Education and the Economy

IDC4140 Final Report

Cole McGuire
Computer Science
Florida State University
Tallahassee Florida United States
cjm20dy@fsu.edu

Reece Gabbett
Computer Science
Florida State University
Tallahassee Florida United States
rmg19@fsu.edu

Kirsten Blair
Computer Science
Florida State University
Tallahassee Florida United States
kmb22@fsu.edu

Austin Miller
Computer Science
Florida State University
Tallahassee Florida United States
ajm19x@fsu.edu

Davone Simmons
Economics, Statistics
Florida State University
Tallahassee Florida United States
ds20b@fsu.edu

ABSTRACT

The purpose of this project is to answer the research question which is "What effect does the level of education have on economic growth?" This question is very important as it can enhance the importance of education to policy makers and possibly even give them further incentive for improving education quality. In order to accomplish this goal, methods were used including data cleansing, Exploratory Data Analysis (EDA), visualization with Matplotlib, linear regression, polynomial regression, logarithmic transformation for linear regression in order to flatten the data, binning to convert numerical data to categorical, and logistic regression. The P-Value is used to test if an attribute is statistically significant. R-Squared is also used, which is a measure of how well the dependent variable is represented by the independent variables [6]. Gross Domestic Product (GDP) per capita is used as a metric of the economy. In the project it was determined that the average years of schooling, the scholarship dollars awarded, and the scholarship headcount are statistically significant; while the average university ranking of a given country is not statistically significant. This research is complex due to the many attributes that make up the economy, and determining the features that result in quality education. This complexity is the main limitation of this project.

INTRODUCTION

Education is considered to be one of the key factors in investing in human capital. Human Capital is defined as a collective set of skills, knowledge, and resources a person can provide to a company, workspace, and or community that results in economic prosperity [20]. Higher education levels are desired by employers and individuals as both believe there is an economic benefit. Employers value higher educated individuals due to increased productivity and knowledge, while individuals want to be educated for better job opportunities and increased personal income. Previous research has shown that higher levels of education are associated with productivity, competitiveness, and innovation, which all have a positive economic result in a region [16].

The aim of this research is to find if the level of education has an impact on GDP levels for both the United States and other countries, then use Data Science techniques to find the results that will be compared between countries.

Education levels in this project are measured by years of schooling, scholarship dollars awarded, scholarship headcount, and average university ranking. The education levels of an area are heavily dependent upon the education system in place for that area, which is shaped through the funding and economic factors, government involvement, and cultural factors of that society.

The motivation for this research is to find the links between education levels and the GDP for a country. Understanding these two variables can allow for individuals to make more informed career decisions for personal economic gain as well as government officials and policymakers to invest appropriate funding into better education systems that will lead to economic development and growth in a country or region.

RELATED WORK

When trying to determine the ideal methodology to approach the research question, it is important to take into consideration a plethora of state-of-the-art methods as well as other analyses that are similar to this one. There are a number of different methods that are relevant to discovering the relationship between our variables, education, and GPD per capita, such as

correlation analysis, machine learning, and regression analysis. In order to conclude which method is ideal, it is important to look at each one analytically as well as other similar research and the methods that were imposed.

Correlation analysis is a method that measures the strength and direction of the linear relationship between two variables. It can be performed using a variety of submethods, such as Pearson's correlation coefficient, Spearman's rank correlation coefficient, or Kendall's tau correlation coefficient. These measures range from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no correlation [17]. This could be used to examine the correlation between the level of education and economic growth. If the correlation is significant, it can be interpreted as evidence of a linear relationship between education and economic growth. However, correlation does not imply causation, and there may be other factors that affect both education and economic growth, such as cultural, political, or natural resources. While this could be the ideal methodology for a simpler problem with fewer variables to take into account, it is not ideal for a problem that is this complex.

Another method to consider would be to take a machine-learning approach. Machine learning is a type of artificial intelligence that enables computers to learn from data without being explicitly programmed. It could predict economic growth based on various education-related factors. The available machine learning algorithms could be used to identify the most important features that contribute to economic growth, such as literacy rates, enrollment rates, education expenditure, and human capital accumulation. Of these available algorithms, there are a plethora to choose from such as decision trees, random forests, and neural networks, depending on the data characteristics [18]. Machine learning seems ideal initially because it can help identify the potential causal effects of education on economic growth and can provide insights for policymakers on the most effective ways to invest in education. While seemingly ideal at the surface, this approach would likely be too daunting as some machine learning models can be complex and difficult to interpret, making it hard to understand how the model is making the predictions. As well as a multitude of possible technical problems such as overfitting, it may be best to look at other methods.

The final method to look at is regression analysis. Regression analysis is a statistical method used to examine the relationship between two or more variables, and it is commonly used to study relationships related to both education and economic growth. Various regression techniques can be used, such as linear regression, logistic regression, and polynomial regression. Regression analysis can provide insights into the direction and strength of the relationship between education and economic growth and can help policymakers understand the potential benefits of investing in education. When looking at research done in the past with similar research questions, it is clear that most if not all are using regression analysis when analyzing education and the economy; especially when relating the two. One example carried out by the Institute of Labor Economics had an almost identical research question, regarding the relationship between education and economic growth and the methodology used in this case was *"a nonparametric local-linear regression estimator and a nonparametric variable relevance test to conduct a rigorous and systematic search for significance of mean years of schooling by examining five of the most comprehensive schooling databases*" [19] which led to the conclusion that regression analysis is perhaps the ideal methodology to approach this question.

Within the scope of regression analysis, there are three different submethods that were used with this particular research question: logistic regression, linear regression, and polynomial regression. Logistic regression is a type of regression analysis used to model binary or categorical dependent variables. The dependent variable is dichotomous, with only two possible outcomes. It uses a logistic function to model the relationship between the independent and dependent variables. The output of the model is a probability value between 0 and 1. Additionally, linear regression is a type of regression analysis used to model the relationship between a dependent variable and one or more independent variables. The dependent variable is continuous, with a range of possible outcomes. It employs a straight line to model the relationship between the independent and dependent variables. The output of the model is a predicted value of the dependent variable. Finally, there is polynomial regression which is a type of regression analysis that models the relationship between a dependent variable and one or more independent variables as an nth-degree polynomial function. This allows us to capture nonlinear relationships between the variables, which may not be captured by linear regression models.

DATA DESCRIPTION AND ANALYSIS

The Datasets

The following is a description of each dataset used for the data analysis in this project:

- Average years of schooling vs GDP per capita
  - This dataset comes from "Our World in Data," a website containing many datasets about important world data. This specific dataset contains the attributes country name, country code, year, GDP per capita, average years of schooling, Purchasing Power Parity (PPP), population, and continent.
- Countries GDP from years 1960 - 2020

- This dataset comes from Kaggle. The dataset is a subset of data from the World Bank national accounts data, and the OECD National Accounts data files [4]. It contains the attributes of the country name, country code, and GDP from the years 1960 to 2020.

- Scholarship money awarded per year in New York
  - This dataset comes from Kaggle. It includes the number of scholarship award recipients and dollar amounts by the Tuition Assistance Program (TAP) in New York from 2009 to 2018 [13]. The scholarship data of New York in this dataset is used as a subset to represent that of the United States. The attributes of this dataset include the academic year, TAP college code, federal school code, TAP college name, TAP sector group, scholarship headcount, scholarship Full Time Equivalents (FTE), and scholarship dollars.
- Global ranking of top universities
  - This dataset comes from Kaggle. The dataset represents the annual publication of university rankings by the Times Higher Education magazine [9]. The dataset includes the attributes university name, country name, ranking by year, average ranking, and ranking based on average for the top 1000 universities.

Average years of schooling vs GDP per capita

Before analyzing this dataset, it needed to be cleaned. This involved dropping null values, selecting only the necessary columns/attributes, and renaming the column names to make them more clear, human readable, and easy to work with. Observe the following code snippet and output, which can be used to help replicate the results.

