Using Linear Regression to Predict Life Expectancy

BA305 A1

Team 9

Ashley Nguyen, Vy Nguyen, Ken Ye, Luke Her

1. Introduction

1.1. Project Context & Purpose

The benefit of knowing life expectancy is to evaluate the performance of the government in improving the welfare of the population in general and improving economic, education, and health status in particular. Moreover, population aging is often depicted as a challenge to economic progress and to the sustainability of public budgets. If global life expectancy is projected to increase, and countries and society do not adjust laws and norms, survival beyond a certain "old age" will be less convenient. When appropriate policies are put in place, population aging can in fact benefit economic development and economic progress. Governments and organizations such as the United Nations (UN) can harness population aging for economic progress and ensure economic security in old age by, for example, embracing diversity in a newly defined old age, removing barriers to access decent work, and improving access to social protection. Those adjustments in policies will impact every single individual.

For this project, our team is modeling the UN diving into different macro-factors provided from World Health Organization (WHO) dataset 2014 for a linear regression prediction of life expectancy (Y). Our team ran the dataset through four linear regression models: (1) on all variables, (2) on statistically significant variables, (3) on variables with high correlation to the target variable, and (4) on variables selected by Stepwise method (Section 4) – to find the best-fitted linear regression prediction. Note that due to the complex dynamics of the pandemic from 2020, our prediction only claims to hold validation in exclusion of direct pandemic effects on people's life span. Data of 2014 was used as it is the closest year with full accessible data.

1.2. Our datasets

The original dataset ("Life_expectancy_data.csv") contains data of 22 features from the years 2000-2015 for 183 countries. They were recorded by the Global Health Observatory (GHO) data repository under WHO – defined as factors influencing one country's life expectancy for the purpose of health data analysis – found on public source Kaggle. Among all categories of health-related factors, only those that are deemed as critical were chosen to represent. Although obtained from WHO, the dataset has some missing data points. Missing data are mostly from countries or features that are difficult to obtain public records such as Vanuatu, North Korea, Libya, etc. or BMI, Thinness, etc. The team has filled in the data manually and decided to drop the ones beyond the reach. The final imported dataset 'Cleaned_data2014.csv' comprises 16 features, whereas only 1 is categorical (Exhibit 1), and 180 observations/countries in 2014.

In addition, for testing purpose, there are 2 self-gathered datasets:

1. US2021
   - "US2021.csv" contains the data of 16 variables as features of our final dataset. Since this dataset presents only statistics of the country United States (U.S.) in 2021, it has only 1 row. The data is self-gathered by the team from GHO and WHO main public webpage.

- Together with the testing dataset, this dataset is also used to test all 4 models. We fit our 4 models to predict the life expectancy of the US in 2021 and compare our results with the gathered true US 2021 life expectancy to see which one gives off the closest value.
- As aforementioned, the models do not consider the direct effect of pandemic on life span as a column although in reality, it may be a critical factor in the year 2020-present. However, there are some underlying indirect effects as the dataset includes features such as GDP or population which are inevitably influenced.

2. Life_Exp_2019
   - "Life_Exp_2019.csv" is our main dataset that we use to make the prediction. In this dataset, we manually gather the life expectancy of 180 countries in 2019.
   - By choosing Model 4 as the main predicted model, the 9 predictors of this dataset are chosen based on the results of the Stepwise method (Sections 4.4. & 6).
   - After fitting Model 4 to predict the life expectancy of all countries, we will compare our prediction result with the self-gathered true target values of 180 countries to evaluate the accuracy of our final model.

1.3. Methodology Overview

After considering all techniques of model-building, we recognize that Regression is the most suitable method for predicting our target variable – life expectancy. Among all regressions, we choose Multiple Linear Regression (MLR) as our primary technique for the following reasons:

- The biggest advantage of linear regression is that it has the simplest estimation procedure and one of the most familiar methods to the majority. By using MLR, the model can be trained quickly, and its results can be easily interpreted. At the same time, it allows users to confidently determine which factors matter the most, which factors can be ignored, and how these factors influence each other.
- As mentioned, most features in our dataset are numerical with only 1 categorical independent variable. Therefore, it is impossible or difficult to apply some advanced or classification techniques such as K-nearest neighbors (KNN) or Neural Networks for our project, not to mention our relatively limited number of observations of 180.
- The original dataset from Kaggle contains only data of 15 years from 2000-2014. If we run Time series analysis, our data points with 15 observations are not enough to build a model and may lead to inaccurate results on prediction. In addition, among over 180 countries, we cannot choose which country to be the representative for building a model since it is inappropriate to use the model that is built from a country to fit on the prediction of another country. Hence, we decided to focus on only one year (2014) and build our model from data of all countries in that year, and running Linear regression is a preferred method for a dataset with all entries in a fixed time.

To run the MLR, our main task is choosing which variables among 15 independent variables to include in our final model. To decide that, we build 4 models as mentioned above; thus, each model has a different set of features but a similar target variable. All models use Ordinary Least Squares (OLS) – a type of linear regression which estimates the relationship between one or more independent variables and a dependent variable by minimizing the sum of the squares in residuals configured as a straight line.

Overall, we choose MLR as our primary analyzing technique for this project due to the characteristics of our dataset and the properties of this technique. Even so, we understand that it might not be the best technique. In the future, after evaluating our results from using MLR, we will work on more complicated regressions or try different techniques to improve our model.

