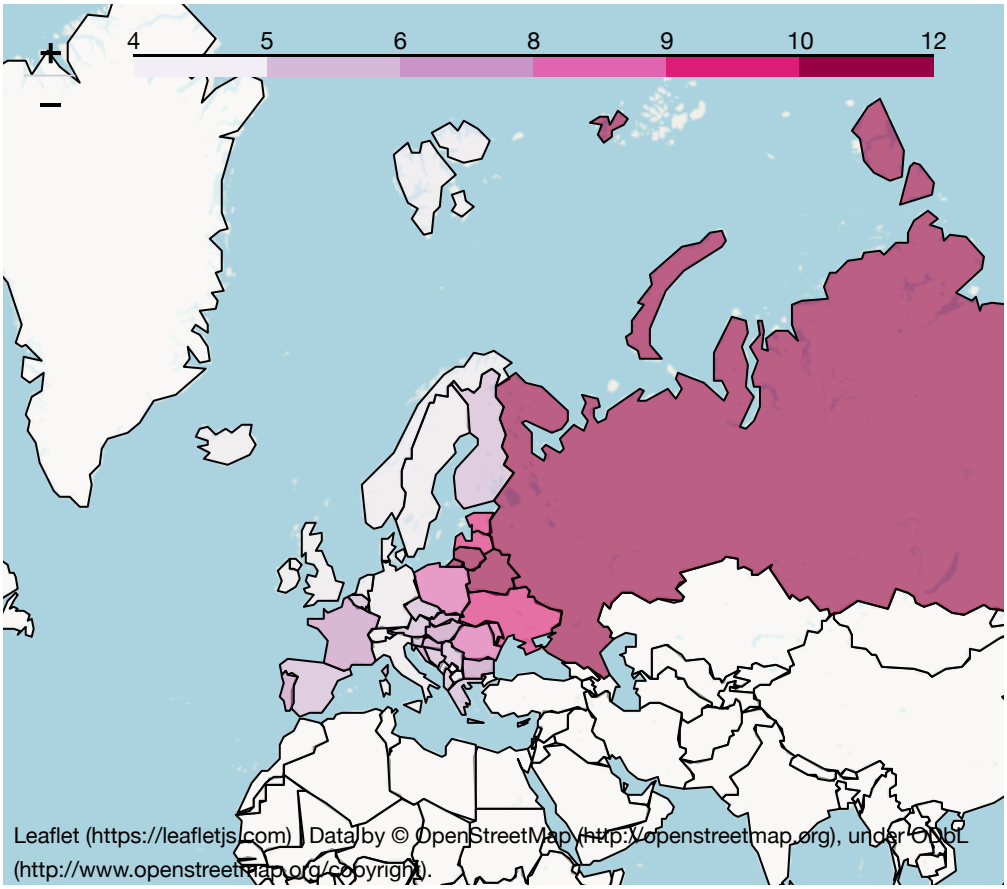
```
#display Table 11.↔
```

Table 11: The top 6 countries, with greatest difference between male and female life expectancy (ascending order).

country	female_male_difference
Russia	11.500000
Belarus	11.200000
Lithuania	11.110000
Ukraine	9.700000
Estonia	9.600000

The following map shows how much longer a female is expected to live compared to a male, within the countries of Europe. The darker the maroon colour, the longer a females lives compared to a male (legend in years).

```
#Figure 8↔
```



Observations:

- Russia has the largest divergence in life expectancy in Europe - females are expected to live 11.5 years longer.
- It seems that in the east of Europe, females live quite a lot longer than males.
- In the centre and north of Europe, females still live longer, but the difference in life expectancy is not as large.