Code Snippet:

```python
# Read the CSV file into a pandas dataframe
df = pd.read_csv("./datasets/average-years-of-schooling-vs-gdp-per-capita.csv")
print(df)

# Clean data
# Drop rows with missing values
df = df.dropna()

# Select only the columns we need for our plot
df = df[['Entity', 'Year', 'Average Total Years of Schooling for Adult Populatic

# Rename the columns for easier use
df.columns = ['country', 'year', 'schooling', 'gdp']
```

This transformed the dataset in the following way:

Original dataset:

	Entity	Code	Year	...	GDP per capita, PPP (constant 2017 international $)	Population (historical estimates)	Continent
0	Abkhazia	OWID_ABK	2015	...	NaN	NaN	Asia
1	Afghanistan	AFG	1870	...	NaN	4142928.0	NaN
2	Afghanistan	AFG	1875	...	NaN	4247357.0	NaN
3	Afghanistan	AFG	1880	...	NaN	4354370.0	NaN
4	Afghanistan	AFG	1885	...	NaN	4464010.0	NaN
...	...	...	...	...	...	...	...
58171	Zimbabwe	ZWE	1987	...	NaN	9277484.0	NaN
58172	Zimbabwe	ZWE	1988	...	NaN	9568745.0	NaN
58173	Zimbabwe	ZWE	1989	...	NaN	9846352.0	NaN
58174	Zimbabwe	ZWE	2021	...	NaN	15993525.0	NaN
58175	Åland Islands	ALA	2015	...	NaN	NaN	Europe

Dataset after data cleansing:

	country	year	schooling	gdp
50	Afghanistan	2015	3.6	2068.265869
641	Albania	2015	9.7	11878.454102
900	Algeria	2015	7.9	11696.950195
1499	Angola	2015	5.0	8036.411133
1882	Antigua and Barbuda	2015	9.2	18594.544922
...	...	...	...	...
55709	Uzbekistan	2015	11.4	6401.115234
55963	Vanuatu	2015	6.8	2915.700439
56716	Vietnam	2015	8.0	6438.259766
57705	Zambia	2015	6.9	3443.553223
57965	Zimbabwe	2015	8.2	3707.622559

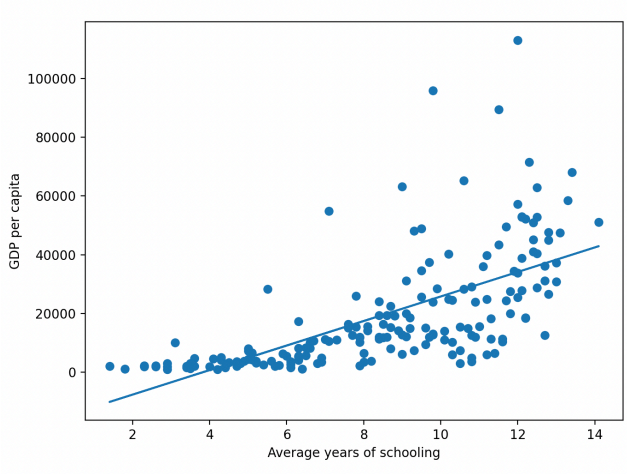
Next, the Exploratory Data Analysis (EDA) was performed. This involved using methods included with PANDAS, along with visualizations using matplotlib and numpy. The description and types were found along with a scatterplot with trendline of the years of schooling vs GDP per capita. Observe the following code snippets and the results.

Code snippet:

```python
# Exploratory data analysis (EDA)
print("\nThe dataframe: \n", df)
print("\nThe dataframe's description: \n", df.describe())
print("\nThe dataframes types: \n", df.dtypes)


# Create a scatter plot of GDP vs schooling
plt.scatter(df.schooling, df.gdp)
# Set the x and y labels
plt.xlabel('Average years of schooling')
plt.ylabel('GDP per capita')

# Get the trendline coefficients
z = np.polyfit(df.schooling, df.gdp, 1)
# Get the polynomial of the trendline
p = np.poly1d(z)
# Plot the trendline
plt.plot(df.schooling, p(df.schooling))

# Show the plot
plt.show()

```

Output:

```
The dataframe:
                  country  year  schooling          gdp
50            Afghanistan  2015        3.6   2068.265869
641               Albania  2015        9.7  11878.454102
900               Algeria  2015        7.9  11696.950195
1499               Angola  2015        5.0   8036.411133
1882   Antigua and Barbuda  2015        9.2  18594.544922
...                    ...   ...        ...           ...
55709           Uzbekistan  2015       11.4   6401.115234
55963              Vanuatu  2015        6.8   2915.700439
56716              Vietnam  2015        8.0   6438.259766
57705               Zambia  2015        6.9   3443.553223
57965             Zimbabwe  2015        8.2   3707.622559

[181 rows x 4 columns]

The dataframe's description:
          year   schooling           gdp
count   181.0  181.000000     181.000000
mean   2015.0    8.491713   19441.335688
std       0.0    3.091384   19973.449943
min    2015.0    1.400000     825.205688
25%    2015.0    6.100000    4488.802734
50%    2015.0    8.800000   11973.353516
75%    2015.0   11.200000   27797.058594
max    2015.0   14.100000  113182.726562

The dataframes types:
 country       object
year           int64
schooling    float64
gdp          float64
dtype: object
```

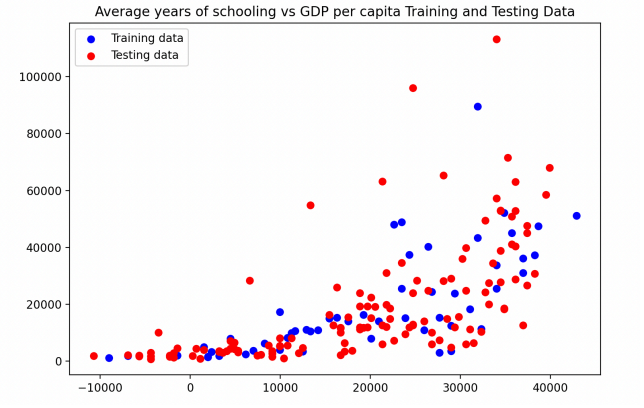


Next , to begin the exploration of the effect of years of schooling on GDP, we completed a linear regression. In order to do so, any categorical variables, such as the country name, were dropped, then the data was split into X and y, or independent and dependent variables respectively. The y variable is the GDP and the independent variables are all other attributes, which in this case is the year, and years of schooling. Next, train_test_split from the scikit-learn module is used to split the data to X and y training and testing sets. After the data is fit using a linear regression, the intercepts of the regression equation, coefficients of the regression equation, root-mean-square-error (RMSE), mean-square-error (MSE), and R-Squared value were analyzed. The linear regression was also plotted. Observe.
Code Snippet:

```python
# Linear regression
# Drop categorical data
print("LINEAR REGRESSION")
df = df.drop(['country'], axis=1)

# Split the data into training and testing sets
# Get the x and y values
y = df['gdp']
X = df.drop(['gdp'], axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.7, train_size=0.3)

# Fit the training data using a strait line
model = LinearRegression()
model.fit(X_train, y_train)

print("\nIntercepts of regression equation: ", model.intercept_)
coefficients = pd.DataFrame(model.coef_, X.columns, columns=['Coefficient'])
print("\nCoefficients of regression equation: \n", coefficients)

# Plot the training and testing data
predictionsTrain = model.predict(X_train)
predictionsTest = model.predict(X_test)
plt.scatter(predictionsTrain, y_train, color='blue', label='Training data')
plt.scatter(predictionsTest, y_test, color='red', label='Testing data')
plt.title('Average years of schooling vs GDP per capita Training and Testing Data')
plt.legend()
plt.show()

# now calculate RMSE and MSE
rmse = np.sqrt(metrics.mean_squared_error(y_test, predictionsTest))
mse = metrics.mean_squared_error(y_test, predictionsTest)
r2 = metrics.r2_score(y_test, predictionsTest)
print("\nRMSE: ", rmse)
print("\nMSE: ", mse)
print("\nR-squared: ", r2)
```

Output:

```
LINEAR REGRESSION

Intercepts of regression equation:  -11514.710780993893

Coefficients of regression equation:
            Coefficient
year           0.000000
schooling   3627.755957

RMSE:  16739.060720423622

MSE:  280196153.80202895

R-squared:  0.36361962350690946
```



P Values:
Code snippet:

```
159   # Get p values
160   model = sm.OLS(y_train, sm.add_constant(X_train)).fit()
161   print("\n P Values: \n", model.pvalues)
```

Output:

```
 P Values:
 year          0.110202
schooling      0.000058
dtype: float64
```

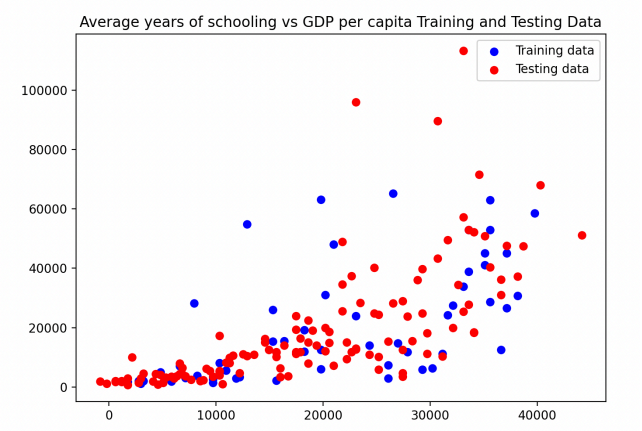
It was then observed that a polynomial regression may better fit this dataset. A polynomial regression is executed using the PolynomialFeatures method from the preprocessing part of the scikit-learn module. This PolynomialFeatures method was initialized with a degree of 2, and then used to fit the X_train and X_test subsets from the original linear regression. Next these polynomial X_train andX_test subsets are fit using a linear regression. Similarly, the intercepts of the regression equation, coefficients of the regression equation, RMSE, MSE, and R-Square values are evaluated, along with a graph of the polynomial regression.