2. Data Preprocessing & Cleaning

2.1. Removing Unnecessary Features

The original data set that we downloaded from Kaggle had 22 columns in total. We then looked into each column and its validity for its overall relevancy. From there, we identified that all of the columns were indeed relevant at this moment for our dataset since they all have some connections to life expectancy. The only column that will potentially have little impact going forward but we decided to keep in for the time being is the "year" column. This is because we are only deciding to use one year out of the many years in the dataset for the model, but it is still good to have early on for tracking purposes.

2.2. Data Cleaning: Choosing Years

Immediately upon inspecting the data, we found many inconsistencies and empty values for values across the board. Our initial idea was to use the latest year from the dataset (2015) to formulate our model, but unfortunately that was one of the years that had a glaring problem with missing values. Instead, we went back a year and decided to work with the 2014 data as it had less holes that we have to manually fill in.

2.3. Data Cleaning 2014: Removing Variables

After deciding on 2014 as the year to go with, we then went through the dataset to manually fill in data that are still missing with the help of the Internet. There, we ran into a few issues with the missing values. The first two variables we deleted were "thinness" related because we were unsure of what exactly it describes and that we could not find any of the data online. Following that, we also dropped "deaths under 5" because we already have infant mortality and it was hard to find data on it. In the end, upon reevaluating the dataset, we dropped the "Total Expenditure," "BMI," and "measles" variables because the values from the dataset were riddled with numbers that made no logical sense (i.e. over 50 BMI for half of the countries).

2.4. Data Cleaning 2014: Finishing Touches & Visualizing

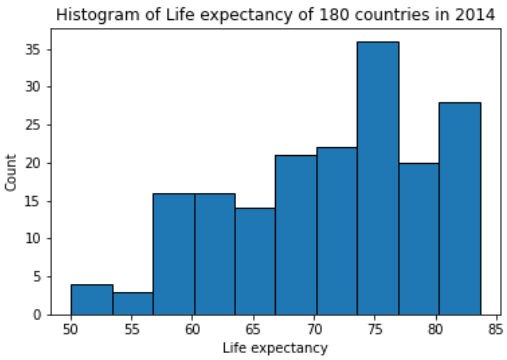


Exhibit 1: Life Expectancy 2014

The next step in our data validation process was to look over the accuracies of the data. We found that there were some inconsistencies with some of the variables such as Life Expectancy and Mortality Rate that were off by a small margin so we manually reinserted data for those values. In the process, there were 3 countries that we could not obtain sufficient data on (North Korea, Somalia, and Libya), so we decided to delete them from the dataset. In total, we have 16 columns (from 22) and 180 countries (from 183) in our final dataset for 2014 (see Appendix 1 & 2).

2.5. Convert "Status" – categorical data – to dummy variables

Our team dropped the column of "Status" from the dataset and created 2 new ones at the end of the dataset: "Status_developed" and "Status_developing." Each of them is binomial categorized with "1" tantamount to "yes" for its applicable status.

2.6. Split data into testing and training datasets

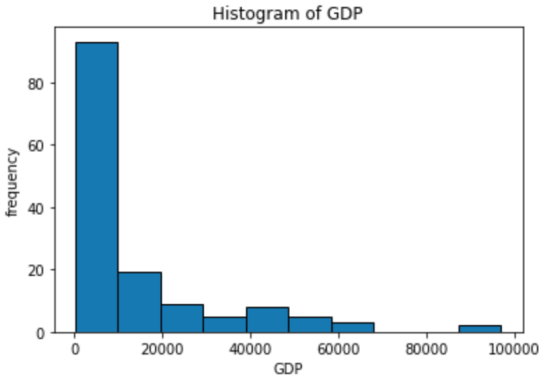
For the purpose of training models and using the trained models for predictions as well as to avoid overfitting, we split our cleaned dataset into a training and a testing dataset with the ratio of 80:20 (the test size is 0.2) by using the "train_test_split" method from the Python Sklearn package. After splitting, our 4 models have 144 observations in the training dataset and 36 observations in the testing dataset. In addition, we use this "train_test_split" method only once for all 4 models. So, these 4 models are run on the same "y_train" and "y_test" but different "X_train" and "X_test" as the number of columns in X datasets are different in each model.

.

3. Data Transformations

3.1. Run histogram plots against each individual independent variable

As the main technique of this project is utilizing linear regression analysis for prediction, one of its assumptions is that all predicted values are multivariate normal. Unlike models ran on large dataset where t-statistic for the test of the slope will converge in probability to the standard normal distribution by the law of large numbers, datasets with small number of datapoints as ours (144 observations in training), departures from normality may have an effect on the linear regression's test. In this case of non-normality, employing a transformation of X variables may result in a more powerful test. After creating some histograms, we can easily figure out there are a lot of distributions with nonnormality. For example (Exhibit 2):



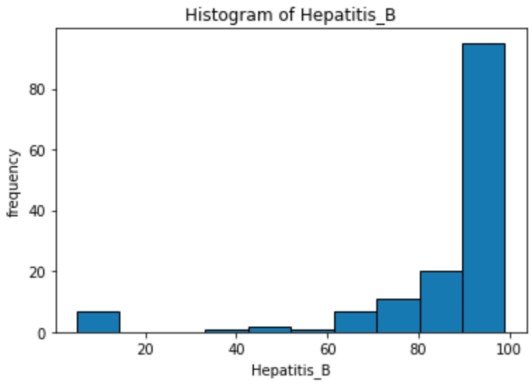


Exhibit 2: Exemplary Histograms of Variables with Skewed Distributions

3.2. Measure of skewness

To find out exactly which variables are fairly, moderately or highly skewed, we use the "agg()" method to calculate the skewness for each variable in the "X_train" and "X_test" datasets. Appendix 6 shows the skewness measurement.