4.6 Worldwide from 1950 to 2020, what is the life expectancy trend per continent?

Below life expectancy is grouped by continents and plotted against time. As showned in Create Combined Collection; not all countries have life expectancy records for each year.

```
# average per year per continent
average_by_year = combined_coll.aggregate([
    {"$unwind": "$yearly_data"},
    {"$group":
        {"_id":
            {"year": "$yearly_data.year",
             "continent": "$continent"
             },
         "avg_life_expectancy":
            {"$avg": "$yearly_data.life_expectancy_dec"}
         }
     },
    {"$project":
        {"_id": 0,
         "year": "$_id.year",
         "continent": "$_id.continent",
         "avg_life_expectancy":
            {"$round": ["$avg_life_expectancy", 2]}
         }
     },
    {"$sort": {"year": 1}},
    {"$limit": 800}
])

plotting_df = pd.DataFrame(average_by_year)
```

```
#plot continents line graph "trend" - Figure 9
```

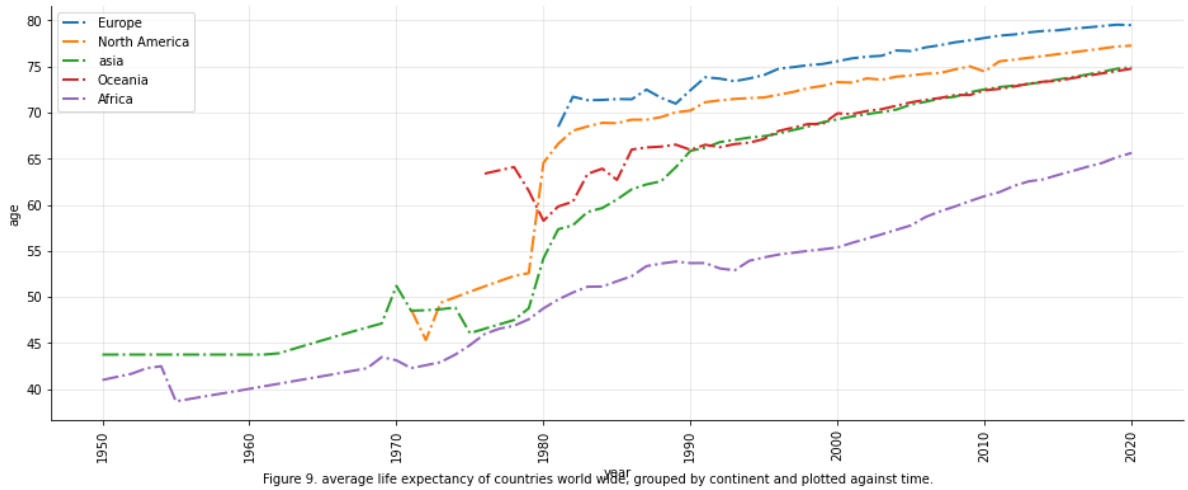


Figure 9. average life expectancy of countries world wide, grouped by continent and plotted against time.

Observations:

- All continents are experiencing a general upward trend.
- Europe have consistently had a higher average of country life expectancy for the last four decades (for all the data points we have in our dataset)