Code Snippet:

```
84   # Polynomial regression
85   print("\nPOLYNOMIAL REGRESSION")
86   poly = PolynomialFeatures(degree=2)
87   X_train_poly = poly.fit_transform(X_train)
88   X_test_poly = poly.transform(X_test)
89
90   model = LinearRegression()
91   model.fit(X_train_poly, y_train)
92
93   print("\nIntercepts of regression equation: ", model.intercept_)
94   coefficients = pd.DataFrame(model.coef_, poly.get_feature_names_out(X.columns), columns=['Coefficient'])
95   print("\nCoefficients of regression equation: \n", coefficients)
96
97   # Plot the training and testing data
98   predictionsTrain = model.predict(X_train_poly)
99   predictionsTest = model.predict(X_test_poly)
100  plt.scatter(predictionsTrain, y_train, color='blue', label='Training data')
101  plt.scatter(predictionsTest, y_test, color='red', label='Testing data')
102  plt.title('Average years of schooling vs GDP per capita Training and Testing Data')
103  plt.legend()
104  plt.show()
105
106  # now calculate RMSE and MSE
107  rmse = np.sqrt(metrics.mean_squared_error(y_test, predictionsTest))
108  mse = metrics.mean_squared_error(y_test, predictionsTest)
109  r2 = metrics.r2_score(y_test, predictionsTest)
110  print("\nRMSE: ", rmse)
111  print("\nMSE: ", mse)
112  print("\nR-squared: ", r2)
```

Output:

```
POLYNOMIAL REGRESSION

Intercepts of regression equation:  -187.99444348294128

Coefficients of regression equation:
                 Coefficient
1               0.000000e+00
year            3.035350e-13
schooling      -1.041607e-04
year^2          0.000000e+00
year schooling -2.098838e-01
schooling^2     2.827382e+02

RMSE:  15580.961777390161

MSE:  242766369.9084932

R-squared:  0.4281794837620567
```



P Values:

Code snippet:

```
159   # Get p values
160   model = sm.OLS(y_train, sm.add_constant(X_train)).fit()
161   print("\n P Values: \n", model.pvalues)
```

Output:

```
 P Values:
 year          0.110202
schooling      0.000058
dtype: float64
```

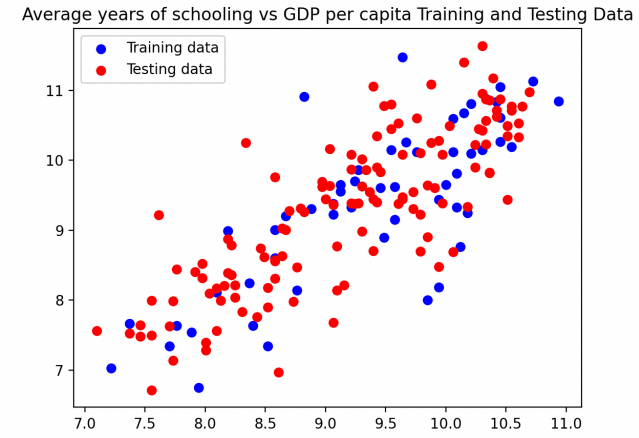
Since the polynomial regression seemed to represent the relationship between the independent and dependent variables better than the linear regression, given the r-squared values, it was hypothesized that a linear regression may perform better if first the data is flattened using logarithmic transformation. In order to accomplish this, the natural log of the GDP was taken using the numpy log method. Then, the same process as the first linear regression was executed, except with the flattened GDP values as the dependent variable. Observe.

Code snippet:

```python
# Use logarithmic transformation for linear regression
print("\nLOGARITHMIC TRANSFORMATION FOR LINEAR REGRESSION")
# Take the natural logarithm of the GDP variable
df['ln_gdp'] = np.log(df['gdp'])

# Split the data into training and testing sets
# Get the x and y values
y = df['ln_gdp']
X = df.drop(['gdp', 'ln_gdp'], axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.7, train_size=0.3)

# Fit the training data using a strait line
model = LinearRegression()
model.fit(X_train, y_train)

print("\nIntercepts of regression equation: ", model.intercept_)
coefficients = pd.DataFrame(model.coef_, X.columns, columns=['Coefficient'])
print("\nCoefficients of regression equation: \n", coefficients)

# Plot the training and testing data
predictionsTrain = model.predict(X_train)
predictionsTest = model.predict(X_test)
plt.scatter(predictionsTrain, y_train, color='blue', label='Training data')
plt.scatter(predictionsTest, y_test, color='red', label='Testing data')
plt.title('Average years of schooling vs GDP per capita Training and Testing Data')
plt.legend()
plt.show()

# now calculate RMSE and MSE
rmse = np.sqrt(metrics.mean_squared_error(y_test, predictionsTest))
mse = metrics.mean_squared_error(y_test, predictionsTest)
r2 = metrics.r2_score(y_test, predictionsTest)
print("\nRMSE: ", rmse)
print("\nMSE: ", mse)
print("\nR-squared: ", r2)
```

Output:

```
LOGARITHMIC TRANSFORMATION FOR LINEAR REGRESSION

Intercepts of regression equation:  6.676240435514124

Coefficients of regression equation:
            Coefficient
year            0.00000
schooling       0.30222

RMSE:  0.6568480430509388

MSE:  0.43144935165984794

R-squared:  0.6551367686056332
```



P Values:

Code snippet:

```python
# Get p values
model = sm.OLS(y_train, sm.add_constant(X_train)).fit()
print("\n P Values: \n", model.pvalues)
```

Output:

```
 P Values:
 year          9.699278e-36
schooling      1.144891e-18
dtype: float64
```

Finally, in order to test using a different type of regression in an attempt to enhance the representation of the data, a logistic regression was used. However, the GDP values are not the proper type for a logistic regression. Therefore, binning was used to split the GDP values into a 'high' and a 'low' bin. If the value is above the midpoint of the data, then it is high, otherwise it is in the low bin. After the binning is complete, the logistic regression can be performed using the LogisticRegression from scikit-learn. To measure how well the logistic regression represents the data, the P value is observed.

Code Snippet:

```python
# Binning
# split the data into 2 bins
range = df['gdp'].max() - df['gdp'].min()
binWidth = range / 2
# Define the bins, that is, each bin's lower and upper limits
bins = [df['gdp'].min()-1, df['gdp'].min() + binWidth, df['gdp'].max()+1]
# Create the names for the two groups
groupNames = ['Low', 'High']
# Create a new column that is the result of the cut function
df['gdp_bin'] = pd.cut(df['gdp'], bins, labels=groupNames)
print("\ndf after binning: \n", df.sort_values(by=['gdp']))
# Drop the original gdp column
df = df.drop(['gdp'], axis=1)
print("\ndf after dropping gdp column: \n", df.sort_values(by=['gdp_bin']))

# Logistic regression
# Split the data into training and testing sets
# Get the x and y values
y = df['gdp_bin']
X = df.drop(['gdp_bin'], axis=1)
X = X.values

# Create the interaction terms
poly = PolynomialFeatures(2, interaction_only=True, include_bias=False)
interactionX = poly.fit_transform(X)

# Fit the logistic regression model
modelInteraction = LogisticRegression(solver='liblinear', random_state=0)
modelInteraction.fit(interactionX, y)

model = LogisticRegression(solver='liblinear', random_state=0)
model.fit(X, y)

# Check for the presence of interactions in the logistic regression model
modelInteractionScore = modelInteraction.score(interactionX, y)
modelScore = model.score(X, y)
df = interactionX.shape[1] - X.shape[1]
ratio = 2 * (modelInteractionScore - modelScore)
pValue = 1 - chi2.cdf(ratio, df)

print("\nP value: ", pValue)
```

Output:

```
df after binning:
        year  schooling           gdp gdp_bin
8281    2015        2.9     825.205688     Low
9628    2015        4.2     852.749207     Low
13153   2015        6.4    1065.242432     Low
37014   2015        1.8    1131.519165     Low
34813   2015        3.5    1262.613159     Low
...      ...        ...            ...     ...
50033   2015       13.4   68025.921875    High
24333   2015       12.3   71508.734375    High
46236   2015       11.5   89519.734375    High
42000   2015        9.8   95965.250000    High
30146   2015       12.0  113182.726562    High

[181 rows x 4 columns]

df after dropping gdp column:
        year  schooling gdp_bin
50      2015        3.6     Low
34813   2015        3.5     Low
35072   2015        4.9     Low
35316   2015        6.7     Low
35813   2015        4.7     Low
...      ...        ...     ...
50033   2015       13.4    High
30146   2015       12.0    High
38529   2015       12.5    High
54822   2015       13.3    High
7504    2015        9.0    High

[181 rows x 3 columns]

P value:  1.0
```

Scholarship Dollars vs GDP per capita

Unlike the Schooling vs GDP, scholarship data and GDP overtime are contained in two separate datasets. This means both need to be read into the program, and joined in a way that accurately reflects the data. In this case, the datasets were merged on the year. Similarly the data needs to be cleaned. This involved dropping null values, selecting only the required attributes, and removing any categorical data. Code snippets will be displayed in order to help with replication of results.