In both tables of skewness in Appendix 5, we can see that most of the variables are highly skewed (skewness out of range [-1.0, 1.0]), for example, "Polio", "Diphtheria", HIV_AIDS", "Status_Developed" and some more. These variables must be changed of scale; otherwise, they would twist the linear relationship and produce errors in our model. For the remaining variables, some of them are moderately skewed (skewness within ranges [-1, -0.5] or [0.5, 1]) and the others are skewed to the left or right (observed from our histogram plots of individual variables). After all, we decided to perform transformations to all variables in our dataset.

3.3. Reduce skewness of variables with Log transformation

As all variables are with numerical types, we can perform a data transformation process in both "X_train" and "X_test" datasets . Observed from the skewness tables as well as distributions of each variable, there are several highly negative and positive skewed variables that need to be transformed. Given the advantage of making a skewed variable to be much less skewed, the logarithm transformation best helps handle non-linear variables, reduces outliers, and restores symmetry to the dataset.

Therefore, our team chooses Log transformation to reduce nonnormality. Specifically, we apply the mathematical function "numpy.log1p" for each individual variable and run it through a for loop. This function would return the natural logarithm of 1 plus input element. Mathematically, it is (`log 1 + x`) and the inverse of `log1p` is `(exp(x) - 1)`. We chose `log1p` since it is accurate even when our data is negative, very small or close to zero. Furthermore, it is not complicated to calculate the inverse function when we need to transform back.

At this step, all independent variables are transformed before we build our regression models. We make plots for each variable (Exhibit 3 - right and Appendix 6) to see how normalized each transformed variable distribution is, especially the previous highly skewed ones. We can also see that the distribution after transformation is much more symmetric than the raw one (Exhibit 3 - left) From these newly transformed distributions (**blue lines**) – appears closer to its true normality (**black lines**) – we can conclude that our dataset has reduced skewness.

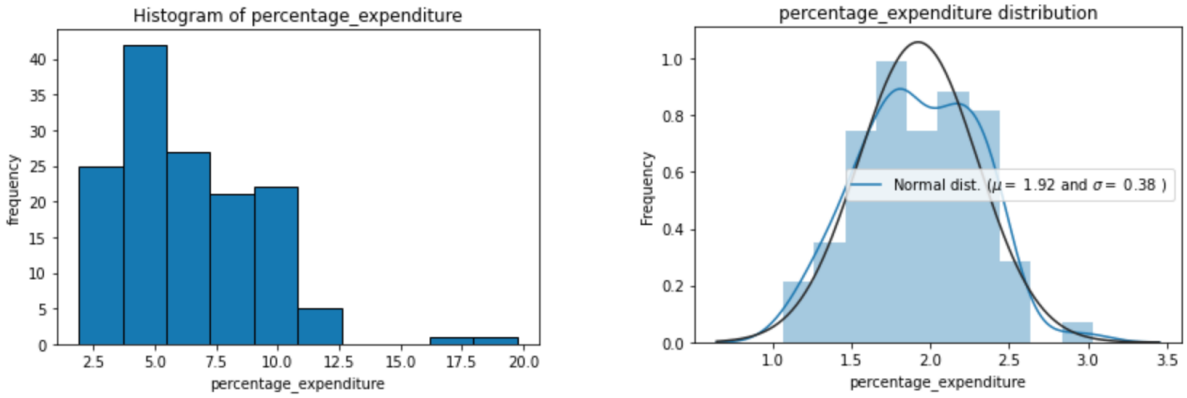


Exhibit 3: Histogram of percentage_expenditure before transformation (left)
Transformed vs. Normal distributions of percentage_expenditure (right)

4. Data Training: Linear Regression

4.1. Model 1: Run Linear Regression on all variables

Exhibit 4: Model 1 Results

OLS Regression Results

Dep. Variable:	Life_expectancy	**R-squared:**	0.957
Model:	OLS	**Adj. R-squared:**	0.953
Method:	Least Squares	**F-statistic:**	222.7
Date:	Mon, 28 Nov 2022	**Prob (F-statistic):**	4.79e-82
Time:	05:29:46	**Log-Likelihood:**	-279.08
No. Observations:	144	**AIC:**	586.2
Df Residuals:	130	**BIC:**	627.7
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Mortality	-4.5628	0.657	-6.940	0.000	-5.864	-3.262
Infant_Mortality	-4.3534	0.493	-8.835	0.000	-5.328	-3.379
Alcohol	0.4459	0.208	2.141	0.034	0.034	0.858
percentage_expenditure	0.8775	0.494	1.777	0.078	-0.100	1.855
Hepatitis_B	-0.8401	0.435	-1.930	0.056	-1.701	0.021
Polio	-0.2172	0.381	-0.570	0.570	-0.971	0.537
Diphtheria	0.9882	0.464	2.129	0.035	0.070	1.906
HIV_AIDS	-3.6241	0.439	-8.253	0.000	-4.493	-2.755
GDP	0.1313	0.355	0.369	0.713	-0.572	0.835
Population	0.0496	0.083	0.597	0.551	-0.115	0.214
Income_composition_of_resources	15.2946	7.615	2.008	0.047	0.229	30.361
Schooling	0.7623	1.286	0.593	0.554	-1.782	3.306
Status_Developed	117.2403	7.767	15.094	0.000	101.873	132.607
Status_Developing	116.2874	7.582	15.337	0.000	101.287	131.288

Omnibus:	2.431	**Durbin-Watson:**	2.289
Prob(Omnibus):	0.297	**Jarque-Bera (JB):**	2.425
Skew:	-0.309	**Prob(JB):**	0.297
Kurtosis:	2.849	**Cond. No.**	1.49e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.49e+03. This might indicate that there are strong multicollinearity or other numerical problems.

