Name: Loh Yee Kai
Matriculation No.: A0238898B

## 1. Exploratory Data Analysis (EDA)

Before creating a model, it is important to understand the dataset given first. EDA is carried out, and here are the findings:

1. 39 columns with 1600 observations
2. There are no NaN/missing values
3. MedianIncome22 has a broad range from $27,326 to $157,778
4. Based on the description of each column, these are how we will be splitting the data and categorising them

```
# Categorical Nominal Columns
cat_nominal_cols = ['LSTATE', 'PopulrVoteParty20']

# Categorical Ordinal Columns
cat_ordinal_cols = ['STRatioRank', 'RaceDiversityRank', 'GenderDiversityRank',
                    'RKGDP2022', 'RKPCGDP22', 'RKPI22', 'RKPCRI22']

# Numerical Columns
num_cols = ['RGDP19','RGDP20', 'RGDP21', 'RGDP22','PNAA','PNC','PPAA','PPC','P20',
            'P21','P22','WaterQualityVPV','NtnlPrkCnt','Unemployment','MajorPer20',
            'AQIGoodPer','CrimeRate2016','LivingCost','CI1p0c','CI1p1c','CI1p2c',
            'CI1p3c','CI1p4c','CI2p0c', 'CI2p1c','CI2p2c','CI2p3c','CI2p4c',
            'MedianIncome22']
```

5. Statistics of each column are displayed using .describe() in pandas
6. The correlation matrix is shown. Most Clipjc columns have a negative correlation with MedianIncome22. Living Cost has the highest positive correlation with MedianIncome22.