Code Snippet:

```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics

# Read the CSV file into a pandas dataframe
scholarship_df = pd.read_csv("./datasets/Scholarship_Recipients_And_D
gdp_df = pd.read_csv("./datasets/CountriesGDP1960-2020.csv")

# Clean the data
scholarship_df = scholarship_df.dropna()
gdp_df = gdp_df.dropna()

# Select only the columns we need for our plot
scholarship_df = scholarship_df[['Academic Year', 'TAP College Name',
# Rename the columns for easier use
scholarship_df.columns = ['year', 'college', 'headcount', 'dollars']

# Select rows where 'Country Name' column is 'United States'
gdp_US_df = gdp_df.loc[gdp_df['Country Name'] == 'United States']

# Remove categorical types
scholarship_df = scholarship_df.select_dtypes(exclude=['object'])
gdp_US_df = gdp_US_df.select_dtypes(exclude=['object'])

# Select only the common years
gdp_US_df = gdp_US_df.loc[:, '2009':'2018']

 # Convert the gdp dataframe so that year and gdp are columns
print(gdp_US_df)
gdp_US_df = gdp_US_df.reset_index()
gdp_US_df = gdp_US_df.melt(var_name='year', value_name='gdp')
gdp_US_df.drop(gdp_US_df.index[0], inplace=True)
gdp_US_df['year'] = gdp_US_df['year'].astype(int)
print(scholarship_df)
print(gdp_US_df)

# Merge the two dataframes
df = pd.merge(scholarship_df, gdp_US_df, on='year')
# Drop the year column
df = df.drop(['year'], axis=1)
print(df)
```

Output:

```
          2009          2010          2011          2012
114  1.440000e+13  1.500000e+13  1.550000e+13  1.620000e+13  1.68
      year  headcount    dollars
0     2018         45   55785.67
1     2018        113  448477.99
2     2018         70  130069.50
3     2018         42   31431.00
4     2018          7   16690.00
...    ...        ...        ...
3041  2009         87   83000.00
3042  2009        320  388024.75
3043  2009        178  180395.99
3044  2009        353  305531.11
3045  2009          1    4895.10

[3046 rows x 3 columns]
    year           gdp
1   2009  1.440000e+13
2   2010  1.500000e+13
3   2011  1.550000e+13
4   2012  1.620000e+13
5   2013  1.680000e+13
6   2014  1.750000e+13
7   2015  1.820000e+13
8   2016  1.870000e+13
9   2017  1.950000e+13
10  2018  2.060000e+13
      headcount    dollars           gdp
0            45   55785.67  2.060000e+13
1           113  448477.99  2.060000e+13
2            70  130069.50  2.060000e+13
3            42   31431.00  2.060000e+13
4             7   16690.00  2.060000e+13
...         ...        ...           ...
3041         87   83000.00  1.440000e+13
3042        320  388024.75  1.440000e+13
3043        178  180395.99  1.440000e+13
3044        353  305531.11  1.440000e+13
3045          1    4895.10  1.440000e+13

[3046 rows x 3 columns]
```

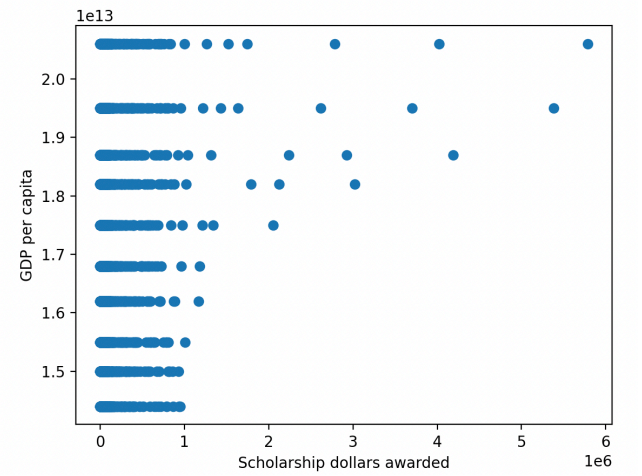
With this new dataset that is created, Exploratory Data Analysis can be performed using methods from PANDAS, along with visualization techniques. To visualize the data, the graph of scholarship dollars awarded vs GDP and the scholarship dollars vs GDP are plotted.

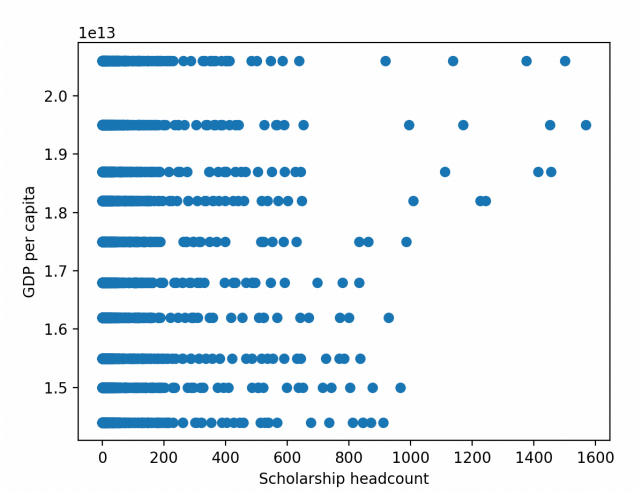
Code snippet:

```
47  # Exploratory data analysis (EDA)
48  print("\nThe dataframe: \n", df)
49  print("\nThe dataframe's description: \n", df.describe())
50  print("\nThe dataframes types: \n", df.dtypes)
51
52
53  # Create a scatter plot of GDP vs scholarship dollars awarded
54  plt.scatter(df.dollars, df.gdp)
55  # Set the x and y labels
56  plt.xlabel('Scholarship dollars awarded')
57  plt.ylabel('GDP per capita')
58  plt.show()
59
60  # Create a scatter plot of GDP vs scholarship headcount
61  plt.scatter(df.headcount, df.gdp)
62  # Set the x and y labels
63  plt.xlabel('Scholarship headcount')
64  plt.ylabel('GDP per capita')
65  plt.show()
```

Output:

```
The dataframe:
      headcount    dollars           gdp
0            45   55785.67  2.060000e+13
1           113  448477.99  2.060000e+13
2            70  130069.50  2.060000e+13
3            42   31431.00  2.060000e+13
4             7   16690.00  2.060000e+13
...         ...        ...           ...
3041         87   83000.00  1.440000e+13
3042        320  388024.75  1.440000e+13
3043        178  180395.99  1.440000e+13
3044        353  305531.11  1.440000e+13
3045          1    4895.10  1.440000e+13

[3046 rows x 3 columns]

The dataframe's description:
          headcount       dollars           gdp
count  3046.000000  3.046000e+03  3.046000e+03
mean     70.163493  1.271665e+05  1.710417e+13
std     149.236615  2.868787e+05  1.921726e+12
min       1.000000  1.225000e+02  1.440000e+13
25%       3.000000  9.776745e+03  1.550000e+13
50%      15.000000  4.526125e+04  1.680000e+13
75%      69.000000  1.231750e+05  1.870000e+13
max    1568.000000  5.785755e+06  2.060000e+13

The dataframes types:
 headcount      int64
dollars       float64
gdp           float64
dtype: object
```





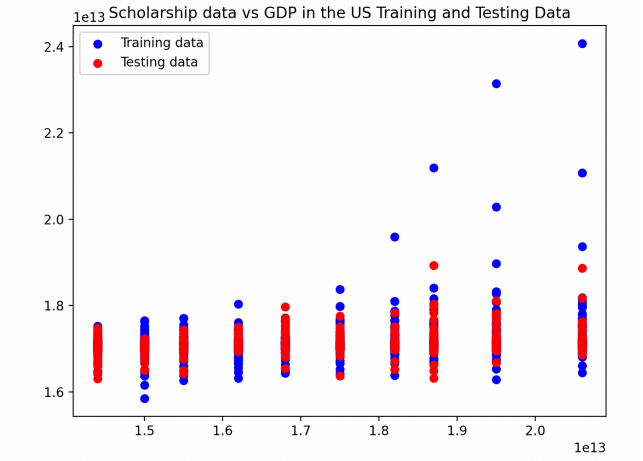
After the two datasets are cleansed and merged together, then the regressions can be executed. Since this data is similar in type to that of schooling vs GDP, it is important to keep the type of regression consistent to guarantee more accurate results. Therefore, the first regression performed is a linear regression. The following code snippet can be used as an aid for replicating results.