After preprocessing and transforming all data for normalization, we have 14 numerically transformed variables to be used for building models. Coming to this process, the first and primary step is running the linear regression with all these variables.

Looking at this summary table (Exhibit 4), we obtain a comparably high R-squared score in both the predicted model and fitted result (0.957). From that, it can be said our data transformation process does lead to a good result on regression. However, there are some variables that are not statistically significant (their p-value > 0.05). Therefore, to improve our model, our next step is to handle these variables.

4.2. <u>Model 2:</u> Run Linear Regression on statistically significant variables

<u>Exhibit 5</u>: Model 3 Results

OLS Regression Results

Dep. Variable:	Life_expectancy	**R-squared:**	0.954
Model:	OLS	**Adj. R-squared:**	0.952
Method:	Least Squares	**F-statistic:**	403.7
Date:	Mon, 28 Nov 2022	**Prob (F-statistic):**	1.13e-87
Time:	17:31:27	**Log-Likelihood:**	-283.87
No. Observations:	144	**AIC:**	583.7
Df Residuals:	136	**BIC:**	607.5
Df Model:	7		
Covariance Type:	nonrobust		

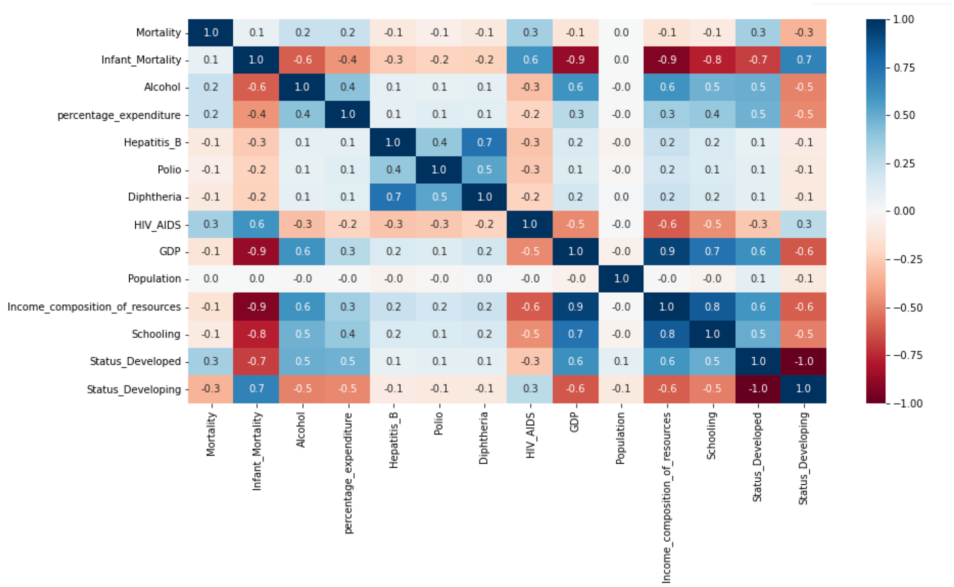
	coef	std err	t	P>|t|	[0.025	0.975]
Mortality	-4.6102	0.630	-7.322	0.000	-5.855	-3.365
Infant_Mortality	-4.3365	0.482	-8.996	0.000	-5.290	-3.383
Alcohol	0.5228	0.201	2.605	0.010	0.126	0.920
Diphtheria	0.3211	0.294	1.092	0.277	-0.260	0.902
HIV_AIDS	-3.4462	0.403	-8.543	0.000	-4.244	-2.648
Income_composition_of_resources	18.5895	4.368	4.256	0.000	9.951	27.228
Status_Developed	120.7511	5.821	20.743	0.000	109.239	132.263
Status_Developing	119.1627	5.802	20.539	0.000	107.689	130.636

Omnibus:	4.056	**Durbin-Watson:**	2.252
Prob(Omnibus):	0.132	**Jarque-Bera (JB):**	3.985
Skew:	-0.406	**Prob(JB):**	0.136
Kurtosis:	2.924	**Cond. No.**	355.

At significance level of 0.05, model 1 results reveal 6 variables that are NOT statistically significant: `['Percentage_expenditure', 'Hepatitis_B', 'Polio', 'GDP', 'Population', 'Schooling']` So, our second model is running linear regression with 8 remaining variables. From the summary table, it is noted that even removing all insignificant variables from the first model, there still exists another insignificant variable "Diphtheria". However, we obtain a lower R-squared value (0.954) (Exhibit 5) than that of the first model. It means there are some problems involved in our model that we have not considered. Thus, we want to look at the relationship and correlation between each independent variable with our target variable to figure out any problems within our model.