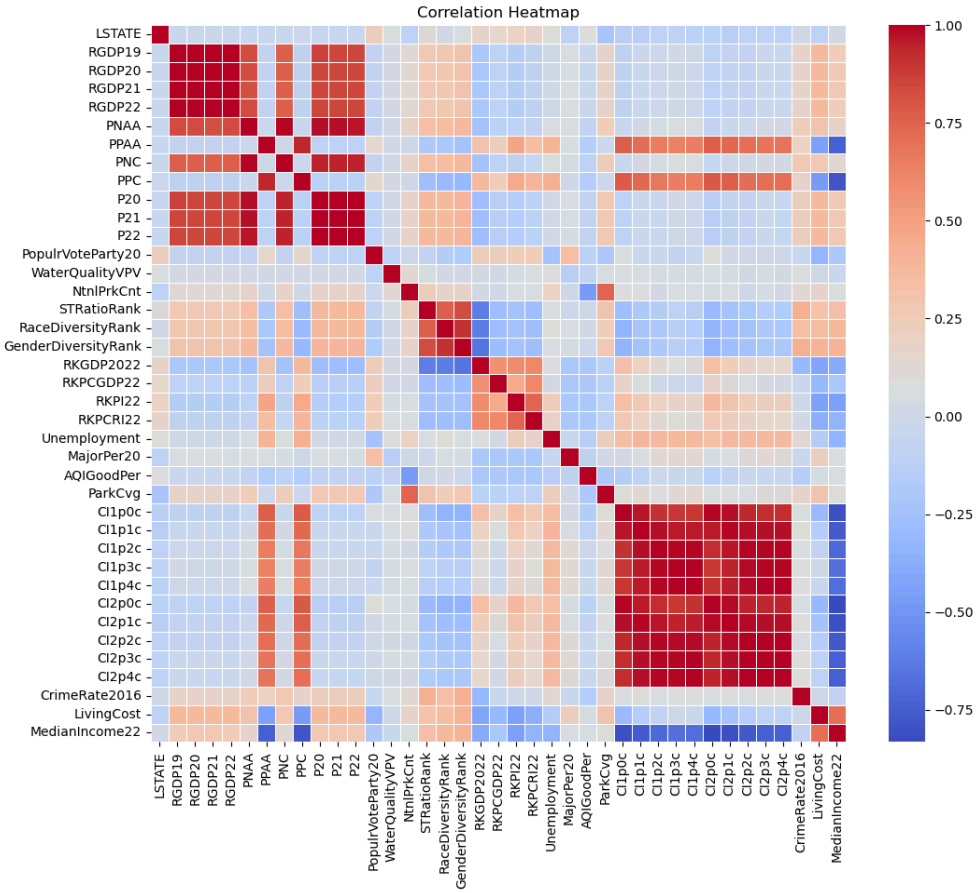


*Fig 1. Correlation Heatmap of Columns Provided*

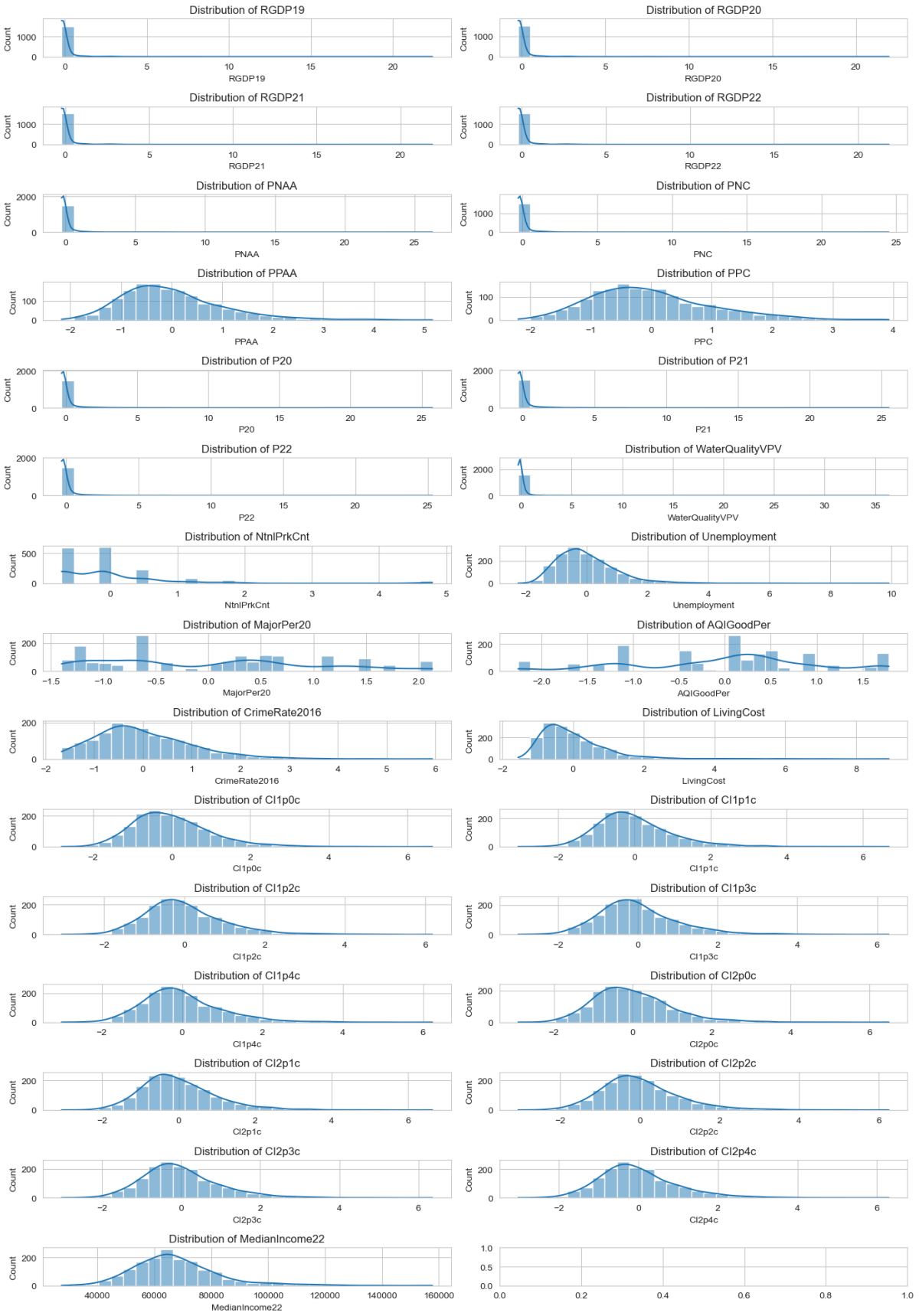


*Fig 2. Distribution of each numerical column*

## 2. Primitive Regression Model

### 2.1 Predictors used

A qualitative and quantitative approach is taken to decide the five predictors in the model.