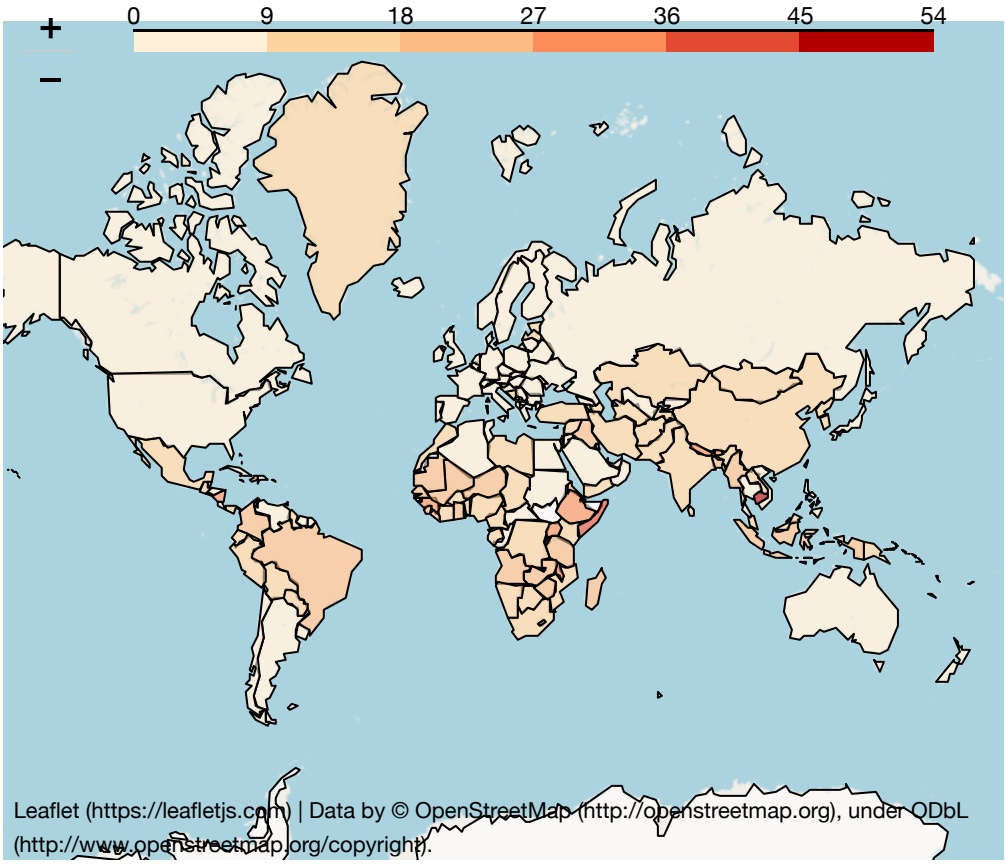
4.7 Worldwide from 1950 to 2020, which country has the biggest change in life expectancy?

Find the biggest difference in life expectancy listed for each country.

```
# country with greatest change in life expectancy
change_in_life_expectancy = combined_coll.aggregate([
    {"$addFields": {"number_years_recorded": {"$size": "$yearly_data"}}},
    {"$unwind": {"path": "$yearly_data"}},
    #group by country - finding max and min life expectancy
    {"$group":
     {"_id": "$name",
      "country_code": {"$last": "$country_code"},
      "number_years_recorded": {"$last": "$number_years_recorded"},
      "max_life_expectancy": {"$max": "$yearly_data.life_expectancy_dec"},
      "min_life_expectancy": {"$min": "$yearly_data.life_expectancy_dec"}
      }
     },
    #calculate life expectancy change per country
    {"$project":
     {"_id": 0,
      "country": "$_id",
      "country_code": 1,
      "number_years_recorded": 1,
      "max_life_expectancy": 1,
      "min_life_expectancy": 1,
      "change_in_life_expectancy":
          {"$subtract": ["$max_life_expectancy", "$min_life_expectancy"]}
      }
     },
    {"$sort": {"change_in_life_expectancy": -1}},
    {"$limit": 300}
])
change_df = pd.DataFrame(change_in_life_expectancy)
change_df["change_in_life_expectancy"] = change_df[
    "change_in_life_expectancy"].astype(str).astype(float)
```

Map shows how much life expectancy worldwide has changed between 1950 and 2020 (based on the data we have). The darker red signifies a larger change in life expectancy (legend is in years).

```
#plot the change in life expectancy - Figure 10 ↔
```



Observations:

- It seems that Asia, Africa and South America experienced the greatest change in life expectancy.

