UNFCCC GREEN CITIES COMMITMENT ANALYSIS

Econometric Analysis of the determinants of Brazillian cities to sign up and undertake actions as part of the UNFCCC

NAME

Nathan Clarke

STUDENT NUMBER

20387306

TABLE OF CONTENTS

EXECUTIVE SUMMARY

RESEARCH PURPOSE

In this research, I will conduct a comprehensive analysis of the relationship between the cities of Brazil and the UNFCCC. I will focus on the key determinants of the cities: 1. Sign up to the UNFCCC, and 2. Undertake individual actions as part of the UNFCCC. Using both supervised and unsupervised techniques, I will assess the significance, magnitude and direction of linear and non-linear relationships between suitable predictor variables and all outcomes of interest.

RESEARCH DESIGN

I selected 16 features applicable to this use case, which I classified into geographic, demographic, wealth and industry groups. After identifying the variables to be used in my research, I matched each of the datasets on city names using Levenshtein Distance to select the most similar match. After dealing with the imbalance in the outcome of interest, I examined the determinants of a city being in the UNFCCC by implementing linear and non-linear models to assess the statistical significance, direction and magnitude of the relationship between my features and the 'inUNFCCC' target variable. I followed this using a clustering algorithm to identify groups of cities with unique characteristics and studied how likely these groups of cities have been to sign up to UNFCCC. Once I had concluded my work on part i, I filtered my data on cities that had signed up to the UNFCCC and examined determinants for their signing up to the individual actions using a similar approach to that of part i.

RESEARCH LIMITATIONS

Throughout my research I faced a number of substantial limitations which hindered the quality of my research.

1. Imbalanced data in the inUNFCCC variable. Only 5.06% of cities were signed up, so it is difficult to draw meaningful interpretations about this class. I overcame this by balancing the data using Random Oversampling.
2. Matching datasets on city names. Given the inconsistent use of syntax across datasets, I could not match directly on the cities names. Instead, I employed Levenshtein distance to match cities across datasets based on their most similar matches.
3. Access to variables. If I had access to more data, I would have achieved better model fits and found more significant determinants to explain more of the variance in cities signing up and taking individual actions.

FINDINGS

I discovered a number of important determinants for a city signing up to the UNFCCC, which were consistent across linear and non-linear models. I found that rural cities with a large dependence on agriculture and industry are less likely to sign up. On the flip side, large, urban, wealthy regions with a large number of government, service and financial service jobs are more likely to sign up. The clustering algorithm was successful in identifying four distinct groups within the full dataset of Brazillian cities that are differentiated by unique sets of characteristics. I found that group zero, with high levels of wealth, quality of living and population, was twice as likely to sign up (10.2%) than the average (5.09%) in Brazil. While group one with low wealth, smaller populations and lower quality of living were half as likely to. Group two displayed characteristics consistent with rural agricultural communities. While this group had similar levels of wealth and quality of living to group zero, the cities in this group were less likely to sign up (3.38%). The final group was dependent on industry features and had 0% in the UNFCCC. I found that, in general, these relationships held for the cities undertaking actions once they had signed up. However, there were some exceptions, namely a change in direction of the relationship for exposure to secondary industries.

RESEARCH IMPLICATIONS

To increase the number of cities signed up to above their current 5.09% and to increase commitments for cities already signed up, I recommend Brazil target the specific groups I have outlined in this report. I recommend incentivising each group using drivers specific to their group.

RESEARCH PURPOSE

This report aims to conduct a comprehensive analysis of the relationship between the cities of Brazil and the UNFCCC. My research will focus on the key determinants of these cities: 1. signing up to the UNFCCC and 2. undertaking individual actions recorded by the UNFCCC.

My analysis will delve into each of the points above, employing a range of econometric, statistical, and machine-learning models and algorithms. Using supervised and unsupervised techniques, I will assess the significance, magnitude and direction of linear and non-linear relationships between suitable predictor variables and all outcomes of interest.

Drawing on academic literature and news articles, I will integrate background knowledge of Brazil's relationship with the UNFCCC and broader climate change initiatives to enrich the intuition behind my findings.

The broader objective of my research is to assess Brazil's commitment to the UNFCCC and its initiatives. To achieve my overall objective, I have defined a list of research questions below.

RESEARCH QUESTIONS

1. What are the main determinants of Brazilian cities signing up to the UNFCCC?
2. What are the main determinants of Brazilian cities undertaking individual actions recorded by the UNFCCC?
3. Are there different subgroups of Brazilian cities with unique characteristics who are more likely to sign up than others?
4. What do these determinants imply for the Brazilian government's Climate Plan?

Throughout my analysis, I will aim to frame my results around answering these core questions, as I feel that they encompass my broader research objective. In the **Conclusion section**, I have provided my findings in relation to these questions, along with some commentary on the implications of these results for the Brazilian Government.

BACKGROUND RESEARCH

What is the UNFCCC?

The United Nations Framework on Climate Change or UNFCCC began in 1994 with the ultimate aim of "preventing 'dangerous' human interference with the climate system". The convention planned to achieve this goal through the following points (UNFCCC, no date) :

1. Recognising the problem – forced members to act in the interest of human safety, even if there was scientific uncertainty.
2. Setting a lofty but specific goal – to limit greenhouse gas levels to a degree at which they would not have a negative impact on the climate system.
3. Begin with a focus on developed countries
4. Direct new funds to climate change activities in developing countries **–** Developed countries signed up would provide financial support for developing countries to reach their targets.
5. Close monitoring of progress **–** Developed countries must provide regular reports on the policies and measures they are undertaking.
6. Strike a balance for developing countries **–** The convention recognised that emissions from developing countries will grow, but with the help of developed countries, this growth can be controlled and limited while not hindering their economic growth.

Critics of the UNFCCC suggest that the effectiveness of the convention was limited by exempting developing countries from the emission reduction targets (UNFCCC, no date). This led to China and India (the first and third largest emitters) experiencing huge increases in emissions over the course of the convention (Shukla, 2023).

However one of the key achievements of the convention has been in developing a reporting framework for information on emissions and other commitments (NAEI, 2023). In fact, it is through the same reporting framework that I have the data for this research.

Brazilian cities commitment to climate change

As my research focuses mainly on the interactions of Brazil's cities with the UNFCCC, it is useful to understand some background on their adaptation to climate change. In their 2019 paper, Di Giulio et al explore the drivers of climate adaptation in six large Brazilian cities. They find that the following factors have a significant impact on the likelihood of cities taking part in climate initiatives (Di Giulio, 2019).

1. administrative practices
2. political will
3. level of commitment
4. mismatch between the scale of urban issues and the extent of local
5. government authority
6. pressures from the private sector

For my research, I plan to explore drivers such as these but across a much larger number of cities and with reference specifically to the climate initiatives of the UNFCCC.

RESEARCH DESIGN

For my analysis, I selected a number of feature variables that I felt could be useful to my overall analysis. The majority of these features came from Christina Parada's 'Brazil Cities' dataset (Parada, 2022). This dataset contains 79 features gathered from several public and government websites. Below, I have presented a table of my selected features and a short description.

Feature Name (Shorthand)	Units	Data Type	Description
Geographic			
Planted Area (plant)	1000 Hectares	Continuous	Planted area associated with the city
Crop Production (crop)	$1000	Continuous	Value of crops produced
Area (area)	Square Kilometers	Continuous	Area of the city
Rural (rural)	-	Binary	Whether the city is considered to be Rural or not
Population Demographic			
Population (pop)	-	Continuous	Population of the city
HDI (hdi)	$1000	Continuous*	HDI Human Development Index
Life Expectancy (life)	-	Continuous*	HDI Life Expectancy Index
Education (edu)	-	Continuous*	HDI Education Index
Wealth			
GDP (gdp)	-	Continuous	GDP per Capita
Municipal Expenditure (munex)	1000 people	Continuous	Municipal Expenditure per 1000 people
Industry			
Primary Proportion (pprop)	-	Continuous*	Proportion of companies focused on Primary Industry
Secondary Proportion (sprop)	-	Continuous*	Proportion of companies focused on Secondary Industry
Tertiary Proportion (tprop)	-	Continuous*	Proportion of companies focused on Tertiary Industry
GVA Primary (gvap)	$1000	Continuous	Gross value added from the primary sector
GVA Secondary (gvas)	$1000	Continuous	Gross value added from the secondary sector
GVA Tertiary (gvat)	$1000	Continuous	Gross value added from the tertiary sector

** Scale between zero and one*

Once I had selected these variables, I carried out feature engineering in the form of encoding, log and square root transformations, and handling of missing values. This ensured that the variables were of sufficient quality to be used in my research. *For more information on the feature engineering, statistical and geographical distribution for each variable, see the* ***Data and Methodology Section***.

For my research, I had 10 target variables, which I have presented in the table below.

Feature Name	Data Type	Description
i) Target		
inUNFCCC	Binary	Indicator for whether a city is signed up with the UNFCCC
ii) Targets		
hasCommitments	Binary	Indicator for whether a city has commitments
hasEmissionInventory	Binary	Indicator for whether a city has an emission investory
hasInitiativePartcipations	Binary	Indicator for whether a city has participation in UNFCCC initiatives
hasActionsUndertaken	Binary	Indicator for whether a city has undertaken any individual actions
hasImpact	Binary	Indicator for whether a city has an emission investory
hasMitigations	Binary	Indicator for whether a city has an emission investory
hasAdaptations	Binary	Indicator for whether a city has an emission investory
hasRiskAssessment	Binary	Indicator for whether a city has an emission investory
hasClimateActionPlans	Binary	Indicator for whether a city has an emission investory

As you can see, all of these variables are binary, and as such, the models which I use throughout my research will be binary classification models. In the first part of my research, I will examine the determinants of a city being in the UNFCCC (1 class of inUNFCCC), and in the second part, I will look at cities that have already signed up and explore the determinants of them undertaking the actions relating to the remaining variables above.

After identifying the variables to be used in my research, I matched each of the datasets using the names of the Brazilian cities. Since the city names were not exactly the same across all datasets, I used Levenshtein distance to quantify the similarity between the city names and select the pairings with the highest similarity (*See* ***Methodology Section*** *for more information on how I used Levenshtein distance*).

Content with the quality of my matching and selection of target and feature variables, I began examining the determinants of a city being in the UNFCCC. My research here was twofold; firstly I implemented linear and non-linear models to assess the statistical significance, direction and magnitude of the relationship between my feature variables and the 'inUNFCCC' target variable. I followed this using a clustering algorithm to identify groups of cities with unique characteristics and studied how likely these groups of cities have been to sign up for UNFCCC historically (See the ***Data and Methodology section*** *for more information on the models and algorithms used*). Before beginning my research for part i, I noted a large imbalance in my dataset between cities that had signed up and those that had not. To resolve this issue, I balanced the dataset so my models could learn as much as possible about my outcome of interest. (*See* ***the Data and Methodology section for more information on*** *how I balanced the dataset*).

Once I had concluded my work on part i, I filtered my data on cities that had signed up to the UNFCCC and examined determinants for signing up to the individual actions listed in the targets table above. I once again employed a logistic regression model to discover the magnitude, direction and significance of the relationship between my features and each of the target variables.

See the **Conclusions** section for the findings and implications of my research. **For my full analysis, see both sections of Appendix I**

RESEARCH LIMITATIONS

Throughout my research I faced a number of substantial limitations which hindered the quality of my research.

Firstly, the classification of inUNFCCC was difficult due to the large imbalance between cities that have signed up and those that have not. While I overcame this issue using Random Oversampling, I could have generated more accurate results if the dataset was not so imbalanced. This imbalance also resulted in a very small number of observations in part ii when I was looking only at cities that had already signed up.

Another key difficulty I faced was matching datasets on city names. Given the different use of syntax across datasets, I could not match directly on the cities. Instead, I employed Levenshtein distance to match cities across datasets based on their most similar match (as long as it was above a tolerance). While this process provided excellent results, any errors could lead to misspecification in my models and biased results.

The main issue which hindered my research was the availability of quality features. After searching government sources and online datasets I found a total of 79 features, however only 16 of those were suitable for this use case. If I had access to more data, for example, political or weather-based features, I would have achieved better model fits, found more significant determinants and been able to explain more of the variance in cities signing up and in taking individual actions. These extra features would increase the robustness of my research and provide better performance across all of my models.

I also found that many of the features I selected had extremely skewed and heavy-tailed distributions. To overcome this, I used square root (when zero was in the dataset) and log transformations.

Overall, while I faced a number of limitations throughout my research, I feel that I have carried out sufficient actions to mitigate these limitations and provide meaningful, insightful results.

CONCLUSION

RESEARCH FINDINGS

Determinants for a city to sign up for the UNFCCC

From my supervised learning models, I discovered a number of important determinants for a city signing up to the UNFCCC, which were consistent across linear and non-linear models. I found that rural cities with a large dependence on agriculture and industry are less likely to sign up, while high levels of wealth, population and jobs in the service industry all make a city more likely to be a member. These relationships are consistent with rural, agricultural communities being against the guidelines of the UNFCCC as they would suggest a reduction in logging, cattle farming and other agricultural activities associated with negative impacts on climate change. Similarly, the negative relationship found with activity in the industrial sector is consistent with companies being hesitant to agree with actions such as reduced emissions. On the flip side, large, urban and wealthy regions with a large number of government, service and financial service jobs are more likely to sign up. These types of cities would have less economic risk from alignment with the UNFCCC guidelines. These cities also have a higher level of municipal expenditure meaning that initiatives which the Brazillian government have aligned themselves with are more likely to be followed. While both the LPM and Logistic models had low R^2, it is clear that there are a number of omitted variables which would help to explain more of the variance in inUNFCCC; I am happy that the features I discovered tell a clear story on some of the important determinants for this problem. My alternative model is the Random Forest, which found that population, proportion of primary companies, crop production and municipal expenditure were the most important features in determining cities that sign up. These findings copper fasten the results of the LPM and Logistic. However, the magnitude of the importance of the population size relative to other features was a stark observation.

Moving on to the unsupervised learning model. The clustering algorithm was successful in identifying distinct groups within the full dataset of Brazillian cities that are differentiated by unique sets of characteristics. By comparing groups zero and one, I found that group zero, with high levels of wealth, quality of living and population, was twice as likely to sign up (10.2%) than the average (5.09%) in Brazil. While group one with low wealth, smaller populations and lower quality of living were half as likely to. This result mirrors the findings from my supervised models in finding that characteristics such as wealth, quality of living, life expectancy, city size, and whether a city is rural or urban are important determinants of its likelihood of signing up to the UNFCCC. By examining group two, I found that this group displayed characteristics consistent with rural agricultural communities mainly focused on primary activities. While this group had similar levels of wealth and quality of living to group zero, the cities in this group were less likely to sign up (3.38%) than the average and much less than group zeros 10.2%. This result is consistent with my previous findings on agricultural communities. Finally, although group three was very small and therefore did not carry much inference, it further copperfastened the importance of industry exposure to the likelihood of being in the UNFCCC as this group had extremely low exposure to tertiary industry and extremely high to primary so as expected had a very low commitment to the UNFCCC (0%).

DETERMINANTS FOR A CITY TO UNDERTAKE INDIVIDUAL ACTIONS RECORDED BY THE UNFCCC

After carrying out my analysis into determinants of other individual actions recorded by the UNFCCC, I found that there were a number of features which were consistently important across different types of actions while others were more important for specific actions. The population and GVA from services features were important across a number of action variables. So cities with large populations and exposure to the services industry are more likely to undertake a range of actions associated with the UNFCCC. These findings mirror what I found for the cities signing up to the UNFCCC. So it appears that once cities have signed up, this relationship continues for determining actions.

Quality of living features such as HDI and Education were important for determining some of the targets, namely city having actions and adaptations. However, both of these models had a poor fit. Exposure to secondary or industrials was important in determining commitments, mitigations and climate action plans. This result is interesting as I found before that exposure to secondary made a city less likely to sign up. So it appears that these cities may have industrial companies who are driving the push towards green actions, making their cities more likely to continue towards actions after signing up. These three models had a good fit relative to the other models and the inUNFCCC models from i.

On the other hand, emissions inventory, initiative participation, and climate action plan each had only one significant determinant: population, population, and GVA from secondary, respectively. This indicates that determining the actions and planning relating to these features may be more complex and diverse than my features could capture. However, the emissions inventory model had the highest R^2 value relative to all other models, which suggests it may just be that its variance can be well explained using the simple population model.

RESEARCH IMPLICATIONS

To increase the number of cities signed up to the UNFCCC to above their current 5.09% average and to increase commitments for cities already signed up, I recommend Brazil target the specific groups I have outlined in this report.

In group zero cities, I believe Brazil has done a good job in growing sign-ups and commitments, as 70% of all cities currently signed up are a part of this group. However, only 10.2% of this group is signed up, so there is still a lot more work to be done to increase uptake for these wealthy, large, high-quality living, urban regions. For group one, I believe it could be more difficult to increase uptake given the socio-economic background of these cities. However, with only 2% of this group signed up, any increase would be helpful. From my research, I discovered that Municipal expenditure has a positive effect on the likelihood of a city signing up, so by offering increased government funding in these areas, Brazil can expect an increased uptake in commitment to the UNFCCC.

Given the strong negative relationship I identified between cities with a primary/agricultural focus and interactions with the UNFCCC, the Brazillian Government should consider providing incentives or regulations for these cities (group 2 type cities) to be more active in relation to signing up to the UNFCCC and its individual actions. By targeting this group I believe Brazil could have success in increasing uptake as this group has similar characteristics to group zero aside from its focus on primary activities. This approach could also be taken more broadly in order to try and mitigate the influence of industry-type features on the likelihood of a city signing up and taking individual actions.

APPENDIX I – RESULTS

*Note: Throughout my research I have used ***,** and * to represent statistical significane at 1%,5% and 10% levels respectively*

DETERMINANTS FOR A CITY TO SIGN UP FOR THE UNFCCC

OVERVIEW

In this section, I will attempt to identify key determinants for a Brazilian city to sign up to the UNFCCC. I will do this by quantifying the statistical significance, direction and magnitude of the relationship between these determinants and whether a city is signed up.

In order to get a broad view of the relationships, I will fit both linear and non-linear classification models as well as a clustering algorithm before comparing the performance of models based on different categories of features.

*For more information on all of the models used, see the **Data and Methodology** section*

SUPERVISED LEARNING MODELS

LPM

I began by fitting a linear probability model using all of my predictor variables. The output of this regression is summarised below. It is worth noting that while this model has the advantage of interpretability of coefficients, it suffers in terms of quality of predictions for the reasons discussed in the **Data and Methodology** section.

LPM	Coefficients
intercept	0.5127***
plant	-0.0611***
crop	0.0385***
area	-0.0054
rural	-0.1144***
hdi	0.0078
life	-0.0436*
pop	0.2011***
edu	0.0336
gdp	0.0909***
munex	0.015*
pprop	0.0112
sprop	-0.0103
tprop	0.0411***
gvap	-0.0414***
gvas	-0.0485***
gvat	0.0249**
Adj R2	0.339
AIC	3112
BIC	3214

Significance

As you can see from this regression summary of the LPM model, 9 out of the 16 features I selected are, in fact, significant at a 5% level or lower. These variables are: plant, crop, rural, pop, gdp, tprop, gvap, gvas and gvat. The intercept was also statistically significant. However, this does not have a meaningful interpretation for this model.

Negative Relationships

Firstly, looking at 'plant', this model suggests that for every additional 1000 hectares of the planted area a city has, its probability of signing up decreases by 6.11 percentage points. Similarly, the gross value added from primary industries, which would encompass forestry as well as other fields of agriculture, also has a negative coefficient. This indicates that for every additional $1000 in gross added value from primary activities, the probability of signing up decreases by 4.14 percentage points. The gross value added from secondary (industrials) also has a negative relationship with a city signing up for the UNFCCC, with a 4.85 point drop for every extra $1000 in GVA added. The 'rural' indicator suggests that if a city is rural or remote then the probability of signing up is 11.44 points lower than non-remote cities. So, at first glance, it appears that rural cities with a large dependence on agriculture and industry are less likely to sign up to the UNFCCC.