1. LivingCost (positive correlation of 0.703): This variable, indicating the cost of living, has the highest positive correlation with MedianIncome22, suggesting that areas with higher living costs tend to have higher median incomes.
2. CI2p0c (negative correlation of -0.832): This variable shows the strongest negative correlation with the median income. Having more children increases families' living costs, thus reducing their median income. CI2p0c is used out of the others due to its higher absolute correlation.
3. PPAA (negative correlation of -0.738): The percentage of the population living in poverty negatively correlates with median income, highlighting poverty rates' impacts on income levels. The higher the poverty rate, the lower the median income.
4. Unemployment (negative correlation of -0.335): Although not the highest in magnitude, the unemployment rate is a crucial economic indicator expected to affect median income inversely.
5. GenderDiversityRank (positive correlation of 0.402): As a representative of social factors, this variable suggests that areas with higher gender diversity rankings may correlate with higher median incomes.

### 2.2 Insights and Diagnostics

The model obtained has the following results:

| Statistic | Value |
|---|---|
| RMSE | 4018.096638190982 |
| Coefficient of Determination, R^2 | 0.9248133330665197 |
| Intercept | 68192.13102906823 |
| LivingCost | 0.8079002677759493 |

| | |
|---|---|
| CI2p0c | -742.8933478778445 |
| PPAA | 42.86203650687341 |
| Unemployment | -237.77555591771386 |
| GenderDiversityRank | -0.5661398366985395 |

*Fig 3. Summary Statistics for Primitive Regression Model*

For each predictor used, a unit change of itself results in the expected median income to change by the associated value, i.e. a unit increase in PPAA results in a unit increase of $42.86 in median income.

## 2.3 Areas of Improvement

To improve this primitive model, we can consider a few things:

1. Enhancing Model Complexity

   We can explore polynomial features for the continuous variables as they might reveal non-linear relationships, improving model accuracy. Lasso or Ridge regression could be explored to prevent overfitting by penalizing larger coefficients, leading to a more generalized model.

2. Variable Selection and Transformation

   Feature engineering might lead to more predictive variables after transformation, i.e., log transformation or square terms.

3. Cross-validation (CV)

   CV can be conducted to provide a more accurate estimate of model performance.

4. Diagnostic Checks

   As seen below, diagnostic checks, such as the QQ plot for the five predictors, can be done. The QQ plot is generally quite linear at the centre and might conclude that the residuals follow a normal distribution.

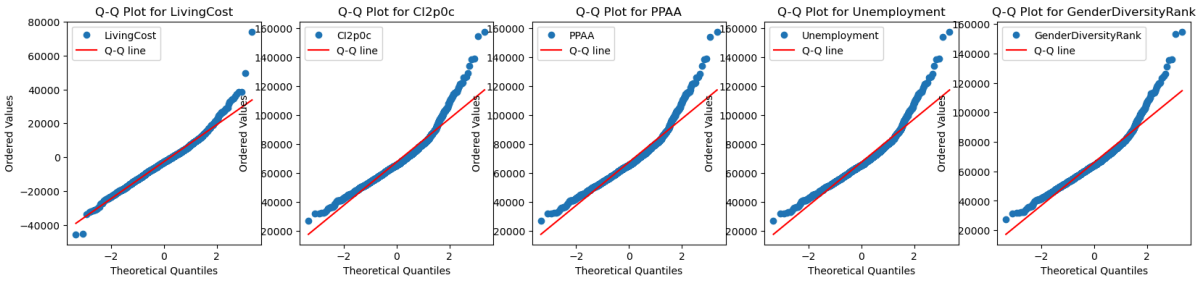


*Fig 4. QQ Plot of the five predictors*

## 3. Advanced Regression Model

To build the advanced regression model, we engineered additional features for our dataset by looking at possible factors that could affect the MedianIncome22 of the county. We used standard scaling and one-hot encoding to preprocess the categorical and numerical data. Subsequently, we sorted the variables into ordinal and nominal features for the model. After this, we selected the different features to be used in our model before running the model.