The country that experienced the greatest change in life expectancy:

```
#display Table ↔
```

Table 12: Country with greatest life expectancy change from 1950 to 2020.

	country	number_years_recorded	change_in_life_expectancy
0	Cambodia	59	54.060000

Plot Cambodia's life expectancy over time and include the average of all countries' life expectancy worldwide.

```
#extract yearly data for cambodia
output = combined_coll.aggregate([
    #filter on Cambodia
    {"$match": {"name": "Cambodia"}},
    #flatten out yearly_data
    {"$unwind": {"path": "$yearly_data"}},
    {"$project":
        {"_id": 0,
         "year": "$yearly_data.year",
         "life_expectancy": "$yearly_data.life_expectancy_dec",
         "female_life_expectancy": "$yearly_data.female_life_expectancy_dec",
         "male_life_expectancy": "$yearly_data.male_life_expectancy_dec"
         }
     },
    {"$sort": {"year": 1}},
    {"$limit": 100}
])
cambodia_df = pd.DataFrame(output)
```

```
# get average life expectancy for all countries (world wide)
average_by_year = combined_coll.aggregate([
    {"$unwind": {"path": "$yearly_data"}},
    {"$group": {"_id": "$yearly_data.year",
                "avg_life_expectancy":
                    {"$avg": "$yearly_data.life_expectancy_dec"}
                }
     },
    {"$project":
        {"_id": 0,
         "year": "$_id",
         "avg_life_expectancy": {"$round": ["$avg_life_expectancy", 2]}
         }
     },
    {"$sort": {"year": 1}},
    {"$limit": 100}
])
average_expectancy = pd.DataFrame(average_by_year)
```

```
#Plot line graph for cambodia and world average Figure 11 ↔
```

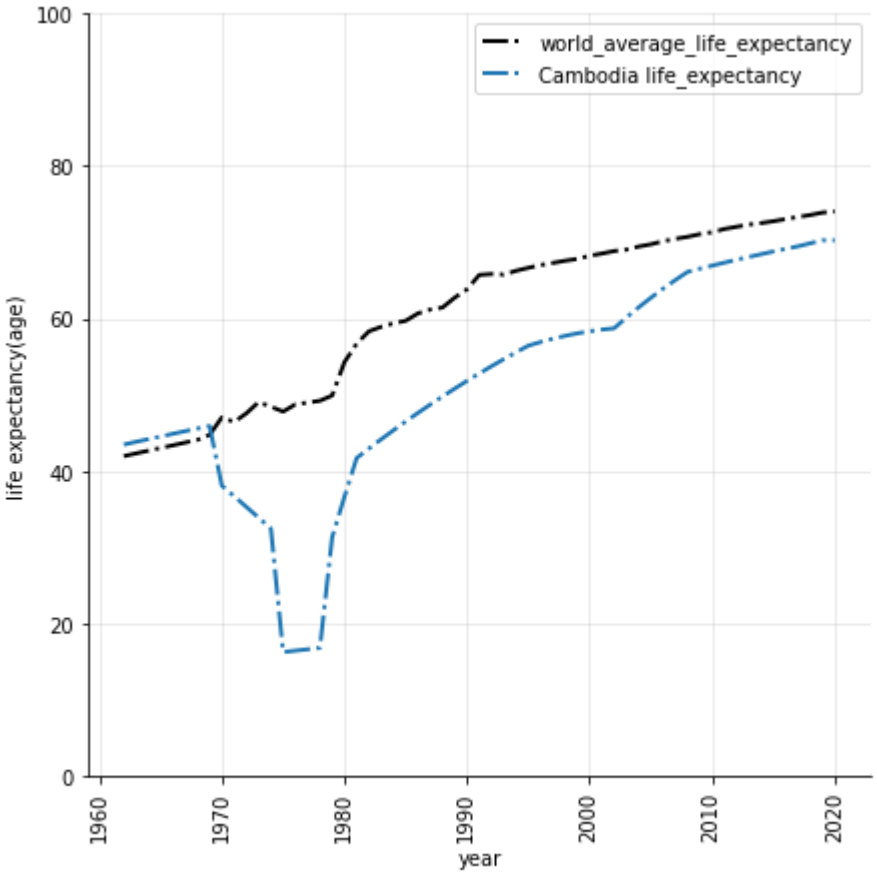


Figure 11. life expectancy of Cambodia and worldwide average country life expectancy.