Positive Relationships

In terms of features which make a city more likely to sign up, this model found that a one-unit increase in log population is associated with a 20.1 point increase. Similarly, a 1 unit increase in GDP per Capita is associated with a 9.09 point increase, and a 1 point increase in the proportion of tertiary companies is associated with a 2.49 point increase. So overall this model suggests that wealth, population size and jobs in services companies all indicate that a city is more likely to sign up.

Note: This model has an adjusted R^2 of 0.339, which indicates that it only accounts for a small portion of the overall variability in the 'inUNFCCC' variable. Therefore there may be other features, which I do not have access to, which are needed to explain more of the variability.

LOGISTIC

After fitting LPM, I moved on to logistic regression. For more on this model, see the **Data and Methodology** section. Below, I have presented the model's output, where inUNFCCC is classified using all predictors.

Significance

As you can see, there are some changes in the significance of the features in this model compared to LPM. This is due to the ability of the logistic model to capture more non-linear relationships than the LPM through its use of the sigmoid function. Newly significant features include the municipal expenditure and gross added value from the tertiary or services industry and the proportion of companies in the primary and secondary sectors.

Relationships

One downside of logistic regression is that the coefficients are now interpreted as the effect on the log odds rather than the probability, so its meaning is more difficult to conceptualise. However like I found in the LPM, the planted area, GVA from agriculture and industry and a city being rural all make it less likely that they will sign up to the UNFCCC. On the other hand, increases in features such as population, the proportion of and gross value added from tertiary companies all increase the log odds of a city signing up. Interestingly, the municipal or government expenditure in the cities increases the log odds of a city signing up.

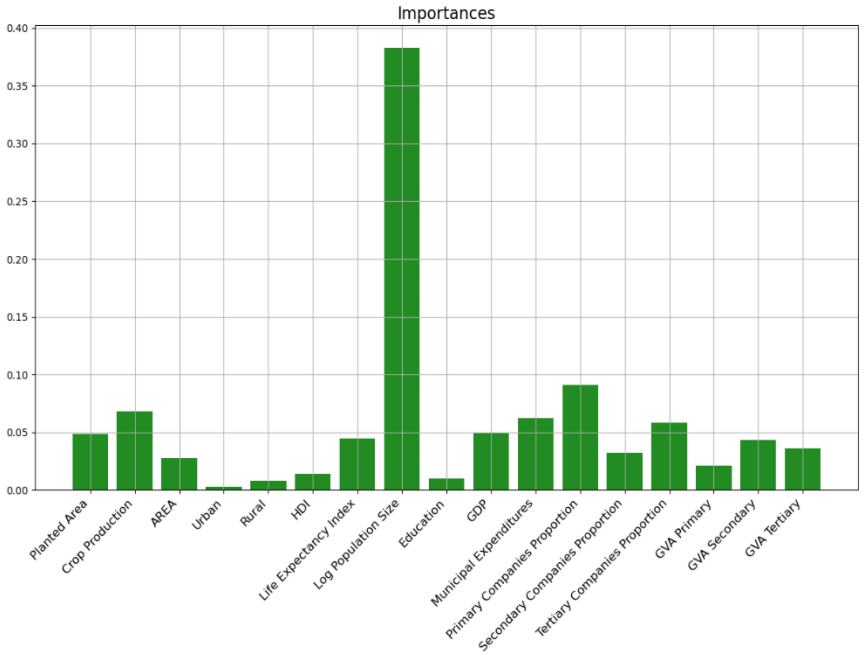
So overall while we see similar results using a model which accounts for non-linear relationships, some interesting additional information is the positive effect of government spending on cities signing up.

Logistic	Coefficients
intercept	0.2684***
plant	-0.3476***
crop	0.0822
area	-0.0915
rural	-0.6605***
hdi	0.3114
life	-0.2958**
pop	1.5054***
edu	-0.1887
gdp	0.4969***
munex	0.156***
pprop	0.1753**
sprop	-0.1273**
tprop	0.3151***
gvap	-0.1263**
gvas	-0.3083***
gvat	0.2387***
Adj R^2	0.3037
AIC	2912
BIC	3014

TREE BASED MODEL

By fitting a tree based model, I hope to explore the overall importance of different features, no matter their direction and significance from the previous models. A tree-based model, such as the random forest I fit here, aims to minimize nodal impurity, meaning they base their classification on a number of rules which separate the feature space in a manner that results in the best classification. Therefore, these models can produce a 'feature importance' statistic, which quantifies how important each of the features is in reducing nodal impurity and improving the model predictions. As this algorithm operates very differently from the previous two probability models I examined, I believe it may yield some contrasting results.

	Importance
pop	0.382535
pprop	0.090686
crop	0.067968
munex	0.060845
tprop	0.057193
gdp	0.054635
plant	0.048953
gvas	0.046019
life	0.044481
gvat	0.035938
sprop	0.029281
area	0.027537
gvap	0.022159
hdi	0.016953
edu	0.011757
rural	0.003059



The main findings from this output are that the population of the city is the most important feature, with almost four times the importance of the next most important feature. Interesting the next two most important features are the proportion of primary companies and the crop production of the cities. This result is consistent with my findings from the previous two models as it echos the importance of wealth, population and exposure to certain industries such as agriculture.

Summary of results from supervised models

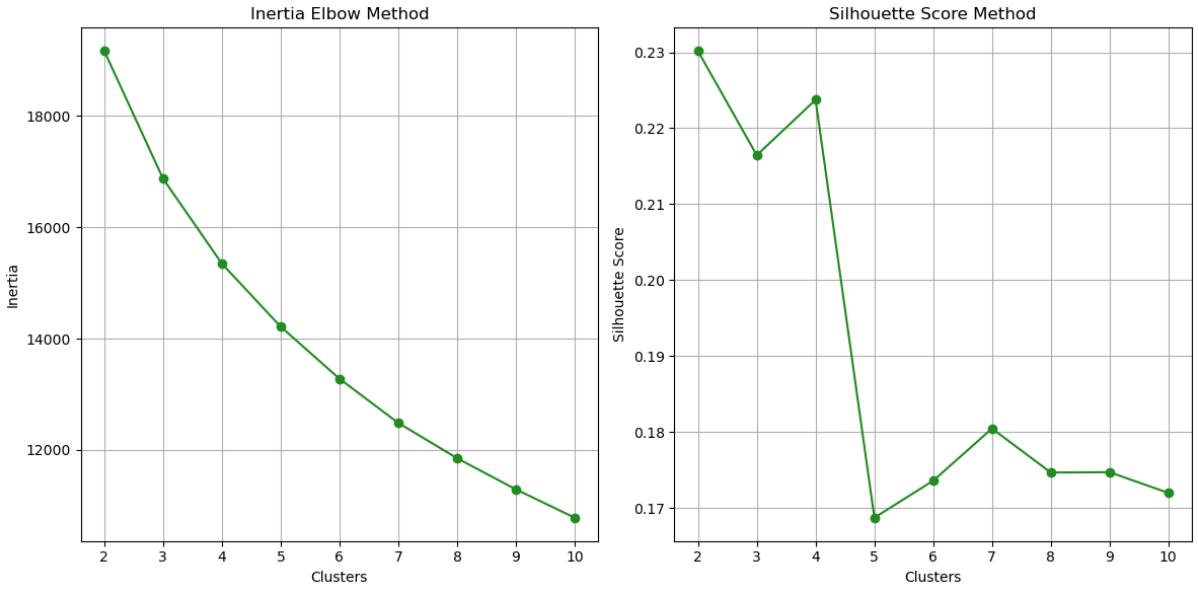
So, to summarise the findings of my supervised learning models, I discovered a number of important determinants for a city signing up to the UNFCCC, which were consistent across linear and non-linear models. I found that rural cities with a large dependence on agriculture and industry are less likely to sign up, while high levels of wealth, population and jobs in the service industry all make a city more likely to be a member. These relationships are consistent with rural, agricultural communities being against the guidelines of the UNFCCC as they would suggest a reduction in logging, cattle farming and other agricultural activities associated with negative impacts on climate change. Similarly, the negative relationship found with activity in the industrial sector is consistent with companies being hesitant to agree with actions such as reduced emissions. On the flip side, large, urban and wealthy regions with a large number of government, service and financial service jobs are more likely to sign up. These types of cities would have less risk of economic decline from alignment with the UNFCCC guidelines. These cities would also have a higher level of municipal expenditure, meaning that initiatives which the Brazillian Govemment have aligned themselves with are more likely to be followed. While both the LPM and Logistic models had low R^2, it is clear that there are a number of omitted variables which would help to explain more of the variance in inUNFCCC; I am happy that the features I discovered tell a clear story on some of the important determinants for this problem. My alternative model is the Random Forest. found that population, proportion of primary companies, crop production and municipal expenditure were the most important features in determining cities that sign up. These findings copper fasten the results of the previous two models. However, the magnitude of the importance of population relative to other features was a streak observation.

UNSUPERVISED LEARNING MODEL

CLUSTERING

Throughout my interpretation of the LPM, Logistic and Random Forest, I noticed that the significant or important features suggested unique city characteristics, which made them more or less likely to sign up for the UNFCCC. For example, cities with an agricultural or rural focus tended to be less likely to sign up, while cities with high GDP per capita, government spending and income from services tended to be more likely to. In order to characterise cities within Brazil which have different characteristics I have used the unsupervised K-Means Clustering algorithm. This algorithm identifies groups of cities with similar characteristics within the dataset based on their values for all of the predictor variables. See the **Data and Methodology** section for more information on this algorithm. In this section I will analyse the results of this algorithm and interpret what characteristics make a city more or less likely to sign up to the UNFCCC. Firstly, I will select the optimal number of clusters or groups for the algorithm to use.

When selecting the optimal number of clusters, I take into account the inertia and silhouette scores.



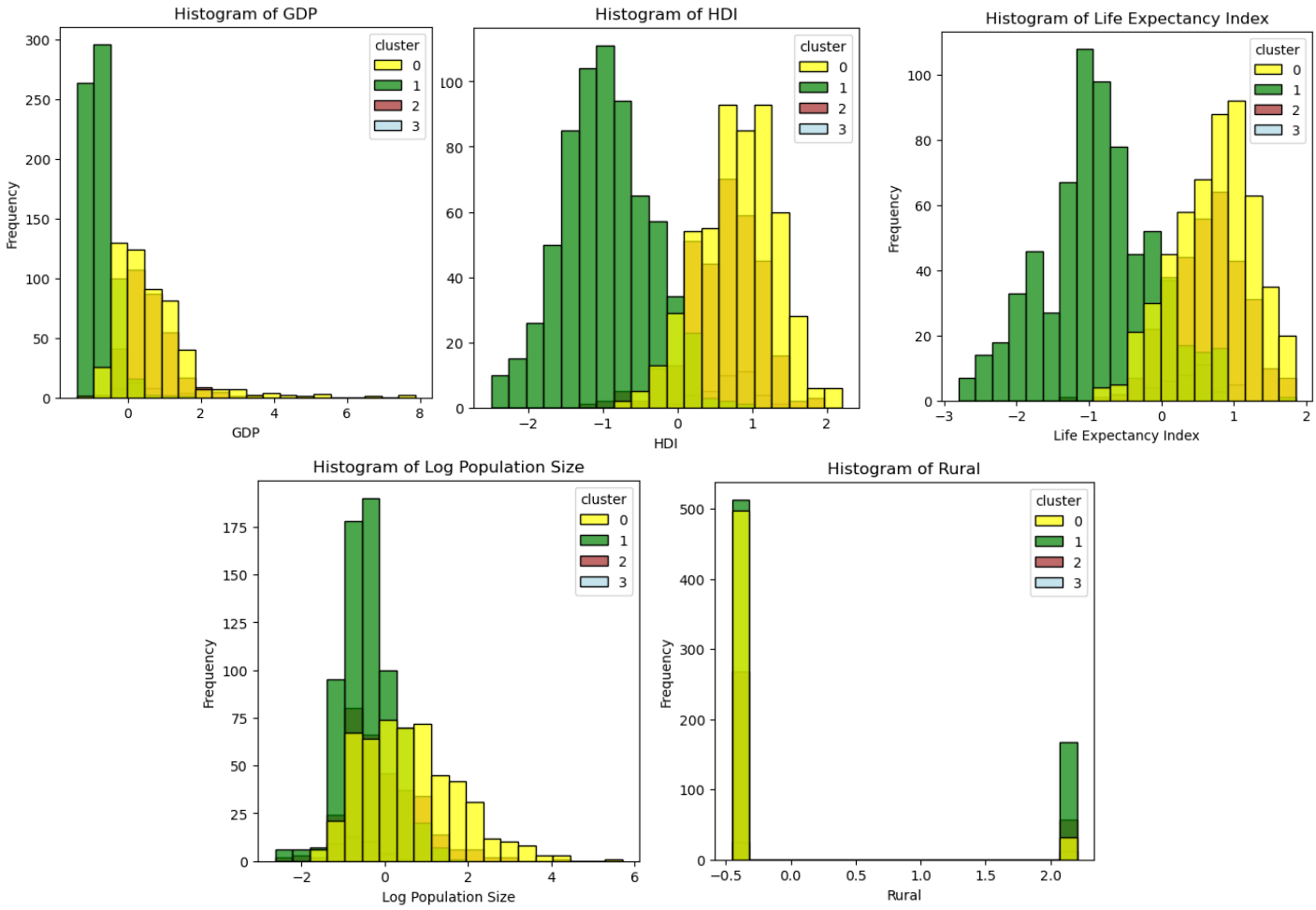
The inertia is a measure of the distance between each observation in a cluster and its centroid. Thus, it is a measure of the quality of the clustering where lower values are better. The aim of selecting K from this method is to find an 'elbow' point on the curve where an increase in clusters only leads to a linear reduction in inertia. The Silhouette score is a measure of how similar observations are to their own cluster relative to others; in this case, we would like to mind a maximum. Given these selection criteria, K=4 arrears is the optimal value. As such I have selected K=4 as my optimal number of clusters. Below I have provided the count of cities in each group and the percentage in each group which is signed up to the UNFCCC. *Note that the percentage signed up to the UNFCCC in the full dataset is 5.09.*

Group	Number of Cities	% in UNFCCC	% in UNFCCC from group*
0	529	10.20%	70%
1	680	2.20%	11.25%
2	325	3.38%	18.75%
3	37	0%	0%

*Of cities signed up, what percentage is from each group

Group 0 and 1

Below, I have provided histograms to display the different characteristics of groups one and two. I also note that there are a similar number of cities in each of these groups (529 in group zero and 680 in group one)

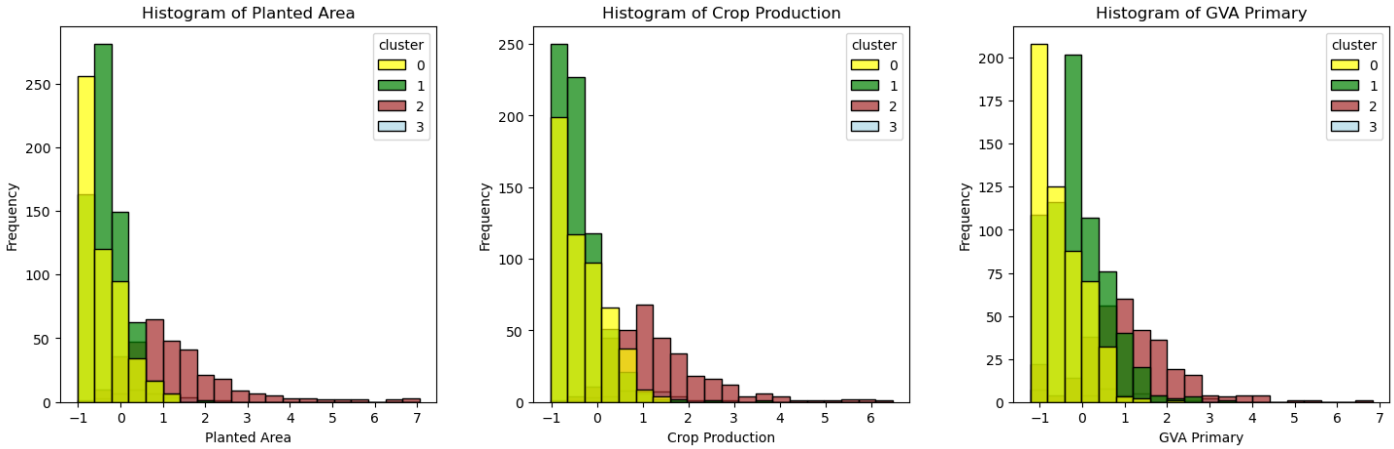


As you can see, these plots display clear differences in group one and group two. Firstly, group zero's GDP per Capita distribution is much further right than group one, indicating a higher average and more extreme values to the right. In terms of HDI, group zero's distribution is also shifted further right than group one; this effect can also be observed from the life expectancy distribution. Group zero also contains more extremely large values of population, and the majority of them are classed as urban rather than rural. So it appears overall that group zero are large urban cities that are, on average, wealthier, with a higher quality of living and a higher life expectancy than the smaller, more rural cities in group one. With relation to the percentage signed up, group zero has 10.2%, which is over double the average of the full dataset, while group one has only 2.2% signed up, which is less than half that of the full dataset. Interestingly, group zero contains 70% of all cities signed up, whereas group one lasts only 11.25%.

So, I conclude that characteristics such as wealth, quality of living, life expectancy, city size, and whether a city is rural or urban are important determinants of its likelihood of signing up to the UNFCCC.

Group 2

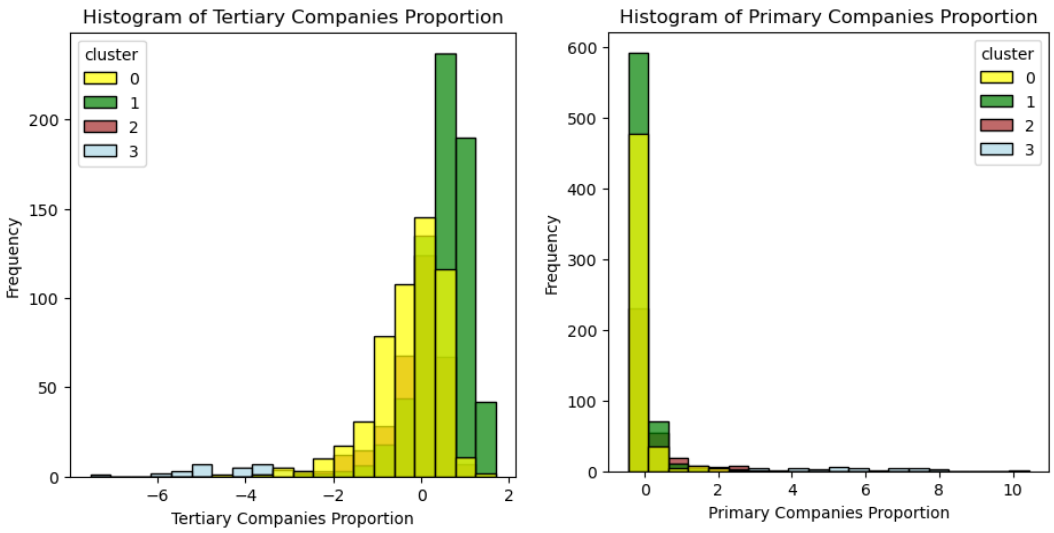
Group two is smaller than the previous two groups, with 325 cities. However, it does have its own unique characteristics, which the plots below display.