Code Snippet:

```
# Linear regression
# Get the x and y values
y = df['gdp']
X = df.drop(['gdp'], axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y)

# Fit the training data using a strait line
model = LinearRegression()
model.fit(X_train, y_train)

print("\nIntercepts of regression equation: ", model.intercept_)
coefficients = pd.DataFrame(model.coef_, X.columns, columns=['Coefficient'])
print("\nCoefficients of regression equation: \n", coefficients)

# Plot the training and testing data
predictionsTrain = model.predict(X_train)
predictionsTest = model.predict(X_test)
plt.scatter(y_train, predictionsTrain, color='blue', label='Training data')
plt.scatter(y_test, predictionsTest, color='red', label='Testing data')
plt.title("Scholarship data vs GDP in the US Training and Testing Data")
plt.legend()
plt.show()

# now calculate RMSE and MSE
rmse = np.sqrt(metrics.mean_squared_error(y_test, predictionsTest))
mse = metrics.mean_squared_error(y_test, predictionsTest)
r2 = metrics.r2_score(y_test, predictionsTest)
print("\nRMSE: ", rmse)
print("\nMSE: ", mse)
print("\nR-squared: ", r2)

```

Output:

```
Intercepts of regression equation:  17041482637988.395

Coefficients of regression equation:
             Coefficient
headcount -2.546013e+09
dollars    1.875952e+06

RMSE:  1915772211687.6274

MSE:  3.670183167074504e+24

R-squared:  0.028316370267666646
```



P Values:

Code snippet:

```
# Get p values
model = sm.OLS(y_train, sm.add_constant(X_train)).fit()
print("\n P Values: \n", model.pvalues)
```

Output:

```
 P Values:
 const          0.000000e+00
headcount     1.794892e-08
dollars       6.188714e-14
dtype: float64
```

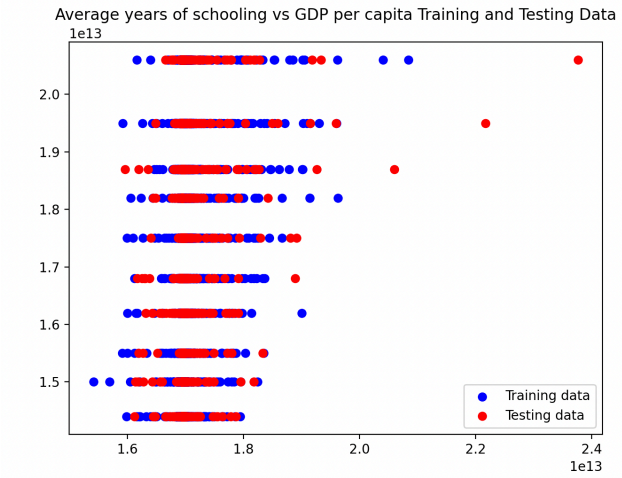
Next, a polynomial regression is executed. Based on the previous results, this is expected to represent the data in a more optimal way than that of the linear regression. Observe.

Code snippet:

```python
# Polynomial regression
print("\nPOLYNOMIAL REGRESSION")
poly = PolynomialFeatures(degree=2)
X_train_poly = poly.fit_transform(X_train)
X_test_poly = poly.transform(X_test)

model = LinearRegression()
model.fit(X_train_poly, y_train)

print("\nIntercepts of regression equation: ", model.intercept_)
coefficients = pd.DataFrame(model.coef_, poly.get_feature_names_out(X.columns), columns=['Coefficient'])
print("\nCoefficients of regression equation: \n", coefficients)

# Plot the training and testing data
predictionsTrain = model.predict(X_train_poly)
predictionsTest = model.predict(X_test_poly)
plt.scatter(predictionsTrain, y_train, color='blue', label='Training data')
plt.scatter(predictionsTest, y_test, color='red', label='Testing data')
plt.title('Average years of schooling vs GDP per capita Training and Testing Data')
plt.legend()
plt.show()

# now calculate RMSE and MSE
rmse = np.sqrt(metrics.mean_squared_error(y_test, predictionsTest))
mse = metrics.mean_squared_error(y_test, predictionsTest)
r2 = metrics.r2_score(y_test, predictionsTest)
print("\nRMSE: ", rmse)
print("\nMSE: ", mse)
print("\nR-squared: ", r2)
```

Output:

```
POLYNOMIAL REGRESSION

Intercepts of regression equation:  16943545779126.973

Coefficients of regression equation:
                    Coefficient
1                  0.000000e+00
headcount         -4.558337e+09
dollars            3.925850e+06
headcount^2        1.390128e+06
headcount dollars -1.281405e+03
dollars^2         -1.229875e-01

RMSE:  1892880008035.5703

MSE:  3.582994724820741e+24

R-squared:  0.03805280064770433
```



P Values:

Code snippet:

```python
# Get p values
model = sm.OLS(y_train, sm.add_constant(X_train)).fit()
print("\n P Values: \n", model.pvalues)
```

Output:

```
 P Values:
 const         0.000000e+00
headcount      4.449993e-07
dollars        2.316054e-13
dtype: float64
```

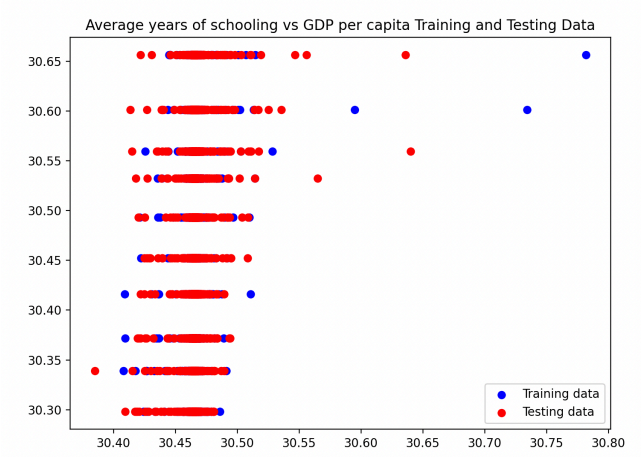
Considering that a polynomial regression performed better than the linear regression, it is also likely that a linear regression with a logarithmic transformation will also outperform a traditional linear regression. Observe:

Code snippet:

```python
# Use logarithmic transformation for linear regression
print("\nLOGARITHMIC TRANSFORMATION FOR LINEAR REGRESSION")
# Take the natural logarithm of the GDP variable
df['ln_gdp'] = np.log(df['gdp'])

# Split the data into training and testing sets
# Get the x and y values
y = df['ln_gdp']
X = df.drop(['gdp', 'ln_gdp'], axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.7, train_size=0.3)

# Fit the training data using a strait line
model = LinearRegression()
model.fit(X_train, y_train)

print("\nIntercepts of regression equation: ", model.intercept_)
coefficients = pd.DataFrame(model.coef_, X.columns, columns=['Coefficient'])
print("\nCoefficients of regression equation: \n", coefficients)

# Plot the training and testing data
predictionsTrain = model.predict(X_train)
predictionsTest = model.predict(X_test)
plt.scatter(predictionsTrain, y_train, color='blue', label='Training data')
plt.scatter(predictionsTest, y_test, color='red', label='Testing data')
plt.title('Average years of schooling vs GDP per capita Training and Testing Data')
plt.legend()
plt.show()

# now calculate RMSE and MSE
rmse = np.sqrt(metrics.mean_squared_error(y_test, predictionsTest))
mse = metrics.mean_squared_error(y_test, predictionsTest)
r2 = metrics.r2_score(y_test, predictionsTest)
print("\nRMSE: ", rmse)
print("\nMSE: ", mse)
print("\nR-squared: ", r2)
```

Output:

```
LOGARITHMIC TRANSFORMATION FOR LINEAR REGRESSION

Intercepts of regression equation:  30.464429647167243

Coefficients of regression equation:
             Coefficient
headcount -1.472449e-04
dollars    9.302588e-08

RMSE:  0.1106712367145698

MSE:  0.012248122635932342

R-squared:  0.024134644613837408
```



P Values:

Code snippet:

```
159  # Get p values
160  model = sm.OLS(y_train, sm.add_constant(X_train)).fit()
161  print("\n P Values: \n", model.pvalues)
```

Output:

```
 P Values:
 const          0.000000
headcount       0.002846
dollars         0.000021
dtype: float64
```

Average University Ranking vs GDP per capita

Lastly, the university ranking vs GDP per capita was analyzed. This is yet another instance where two datasets must be combined in order to evaluate the relationship of the data. After reading the data into two PANDAS dataframes, the null values are dropped. The dataframe of university rankings is modified, by taking the average ranking of universities by country, and filtering the attributes to only the country and average university ranking for that given country. The GDP dataframe is a dataframe where each year is a column name and the record of that year is the GDP value. This is modified so that the dataframe contains the column names year, country, and the gdp value for that given year and country. After this step is completed, we use the average GDP for each year and country to represent that year and country; this is done for the years 2004 - 2020. Finally, the dataframes can be merged together based on the country. After the dataframes are merged on the country, the country column is dropped so that the university ranking and GDP per capita can be accurately analyzed. Observe the following for replication purposes.