4.3. <u>Model 3:</u> Run Linear Regression on variables with high correlations to the target variable

<u>Exhibit 6</u>: Correlation Matrix of all variables



In order to check the variables' intertwined correlations, our team conducts a heatmap of bivariate correlation matrix (Exhibit 6) among all independent variables where the correlation coefficients are smaller than 1. The darker the shades of the cell are, the more correlated the two variables are.

a. Variables with high correlation with 'Life_expectancy':

After making the correlation matrix, to see the exact values of correlation, we use Pearson's method to sort the values of correlation in descending order based on their absolute values. From this table of correlation (Appendix 3), we have 5 variables variables whose absolute values of correlation with 'Life_expectancy' are above 0.6: `['Infant_Mortality', Income_composition_of_resources', 'GDP', 'Schooling', 'HIV_AIDS']`

b. Remove multicollinearity

Normal distribution is not the sole statistical assumption in constructing the regression models. Especially, since the features are inextricable socioeconomic factors surrounding a country's performance, there is a high chance that the independent variables are correlated to one another – hence multicollinearity.

Among the chosen variables of above 0.6 correlation, we can notice that 'Income_composition_of_resources' is highly correlated with 'GDP' (0.9), 'Infant_Mortality' (-0.9), 'Schooling' (0.8), and 'HIV/AIDS' (-0.6). At the same time, 'Infant_Mortality' is highly correlated with 'GDP', 'Schooling'. Therefore, we decided to remove 'Infant_Mortality' 'Income_composition_of_resources' and in our model to avoid multicollinearity. Hence, `['GDP', 'Schooling', 'HIV_AIDS']` is our set of variables for this model.

c. Model 3 results (Appendix 4):

Although the independent variables are highly correlated to 'Life Expectancy' and multicollinearity is also removed from the model, we still do not get as high R-squared score as previous models and even have higher errors when fitting our model to the testing dataset. It can tell that the set of high correlation variables used in this model might not be the most selective set of variables for our predicted model. Therefore, we think of utilizing the Stepwise method to obtain the best set of independent variables.

4.4. Model 4: Run Linear Regression on variables selected by Stepwise method

a. Backward Stepwise Regression:

Stepwise regression is the step-by-step iterative construction of a regression model that involves the selection of independent variables to be used in a final model. It involves adding (Forward) or removing (Backward Elimination) potential explanatory variables in testing for statistical significance after running each regression.

Since we already run the regression with all variables in our first model, to better save our time, we choose Backward instead of Forward selection and start with all possible explanatory variables. Then, we self-programmed a recursive function (Appendix 7) to rerun regression and remove the least statistically significant variables (or insignificant variable with max p-value)

one by one. The function terminates with the regression model where all variables are statistically significant, and these variables will be our ultimate selected set of variables.

b. Final selection of variables

After running Stepwise method, we obtain the final set with 9 variables for Model 4:

```
['Mortality', 'Infant_Mortality',
'Alcohol', 'Hepatitis_B',
'Diphtheria', 'HIV_AIDS',
'Income_composition_of_resources',
'Status_Developed',
'Status_Developing'].
```

c. Model 4 Results

From this summary table (Exhibit 7), we can see that the R-squared of this model is higher than the previous one (0.956). Also, there are no statistically insignificant variables in this model.

Exhibit 7: Model 4 Results

OLS Regression Results

Dep. Variable:	Life_expectancy	**R-squared:**	0.956
Model:	OLS	**Adj. R-squared:**	0.953
Method:	Least Squares	**F-statistic:**	362.5
Date:	Mon, 28 Nov 2022	**Prob (F-statistic):**	2.75e-87
Time:	17:31:30	**Log-Likelihood:**	-281.57
No. Observations:	144	**AIC:**	581.1
Df Residuals:	135	**BIC:**	607.9
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Mortality	-4.5289	0.623	-7.267	0.000	-5.761	-3.296
Infant_Mortality	-4.4599	0.480	-9.295	0.000	-5.409	-3.511
Alcohol	0.5113	0.198	2.578	0.011	0.119	0.903
Hepatitis_B	-0.8997	0.430	-2.093	0.038	-1.750	-0.050
Diphtheria	0.9584	0.421	2.278	0.024	0.126	1.791
HIV_AIDS	-3.5618	0.402	-8.854	0.000	-4.357	-2.766
Income_composition_of_resources	17.8563	4.329	4.125	0.000	9.295	26.418
Status_Developed	123.1396	5.863	21.005	0.000	111.545	134.734
Status_Developing	121.7366	5.861	20.769	0.000	110.145	133.329

Omnibus:	5.019	**Durbin-Watson:**	2.286
Prob(Omnibus):	0.081	**Jarque-Bera (JB):**	4.773
Skew:	-0.444	**Prob(JB):**	0.0920
Kurtosis:	3.076	**Cond. No.**	453.

5. Model Results Evaluation

a. Observations from comparison of all models' statistics. Model 4 is the final chosen model

Exhibit 8: Summary Result for the 4 Linear Regression Models

	R^2 score	**Mean absolute error**	**Mean square error**	**Median absolute error**	**Explained variance score**
Model 1: all variables	0.957	1.368	2.885	1.155	0.957
Model 2: statistical significance	0.951	1.475	3.225	1.399	0.951
Model 3: high correlation	0.860	2.502	9.369	2.252	0.860
Model 4: Stepwise	0.953	1.455	3.117	1.405	0.954

Looking at the results table (Exhibit 8), Model 1 has the highest 'R-squared Score' which indicates that the model's actual life expectancy observations move in line the closest with the prediction line among all models. It is followed respectively by Model 4, 2, and 3. The same goes for 'Explained Variance Score' where higher percentages of explained variance indicates a better-fitting model. The preferred ranking is applied the same for 'Mean Absolute Error' and 'Mean Square Error' from the lowest order – as the lower the error index the better. Only with 'Median Absolute Error' that Model 2's appears to be lower than Model 4's.