Group two appears to represent the agricultural cities of Brazil. This group's distribution of planted area, crop production and gross value added from primary industries are shifted further right than the corresponding distributions for groups zero and one. By observing the plots used for group one, I also note that these cities have a similar GDP, HDI and life expectancy to group one, which indicates that while these farming cities may be more rural, they still have similar wealth and quality of living factors to group zero. However, there is a stark difference in groups zero and one's uptake to the UNFCCC, with group two only having 3.38% (below the average of the full dataset) signed up relative to group zero's 10.2%. This is an interesting observation, as it shows a clear distinction in sign-ups from cities with similar levels of wealth and quality of living but who focus on agriculture or primary industry. This group contains only 18.75% of all cities committed to the UNFCCC.

Group 3

The final group that this unsupervised clustering algorithm identified is the smallest of the four, with only 37 cities, but interestingly, the percentage signed up to the UNFCCC in this group is 0.



This group does not appear to stand out from other groups in terms of wealth, geography, or population features. However, it does take extreme values in some industry features. For example, this group has a very low proportion of tertiary companies and a very large proportion of primary companies. As this group is so small, it is difficult to draw meaning from its differences from other groups; however, its distinction in these industry features indicates the importance of a city's industry as a determinant of their likelihood of being a member of the UNFCCC.

Summary of results from unsupervised models

To conclude, my unsupervised clustering algorithm was successful in identifying distinct groups within the full dataset of Brazillian cities that are differentiated by unique sets of characteristics. By comparing groups zero and group one, I found that group zero, with high levels of wealth, quality of living and population, was twice as likely to sign up (10.2%) than the average (5.09%) in Brazil. While group one with low wealth, smaller populations and lower quality of living were half as likely to. This result mirrors the findings from my supervised models in finding that characteristics such as wealth, quality of living, life expectancy, city size, and whether a city is rural or urban are important determinants of its likelihood of signing up for the UNFCCC. By examining group two, I found that this group displayed characteristics consistent with rural agricultural communities mainly focused on primary activities. While this group had similar levels of wealth and quality of living to group zero, the cities in this group were less likely to sign up (3.38%) than the average and much less than group zeros 10.2%. This result is consistent with my previous findings on agricultural communities. Finally, although group three was very small and therefore did not carry much inference, it further copperfastened the importance of industry exposure to the likelihood of being in the UNFCCC as this group had extremely low exposure to tertiary industry and extremely high to primary so as expected had a very low commitment to the UNFCCC (0%).

DETERMINANTS FOR CITY TO UNDERTAKE INDIVIDUAL ACTIONS RECORDED BY THE UNFCCC

The second part of my research is in relation to the determinants of cities that have signed up to the UNFCCC to undertake individual actions which are recorded by the UNFCCC. The UNFCCC cities dataset contains a number of variables which relate to the different actions a city may or may not have taken part in. After removing any such variables which were indiscriminate (entirely zero or one), I was left with a number of binary variables, each indicating whether or not a city has signed up to a specific commitment.

To isolate the key determinants for each of these variables, I ran a logistic regression using my features from part i. Below, I have presented any coefficients that were statistically significant at a 10% level or better, as well as the measures of fit: pseudo R^2, AIC, and BIC.

	Binary Targets	intercept	pop	gdp	hdi	edu	gvas	gvat	tprop	Pseudo R2	AIC	BIC
Target	hasCommitments		0.8628*				0.8738**			0.3587	86.7017	124.8141
	hasEmissionInventory	-52.22*	1.4667**							0.4306	64.0371	102.1495
	hasInitiativeParticipations		0.9735**							0.2084	95.3721	133.4845
	hasActionsUndertaken				102.7855***	-65.5534***				0.2319	112.4753	150.5877
	hasImpact									0.2718	63.0351	101.1475
	hasMitigations			-0.0628**			0.7157**	2.2761**		0.3386	100.5103	138.6228
	hasAdaptations				92.079***	-61.6849***				0.2357	113.6361	151.7486
	hasRiskAssessments	-81.9728**						-0.6567**	54.183**	0.4245	70.9219	109.0344
	hasClimateActionPlans						0.9427*			0.3045	76.5592	114.6716

Logistic Regression Summary for binary targets (including features significant at a 10% level and below)

At first glance, I note that many of the features which I found to be important determinants for a city to sign up to the UNFCCC are also important here, such as gdp, pop, tprop and gvat. These are also the variables which were useful in identifying Group 0, which is to be expected as 70% of cities signed up to the UNFCCC are from that group. As such, features which were important to other groups, such as Group 2 (the agricultural group, only 18.75% of sign-ups are from this group), are not statistically significant in any of the regressions above. This biased selection of groups explains why features like crop production and GVA from Primary do not appear in the summary table. Below I will discuss the determinants of each commitment type as well as the quality of fit which they provide.

Commitments

The 'hasCommitments' variable indicates whether or not a city has signed up for any of the commitments associated with the UNFCCC. When using my features to determine this variable, I found that only log population and GVA from services were the only statistically significant. A one unit increase in both leads to a circa 0.87 unit increase in the log-likelihood of a city signing up. With an Adjusted R^2 of only 35.87%, this model only explains a small portion of the variation in 'hasCommitments', however it is clear that the population and size of the services industry are important determinants. It is also worth noting that pop was only significant at a 10% level and gvas at 5%.

Emissions Inventory

The 'hasEmissionInventory' variable indicates whether or not a city has and reports an Emission Inventory to the UNFCCC. When using my features to determine this variable, I found that only the log population was statistically significant (at a 5% level). However, the model also had a significant intercept (10%). A one-unit increase in log population leads to a 1.4667 unit increase in the log-likelihood of a city keeping an emission inventory. With an Adjusted R^2 of only 43.06%, this model only explains a small portion of the variation in 'hasEmissionInventory'; however, it is clear that the population is an important determinant.

Initiative Participation

The 'hasInitativeParticipations' variable indicates whether or not a city has participated in any of the UNFCCC's initiatives. When using my features to determine this variable, I found that, once again, only the log population was statistically significant (at a 5% level. A one-unit increase in log population leads to a 0.9735 unit increase in the log-likelihood of a city

participating. With an Adjusted R^2 of only 20.84%, this model only explains a small portion of the variation in 'hasInitativeParticipations'; however, it is clear that the population is an important determinant.

Actions Undertaken

The 'hasActionsUndertaken' variable is probably the most important as it indicates whether or not a city has undertaken any actions as part of the UNFCCC. When using my features to determine this variable, I found that both HDI and Education are highly statistically significant (1% level). This is interesting as most other variables are mainly determined by population or industry features. For every one unit increase in HDI there is an increase of 102.7855 units in the log-likelihood of the city undertaking actions. However, this model suggests that a one-unit increase in score on the education index actually results in a decrease of 65.5525 units of log-likelihood. This is a very important observation as it suggests that cities with higher levels of education tend to be less likely to partake in the actions of the UNFCCC. However, the Adjusted R^2 of this model was only 23.19%, so it only explains a small portion of the variation in 'hasActionsUndertaken'; therefore, there may be other variables needed to truly understand how a city undertaking actions can be determined.

Impact

The 'hasImpact' variable indicates whether or not the city has had an impact as defined by the UNFCCC. When using my features to determine this variable, I found that none of them were significant below a 10% level. Therefore, the likelihood of a city having an impact, as defined by the UNFCCC, may require a set of variables that I do not have access to.

Mitigations

The 'hasMitigations' variable indicates whether or not a city has any mitigations as defined by the UNFCCC. When using my features to determine this variable, I found that GVA from both secondary and tertiary industries were significant (5%) determinants, with an increase in both leading to an increase in the likelihood of a city having mitigations. Interestingly, this was the only variable for which log GDP per capita was a significant feature, with a one-unit increase leading to a 6.28% decrease in the likelihood of a city having mitigations. So, it appears that cities with a large focus on services and industry are more likely than cities with a high public wealth to be classified as having mitigations. once again, only the log population was statistically significant (at a 5% level. A one-unit increase in the log population leads to a 0.9735 unit increase in the log-likelihood of a city participating. With an Adjusted R^2 of 33.86%, this model only explains a small portion of the variation in 'hasMitigations'.

Adaptations

The 'hasAdaptations' variable indicates whether or not a city has made any adaptations as defined by the UNFCCC. Interestingly, the model for this variable is very similar to the 'hasActionsUndertaken' variable. Both models have the same significant features, HDI and education, both with the same direction and very similar magnitude. So, it appears that cities that have actions undertaken also tend to have adaptations, which results in the similarity of these models. However, both models are also similar in terms of their low Adjusted R2, which is important to keep in mind when trying to draw inferences.

Risk Assessment

The 'hasRiskAssessment' variable indicates whether or not a city has undertaken a risk assessment as part of the UNFCCC. When using my features to determine this variable, I found that GVA from both tertiary industries and the intercept were significant at a 5% level. However, unlike the 'hasMitigations' variable, an increase in GVAT leads to a decrease in the likelihood of a city having a risk assessment. This model was also the only one to have the proportion of service companies as a significant and very positive feature. Interestingly, this suggests that cities with a large focus on services but with a low value-added from this industry are the most likely to have a risk assessment. This might suggest that cities like this have an abundance of low-income office workers. This type of labour force may be tasked with carrying

out tasks such as risk assessments. However, once again, I must caveat these results as the model explains less than half (42.45%) of the variation in 'hasRiskAssessment'.

Climate Action Plan

The 'hasClimateActionPlans' variable indicates whether or not a city has and reports a Climate Action Plan to the UNFCCC. When using my features to determine this variable, I found that only GVA from services was significant. However, this was only at a 10% level. So, overall, I cannot draw inferences from this model. This lack of explanatory power is also reflected by the low Adjusted R^2. Therefore, there may be other variables needed to truly understand how a city having Climate Action Plans can be determined.

Summary

So overall, I found that there were a number of features which were consistently important across different types of actions, while others were more important for specific actions. The population and GVA from services were important across a number of action variables. So cities with large populations and exposure to the services industry are more likely to undertake a range of actions associated with the UNFCCC. These findings mirror what I found for the cities signing up to the UNFCCC. So it appears that once cities have signed up, this relationship continues for determining actions.

Quality of living features such as HDI and Education were important for determining some of the targets. Namely, cities having actions and adaptations. However, both of these models had a poor fit. Exposure to secondary or industrials was important in determining commitments, mitigations and climate action plans. This result is interesting as I found before that exposure to secondary made a city less likely to sign up. So, it appears that these cities may have industrial companies who are driving the push towards green actions, making their cities more likely to continue towards actions after signing up. These three models had a good fit relative to the other models and the inUNFCCC models from i.

On the other hand, emissions inventory, initiative participation and climate action plan each only had one significant determinant, population, population and GVA from secondary, respectively. This indicates that the actions and planning relating to these features may be more complex and diverse than my features could capture. However, the emissions inventory model had the highest R2 value relative to all other models, which suggests it may just be that its variance can be well explained using the simple population model.

However, I caveat my results with the fact that these models only had 80 data points to train on, relative to the 5296 points the inUNFCCC models had. This is reflected by their overall low R^2 values, which will also be driven by the need for more variables, which I discussed in part I.

APPENDIX II

DATA AND METHODOLOGY

DATA

This section is dedicated to my selected feature variables and any methodology related to data engineering that I used during my research.

VARIABLES

In this section, I will discuss the geographic and statistical distribution of each of my feature variables.

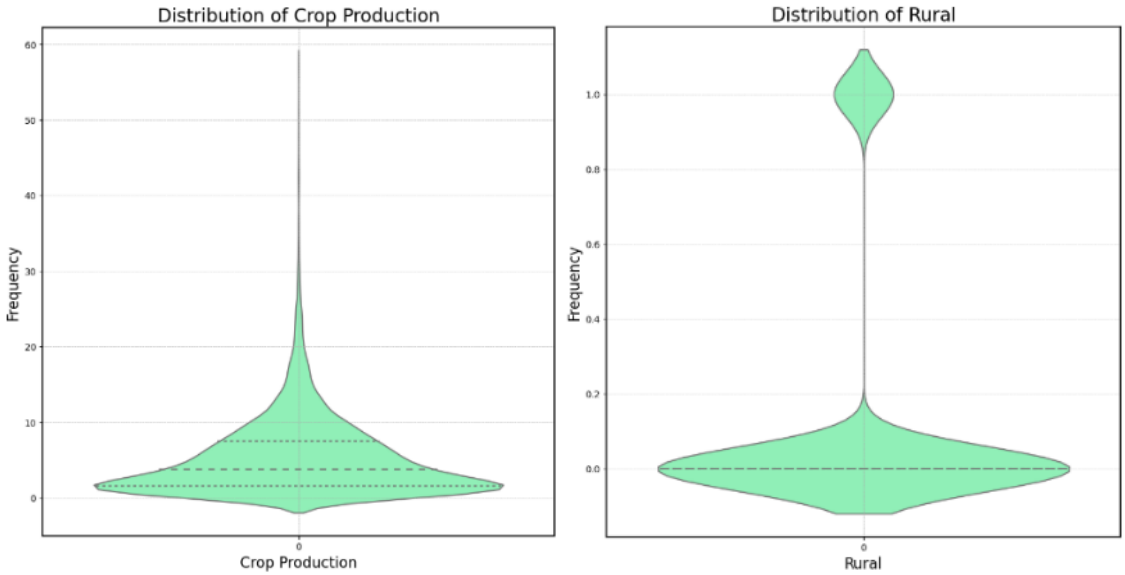
GEOGRAPHIC

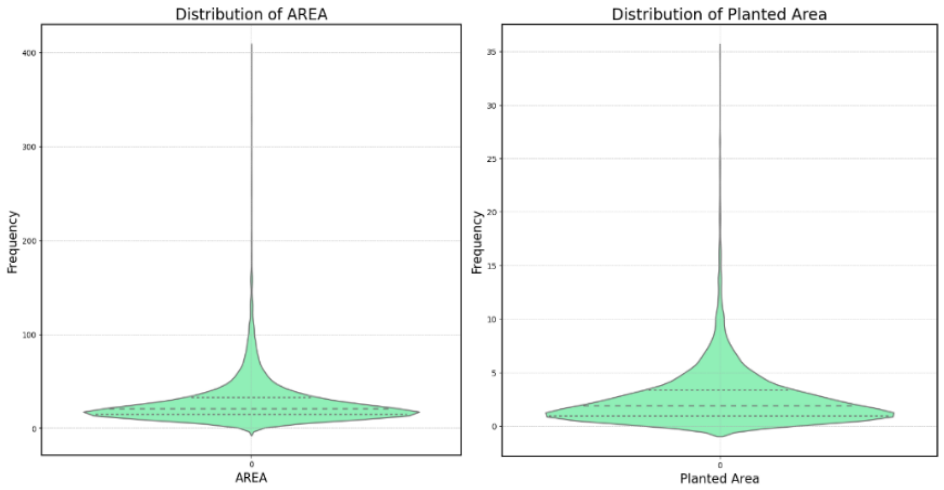
In this section, you can see information on the distribution of the geographic variables used throughout my research. I have attempted to characterise their distributions using summary statistics and violin plots. I have also provided geographic distributions of the variables across Brazil.

In terms of feature engineering, I used a square root transformation on the plant feature in order to stabilise variance, reduce right skewness and increase robustness to outliers. I was unable to use a log transformation for this as the plant feature contains zero values.

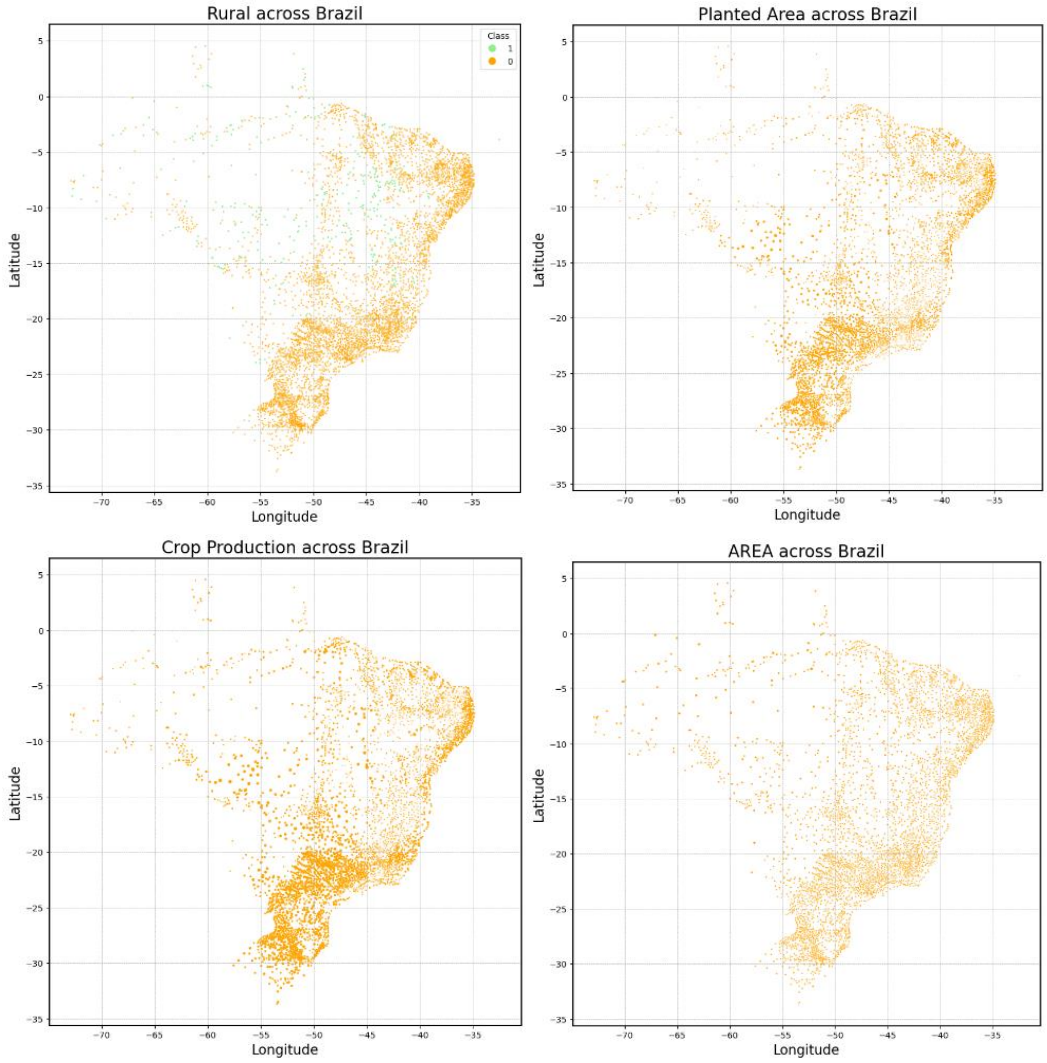
feature	count	mean	std	min	25%	50%	75%	max
plant	5296.00	2.65	2.72	0.00	0.97	1.89	3.36	34.72
crop	5296.00	5.39	5.44	0.00	1.56	3.77	7.51	57.23
area	5296.00	28.61	27.22	1.89	14.35	20.66	32.53	399.42
rural	5296.00	0.13	0.33	0.00	0.00	0.00	0.00	1.00

summary statistics for geographic features





As you can see, crop production, planted area, and area all still experience large outliers even after the transformations. However, the observations around the main body of the violin (the interquartile range) are well-centred around their mean and less tightly compacted than before the transformations. So, these variables are suitable for use in my research. As rural is a binary classifier, we see two main nodes at zero and one; I also note the size difference in the nodes, which indicates the imbalance between rural and urban cities.