Code snippet:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.preprocessing import PolynomialFeatures

# Read the CSV file into a pandas dataframe
ranking_df = pd.read_csv("./datasets/universityRankings.csv")
gdp_df = pd.read_csv("./datasets/CountriesGDP1960-2020.csv")

# Clean the data
ranking_df = ranking_df.dropna()
gdp_df = gdp_df.dropna()

# Get average ranking by country
ranking_df = ranking_df.groupby('Country')['Ranking based on the average'].mean().reset_index()
# Rename the columns for easier use
ranking_df.columns = ['country', 'ranking']
# Sort by ranking
ranking_df = ranking_df.sort_values(by=['ranking'])

# Convert the gdp dataframe so that year, country and gdp are columns
gdp_df = gdp_df.reset_index()
gdp_df = gdp_df.rename(columns={'Country Name': 'country'})
gdp_df = gdp_df.melt(id_vars=['country'], var_name='year', value_name='gdp')
# Remove invalid records
gdp_df = gdp_df[gdp_df['year'] != 'index']
gdp_df = gdp_df[gdp_df['year'] != 'Country Name']
gdp_df = gdp_df[gdp_df['year'] != 'Country Code']
gdp_df['year'] = gdp_df['year'].astype(int)

# Get average gdp by country from 2004 to 2020
gdp_df = gdp_df[gdp_df['year'].between(2004, 2020)]
gdp_df = gdp_df.groupby('country')['gdp'].mean().reset_index()
gdp_df = gdp_df.sort_values(by=['gdp'])


print(ranking_df)
print(gdp_df)

# Merge the two dataframes
df = pd.merge(ranking_df, gdp_df, on='country')
print(df)

# Drop the country column
df = df.drop(['country'], axis=1)
print(df)

```

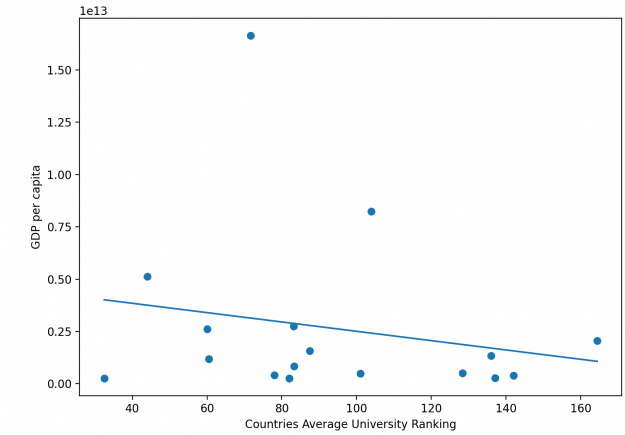
Output:

	ranking	gdp
0	32.500000	2.622353e+11
1	44.000000	5.134706e+12
2	60.000000	2.611765e+12
3	60.444444	1.181706e+12
4	71.696429	1.664706e+13
5	78.000000	4.108235e+11
6	82.000000	2.520000e+11
7	83.181818	2.755882e+12
8	83.250000	8.361765e+11
9	87.500000	1.568824e+12
10	101.000000	4.838824e+11
11	103.909091	8.234118e+12
12	128.333333	5.075882e+11
13	136.000000	1.343529e+12
14	137.000000	2.786471e+11
15	142.000000	4.001765e+11
16	164.333333	2.052353e+12

Then, Exploratory Data Analysis can be performed on this new dataframe. This involves using PANDAS methods, along with a plot of Countries Average University Ranking vd GDP.
Code snippet:

```
# Exploratory data analysis (EDA)
print("\nThe dataframe: \n", df)
print("\nThe dataframe's description: \n", df.describe())
print("\nThe dataframes types: \n", df.dtypes)


# Create a scatter plot of GDP vs the Countries Average University Ranking
plt.scatter(df.ranking, df.gdp)
# Set the x and y labels
plt.xlabel('Countries Average University Ranking')
plt.ylabel('GDP per capita')

# Get the trendline coefficients
z = np.polyfit(df.ranking, df.gdp, 1)
# Get the polynomial of the trendline
p = np.poly1d(z)
# Plot the trendline
plt.plot(df.ranking, p(df.ranking))

# Show the plot
plt.show()

```

Output:

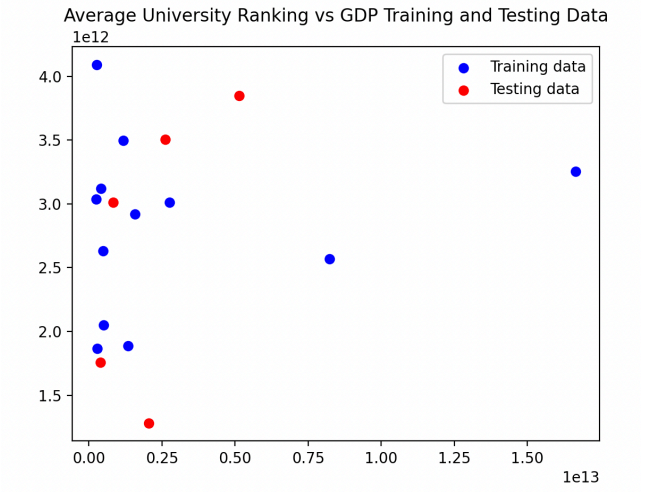


Then, a linear regression is executed. Observe.
Code Snippet:

```
# Linear regression
# Get the x and y values
y = df['gdp']
X = df.drop(['gdp'], axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y)

# Fit the training data using a strait line
model = LinearRegression()
model.fit(X_train, y_train)

print("\nIntercepts of regression equation: ", model.intercept_)
coefficients = pd.DataFrame(model.coef_, X.columns, columns=['Coefficient'])
print("\nCoefficients of regression equation: ", coefficients)

# Plot the training and testing data
predictionsTrain = model.predict(X_train)
predictionsTest = model.predict(X_test)
plt.scatter(y_train, predictionsTrain, color='blue', label='Training data')
plt.scatter(y_test, predictionsTest, color='red', label='Testing data')
plt.title("Average University Ranking vs GDP Training and Testing Data")
plt.legend()
plt.show()

# now calculate RMSE and MSE
rmse = np.sqrt(metrics.mean_squared_error(y_test, predictionsTest))
mse = metrics.mean_squared_error(y_test, predictionsTest)
r2 = metrics.r2_score(y_test, predictionsTest)
print("\nRMSE: ", rmse)
print("\nMSE: ", mse)
print("\nR-squared: ", r2)

```

Output:

```
Intercepts of regression equation:  4783404833570.826

Coefficients of regression equation:            Coefficient
ranking -2.131019e+10

RMSE:  1386921477494.2478

MSE:  1.9235511847348273e+24

R-squared:  0.30822343838643385
```



P Values:
Code snippet:

```
# Get p values
model = sm.OLS(y_train, sm.add_constant(X_train)).fit()
print("\n P Values: \n", model.pvalues)
```

Output:

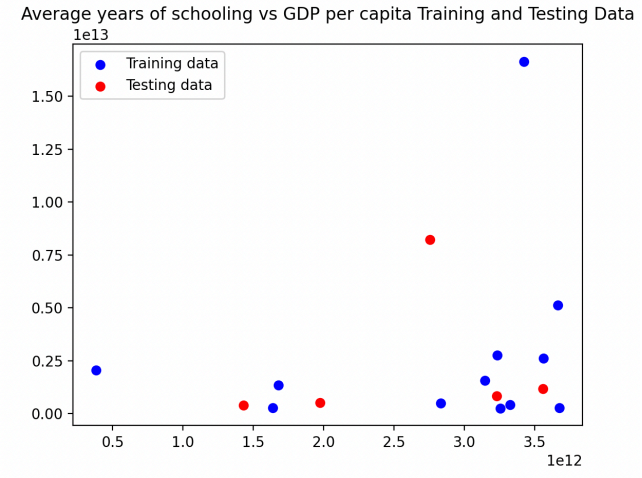
```
P Values:
const       0.268803
ranking     0.632904
dtype: float64
```

Next, a polynomial regression is executed.
Code Snippet:

```
# Polynomial regression
print("\nPOLYNOMIAL REGRESSION")
poly = PolynomialFeatures(degree=2)
X_train_poly = poly.fit_transform(X_train)
X_test_poly = poly.transform(X_test)

model = LinearRegression()
model.fit(X_train_poly, y_train)

print("\nIntercepts of regression equation: ", model.intercept_)
coefficients = pd.DataFrame(model.coef_, poly.get_feature_names_out(X.columns), columns=['Coefficient'])
print("\nCoefficients of regression equation: \n", coefficients)

# Plot the training and testing data
predictionsTrain = model.predict(X_train_poly)
predictionsTest = model.predict(X_test_poly)
plt.scatter(predictionsTrain, y_train, color='blue', label='Training data')
plt.scatter(predictionsTest, y_test, color='red', label='Testing data')
plt.title('Average years of schooling vs GDP per capita Training and Testing Data')
plt.legend()
plt.show()

# now calculate RMSE and MSE
rmse = np.sqrt(metrics.mean_squared_error(y_test, predictionsTest))
mse = metrics.mean_squared_error(y_test, predictionsTest)
r2 = metrics.r2_score(y_test, predictionsTest)
print("\nRMSE: ", rmse)
print("\nMSE: ", mse)
print("\nR-squared: ", r2)
```

Output:

```
POLYNOMIAL REGRESSION

Intercepts of regression equation:  3415839987249.0703

Coefficients of regression equation:
             Coefficient
1           0.000000e+00
ranking     1.441791e+10
ranking^2 -2.000177e+08

RMSE:  2986712895463.147

MSE:  8.920453919925855e+24

R-squared:  0.01770163704000116
```



P Values:
Code snippet:

```
# Get p values
model = sm.OLS(y_train, sm.add_constant(X_train)).fit()
print("\n P Values: \n", model.pvalues)
```

Output:

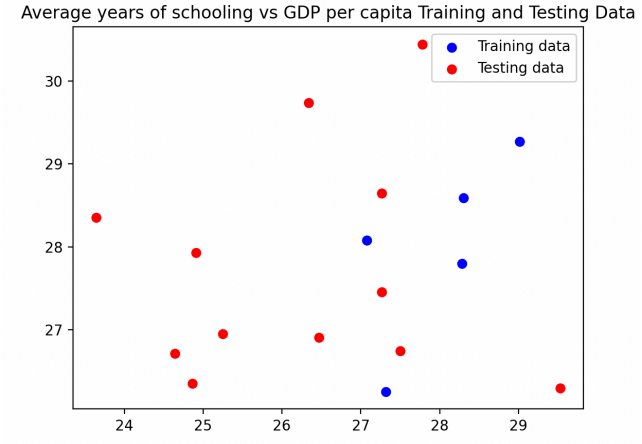
```
P Values:
const       0.268803
ranking     0.632904
dtype: float64
```

Next, similar to the other datasets, a second linear regression is performed where a logarithmic transformation is executed first.
Code Snippet:

```python
# Use logarithmic transformation for linear regression
print("\nLOGARITHMIC TRANSFORMATION FOR LINEAR REGRESSION")
# Take the natural logarithm of the GDP variable
df['ln_gdp'] = np.log(df['gdp'])

# Split the data into training and testing sets
# Get the x and y values
y = df['ln_gdp']
X = df.drop(['gdp', 'ln_gdp'], axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.7, train_size=0.3)

# Fit the training data using a strait line
model = LinearRegression()
model.fit(X_train, y_train)

print("\nIntercepts of regression equation: ", model.intercept_)
coefficients = pd.DataFrame(model.coef_, X.columns, columns=['Coefficient'])
print("\nCoefficients of regression equation: \n", coefficients)

# Plot the training and testing data
predictionsTrain = model.predict(X_train)
predictionsTest = model.predict(X_test)
plt.scatter(predictionsTrain, y_train, color='blue', label='Training data')
plt.scatter(predictionsTest, y_test, color='red', label='Testing data')
plt.title('Average years of schooling vs GDP per capita Training and Testing Data')
plt.legend()
plt.show()

# now calculate RMSE and MSE
rmse = np.sqrt(metrics.mean_squared_error(y_test, predictionsTest))
mse = metrics.mean_squared_error(y_test, predictionsTest)
r2 = metrics.r2_score(y_test, predictionsTest)
print("\nRMSE: ", rmse)
print("\nMSE: ", mse)
print("\nR-squared: ", r2)
```

Output:

```
LOGARITHMIC TRANSFORMATION FOR LINEAR REGRESSION

Intercepts of regression equation:  30.97867043609652

Coefficients of regression equation:
          Coefficient
ranking     -0.044628

RMSE:  2.4573336358347615

MSE:  6.038488597804888

R-squared:  -2.625917028426479
```



P Values:
Code snippet:

```python
# Get p values
model = sm.OLS(y_train, sm.add_constant(X_train)).fit()
print("\n P Values: \n", model.pvalues)
```

Output:

```
 P Values:
 const      0.000378
ranking     0.145061
dtype: float64
```

MODEL DESIGN

Throughout the process of designing and executing this research, we decided to place heavy emphasis on the impact education could, and would make on an economy, both domestically and internationally. When choosing models that would best represent the outcome we sought to uncover, there were some that became advantageous, and others that were not a correct fit for the type of data we used.

We decided to go with a linear regression, a polynomial regression, a linear regression with logarithmic transformation, and a logistic regression. These models became the pinnacle of our research because that is what our data type called for. Other models we could have done include decision trees and neural networks. Both decision tree models and neural network models are models that many use when it comes to nonlinear relationships, and complex data. These models would not have proved sufficient in our case because the type of data we began to work with, do not require complex algorithms, and some correlations were proven to be linear within our first model of a linear regression, no further analysis of the type of relationship was required, which wouldn't have been the case with a decision tree or a neural network.

When dealing with a common labor economics problem like this, it is usually best to stay with models that will produce a streamlined visual outcome, rather than a complex one that would require deeper understanding. Both the polynomial regression, and the linear regression with logarithmic transformation models performed with the most accuracy, precision, and recall.

The initial linear regression was run to test the average years of schooling and the GDP per capita in given countries. This regression used data that had been pre-processed, and split into testing and training categories previously. The result of this model gave a value of 0.36 for the R^2 value. This showed us that there was some significance, but there had to be another way to generalize the outcome with more accuracy. We ultimately decided to drop this version of the model in our future tests to ensure accuracy.

The polynomial regression was set up to assess the correlation between the average years of schooling and the GDP per capita in given countries, the same way the linear regression was. The data was split into training and testing categories, and a polynomial feature was executed. The model produced an R^2 value of 0.43 versus the original linear regression's value of 0.36, and this showed us that this second form of regression made for a better representation of the data, and the outcome we desired. The higher value plays into our need for a model with high accuracy, and precision. The model also showcased a higher probability of recall by eliminating irrelevant correlations, and streamlining the outcome.

We then utilized a logistic regression to run the same comparison, and was met with an R^2 value of 0.65. This value was the highest of the three thus giving us accuracy and precision when testing the average years of schooling and the GDP per capita. This is a consistent outcome when it comes to the majority of the other tests we run during the tenure of this research.

There are many other models we could have used when it comes to this kind of research, but as stated before, the type of issue we aimed to investigate created limitations in our liberty of choosing the correct model.

EVALUATION METRICS

The methods of modeling we chose to use were linear, polynomial, and a logarithmic transformation for linear regression, meaning that the metrics chosen to evaluate those models include RMSE, MSE, and R^2 values. These evaluation metrics will allow us to see/rate the performance of our models. Viewing the performance of the model is critical to understanding the data and outputs. If the model is not seen as accurate then our data and conclusion may be invalid.

Below shows the condensed metrics for each dataset:

Linear Regression Evaluation Metrics

Dataset	RMSE	MSE	R^2
Avg. Yr. School VS GDP	16739.06	2.8e+8	0.36
Schol. $ VS GDP	1.9e+12	3.67e+12	0.0283
Avg. Uni. Ranking VS GDP	1.38e+12	1.92e+24	0.308

Polynomial Regression Evaluation Metrics

Dataset	RMSE	MSE	R^2
Avg. Yr. School VS GDP	15580.9	2.4e+12	0.42
Schol. $ VS GDP	1.89e+12	3.58e+24	0.038
Avg. Uni. Ranking VS GDP	2.98e+12	8.9e+24	0.0177

Logarithmic Transformation For Linear Regression Metrics

Dataset	RMSE	MSE	R^2
Avg. Yr. School VS GDP	0.065	0.431	0.65
Schol. $ VS GDP	0.1106	0.0122	0.024
Avg. Uni. Ranking VS GDP	2.45	6.0384	-2.62

Background Info on Metrics

Before we delve into the metrics for each dataset. We must first discuss what the standard for each of the metrics is. For each metric there is a standard that allows us to evaluate the metric to ascertain what it means. For RMSE and MSE the values can range from 0-∞. The closer to 0 the value of each of these is, the better the model. For R^2 the values can range between 0-1, the closer to 1 we are the better the model[14][15]. However for R^2 we can possibly have negative values which will be discussed later. Now that we know a little about the metrics themselves we can begin to take a look at the individual modeling types.

Linear Regression Models

After performing linear regressions on all the models we can see that all of the models on data sets did not produce favorable outcomes. As a general rule of thumb it can be said that good RMSE values fall between 0.2 and 0.5. After looking at the individual scores for each dataset we can see that the RMSE, MSE, and R^2 values for these datasets fall out of the "good"

range. The worst of them being Scholarship $ VS GDP per capita. The R^2 values of all the models show us that the models do not explain variation in the independent variable. The RMSE and MSE being high mean that the models are no better or worse than taking the mean values of the datasets.

After reviewing the evaluation metrics of the models, it can be clearly stated that the models are far from perfect and therefore do not offer much information on the effect of education on GDP. Next we will look at the polynomial regressions.

Polynomial Regression Models

When looking at the polynomial regression models we can see that the values as before with the linear regression fall short of what we were hoping for. Each model has a high MSE and RMSE, meaning that it is no better than taking the mean value. It can also be noted that the R^2 value of each model being less than 0.5 means that the models are also far from perfect. As stated before the metrics for each model show that polynomial regression does not predict well when compared with the actual observed values. Therefore meaning that the models do not offer us much help.

Logarithmic Transformation For Linear Regression Models

The evaluation metrics for these models are much more favorable than those previous. However there are still some shortcomings for two of the models.

It is obvious to see that the logarithmic transformation for Average Years of Schooling vs GDP per capita was very fruitful. The MSE and RMSE values are much more favorable than previous models. The R^2 being 0.65 shows us that the model is close to 65% when predicting values compared to the observed. Therefore this means that this model is most likely the best out of all the models presented.

However, the same cannot be said for the other two models. While their values may be more favorable than previous models, we can still see that they fall short in a few categories. The RMSE and MSE for each of these 2 models are lower than the other modeling methods but still fall short of being good values. The R^2 value of Scholarship $ VS GDP per capita, shows us that the model is not doing well when compared to the actual observed values. The negative R^2 value of Average University Ranking VS GDP per capita shows us that the chosen model does not follow the trend of data.

In conclusion while the Logarithmic Transformation may give us more favorable results, there are still some shortcomings of these models.

Evaluation Metrics Conclusions

After discussing and viewing the evaluation metrics of the models and methods for modeling, we can easily see that while the most fruitful method for modeling is the Logarithmic Transformation for Linear Regression. Also it can be seen that Average Years of Schooling seems to be the most impactful on GDP, however due to the low scores of all the models there must be other factors at play. The model performance will therefore have to be discussed further due to the fact that the models did not perform optimally.

DISCUSSION

Results

The average years of schooling had p values of 0.000058 for the linear regression, 0.000058 for the polynomial regression, 1.144891x10^-18 for the linear regression with logarithmic transformation, and 1.0 for the logistic regression. The scholarship dollars awarded had p values of 6.188714x10^-14 for the linear regression, 2.316054x10^-13 for the polynomial regression, and 0.000021 for the linear regression with logarithmic transformation. The scholarship headcount had p values of 1.794892x10^-8 for the linear regression, 4.449993x10^-7 for the polynomial regression, and 0.002846 for the linear regression with logarithmic transformation. Lastly, The average university rankings had p values of 0.632904 for the linear regression, 0.632904 for the polynomial regression, and 0.145061for the linear regression with logarithmic transformation. So, out of the factors tested, the average years of schooling, the scholarship dollars awarded, and the scholarship headcount seem to be statistically significant; while the average university rank does not seem to be statistically significant.

Further Discussion

There are a vast amount of factors that impact the economy. Aside from education, there is inflation and deflation, foreign policy changes, fiscal and monetary policy, and the stock market just to name a few [1]. Given all the factors that influence the economy, it would be quite ignorant to conclude that one variable could rule them all in regard to economic influence. The findings show that of the tested variables, years of schooling, scholarship dollars awarded, and scholarship headcount were statistically significant; while average university ranking was not statistically significant. It must be noted that the economy is similar to a living creature where all parts must work together for the entire body to flourish. In this specific circumstance, it can be seen that years of schooling, scholarship dollars awarded, and scholarship headcount have a statistically significant impact on the GDP per capita, but biases should be considered. For instance, the data seems to be very clean cut at surface level, but in order to really understand which variable is the most influential, one must address bias. In other words, the biases are not clearly accounted for, which is why the data at its core is inconclusive. In order to gain

confidence in the findings of this study, it must be understood how each variable affects the other, and to what extent.

Education is also highly complex. There is the difficult question of what exactly is good education? Is quality or quantity more important? For example, in this project, years of schooling had the greatest impact on GDP; but what if those years were of poor quality? According to Our World In Data, "A general difficulty in educational data is to find a good measure to express the share of children that are taken into, enrolled, or attending educational institutions" [12]. Attributes to consider also include the intake rate, which is essentially the number of students entering first grade in a given year; enrollment, where enrollment rates measure enrollment across multiple different age groups; attendance rates, which measure attendance; the Out-Of-School metric, which measures the number of students who are out of school, drop-out rates, promotions rates, which measures the number of students in a given grade who are moving up to the next grade; and transition rates, which measures the number of students who transition to the next level [12]. To clarify, "Promotion refers to the progression from one grade to the next while transition refers to the promotion from one level of education to the next (primary to secondary or secondary to tertiary)" [12]." Other attributes include survival, students who repeat the same grade, school life expectancy, completion rates, and the gross graduation ratio [12]. The above examples are meant to serve as an example to represent the complexity involved in measuring education. As demonstrated, there are many attributes that could have been analyzed to judge what is good education other than what is used in this project; however, the attributes used in this project reprepresent the best options given the datasets available.
In the future, in order to determine which attributes have the most optimal representation of the GDP, more data needs to be acquired. This would allow for a greater variety of metrics for education and for the economy. Then Principal Component Analysis can be performed to determine which attributes have the highest impact on the economy.

CONCLUSION

In this project, data was analyzed to answer the research question of "What effect does the level of education have on economic growth?" Through this analysis, using various metrics to represent education, it was determined that the average years of schooling, scholarship dollars awarded, and scholarship headcount are statistically significant on the dependent variable which is the GDP. On the other hand, the average ranking of universities in a given country was not statistically significant. As previously stated, a difficulty of analyzing both the education and the economy is that these are very complex entities, with many parts which work in unison. This makes it difficult to analyze any specific metric. As a result, in the future, this research question could be broken up into more specific variations of the question, and more metrics of education could be tested to determine the impact on GDP per capita. Although this topic has the potential for many biases, it is clear that the metrics stated do in fact impact the GDP per capita.

REFERENCES

[1] "7 Factors of How the U.S. Economy Works." *Money & Markets, LLC*, 16 Mar. 2022, https://moneyandmarkets.com/7-factors-of-how-the-us-economy-works/.

[2] "API Reference" *Statsmodels*, https://www.statsmodels.org/stable/api.html.

[3] "Average Years of Schooling vs. GDP per Capita." *Our World in Data*, https://ourworldindata.org/grapher/average-years-of-schooling-vs-gdp-per-capita?tab=table.

[4] Christy, Rini. "Countries GDP 1960-2020." *Kaggle*, 8 Apr. 2022, https://www.kaggle.com/datasets/rinichristy/countries-gdp-19602020.

[5] "Education." *USDA ERS - Data Products*, https://data.ers.usda.gov/reports.aspx?ID=17829.

[6] Fernando, Jason. "R-Squared: Definition, Calculation Formula, Uses, and Limitations." *Investopedia*, Investopedia, 15 Apr. 2023, https://www.investopedia.com/terms/r/r-squared.asp.

[7] "Gross Domestic Product." *Wikipedia*, Wikimedia Foundation, 1 May 2023, https://en.wikipedia.org/wiki/Gross_domestic_product.

[8] Hair, Joseph F., et al. "An Introduction to Structural Equation Modeling." *SpringerLink*, Springer International Publishing, 1 Jan. 1970, https://link.springer.com/chapter/10.1007/978-3-030-80519-7_1.

[9] Pandey, Raj Kumar. "Highest Ranked Universities in Global Rankings." *Kaggle*, 21 Mar. 2023, https://www.kaggle.com/datasets/rajkumarpandey02/highest-ranked-universities-in-global-rankings?select=1000%2Bhighest%2Branked%2Buniversities%2Bin%2Bglobal%2Brankings.csv.

[10] Pant, Anjali. "Unemployment Dataset." *Kaggle*, 8 Sept. 2022, https://www.kaggle.com/datasets/pantanjali/unemployment-dataset.

[11] "Poverty." *USDA ERS - Data Products*, https://data.ers.usda.gov/reports.aspx?ID=17826.

[12] Roser, Max. "Measuring Education: What Data Is Available?" *Our World in Data*, 23 Feb. 2018, https://ourworldindata.org/measuring-education-what-data-is-available.

[13] Srinivasan, Sripaad. "New York Scholarship Dataset." *Kaggle*, 28 Nov. 2020, https://www.kaggle.com/datasets/sripaadsrinivasan/new-york-scholarship-dataset?resource=download.

[14] Team, Great Learning. "Mean Squared Error - Explained: What Is Mean Square Error?" *Great Learning Blog: Free Resources What Matters to Shape Your Career!*, 18 Nov. 2022, https://www.mygreatlearning.com/blog/mean-square-error-explained/#:~:text=RMSE%20is%20better.-,Conclusion,value%20indicates%20a%20better%20fit.

[15] Yashwanth, NVS. "Evaluation Metrics & Model Selection in Linear Regression." *Medium*, Towards Data Science, 1 Jan. 2021, https://towardsdatascience.com/evaluation-metrics-model-selection-in-linear-regression-73c7573208be.

[16] Hanushek, A. E. A. (n.d.). *Higher grades, higher GDP*. Higher Grades, Higher GDP | Eric A. Hanushek. Retrieved May 2, 2023, from http://hanushek.stanford.edu/publications/higher-grades-higher-gdp

[17] LaMorte, W. (2021, October 7). Correlation and Linear Regression. Correlation Analysis. Retrieved May 2, 2023, from https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_correlation-regression/bs704_correlation-regression2.html

[18] What is machine learning? IBM. (n.d.). Retrieved May 2, 2023, from https://www.ibm.com/topics/machine-learning

[19] Delgado, M. S., Henderson, D. A. J., & Parmeter, C. F. (2014). Does Education Matter for Economic Growth? Oxford

Bulletin of Economics and Statistics, 76(3), 334–359. Retrieved May 2, 2023, from https://docs.iza.org/dp7089.pdf.

[20] World Bank Group. (2023, March 30). *The Human Capital Project: Frequently Asked Questions*. World Bank. Retrieved May 2, 2023,fromhttps://www.worldbank.org/en/publication/human-capital/brief/the-human-capital-project-frequently-asked-questions#:~:text=Human%20capital%20consists%20of%20the,as%20productive%20members%20of%20society.