Observations:

- Cambodia is the country with the biggest life expectancy change worldwide (for the data we have) between 1950-2020
- The lowest life expectancy for Combodia happend between 1975 and 1980. This seems to correspond to the same time period as the Cambodian genocide. (https://en.wikipedia.org/wiki/Cambodian_genocide)

4.8 In 2020, is there a correlation between a country's Gini Index and its life expectancy?

Reuse the worldwide dataframe from the first section rather than doing another MongoDB query.

The Gini Index values (that are available in our dataset) are plotted against the country's life expectancy for the year 2020.

The **Gini Index** is a measure of statistical dispersion intended to represent the income inequality or the wealth inequality within a nation or a social group. The Gini Index (coefficient) was developed by statistician and sociologist Corrado Gini. For details see Gini Index (https://en.wikipedia.org/wiki/Gini_coefficient)

```
#graph scatter between life expectancy and the GINI Index - figure 12
```

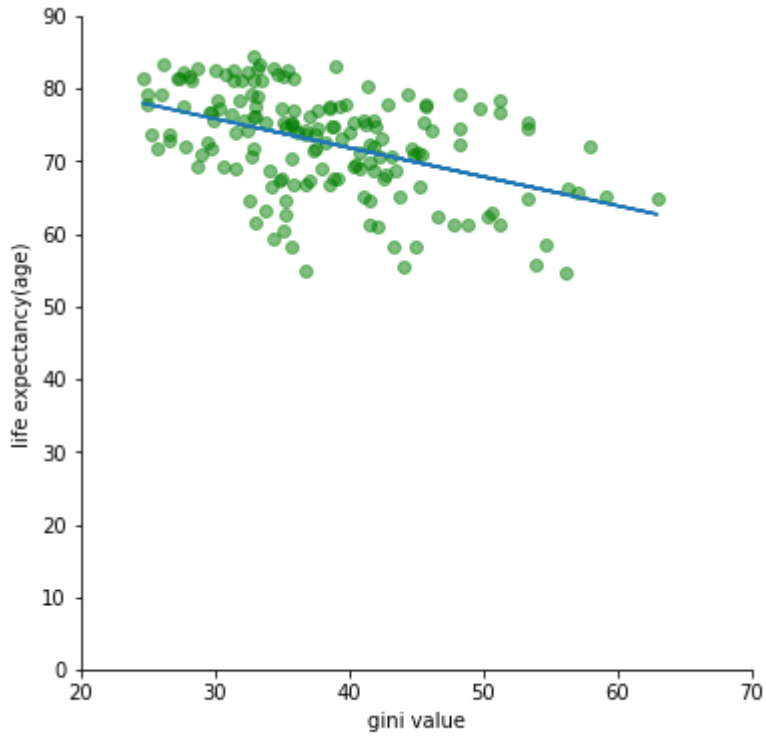


Figure 12. correlation between life expectancy and Gini Index.

```
The pearson correlation coefficient is -0.453513986915165
```

Observations:

- There appears to be a negative covariance between countries average life expectancy and the countries Gini Index.
- The correlation appears to be presents but would generally not be considered very strong.
- Further investigation would be required (with more Gini data) to explore this potential correlation.

5 Conclusions

The interesting observations for each questions are contained in the "green" boxes above. Here are some general conclusions:

- The continent of Europe has consistently had a higher average country life expectancy in the last four decades
- All the contients seem to have an average country life expectancy that is trending upwards.
- Monaco is the overall highest life expectancy in the world in 2020