From the green and orange points (rural and urban, respectively), it is clear that the majority of the urban regions hug the east coast of the country, with particularly high density in the south. While the rural cities are mainly concentrated in the centre of the country. A similar pattern is observed in the area variable, as there is a correlation between the area of a city and whether it is rural or urban. As expected, the geographic distributions of planted area and crop production are very similar, with a high density in the south of the country.

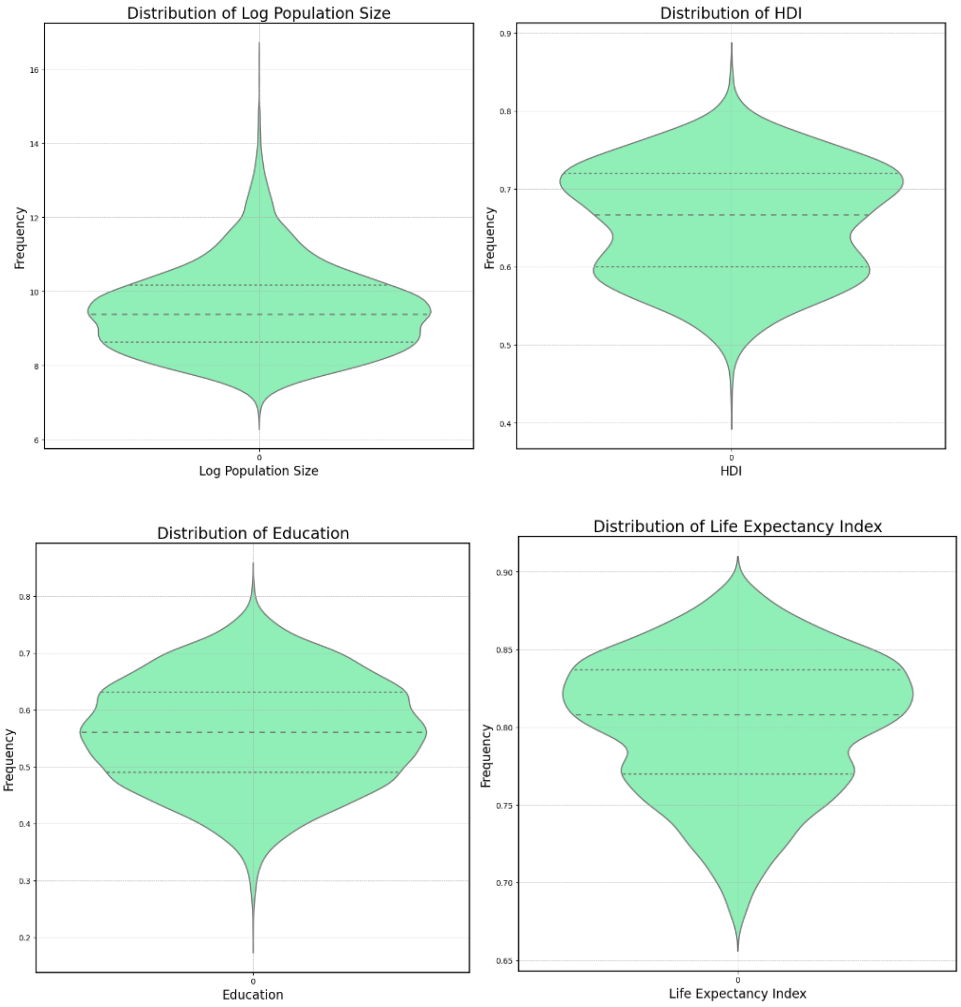
POPULATION DEMOGRAPHIC

In this section, you can see information on the distribution of the population demographic variables used throughout my research. I have attempted to characterise their distributions using summary statistics and violin plots. I have also provided geographic distributions of the variables across Brazil.

In terms of feature engineering, I used a log transformation on the population feature in order to stabilise variance, reduce right skewness and increase robustness to outliers.

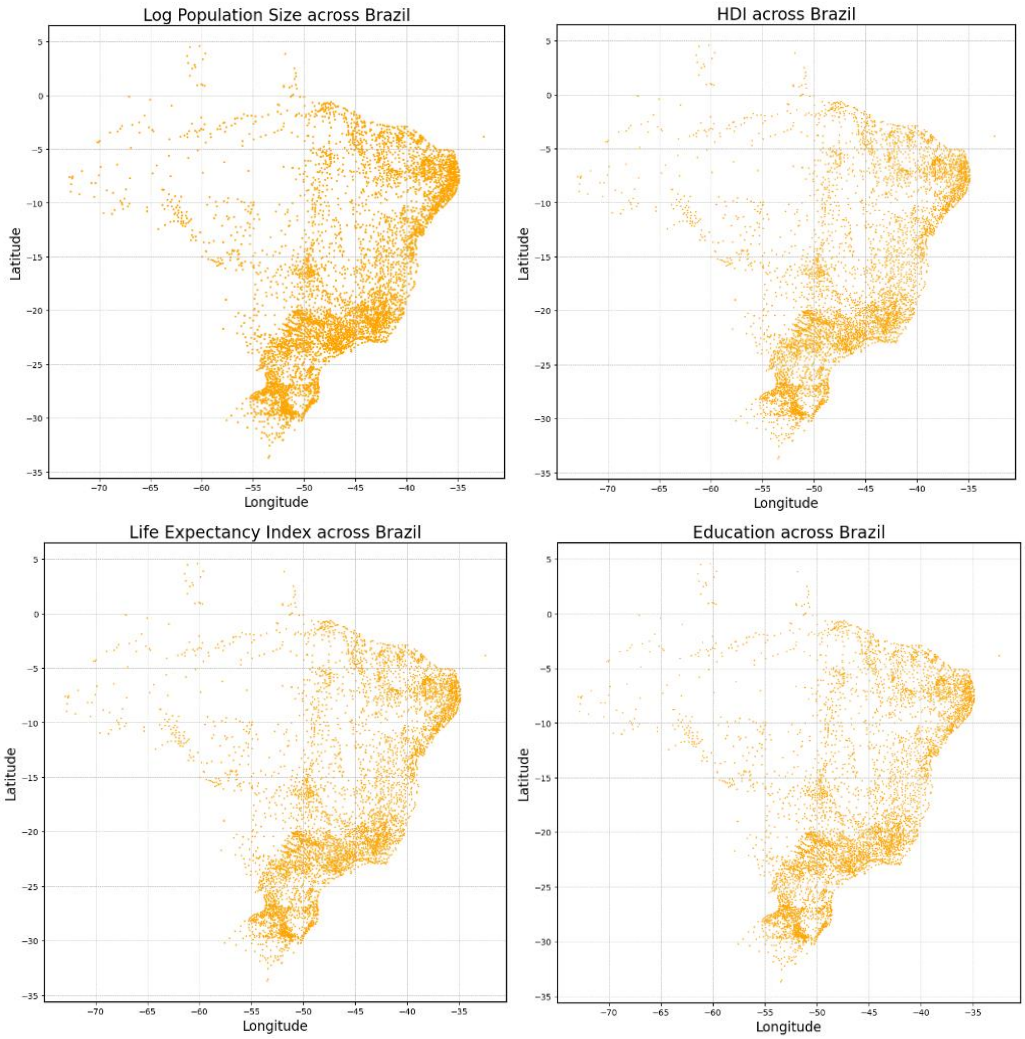
feature	count	mean	std	min	25%	50%	75%	max
pop	5296.00	9.50	1.18	6.70	8.63	9.39	10.16	16.30
hdi	5296.00	0.66	0.07	0.42	0.60	0.67	0.72	0.86
life	5296.00	0.80	0.04	0.67	0.77	0.81	0.84	0.89
edu	5296.00	0.56	0.09	0.21	0.49	0.56	0.63	0.83

summary statistics for population demographic features



Unlike the geographic variables, the population demographic variables have much less outliars and are more symmetric. While the population was highly skewed, the introduction of the log-transformation reduced the overall skewness and

variance of the distribution. The three quality of living features are on a scale from zero to one and their values are symmetric and appear to have normal levels of kurtosis as they do not exhibit particularly heavy long tails in either direction. Therefore, these features will perform well in my analysis.



All of these features appear to be distributed similarly across Brazil. So it seems that cities with high populations and quality of living are heavily focused on the east of the country and especially in the southeast. This distribution is similar to what we saw foir urban cities in the previous section.

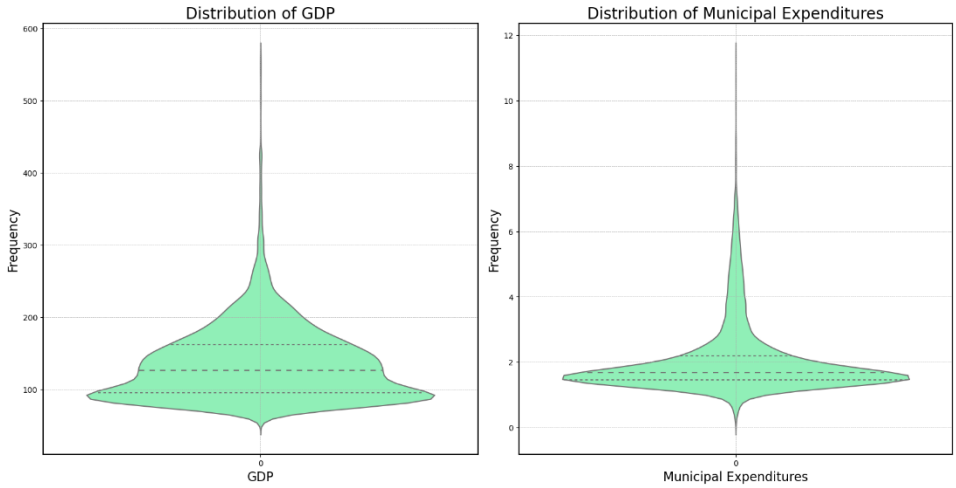
WEALTH

In this section you can see information on the distribution of the wealth variables used throughout my research. I have attempted to characterise their distributions using summary statistics and violin plots. I have also provided geographic distributions of the variables across Brazil.

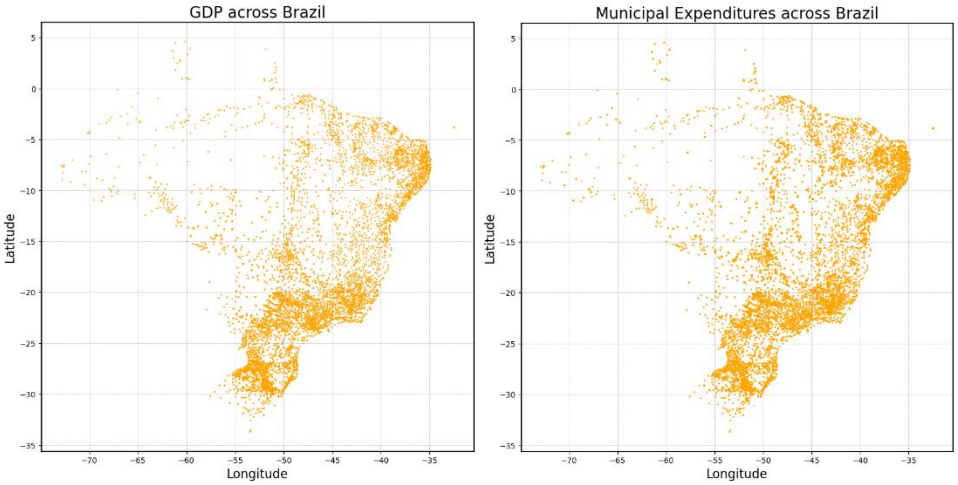
In terms of feature engineering, I used a square root transformation on both features to stabilize variance, reduce right skewness, and increase robustness to outliers.

feature	count	mean	std	min	25%	50%	75%	max
gdp	5296.00	136.03	52.32	56.49	95.64	126.39	161.90	560.93
munex	5296.00	2.08	1.12	0.19	1.45	1.68	2.20	11.36

summary statistics for wealth features



Now, looking at GDP per Capita and Municipal expenditure per capita, I note that both have large, thick tails in the positive direction, which indicates right-skewed and heavy-tailed distributions even after using a square root transformation. This characteristics the wealth imbalance present in Brazilian cities.



Once again, we see the majority of cities with high GDP per Capita and Municipal expenditures focused on the left and south of the country. However, municipal expenditures remain high further inland, which may indicate government spending in more rural areas. These distributions are overall consistent with what I would have expected from previous sections.

INDUSTRY

In this section you can see information on the distribution of the industry variables used throughout my research. I have attempted to characterise their distributions using summary statistics and violin plots. I have also provided geographic distributions of the variables across Brazil.

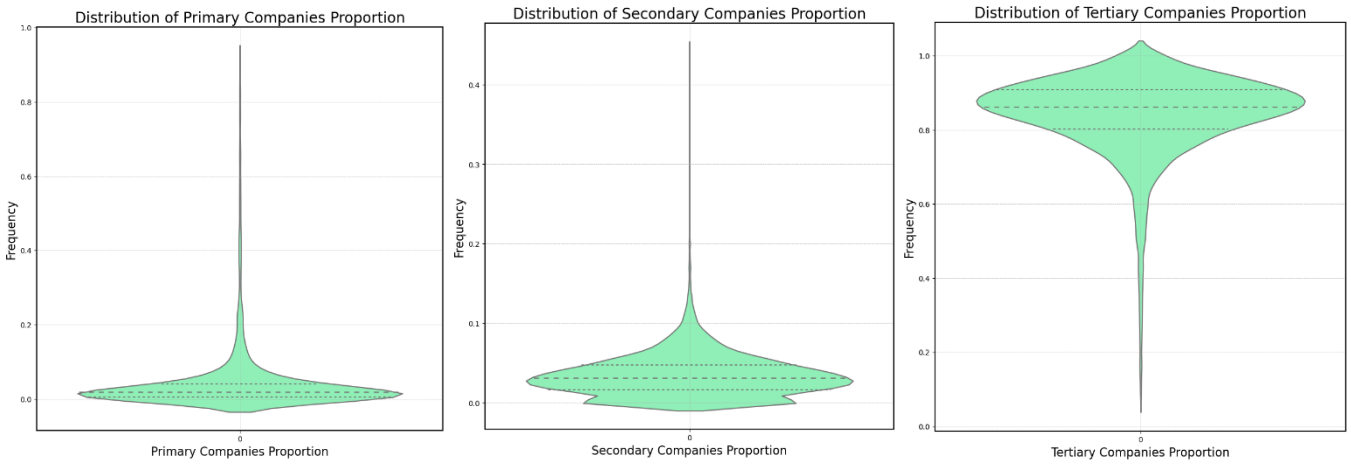
In terms of feature engineering, I formed the proportion variables by grouping companies of different types into primary, secondary, and tertiary-related companies. I then divided these by the total number of companies to remove the effect of

cities that just have a large number of companies. I also left out some company types from the groupings to prevent multicolinearity among these new features.

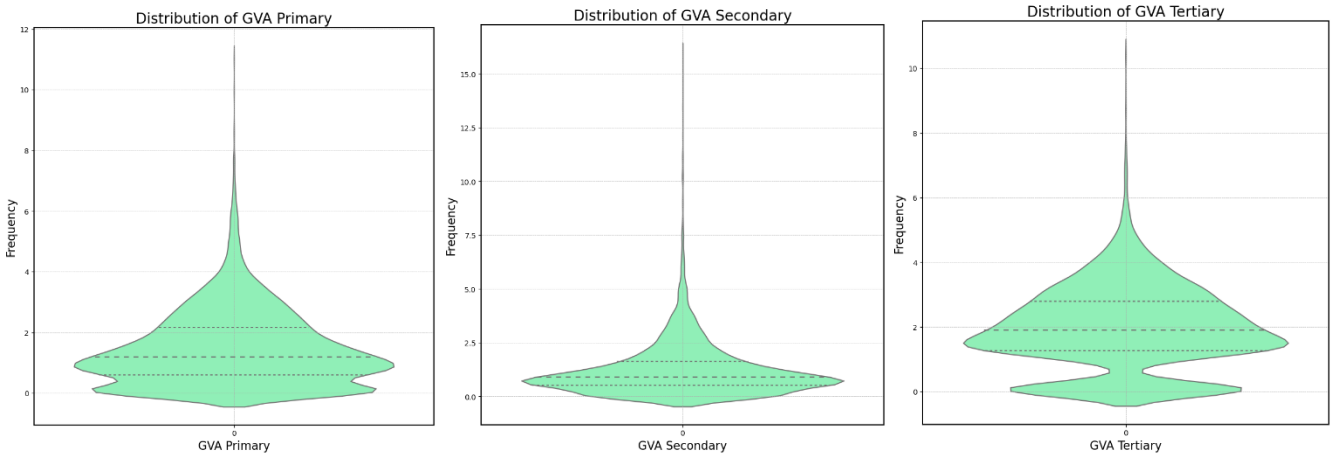
I also formed the Gross Value Added per Capita features by dividing the GVA from each sector by the city's population size to remove the effect of large populations generating more value added. I used a square root transformation on these features to stabilize variance, reduce right skewness, and increase robustness to outliers.

feature	count	mean	std	min	25%	50%	75%	max
pprop	5296.00	0.05	0.10	0.00	0.01	0.02	0.04	0.92
sprop	5296.00	0.03	0.03	0.00	0.02	0.03	0.05	0.44
tprop	5296.00	0.84	0.11	0.08	0.80	0.86	0.91	1.00
gvap	5296.00	1.47	1.26	0.00	0.59	1.19	2.16	11.00
gvas	5296.00	1.26	1.32	0.01	0.51	0.89	1.62	15.97
gvat	5296.00	2.00	1.30	0.02	1.27	1.89	2.80	10.43

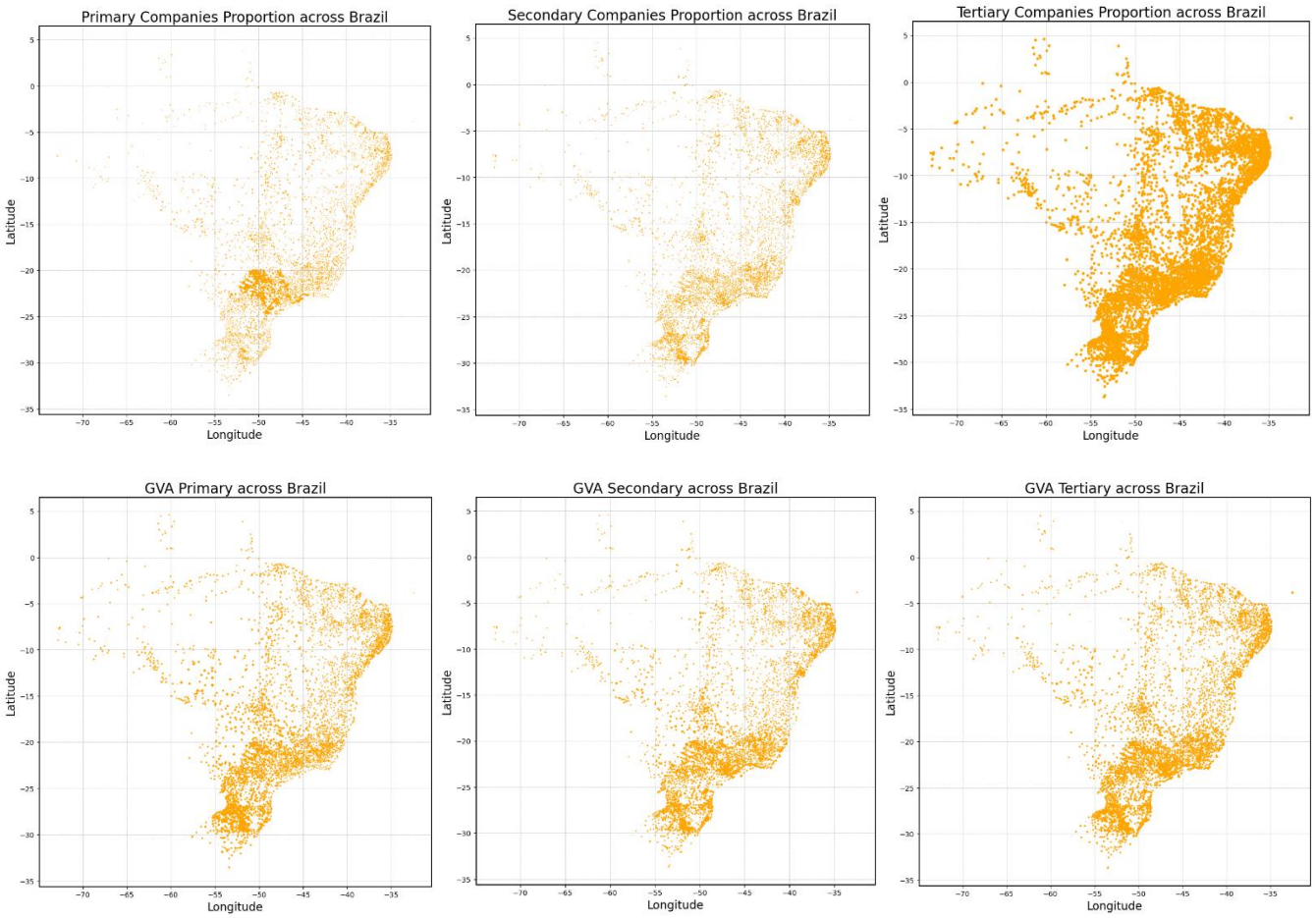
summary statistics for industry features



Here I note the difference in distribution between exposure to primary/secondary and tertiary industries. Companies within Brazillian cities appear to, on average, be focused on tertiary industry followed by secondary, with primary having the lowest mean. However, I observe large upper tails on the primary and secondary distributions and a large lower tail on the tertiary proportion, which is evidence of the extremes in each category. Every after square root transformations these variables are still highly skewed.