### 3.1 Feature Engineering

**Feature Engineering**

```
# Feature Engineering Steps
# 1. Aggregate RGDP
train_data['AverageRGDP'] = train_data[['RGDP19', 'RGDP20', 'RGDP21', 'RGDP22']].mean(axis=1)

# 2. Interaction Between Living Cost and an average CIipjc metric (demonstration purposes only)
# Assuming 'CI1p0c', 'CI1p1c' ... 'CI2p4c' represent different CIipjc metrics
ciipjc_columns = [col for col in train_data.columns if 'CI' in col and 'c' in col]
train_data['AverageCIipjc'] = train_data[ciipjc_columns].mean(axis=1)
train_data['LivingCost_CIipjc_Interaction'] = train_data['LivingCost'] / train_data['AverageCIipjc']

# 3. Economic and Demographic Ratios
train_data['GDPPerCapita'] = train_data['AverageRGDP'] / train_data['P22']

# Display the first few rows to verify the new features
new_features_preview = train_data[['AverageRGDP', 'LivingCost_CIipjc_Interaction', 'GDPPerCapita']].head()
new_features_preview
```

*Fig. 5 Feature Engineering in JupyterNotebook*

1. AverageRGDP - This feature was created by taking the mean of all four columns of RGDP19 till RGDP22.
2. GDPPerCapita - To create a more concise feature, we create a GDPPerCapita feature by dividing AverageRGDP and dividing by the 3-year average population of the county.
3. AverageCIipjc - This feature is created by taking the average of all CIipjc columns. This was done due to the high correlation with each other, and it would be better to take the aggregate of it.
4. LivingCost_CIipjc_Interaction - Since CIipjc is the average cost-to-income ratio for families with i parents and j children, there should be some interaction between CIipjc and LivingCost. By trial and error, we find that dividing the LivingCost by the Average value of CIipjc helps to create a model with lower MSE

With these four features created, we can proceed to feature selection.

## 3.2 Data Processing

The data was further processed to treat ordinal and nominal categorical features differently. By doing so, the model would be configured to treat them differently during the training phase.

The following columns have been categorised to their data type as seen below. It is important to fish out such data as preprocessing needs to be done so that they can be useful in model building. Mainly, categorical data will be preprocessed using One Hot Encoding, while the numerical data will be preprocessed using Standard Scaling.

```
# Categorical Nominal Columns
cat_nominal_cols = ['LSTATE', 'PopulrVoteParty20']

# Categorical Ordinal Columns
cat_ordinal_cols = ['STRatioRank', 'RaceDiversityRank', 'GenderDiversityRank',
                    'RKGDP2022', 'RKPCGDP22', 'RKPI22', 'RKPCRI22']

# Numerical Columns
num_cols = ['RGDP19','RGDP20', 'RGDP21', 'RGDP22','PNAA','PNC','PPAA','PPC','P20',
            'P21','P22','WaterQualityVPV','NtnlPrkCnt','Unemployment','MajorPer20',
            'AQIGoodPer','CrimeRate2016','LivingCost','CI1p0c','CI1p1c','CI1p2c',
            'CI1p3c','CI1p4c','CI2p0c', 'CI2p1c','CI2p2c','CI2p3c','CI2p4c',
            'MedianIncome22']
```

*Fig 6. Different Data Columns Categorised*

## 3.3 Feature Selection

To prevent overfitting, we must choose the features carefully. Selecting too many features naturally results in overfitting of the data.

```
col_to_drop = ['RGDP19', 'RGDP20', 'RGDP21', 'RGDP22','PNAA', 'PNC', 'P20', 'P21', 'P22','MedianIncome22',
               'CI1p0c','CI1p1c','CI1p2c','CI1p3c','CI1p4c','CI2p0c','CI2p1c','CI2p2c','CI2p3c','CI2p4c','MajorPer20',
              'AverageRGDP','AverageCIipjc','LivingCost','PPAA','PPC','ParkCvg','STRatioRank',
              'RaceDiversityRank','RKGDP2022','RKPCGDP22', 'RKPI22', 'RKPCRI22','WaterQualityVPV','NtnlPrkCnt',
              'AQIGoodPer','GenderDiversityRank', 'Unemployment', 'CrimeRate2016']

# Adjusting feature set and redefining X and y with correct handling
X = train_data.drop(columns=col_to_drop)
y = train_data['MedianIncome22']

# Splitting the dataset into training and validation sets again
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)
```

*Fig 7. Feature selection and splitting of data into training and testing*

The advanced regression model only needs four features: LSTATE, PopulrVoteParty20, LivingCost_Clipjc_Interaction, and GDPPerCapita. This was done through trial and error.