Overall, Model 1 gives off the best statistics and the least error (**red** highlighted). Yet the nature of this model is that it contains statistically insignificant variables (p-value larger than 0.05). Therefore, if Model 1 is used on a larger dataset, i.e. with thousands observations, it may not give off the lowest error among the 4.

Model 2 also gives a comparably good result for prediction (high R-squared score and second lowest median absolute error). However, the set of variables in this model is obtained by running regression only once; so, there may still exist other insignificant variables if we continue running regression for this model. To the contrary, the set of variables in Model 4 is obtained by running regression multiple times until there are no insignificant variables. Therefore, theoretically, Model 4 must be more accurate and give better results on predictions than Model 2.

That is to say, using the stepwise method in Model 4 which excludes all those mentioned and gives off the second-best results (**blue** highlighted) is the ultimate option for this project.

b. Test all models against the US2021 dataset (Exhibit 9)

Having all 4 models tested against the actual US2021 dataset, Model 4 now is the closest to the actual record of US2021 life expectancy with 0.108 difference (0.14%), followed by Model 2, 1, and 3. It further strengthens the use of Model 4 in this project instead of other models.

Actual US2021 Life_expectancy	**Model 1**	**Model 2**	**Model 3**	**Model 4**
78.99*	79.887	**79.274**	79.296	**79.098**
% difference	+1.14%	**+0.36%**	+0.39%	**+0.14%**

Exhibit 9: US2021 Results from All 4 Models.
*(*78.99 is the UN projected life expectancy without COVID & 76.1 with the inclusion of COVID).*

Our final model can be interpreted as:

`Life_expectancy = -4.5289 * Mortality - 4.4599 * Infant_Mortality + 0.5113 * Alcohol - 0.8997 * Hepatitis_B + 0.9584 * Diphtheria - 3.5618 * HIV_AIDS + 17.8563 * Income_composition_of_resources + 123.1396 * Status_Developed + 121.7366 * Status_Developing`

6. Final Result – Run Model 4 Against 2019's Data

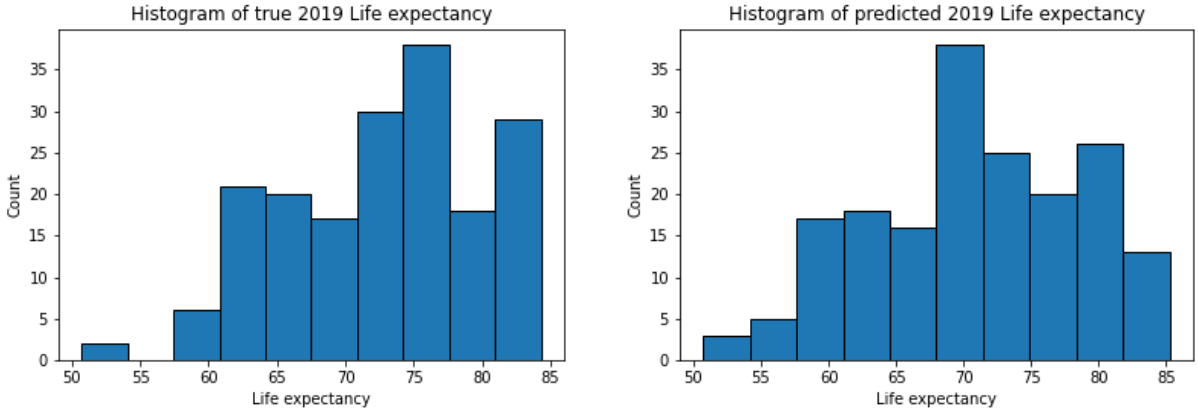


Exhibit 10: Histograms of true vs predicted 2019 life expectancy values

To truly put the Model 4 to the test, we decided to use it to predict the life expectancy of our 180 countries in 2019. From the results, we reached a R-squared value of 0.634 and a variance score of 0.703. While it does not boast the highest confidence in its correlation and predictability, we can nevertheless conclude that it will predict life expectancy (in 2019) with a mean absolute error of 3.198 and a median absolute error of 2.278. Overall as evident in Exhibit 10, our model shifted more to the left compared to the actual life expectancy values of 2019. This can mainly be attributed to the scarcity of information available on some of the countries that are on the lower end, which contributed to potential inaccuracy and inconsistency of the data collected.

7. Conclusion

7.1. Practical Application of Concepts from BA305

The very first thing that was taught in BA305 is that when handling data, data analysts have to dedicate the majority of their time cleaning and preprocessing data before actual usage. Our project was no exception; we spent most of our time cleaning and preprocessing data. As there is only limited access to English-published reports of all countries for 2014 and 2019, it required us to spend more than 80% of our time processing data despite our relatively small dataset. The outcome of the cleaned data is fruitful as we were able to fill in all necessary null data points for 180 out of 183 countries given.

There are practical topics that go beyond the classroom along the way such as normality assumption of linear regression. Since our dataset is small with only 144 observations in training, it is suggested to reduce each independent variable's skewness for the best model outcome – using Log transformation. Or applying the Stepwise method as another regression model that works better for the dataset. All tools added on are nothing over-advanced but essential applications as long as they are useful to reach desirable results.