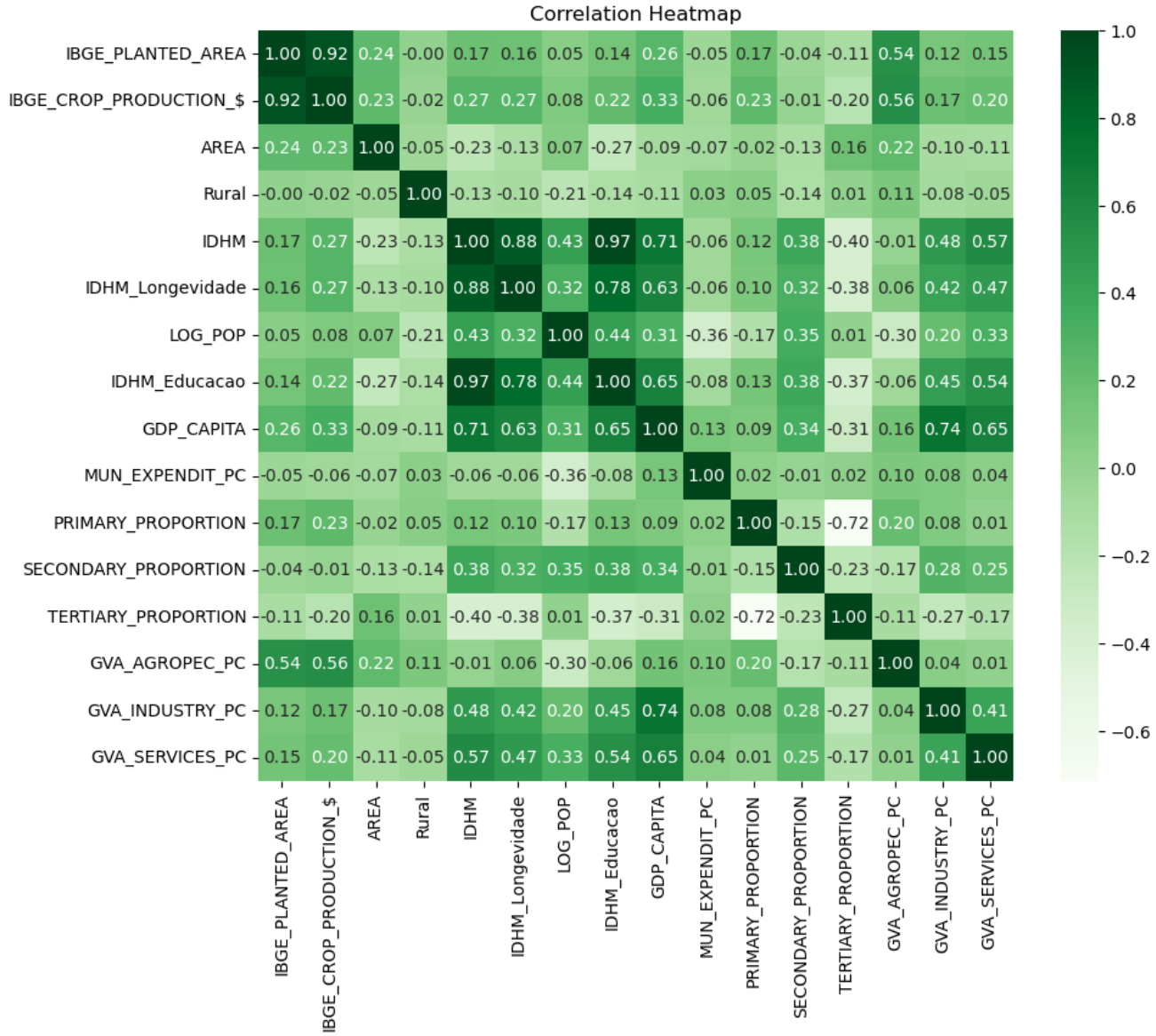
The distribution of gross value added per industry category tells an interesting story when compared to the previous three distributions. We had previously observed the large focus on tertiary companies within Brazillian cities. However, I now observe that the distribution of value from this sector is quite similar to that of the primary sector. This tells the story that while there are more tertiary companies they make less money on average than primary companies. This also provides evidence for the importance of industries relating to agriculture, for example, in Brazil's economy.



The geographic distributions of this feature are similar to what I observed in previous sections. As expected the distribution of primary companies is similar to that of crop production and planted area. The distributions relating to tertiary industries appear similar to those of the wealth and size features, which were highly focused on the east and southeast of the country. This is consistent with large, wealthy, urban cities with a focus on tertiary companies such as municipal jobs and financial services being mainly located in the south-east of the country

CORRELATION ANALYSIS

Below I have presented a correlation heatmap between my feature variables



As you can see, there is a high correlation between planted area and crop production and also between the quality of living measures such as HDI, Education Index and Life Expectancy. For my research, I decided not to remove features with high correlation as many of the non-linear techniques I use throughout do not suffer as a result of high correlation. Removing variables like these could also lead to a loss of information on potential interactions between these features. More importantly, keeping highly correlated variables can improve the interpretability of the model, which is the most important aspect of my research; for example, having more similar features to compare is very useful for the clustering algorithm.

DATA METHODOLOGY

COMBINING THE DATASETS

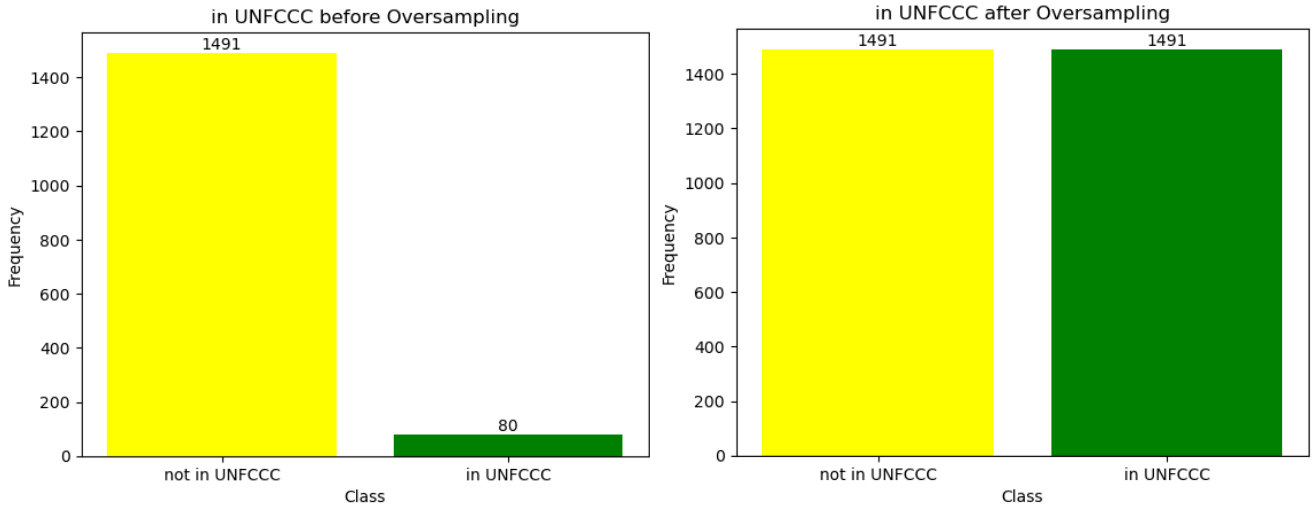
To combine my UNFCCC, Brazil Cities and Features datasets on the city names, I used Levenshtein distance to quantify the similarity between the city names. This was necessary to overcome the differences in syntax and capitalization across datasets.

Levenshtein distance is a measure of how similar two strings are by taking into account the number of insertion, deletion and substitution operations needed to transform one of the strings into the other (GeeksforGeeks,2024). In practice, I used the Python package 'difflib' in order to calculate the similarity between a city name in dataset A and all other city names in dataset B. For each city in dataset A, I selected the match which had the highest similarity score as long as it was above a selected tolerance level.

BALANCING DATASETS

When dealing with imbalanced data in a binary classification setting, it is common practice to use techniques to balance the data, resulting in an equal or very similar number of observations in each class. This allows the models to derive as much insight or learn as much as possible as the outcome of interest.

In my case, I decided to use Random Oversampling, which constructs a new dataset by sampling from both the 1 and 0 classes but taking more observations from the class with less. This results in a dataset with circa 50% 1s and 50% 0s. Below, you can see the result of my balancing.



DEALING WITH MISSING VALUES

The only one of my features which contained Na values was the Municipal Expenditure variable. While there are many approaches to solving this issue, such as forward/backward fill or dropping the rows, I felt that filling with the mean was the best solution in this particular case. This is because the number of Na values in this variable was very low, so filling with the mean would not have a significant impact on the distribution of the variable. Also, logically, it would not make sense to fill these missing values with 0 in this context and dropping the city if it had Na municipal expenditure would lead to information loss by not including the other variables.

STATISTICAL MODELS

In this section, I will briefly discuss my motivation for the models and algorithms used in my analysis. These include linear and non-linear supervised learning models and an unsupervised clustering algorithm.

LINEAR PROBABILITY MODEL

The LPM (Linear Probability Model) is an application of the Linear Regression Model in a classification setting. In this case, the endogenous variable is a binary or indicator variable. The model takes the following form for each observation i:

$$Y_i = B_0 + B_1 X_{i,i} + \cdots + B_p X_{p,i} + u_i$$
$$E(ui) = 0$$

Since Y is a binary variable, the conditional probability of Y_i given each of $X_{j,I}$ can be interpreted as a probability. While this model has some shortcomings, such as the predictions of Y_i not being bound between zero and one, it does have the advantage of interpretability. Each B_j corresponding to feature $X_{,ij}$ can be interpreted as " for a one unit increase in $X_{,ij}$, the probability of Y_i being equal to 1_i will increase by $B_{j,i}$*100". I will use this interpretability as well as the statistical significance of the betas to examine the magnitude, direction and significance of the feature variables with the regressand.

LOGISTIC REGRESSION

The logistic regression model introduces an exponential transformation of the LPM which solves some of the predictability issues that LPM faces. All predicted values of the Logistic Model are bound between zero and one. The model is defined as follows:

$$P(X_1, \ldots, X_p) = \frac{e^{B_o + B_1 X_1 + \cdots + X_p}}{1 + e^{B_o + B_1 X_1 + \cdots + X_p}}$$

While this model improves on LPM in terms of predictability, the introduction of the logistic function reduces the interpretability. Now an increase in a feature results in an increase in the "log-odds" of Y. However, the magnitude, direction and significance is still comparable across features.

RANDOM FOREST ALGORITHM

Both previous models attempt to quantify linear relationships between the target and feature variables. To examine non-linear relations, I have used a random forest classification model. A decision tree aims to partition the feature space with the aim of minimizing nodal impurity in order to predict the outcome of interest. A random forest improves on this by fitting multiple uncorrelated trees on randomly sampled subsets of the original dataset. This process reduces the variance of predictions. While this model is quite sophisticated in terms of classification models. My main use case is the 'feature importances' it generates.

After fitting a random forest, the model can produce an 'importance' for each variable used in prediction. This statistic can be used to rank features on how good they were at improving the model performance. Therefore, in this way, I will use this model to assess how useful each of my features are in determining my target variables in a non-linear sense.

K-MEANS CLUSTERING ALGORITHM

K-Means Clustering is an unsupervised learning technique which attempts to divide each of the feature spaces into a specified number of 'clusters'. It does this separation by attempting to minimize the sum of squared distances between each observation in a cluster and the centroid of the cluster. In doing this the algorithm can identify groups within the data with unique characteristics.

When selecting the optimal number of clusters, I take into account the inertia and silhouette scores. The inertia is a measure of the distance between each observation in a cluster and its centroid. Thus, it is a measure of the quality of the clustering where lower values are better. The aim of selecting K from this method is to find an 'elbow' point on the curve where an increase in clusters only leads to a linear reduction in inertia (a point of diminishing marginal decrease in inertia). The Silhouette score is a measure of how similar observations are to their own cluster relative to others; in this case, we would like to mind a maximum.

BIBLIOGRAPHY

Di Giulio, B. and Munson, J. (2019) *Bridging the gap* [Preprint]. doi:10.1142/11376.

GeeksforGeeks (2024) *Introduction to levenshtein distance*, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/introduction-to-levenshtein-distance/ (Accessed: 28 April 2024).

NAEI (2023) *United Nations Framework Convention on Climate Change (UNFCCC) - defra, UK*, *United Nations Framework Convention on Climate Change (UNFCCC) - NAEI, UK*. Available at: https://naei.beis.gov.uk/about/why-we-estimate?view=unfccc#:~:text=One%20of%20the%20achievements%20of,are%20party%20to%20the%20Convention (Accessed: 28 April 2024).

Parada, C. (2022) *Brazilian cities*, *Kaggle*. Available at: https://www.kaggle.com/datasets/crisparada/brazilian-cities (Accessed: 28 April 2024).

Shukla, N. (2023) *Explainer: What is the UNFCCC?*, *Earth.Org*. Available at: https://earth.org/explainer-what-is-the-unfccc/#:~:text=Criticisms%20to%20the%20UNFCCC&text=However%2C%20by%20exempting%20developing%20countries,in%20the%20decades%20that%20followed (Accessed: 28 April 2024).

UNFCCC (no date) *What is the United Nations Framework Convention on Climate Change?*, *Unfccc.int*. Available at: https://unfccc.int/process-and-meetings/what-is-the-united-nations-framework-convention-on-climate-change (Accessed: 28 April 2024).

APPENDIX III - CODE

code-final

April 28, 2024

1 Set-up

Import packages and set palette. I have imported the generateData class from the dataGenerator module. The code for this module is also attached.

[1]:
```
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import requests

from bs4 import BeautifulSoup
from unidecode import unidecode
from dataGenerator import generateData
from fuzzywuzzy import fuzz
from fuzzywuzzy import process
from imblearn.over_sampling import RandomOverSampler
from sklearn.ensemble import RandomForestRegressor
from IPython.display import display, HTML
from sklearn.cluster import KMeans
from sklearn.metrics import confusion_matrix,silhouette_score
from matplotlib.lines import Line2D

pd.options.mode.chained_assignment = None

# Define the custom color palette
custom_palette = ['#228B22', '#DAA520', '#0077BE', '#FFA500', '#8B4513',␣
 ↪'#FF0000']

# Set the default color palette to the custom one
sns.set_palette(custom_palette)
```

Initializing the generateData class

[2]:
```
gd = generateData('Brazil')
```

```
213
```

Loading the UNFCC data from the generateData class

[3]:
```
UNFCC = gd.inUNFCCData()
```

```
Porto Alegre -> Porto Alegre with similarity 1.0
Belo Horizonte -> Belo Horizonte with similarity 1.0
Caruaru -> Caruaru with similarity 1.0
Rio Branco -> Rio Branco with similarity 1.0
Maua -> Maua with similarity 1.0
Diadema -> Diadema with similarity 1.0
Sao Jose dos Campos -> Sao Jose Dos Campos with similarity 1.0
Sao jose dos Campos -> Sao Jose Dos Campos with similarity 1.0
Palmas -> Palmas with similarity 1.0
Goias -> Goias with similarity 1.0
Florianopolis -> Florianopolis with similarity 1.0
Cuiaba -> Cuiaba with similarity 1.0
Brasileia -> Brasileia with similarity 1.0
Brasilia -> Brasilia with similarity 1.0
Camocim -> Camocim with similarity 1.0
Franco da Rocha -> Franco Da Rocha with similarity 1.0
Duque de Caxias -> Duque De Caxias with similarity 1.0
Caieiras -> Caieiras with similarity 1.0
Boa Vista -> Boa Vista with similarity 1.0
Petrolina -> Petrolina with similarity 1.0
petrolina -> Petrolina with similarity 1.0
Tupaciguara -> Tupaciguara with similarity 1.0
Niteroi -> Niteroi with similarity 1.0
niteroi -> Niteroi with similarity 1.0
Jundiai -> Jundiai with similarity 1.0
Capivari -> Capivari with similarity 1.0
Natal -> Natal with similarity 1.0
Igarassu -> Igarassu with similarity 1.0
Jaguariuna -> Jaguariuna with similarity 1.0
Canoas -> Canoas with similarity 1.0
Campo Grande -> Campo Grande with similarity 1.0
Campina Grande -> Campina Grande with similarity 1.0
Cruzeiro do Sul -> Cruzeiro Do Sul with similarity 1.0
Aparecida de Goiania -> Aparecida De Goiania with similarity 1.0
nAtAl -> Natal with similarity 1.0
Cumaru -> Cumaru with similarity 1.0
Sao Caetano do Sul -> Sao Caetano Do Sul with similarity 1.0
Guarulhos -> Guarulhos with similarity 1.0
Manaus -> Manaus with similarity 1.0
Jaboatao dos Guararapes -> Jaboatao Dos Guararapes with similarity 1.0
Sao Paulo -> Sao Paulo with similarity 1.0
Sao Jose do Rio Preto -> Sao Jose Do Rio Preto with similarity 1.0
Joao Pessoa -> Joao Pessoa with similarity 1.0
Aracatuba -> Aracatuba with similarity 1.0
```

```
Uberaba -> Uberaba with similarity 1.0
uberaba -> Uberaba with similarity 1.0
Londrina -> Londrina with similarity 1.0
londrina -> Londrina with similarity 1.0
Cordeiropolis -> Cordeiropolis with similarity 1.0
Aracaju -> Aracaju with similarity 1.0
Teresina -> Teresina with similarity 1.0
teresina -> Teresina with similarity 1.0
Sao Luis de Montes Belos -> Sao Luis De Montes Belos with similarity 1.0
Coracao de Jesus -> Coracao De Jesus with similarity 1.0
Bertioga -> Bertioga with similarity 1.0
Penapolis -> Penapolis with similarity 1.0
Piracicaba -> Piracicaba with similarity 1.0
Osasco -> Osasco with similarity 1.0
Guaruja -> Guaruja with similarity 1.0
guaruja -> Guaruja with similarity 1.0
Guanhaes -> Guanhaes with similarity 1.0
Brumadinho -> Brumadinho with similarity 1.0
Santo Andre -> Santo Andre with similarity 1.0
Limeira -> Limeira with similarity 1.0
Indiaroba -> Indiaroba with similarity 1.0
Extrema -> Extrema with similarity 1.0
Itacoatiara -> Itacoatiara with similarity 1.0
Sorocaba -> Sorocaba with similarity 1.0
Fortaleza -> Fortaleza with similarity 1.0
Joinville -> Joinville with similarity 1.0
Brusque -> Brusque with similarity 1.0
Salvador -> Salvador with similarity 1.0
Porto Velho -> Porto Velho with similarity 1.0
Tremembe -> Tremembe with similarity 1.0
Pau Brasil -> Pau Brasil with similarity 1.0
Sao Leopoldo -> Sao Leopoldo with similarity 1.0
Lorena -> Lorena with similarity 1.0
lorena -> Lorena with similarity 1.0
Campinas -> Campinas with similarity 1.0
Abaetetuba -> Abaetetuba with similarity 1.0
Venancio Aires -> Venancio Aires with similarity 1.0
Itatiba -> Itatiba with similarity 1.0
Botucatu -> Botucatu with similarity 1.0
Sumare -> Sumare with similarity 1.0
sumare -> Sumare with similarity 1.0
Ribeirao Pires -> Ribeirao Pires with similarity 1.0
Itu -> Itu with similarity 1.0
itu -> Itu with similarity 1.0
Lauro de Freitas -> Lauro De Freitas with similarity 1.0
Recife -> Recife with similarity 1.0
Contagem -> Contagem with similarity 1.0
Betim -> Betim with similarity 1.0
```

```
Maceio -> Maceio with similarity 1.0
Aparecida -> Aparecida with similarity 1.0
Sao Cristovao -> Sao Cristovao with similarity 1.0
Birigui -> Birigui with similarity 1.0
birigui -> Birigui with similarity 1.0
Serra Talhada -> Serra Talhada with similarity 1.0
Juruti -> Juruti with similarity 1.0
juruti -> Juruti with similarity 1.0
Promissao -> Promissao with similarity 1.0
Presidente Prudente -> Presidente Prudente with similarity 1.0
presidente prudente -> Presidente Prudente with similarity 1.0
Goiania -> Goiania with similarity 1.0
goiania -> Goiania with similarity 1.0
Sao Bernardo do Campo -> Sao Bernardo Do Campo with similarity 1.0
Maringa -> Maringa with similarity 1.0
maringa -> Maringa with similarity 1.0
Sao Luis -> Sao Luis with similarity 1.0
Sertaozinho -> Sertaozinho with similarity 1.0
Rio de Janeiro -> Rio De Janeiro with similarity 1.0
Rio De Janeiro -> Rio De Janeiro with similarity 1.0
Curitiba -> Curitiba with similarity 1.0
Angra dos Reis -> Angra dos Reis with similarity 1.0
```