6 Appendix

6.1 Document formats

6.1.1 One Document from the "raw countries collection"

```python
# single record from raw data
raw_countries_data = raw_countries_coll.aggregate([
        {"$project": {"_id":0, "translations":0}},
        {"$limit": 1}
])

for record in raw_countries_data:
    pp.pprint(record)
```

```
{'altSpellings': ['UY',
                  'Oriental Republic of Uruguay',
                  'República Oriental del Uruguay'],
 'area': 181034.0,
 'borders': ['ARG', 'BRA'],
 'capital': ['Montevideo'],
 'capitalInfo': {'latlng': [-34.85, -56.17]},
 'car': {'side': 'right', 'signs': ['ROU']},
 'cca2': 'UY',
 'cca3': 'URY',
 'ccn3': '858',
 'cioc': 'URU',
 'coatOfArms': {'png': 'https://mainfacts.com/media/images/coats_of_ar
ms/uy.png',
                'svg': 'https://mainfacts.com/media/images/coats_of_ar
ms/uy.svg'},
 'continents': ['South America'],
 'currencies': {'UYU': {'name': 'Uruguayan peso', 'symbol': '$'}},
 'demonyms': {'eng': {'f': 'Uruguayan', 'm': 'Uruguayan'},
              'fra': {'f': 'Uruguayenne', 'm': 'Uruguayen'}},
 'fifa': 'URU',
 'flag': '🇺🇾',
 'flags': {'png': 'https://flagcdn.com/w320/uy.png',
           'svg': 'https://flagcdn.com/uy.svg'},
 'gini': {'2019': 39.7},
 'idd': {'root': '+5', 'suffixes': ['98']},
 'independent': True,
 'landlocked': False,
 'languages': {'spa': 'Spanish'},
 'latlng': [-33.0, -56.0],
 'maps': {'googleMaps': 'https://goo.gl/maps/tiQ9Baekb1jQtDSD9',
          'openStreetMaps': 'https://www.openstreetmap.org/relation/28
7072'},
 'name': {'common': 'Uruguay',
          'nativeName': {'spa': {'common': 'Uruguay',
                                 'official': 'República Oriental del U
ruguay'}},
          'official': 'Oriental Republic of Uruguay'},
 'population': 3473727,
 'postalCode': {'format': '#####', 'regex': '^(\\d{5})$'},
 'region': 'Americas',
 'startOfWeek': 'monday',
 'status': 'officially-assigned',
 'subregion': 'South America',
 'timezones': ['UTC-03:00'],
 'tld': ['.uy'],
 'unMember': True}
```

6.1.2 One Document from the "raw census collection"

```
# single record from raw data
raw_census_data = raw_census_coll.aggregate([
        {"$project": {"_id":0}},
        {"$limit": 1}
])

for document in raw_census_data:
    pp.pprint(document)
```

```
{'AREA_KM2': '468',
 'FPOP': None,
 'GENC': 'AD',
 'MPOP': None,
 'NAME': 'Andorra',
 'POP': '6176',
 'female_life_expectancy': None,
 'life_expectancy': None,
 'male_life_expectancy': None,
 'time': '1950'}
```

6.1.3 One Document from the "combined collection"

> **NOTE**: In the case of the US (below), it has only one yearly data document associated. For most other countries there are many documents contained in the yearly_data array.

```
# single record from raw data
raw_combined_data = combined_coll.aggregate([
        {"$match": {"_id":"US"}},
        {"$limit": 1}
])

for document in raw_combined_data:
    pp.pprint(document)
```

```
{'_id': 'US',
 'area_km2_int': 9150541,
 'continent': 'North America',
 'country_code': 'USA',
 'gini_value': 41.4,
 'name': 'United States',
 'region': 'Americas',
 'subregion': 'North America',
 'yearly_data': [{'_id': ObjectId('624f030838914fe7849eb340'),
                  'area_km2': '9150541',
                  'country_2_code': 'US',
                  'female_life_expectancy_dec': Decimal128('82.51'),
                  'life_expectancy_dec': Decimal128('80.27'),
                  'male_life_expectancy_dec': Decimal128('77.99'),
                  'name': 'United States',
                  'population': 332639102,
                  'year': 2020}]}
```