7.2. Potential Areas of Improvements

After going through our whole procedure and analysis for the report, our team recognized some areas of improvements. First, our team will learn more about the data transformation and test out multiple data transformation methods (Standardize, Square root, Box-cox, etc.) to pick out the one that is suitable for our dataset, further reduce skewness, and best used for our training model.

Our final result may have overlooked Model 4 as it still contains multicollinearity. 'Infant_Mortality' and 'Income_composition_of_resources' have a high correlation of 0.9. Yet it is noteworthy that the final decision may not be altered as Model 4 is still the best fit in this scale. We understand that our final model might not be the best model in a larger dataset with multicollinearity in place.

Besides Model 1 that runs on all raw variables, both Model 2 and 4 with multicollinearity may be overfitting which the algorithm unfortunately cannot perform accurately against unseen data, defeating the whole project's purpose. We recognized those when it was too short-noticed for improvement directly. In the scenario that there will be an improvement with extended time, Model 5 and 6 may cover Model 2 and 4's multicollinearity and return better results.

7.3. Project Usages in the Serving the Purpose

As previously mentioned in the introduction, our purpose is to predict life expectancy of countries so that governments are more prepared to allocate resources in adjustment to the increase in the elderly population. However, a byproduct of the project also showed us the different impactful variables such as disease vaccinations to the Human Development Index (HDI) with regards to income that are affecting different country's life expectancies.

To expand on our purpose, we would also like to advocate avenues where organizations like the UN can allocate more funding in order to elevate the living standards of people and their life expectancy. Going forward with this information, we can raise awareness for organizations that support the variables that we've deemed important in our equation, in addition to having a model to assist with countries allocation of funds for the elderly.

Works cited

1. Blankespoor, Brian, Nishant Yonzan, John Baffes, and Calogero Carletto. "World Bank Open Data." Data, November 28, 2022. https://data.worldbank.org/.
2. Central Intelligence Agency. Central Intelligence Agency. Accessed November 29, 2022. https://www.cia.gov/the-world-factbook/field/hiv-aids-adult-prevalence-rate/country-comparison.
3. Erik Marsja. "How to Use Square Root, Log, & Box-Cox Transformation in Python." Erik Marsja, November 20, 2020. https://www.marsja.se/transform-skewed-data-using-square-root-log-box-cox-methods-in-python/?fbclid=IwAR3ta_xPtR7qtHLvOKUNirNwuCTzIxfJSX93cCMFvoaVrLJOorwr_Dq7xl0.
4. Kwok, Ryan. "Stepwise Regression Tutorial in Python." Medium. Towards Data Science, March 9, 2021. https://towardsdatascience.com/stepwise-regression-tutorial-in-python-ebf7c782c922.
5. "Numpy.log1p." numpy.log1p - NumPy v1.23 Manual. Accessed November 29, 2022. https://numpy.org/doc/stable/reference/generated/numpy.log1p.html.
6. Oecd. "OECD Statistics." OECD Statistics. Accessed November 29, 2022. https://stats.oecd.org/.
7. "The Statistics Portal." Statista. Accessed November 29, 2022. https://www.statista.com/.
8. "Undata." United Nations. United Nations. Accessed November 29, 2022. https://data.un.org/.
9. "Worlddata: The World in Numbers." Worlddata.info. Accessed November 29, 2022. https://www.worlddata.info/.

Appendix

Appendix 1: Condensed overview of Cleaned Dataset 2014

```
Int64Index: 180 entries, 0 to 179
Data columns (total 14 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   Status                           180 non-null    object
 1   Life_expectancy                  180 non-null    float64
 2   Mortality                        180 non-null    float64
 3   Infant_Mortality                 180 non-null    float64
 4   Alcohol                          180 non-null    float64
 5   percentage_expenditure           180 non-null    float64
 6   Hepatitis_B                      180 non-null    int64
 7   Polio                            180 non-null    int64
 8   Diphtheria                       180 non-null    int64
 9   HIV_AIDS                         180 non-null    float64
 10  GDP                              180 non-null    float64
 11  Population                       180 non-null    int64
 12  Income_composition_of_resources  180 non-null    float64
 13  Schooling                        180 non-null    float64
dtypes: float64(9), int64(4), object(1)
memory usage: 21.1+ KB
```

Appendix 2: *Description of original dataset's 16 features of: ['Country', 'Year', 'Status', 'Life expectancy ', ' Mortality', 'Infant Mortality', 'Alcohol', 'percentage expenditure', 'Hepatitis B', 'Polio', 'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population', 'Income composition of resources', 'Schooling']. All are numerical variables except 'Country' and 'Status']*

Country	Name of the country	Schooling	Number of years of schooling
Year	The year the data is on	Population	Population of the country
Status	The economic status of the country denoted by "developing" or "developed"	Diphtheria	Diphtheria tetanus toxoid and pertussis immunization coverage for 1-year-olds (%)
Life Expectancy	The average years one is expected to live in that country	HIV/AIDS	Deaths per 1,000 live births HIV/AIDS
Mortality	The probability of dying (both sexes) of a person per 1,000	Income Composition of Resources	Human Development Index in terms of income composition of resources
Infant Mortality	Number of Infant Deaths per 1,000 population (ages 0-1)	Polio	Polio immunization coverage among 1-year-olds (%)
Alcohol	Alcohol, recorded per capita in liters	GDP	Gross Domestic Product per capita (in USD)

Percentage Expenditure	Expenditure on healthcare as a percentage of GDP per capita (%)	Hepatitis B	Hepatitis B immunization coverage among 1-year-olds (%)