[4]:
```
df_y = UNFCC.copy()
```

[5]:
```
df_y.dropna(axis = 1,inplace=True, how = 'all') # removing any columns that are␣
↪entirely NaN
```

Setting the table style

[6]:
```
table_style = """
    <style>
        table {
            font-family: Arial, sans-serif;
            border-collapse: collapse;
            width: 50%;
            margin: auto;
        }
        th {
            background-color: #f2f2f2;
            color: #333333;
            font-weight: bold;
            font-size: 14px;
            border: 1px solid #dddddd;
            text-align: left;
            padding: 8px;
        }
        td {
```

```
            background-color: #ffffff;
            color: #333333;
            border: 1px solid #dddddd;
            text-align: left;
            padding: 8px;
        }
        tr:nth-child(even) {
            background-color: #f2f2f2;
        }
        .title {
            text-align: center;
            font-size: 18px;
            font-weight: bold;
            color: #333333;
            padding: 10px;
            margin-bottom: 20px;
        }
    </style>
"""
```

General function to create all geographic and statistially summaries for each variable

[7]:
```
def feature_summary(feature, feature_set, df, table_style=table_style,␣
 ↪size=5000):
    # Get data for the specified feature
    data = df[feature_set[feature]]

    # Summary statistics
    summary_table = pd.DataFrame(data.describe()).transpose()

    # Display summary table as HTML with styling
    html_content = f"<div class='title'>Summary Statistics for {feature}</div>"
    html_content += table_style
    html_content += summary_table.to_html(index=False, classes='table')
    display(HTML(html_content))

    # Create a figure for the violin plot
    plt.figure(figsize=(10, 10))

    # Visualization of the distribution using a violin plot
    sns.violinplot(data=data, inner='quartile', palette='rainbow')
    plt.title(f'Distribution of {feature}', fontsize=20)
    plt.xlabel(feature, fontsize=16)
    plt.ylabel('Frequency', fontsize=16)
    plt.gca().set_facecolor('white')  # Remove grey background
    plt.grid(True, linestyle='--', linewidth=0.5)
    plt.gca().spines['top'].set_linewidth(1.5)
```

```
    plt.gca().spines['bottom'].set_linewidth(1.5)
    plt.gca().spines['left'].set_linewidth(1.5)
    plt.gca().spines['right'].set_linewidth(1.5)

    plt.show()

    # Create a figure for the scatter plot
    plt.figure(figsize=(10, 10))

    # Scatter plot for Latitude and Longitude
    mask1 = df["LONG"] != 0
    mask2 = df["LAT"] != 0
    x = df[mask1 & mask2]["LONG"]
    y = df[mask1 & mask2]["LAT"]
    z = df[mask1 & mask2][feature_set[feature]]

    plt.scatter(x, y, s=z/size, alpha=1, c='orange')
    plt.title(f"{feature} across Brazil", fontsize=20)
    plt.xlabel("Longitude", fontsize=16)
    plt.ylabel("Latitude", fontsize=16)
    plt.gca().set_facecolor('white')  # Remove grey background
    plt.grid(True, linestyle='--', linewidth=0.5)
    plt.gca().spines['top'].set_linewidth(1.5)
    plt.gca().spines['bottom'].set_linewidth(1.5)
    plt.gca().spines['left'].set_linewidth(1.5)
    plt.gca().spines['right'].set_linewidth(1.5)

    plt.show()
```

Alternative to the previous function which I used only for binary variables

[122]:
```
def binary_feature_summary(feature, feature_set, df, table_style=table_style,␣
↪size=5000):
    # Get data for the specified feature
    data = df[feature_set[feature]]

    # Summary statistics
    summary_table = pd.DataFrame(data.describe()).transpose()

    # Display summary table as HTML with styling
    html_content = f"<div class='title'>Summary Statistics for {feature}</div>"
    html_content += table_style
    html_content += summary_table.to_html(index=False, classes='table')
    display(HTML(html_content))

    # Create a figure for the violin plot
    plt.figure(figsize=(10, 10))
```

```
    # Visualization of the distribution using a violin plot
    sns.violinplot(data=data, inner='quartile', palette='rainbow')
    plt.title(f'Distribution of {feature}', fontsize=20)
    plt.xlabel(feature, fontsize=16)
    plt.ylabel('Frequency', fontsize=16)
    plt.gca().set_facecolor('white')  # Remove grey background
    plt.grid(True, linestyle='--', linewidth=0.5)
    plt.gca().spines['top'].set_linewidth(1.5)
    plt.gca().spines['bottom'].set_linewidth(1.5)
    plt.gca().spines['left'].set_linewidth(1.5)
    plt.gca().spines['right'].set_linewidth(1.5)

    plt.show()

    # Create a figure for the scatter plot
    plt.figure(figsize=(10, 10))

    # Scatter plot for Latitude and Longitude
    mask1 = df["LONG"] != 0
    mask2 = df["LAT"] != 0
    x = df[mask1 & mask2]["LONG"]
    y = df[mask1 & mask2]["LAT"]
    z = df[mask1 & mask2][feature_set[feature]]

    # Scatter plot for class 0 (light green)
    plt.scatter(x[z == 0], y[z == 0], s=size, alpha=0.6, c='orange', label='0')

    # Scatter plot for class 1 (orange)
    plt.scatter(x[z == 1], y[z == 1], s=size, alpha=1, c='lightgreen',␣
↪label='1')

    plt.title(f"{feature} across Brazil", fontsize=20)
    plt.xlabel("Longitude", fontsize=16)
    plt.ylabel("Latitude", fontsize=16)
    plt.gca().set_facecolor('white')  # Remove grey background
    plt.grid(True, linestyle='--', linewidth=0.5)
    plt.gca().spines['top'].set_linewidth(1.5)
    plt.gca().spines['bottom'].set_linewidth(1.5)
    plt.gca().spines['left'].set_linewidth(1.5)
    plt.gca().spines['right'].set_linewidth(1.5)

    # Create legend
    legend_elements = [Line2D([0], [0], marker='o', color='w', label='1',␣
↪markerfacecolor='lightgreen', markersize=10),
                       Line2D([0], [0], marker='o', color='w', label='0',␣
↪markerfacecolor='orange', markersize=10)]
```

```
    plt.legend(handles=legend_elements, title="Class", loc='upper right')

    plt.show()
```

2 Exploratory Data Analysis

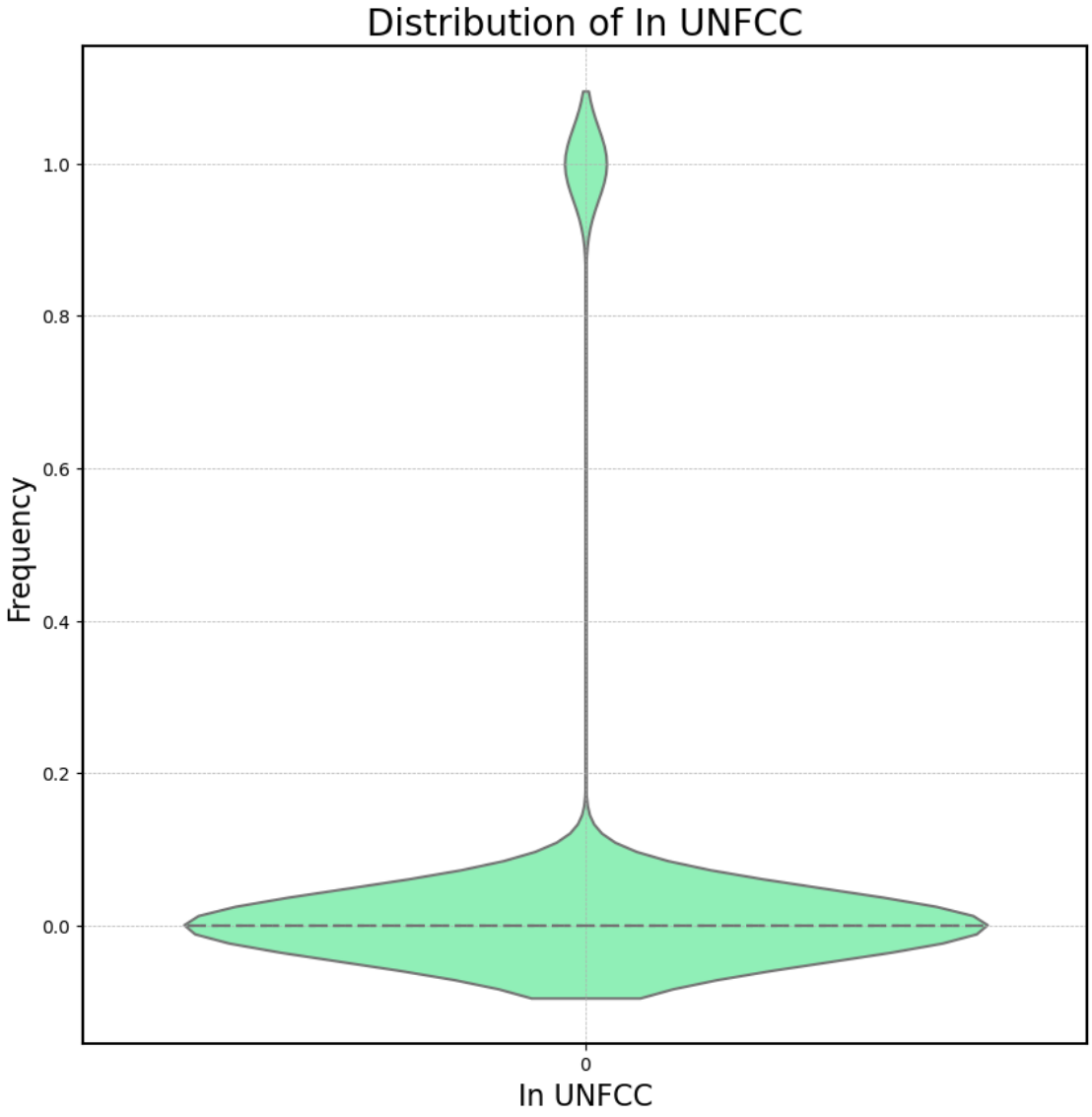
[8]:
```
targets_variables = {'In UNFCC':'inUNFCC'}
```

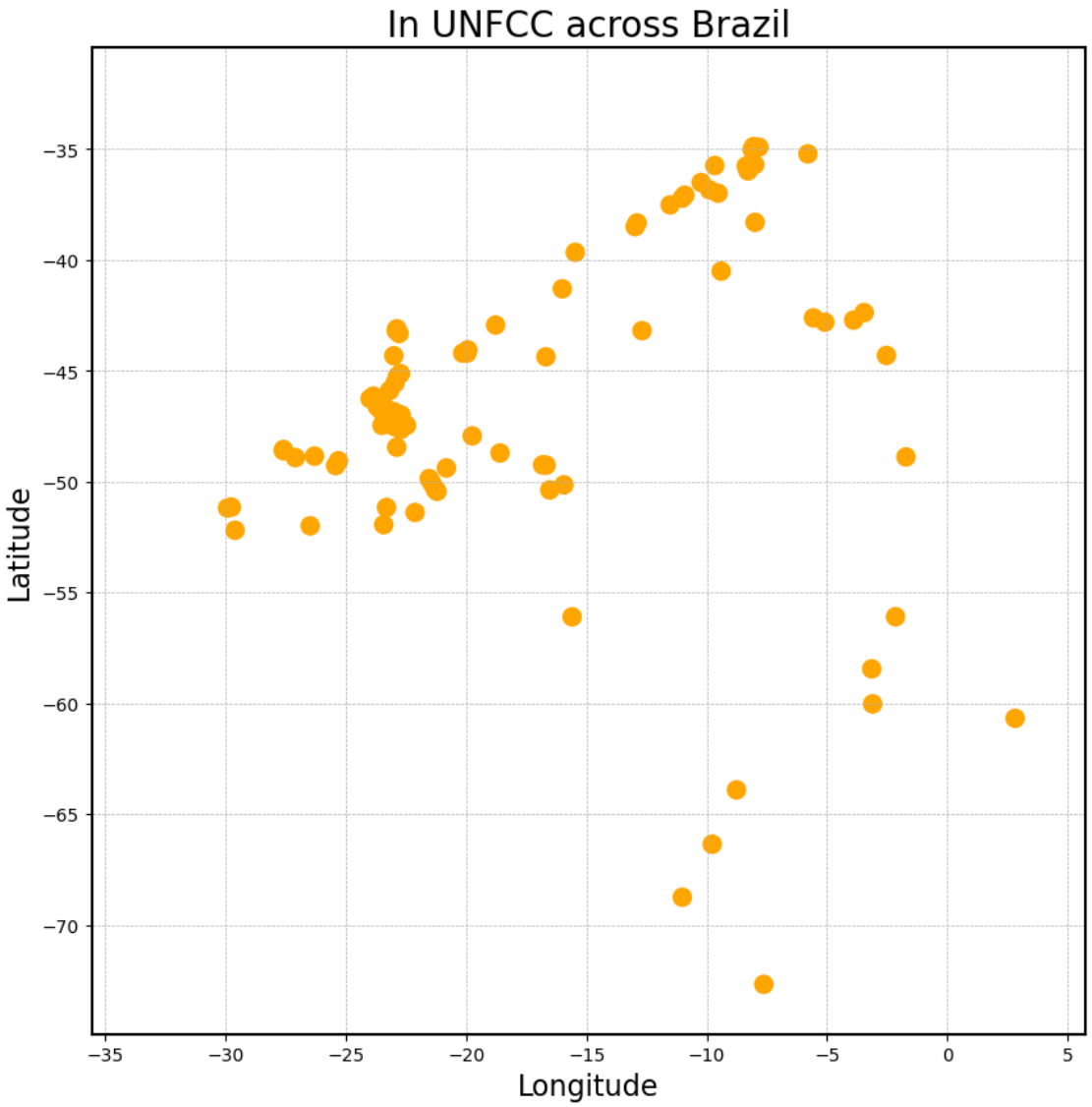
[9]:
```
df_y['LONG'] = [float(i[0]) for i in df_y['Coordinates'].str.split(', ')]
df_y['LAT'] = [float(i[1]) for i in df_y['Coordinates'].str.split(', ')]
```

2.0.1 In UNFCC

[10]:
```
feature_summary('In UNFCC',targets_variables,df = df_y,size = 0.01)
```

```
<IPython.core.display.HTML object>
```


Distribution of In UNFCC
Frequency
1.0
0.8
0.6
0.4
0.2
0.0
0
In UNFCC



2.1 My feature variables

See the report for more info on the modifications and feature engineering carried out

Loading the Brazil data which I gathered

[11]:
```
df = gd.datasets['brazil_cities'].copy()
```

2.1.1 Geographic

[12]:
```
df['IBGE_CROP_PRODUCTION_$'] = df['IBGE_CROP_PRODUCTION_$']/1000 # making per␣
↪1000 dollars
df['IBGE_PLANTED_AREA'] = df['IBGE_PLANTED_AREA']/1000 #making per 1000 hectares
```