## 3.4 Model Results

After feature engineering and data preprocessing, we can finally build our model. The training features will be those in numeric features and the categorical features.

We utilise ColumnTransformer and Pipeline to streamline the model-building process. The preliminary results yield a final RMSE score of 0.64527, thus showing that the model is accurate and useful in forecasting. RMSE measures the average magnitude of the errors between the values predicted by the model and the actual values. An RMSE of 0.64527 suggests that, on average, the model's predictions deviate from the actual values by approximately 0.64527 units.

For further confirmation, we perform a 10-fold cross-validation on the model. After performing CV, we get an average RMSE of 0.6275. This boosts the confidence of the usability and accuracy of the model.

## 3.5 Model Diagnostics

After getting the results, we perform diagnostics for the advanced model using Q-Q plots.

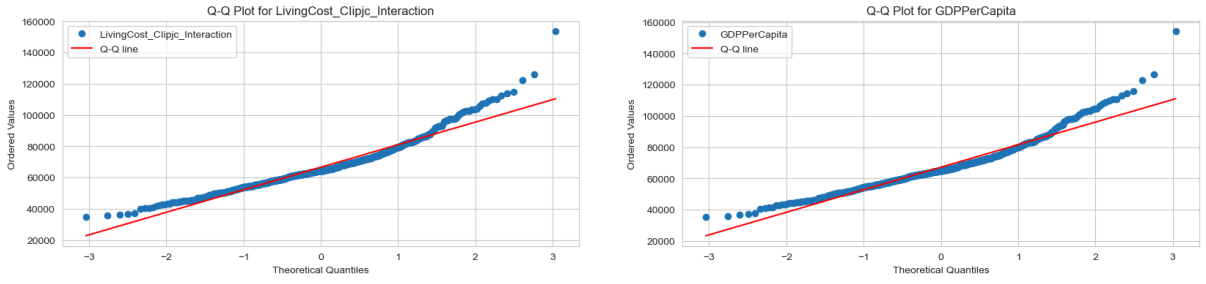


*Fig 8. Q-Q Plot of the predictors against MedianIncome22 on test set*

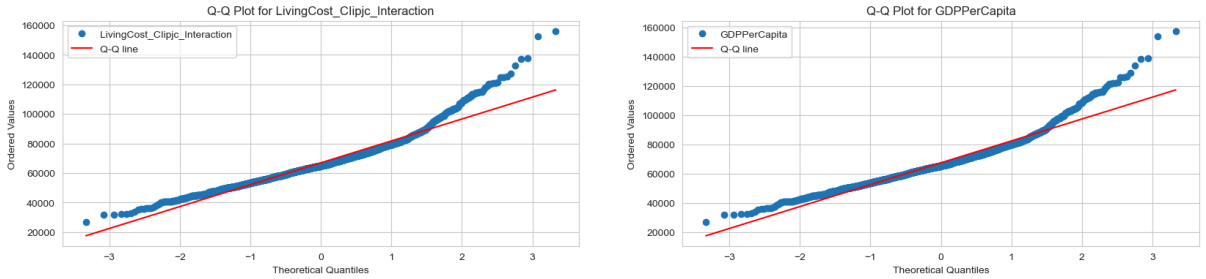


*Fig 9. Q-Q Plot of the predictors against MedianIncome22 on train set*

1. LivingCost_Clipjc_Interaction:

The plot shows that the lower and middle quantiles of the data align reasonably well with the red line, which represents the theoretical quantiles if the data were normally distributed.

However, in the upper quantiles, the data points deviate significantly from the line, indicating that there are more extreme values in your data than would be expected in a normal distribution (right-skewed).

This suggests the presence of outliers or a long tail on the right side of the distribution, which is typical for data involving income or costs where a small number of cases can have very high values. Such outliers might have a high influence on the model.

2. GDPPerCapita:

Similar to the first plot, the lower and middle quantiles align well with the theoretical normal distribution. The upper quantiles again show a deviation from the line, indicating the data is right-skewed with outliers or extremely high values.

This skewness in the upper tail is often observed in economic data where wealth or production is concentrated among a smaller portion of the population or regions.