Appendix 3: Model 3 - Correlation table

```
Life_expectancy                   1.000000
Infant_Mortality                 -0.930508
Income_composition_of_resources   0.917236
GDP                               0.843398
Schooling                         0.759171
HIV_AIDS                         -0.748277
Status_Developing                -0.562744
Status_Developed                  0.562744
Alcohol                           0.553946
percentage_expenditure            0.390019
Mortality                        -0.289338
Diphtheria                        0.260493
Hepatitis_B                       0.254861
Polio                             0.237752
Population                        0.015108
Name: Life_expectancy, dtype: float64
```

Appendix 4: Model 3 Results Summary

OLS Regression Results

Dep. Variable:	Life_expectancy	**R-squared (uncentered):**	0.997
Model:	OLS	**Adj. R-squared (uncentered):**	0.997
Method:	Least Squares	**F-statistic:**	1.792e+04
Date:	Tue, 29 Nov 2022	**Prob (F-statistic):**	8.17e-182
Time:	07:22:48	**Log-Likelihood:**	-391.16
No. Observations:	144	**AIC:**	788.3
Df Residuals:	141	**BIC:**	797.2
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Schooling	19.5098	1.147	17.013	0.000	17.243	21.777
HIV_AIDS	-4.2959	0.631	-6.813	0.000	-5.542	-3.049
GDP	2.5174	0.334	7.548	0.000	1.858	3.177

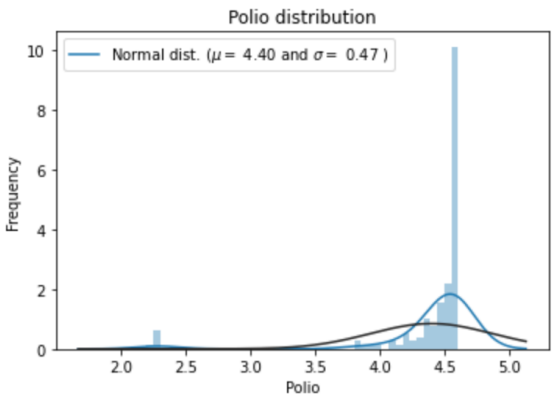
Omnibus:	20.232	**Durbin-Watson:**	2.092
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	33.599
Skew:	0.689	**Prob(JB):**	5.06e-08
Kurtosis:	4.924	**Cond. No.**	36.3

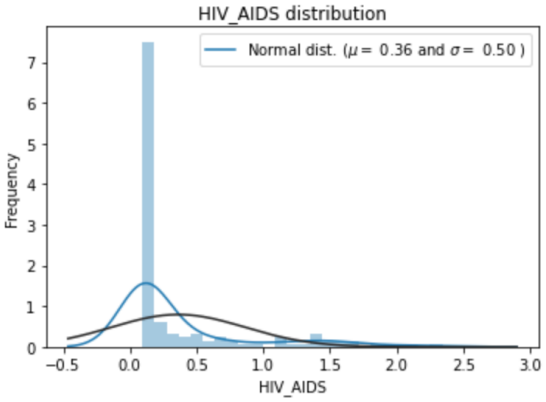
Appendix 5: Tables of Skewness for variables in X_train and X_test datasets

	skew	kurtosis
Mortality	-0.536379	1.540261
Infant_Mortality	-0.148636	-1.102083
Alcohol	0.336722	-1.656497
percentage_expenditure	-0.004885	-0.383572
Hepatitis_B	-3.777857	13.644566
Polio	-3.862717	14.749296
Diphtheria	-3.722573	13.262422
HIV_AIDS	2.071785	3.418243
GDP	0.045424	-0.907076
Population	-0.317158	0.045303
Income_composition_of_resources	-0.418374	-0.872826
Schooling	-1.008368	1.632776
Status_Developed	1.504883	0.269287
Status_Developing	-1.504883	0.269287

	skew	kurtosis
Mortality	-1.308631	3.379055
Infant_Mortality	0.140990	-1.262723
Alcohol	0.487145	-1.573131
percentage_expenditure	-0.339972	0.247974
Hepatitis_B	-2.728965	7.335900
Polio	-3.177055	11.863974
Diphtheria	-2.768442	7.505101
HIV_AIDS	2.234068	3.785353
GDP	0.133997	-0.646205
Population	0.048628	-0.038100
Income_composition_of_resources	-0.585621	-0.708818
Schooling	-0.477736	-0.653402
Status_Developed	1.867188	1.572266
Status_Developing	-1.865234	1.568359

Appendix 6: Distributions of some variables that originally have high skewness





Appendix 7: Function of running Stepwise

```
x = ['Mortality', 'Infant_Mortality', 'Alcohol', 'percentage_expenditure',
     'Hepatitis_B', 'Polio', 'Diphtheria', 'HIV_AIDS', 'GDP', 'Population',
     'Income_composition_of_resources', 'Schooling', 'Status_Developed',
     'Status_Developing']

def stepwise(x):
    x_train = X_train[x]
    results = sma.OLS(y_train, x_train).fit()

    pvalue = results.pvalues.values
    var = results.pvalues.index

    max_p = pvalue.max()
    if max_p > 0.05:
      index = np.where(pvalue == max_p)
      removed = var[index[0]]
      x.remove(removed[0])
      stepwise(x)

    else:
      print(results.summary())
      print('Selected variables: ', x)

stepwise(x)
```