[13]:
```
df['AREA'] = pd.to_numeric(df['AREA'].str.replace(',',''))
```

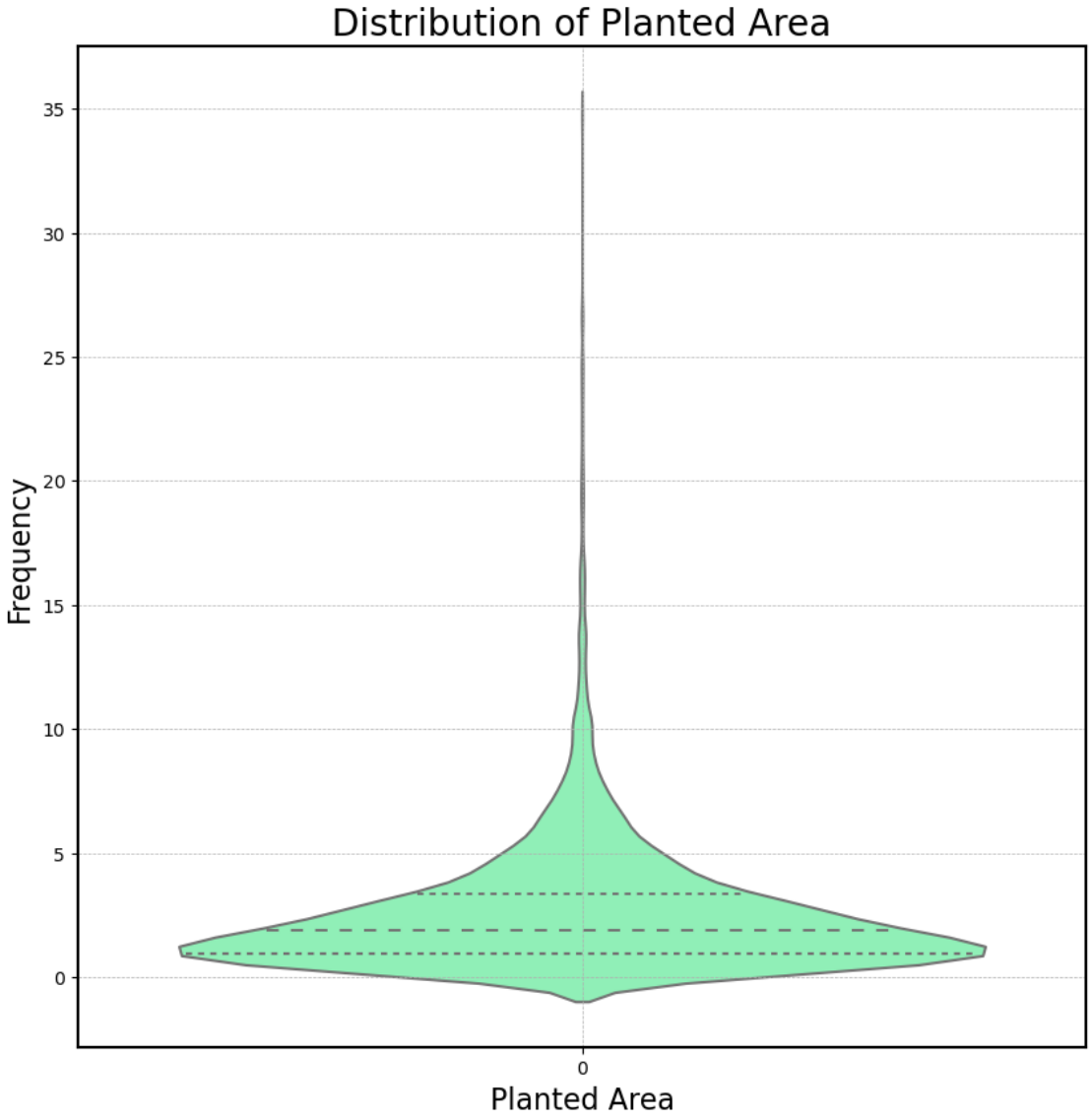
[14]:
```
geographic = {'Planted Area':'IBGE_PLANTED_AREA',
              'Crop Production':'IBGE_CROP_PRODUCTION_$',
               'AREA':'AREA'
             }
```

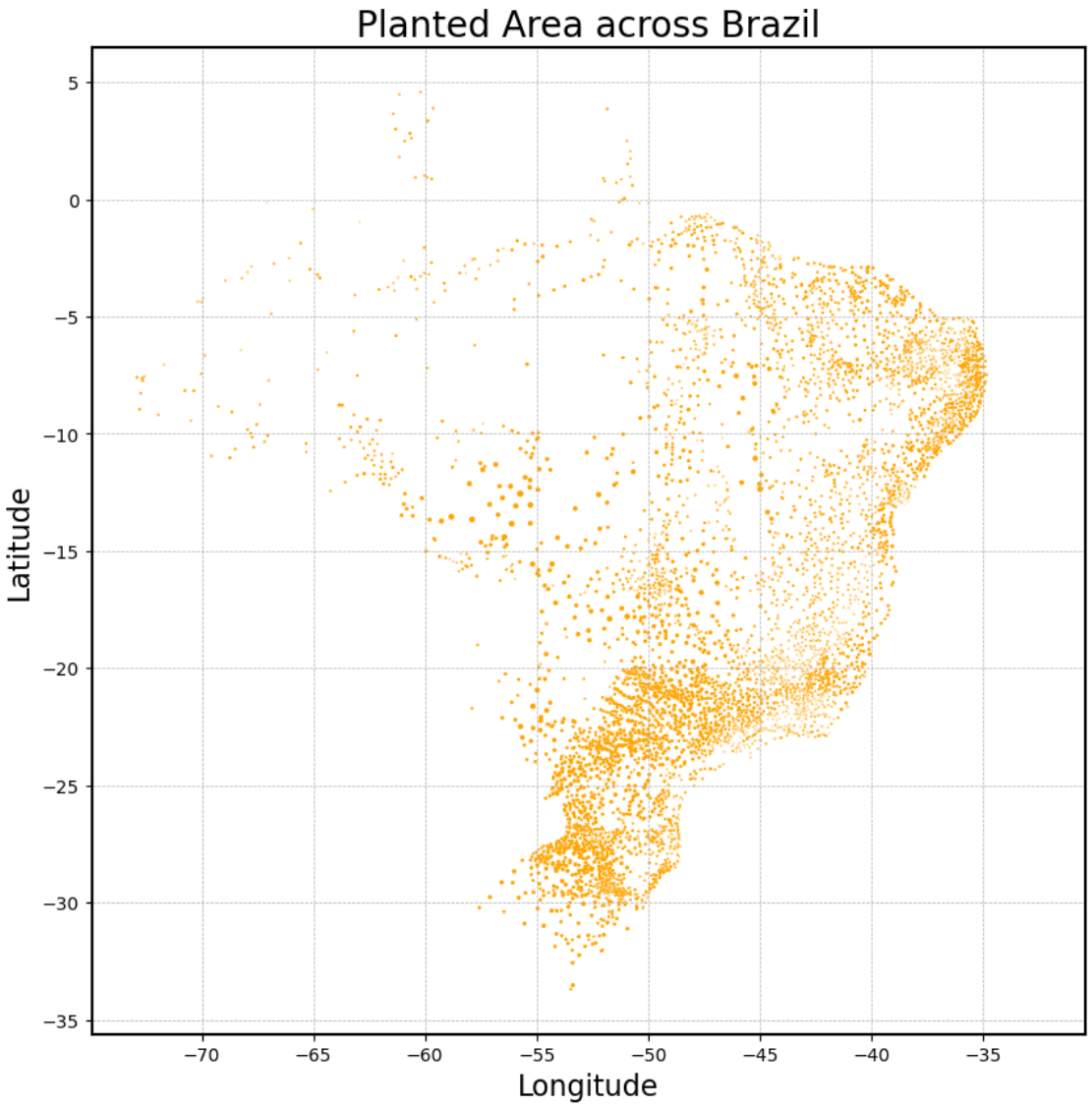
Planted Area

[15]:
```
df['IBGE_PLANTED_AREA'] = np.sqrt(df['IBGE_PLANTED_AREA'])
```

[109]:
```
feature_summary('Planted Area',geographic, df,size = 5)
```

```
<IPython.core.display.HTML object>
```


Distribution of Planted Area
Frequency
35
30
25
20
15
10
5
0
0
Planted Area

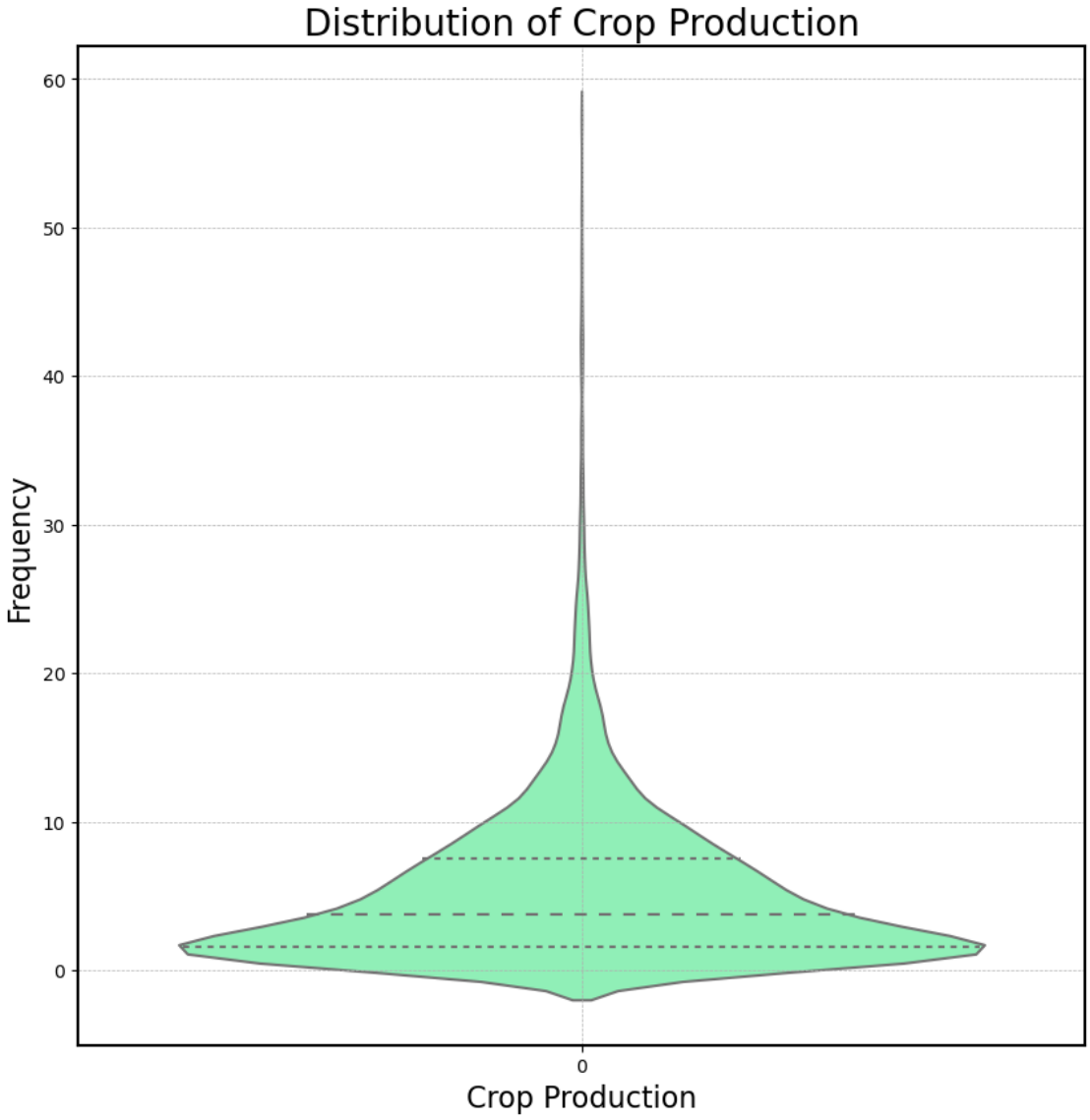


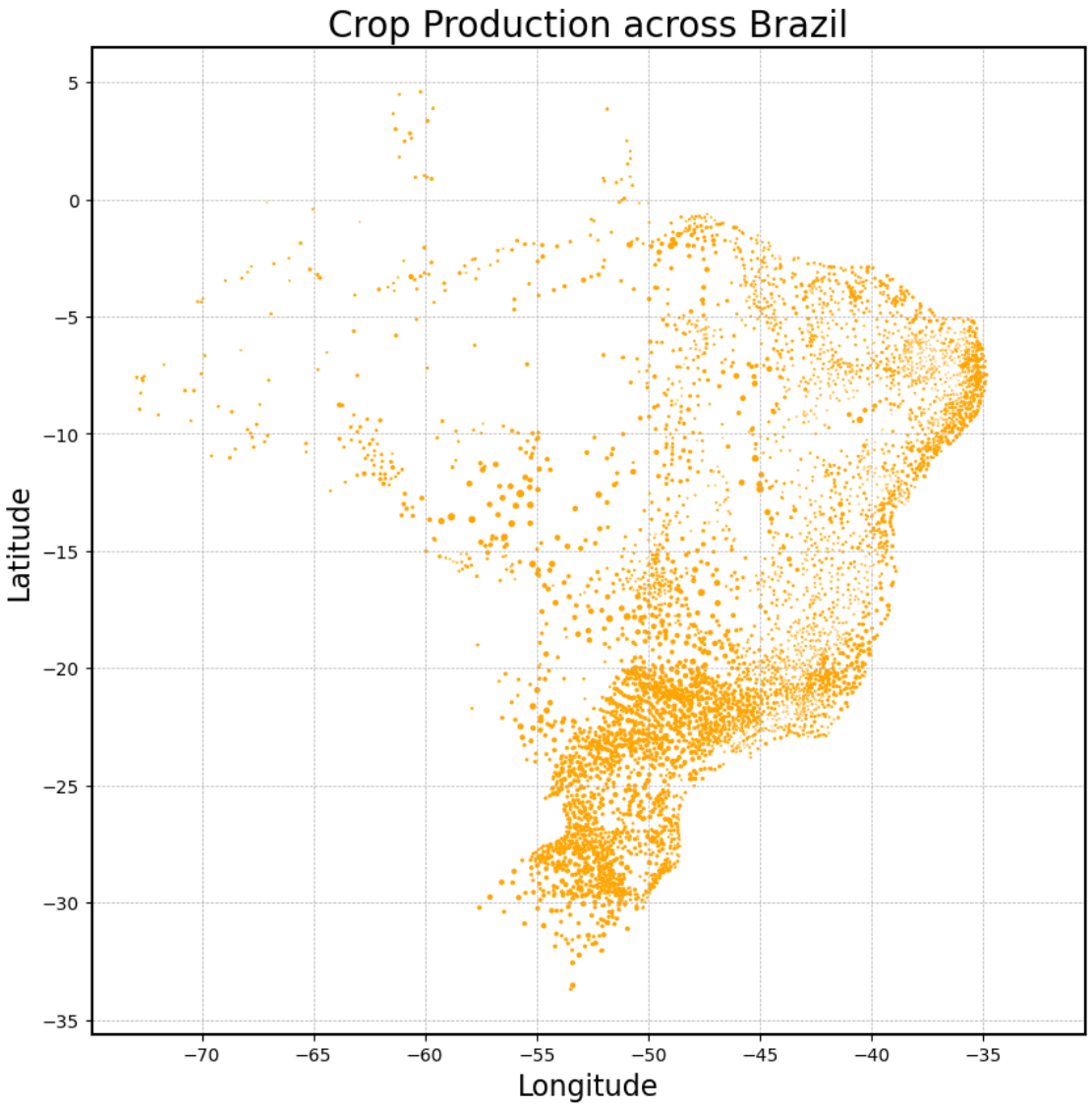
Crop Production

[17]:
```
df['IBGE_CROP_PRODUCTION_$'] = np.sqrt(df['IBGE_CROP_PRODUCTION_$'])
```

[18]:
```
feature_summary('Crop Production',geographic, df,size = 5)
```

```
<IPython.core.display.HTML object>
```


Distribution of Crop Production
Frequency
60
50
40
30
20
10
0
0
Crop Production

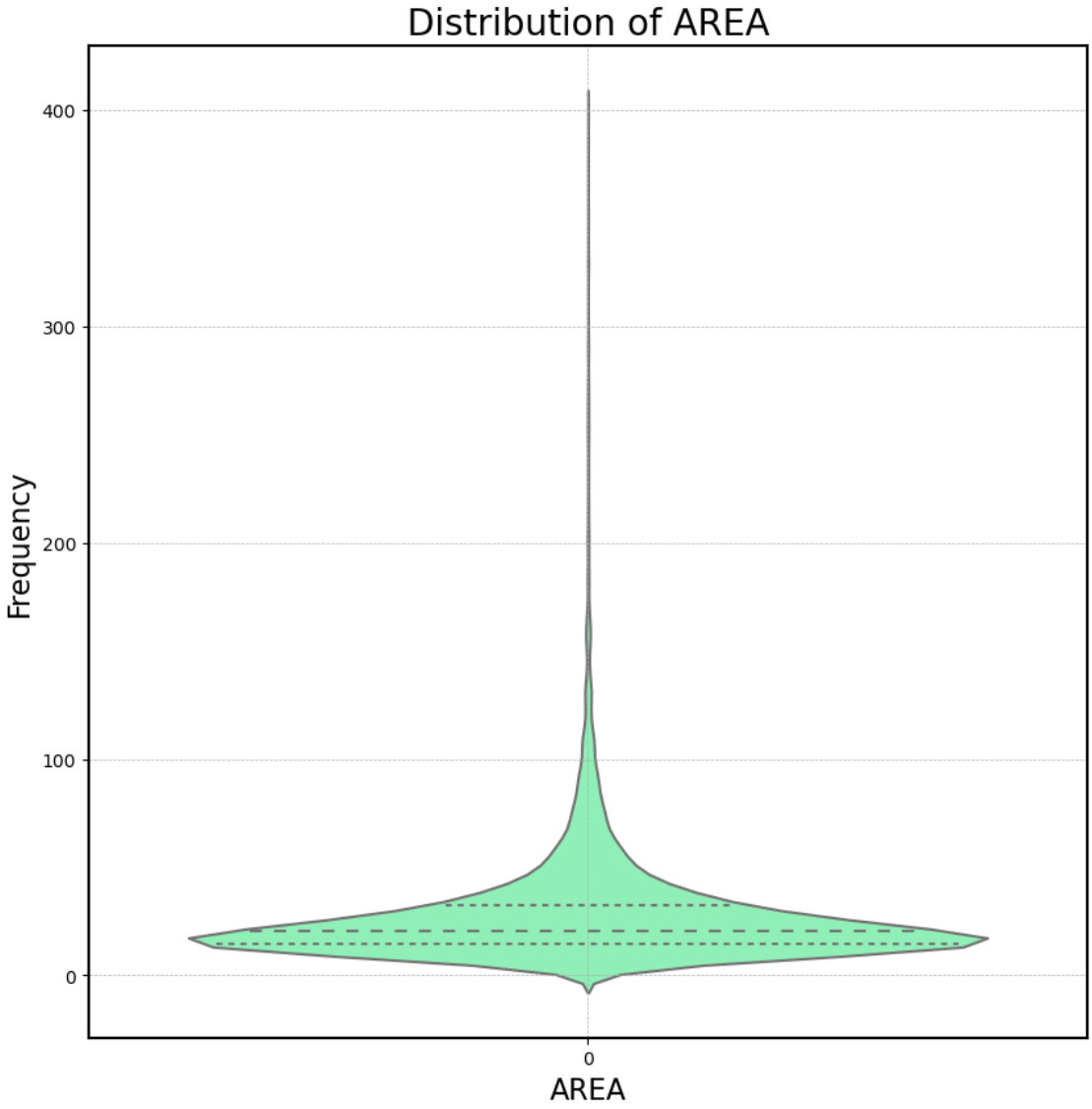


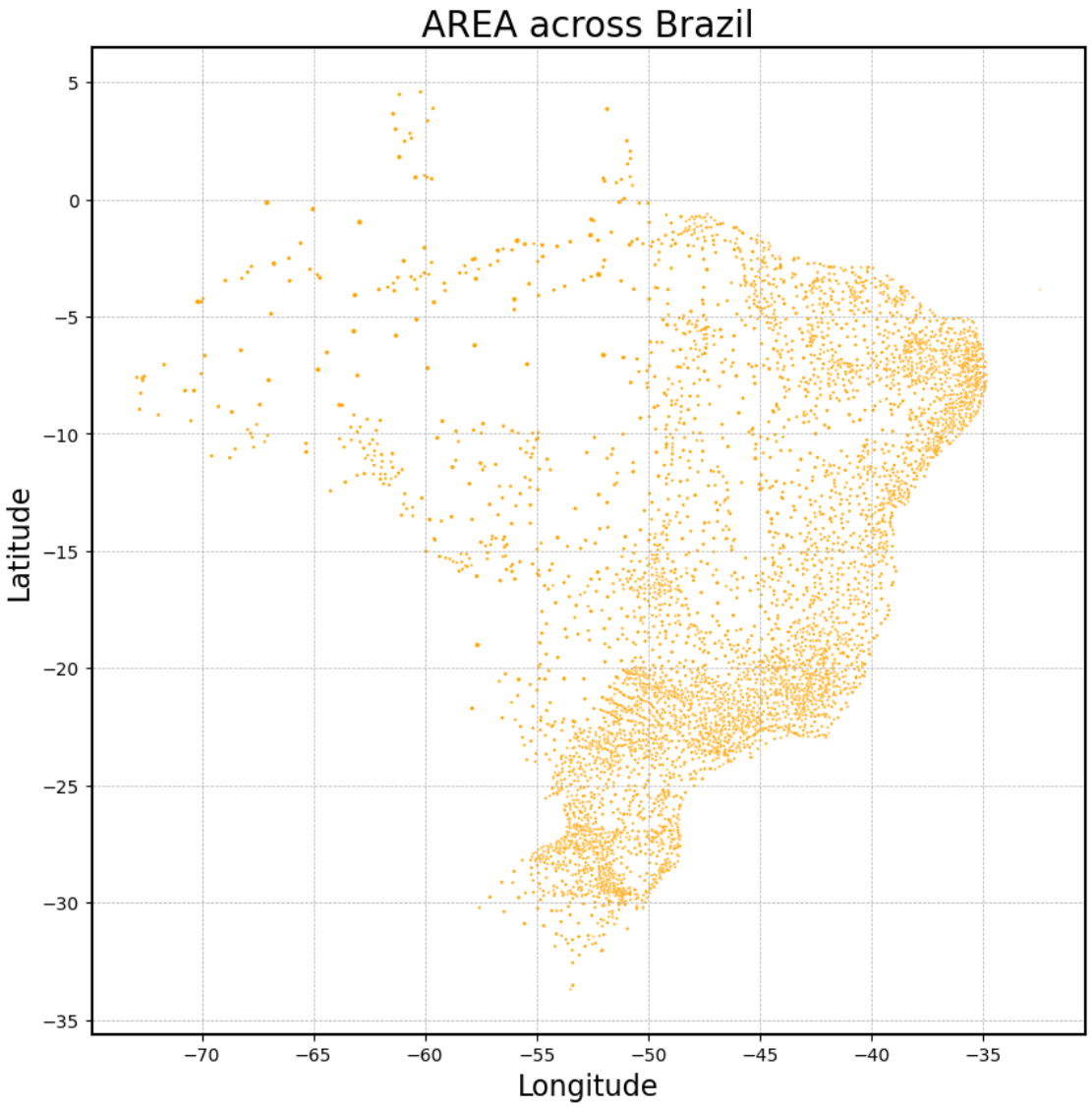
Area

[19]:
```
df['AREA'] = np.sqrt(df['AREA'])
```

[110]:
```
feature_summary('AREA',geographic, df,size = 100)
```

```
<IPython.core.display.HTML object>
```


Distribution of AREA
Frequency
400
300
200
100
0
0
AREA



Rural or Urban

[21]:
```
rural_urban_translations = {'Sem classificação':'Unclassified',
                            'Intermediário Remoto':'Intermediate Remote',
                            'Rural Remoto':'Rural Remote',
                            'Intermediário Adjacente':'Adjacent Intermediate',
                            'Urbano':'Urban',
                            'Rural Adjacente':'Adjacent Rural'}
```

[22]:
```
df['RURAL_URBAN'] = df['RURAL_URBAN'].replace(rural_urban_translations)
```

Encoding the catregories as dummies and combining them

[23]:
```
df['RURAL_URBAN'].value_counts()
```

[23]:
```
Adjacent Rural           2833
Urban                    1420
Adjacent Intermediate     666
Rural Remote              314
Intermediate Remote        58
Unclassified                5
Name: RURAL_URBAN, dtype: int64
```

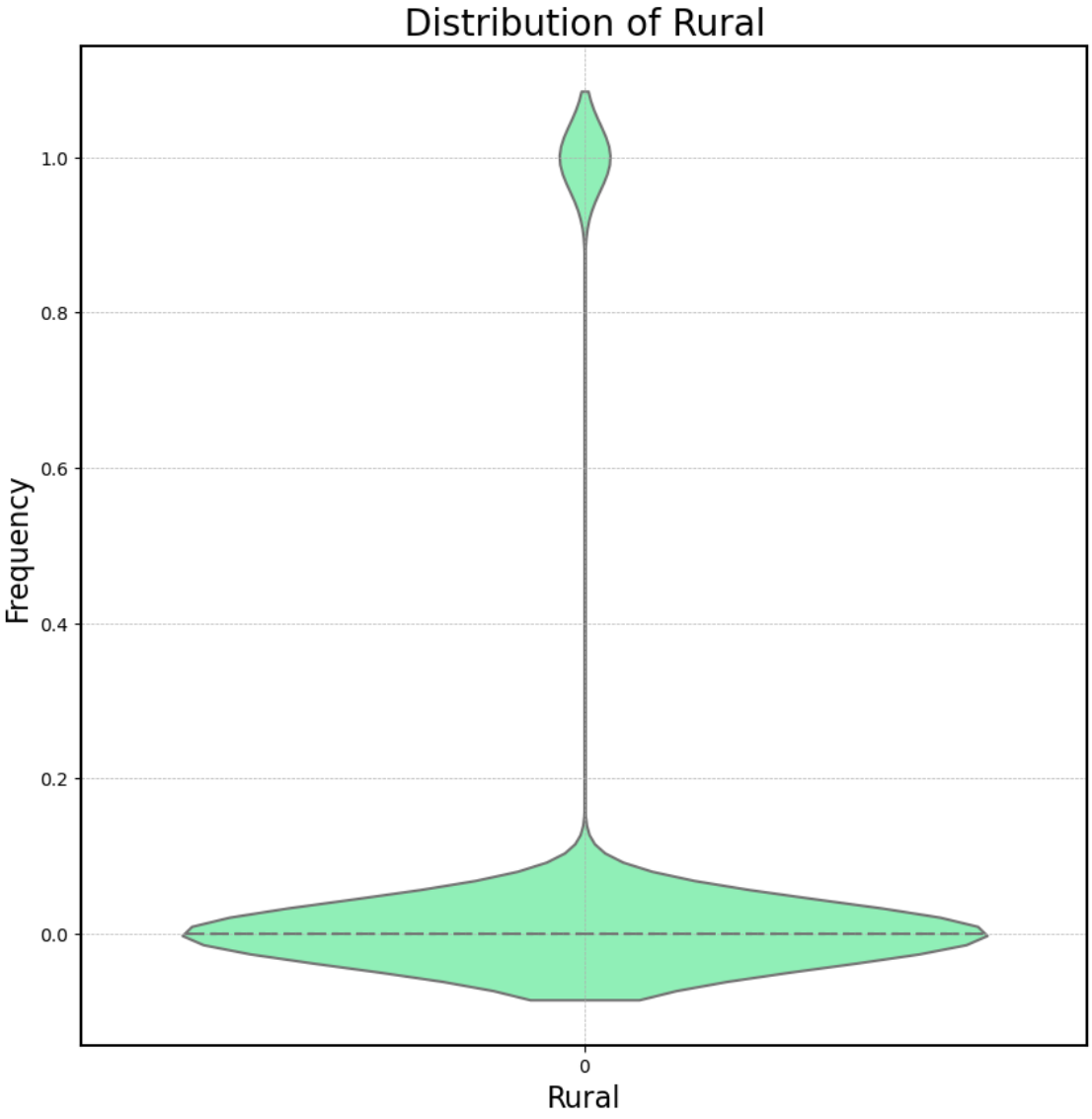
[24]:
```
df = pd.get_dummies(df,columns = ['RURAL_URBAN'])
```

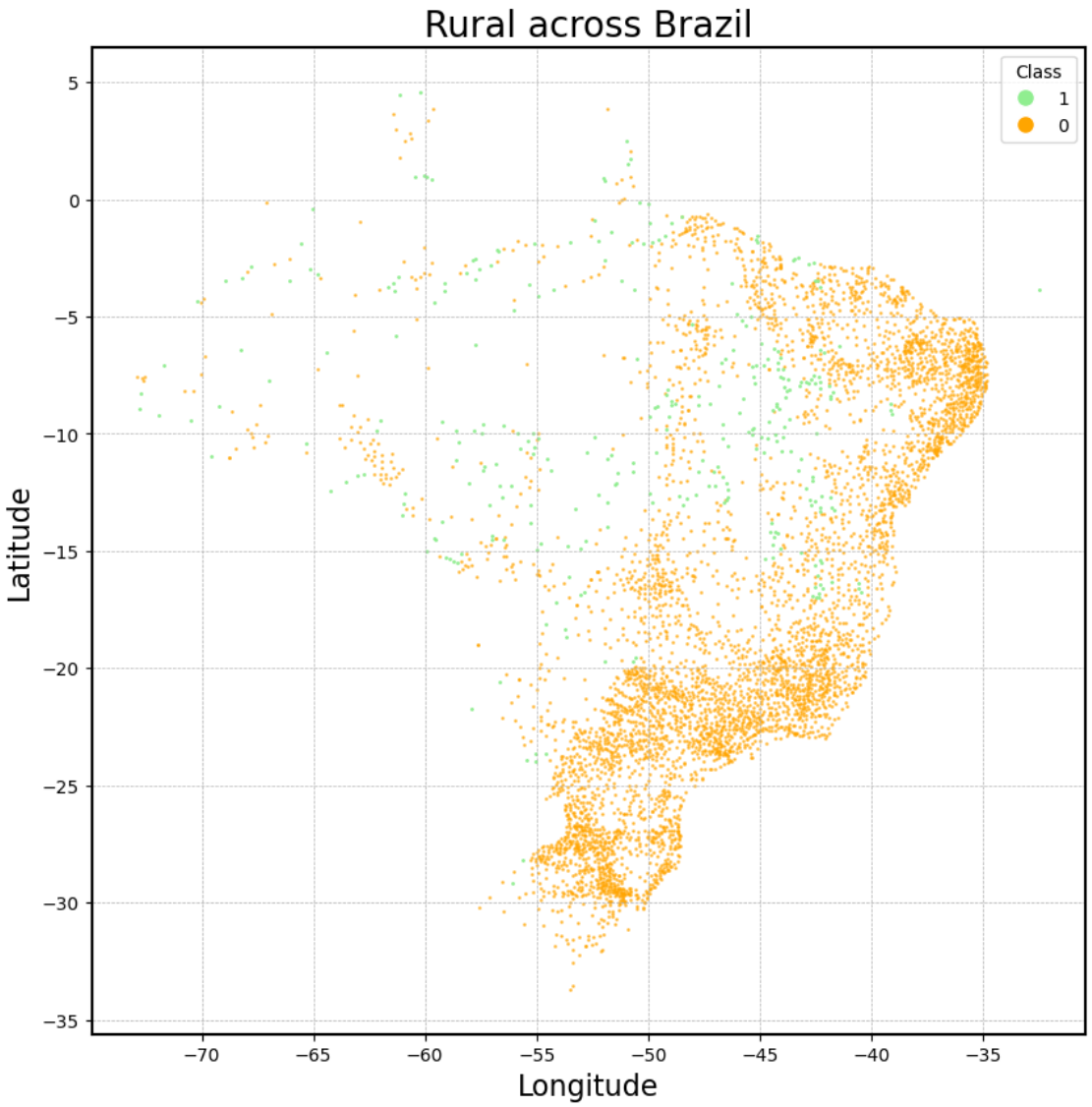
[115]:
```
df['Rural'] = df[['RURAL_URBAN_Rural Remote']]
```

[26]:
```
geographic['Rural'] = 'Rural'
```

[123]:
```
binary_feature_summary('Rural',geographic, df,size = 1)
```

```
<IPython.core.display.HTML object>
```


Distribution of Rural
Frequency
1.0
0.8
0.6
0.4
0.2
0.0
0
Rural



2.1.2 Population Demographic Features

[28]:
```
df = df.dropna(subset = ['POP_GDP'])
```

[30]:
```
df['LOG_POP'] = np.log(df['POP_GDP'])
```

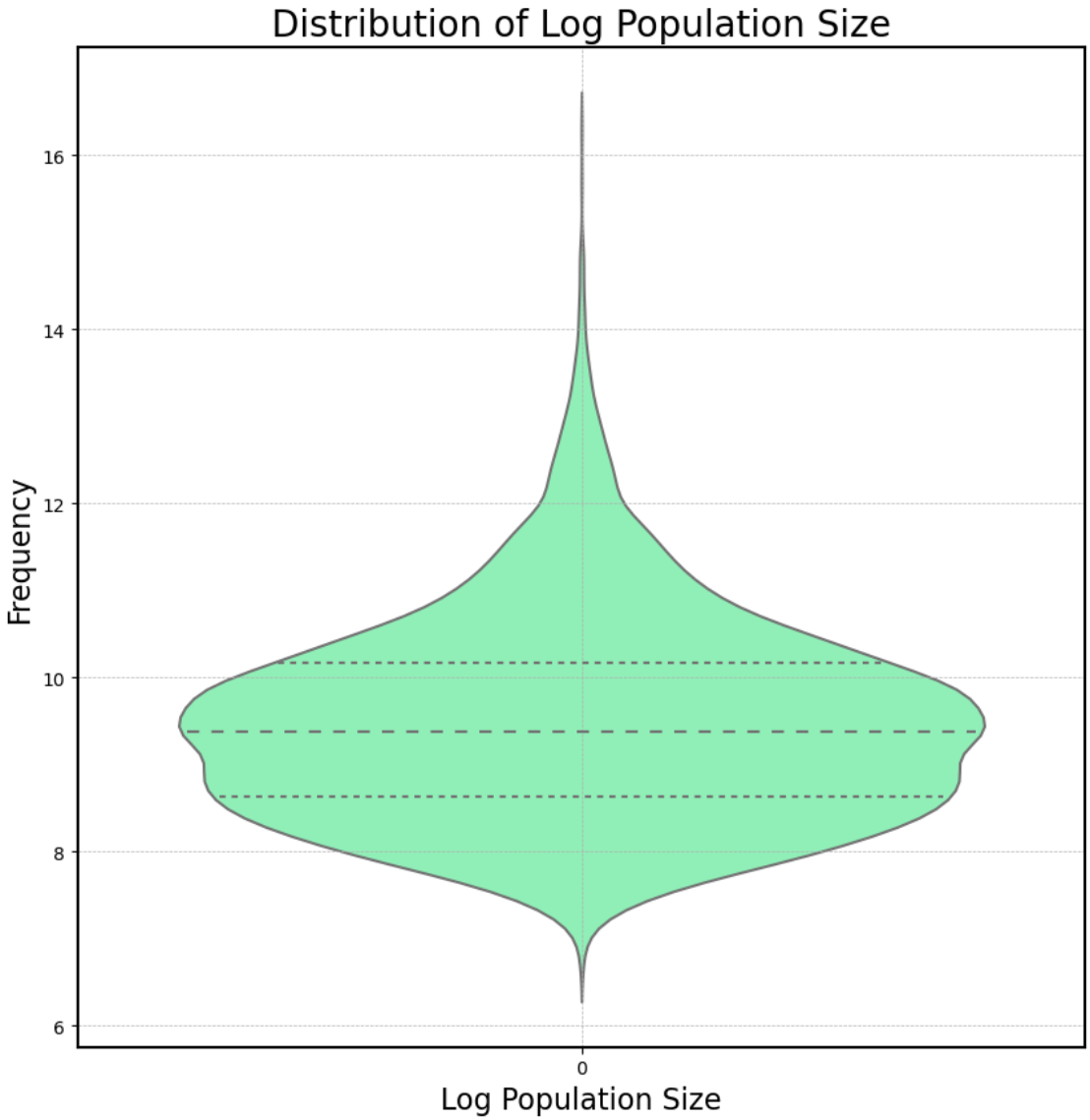
[31]:
```
population = {'Population Size':'POP_GDP',
              'HDI':'IDHM',
              'Life Expectancy Index':'IDHM_Longevidade',
              'Log Population Size':'LOG_POP',
              'Education':'IDHM_Educacao'
```

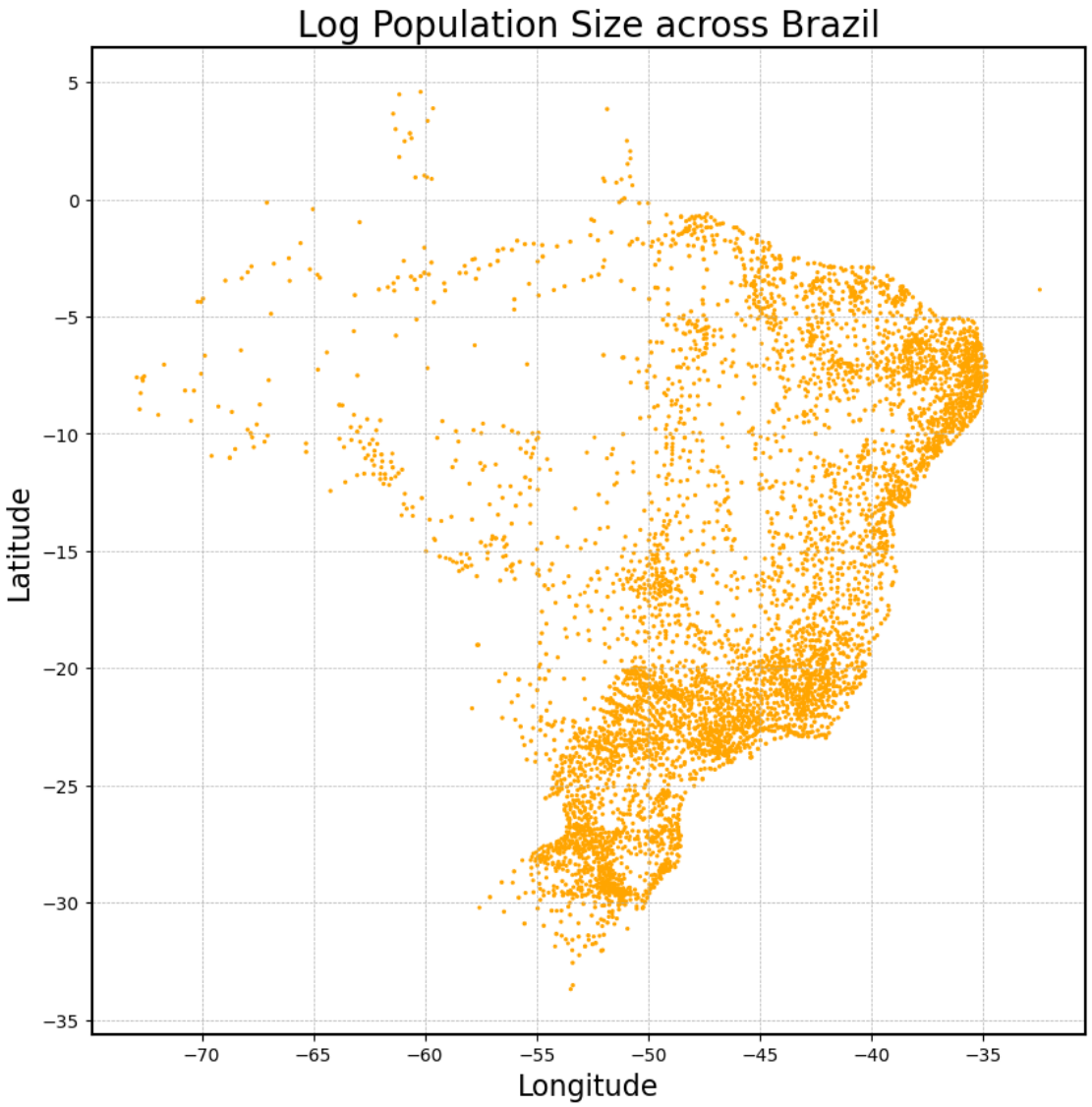
```
        }
```

Population Size

[126]:
```
feature_summary('Log Population Size',population,df,size = 5)
```

```
<IPython.core.display.HTML object>
```

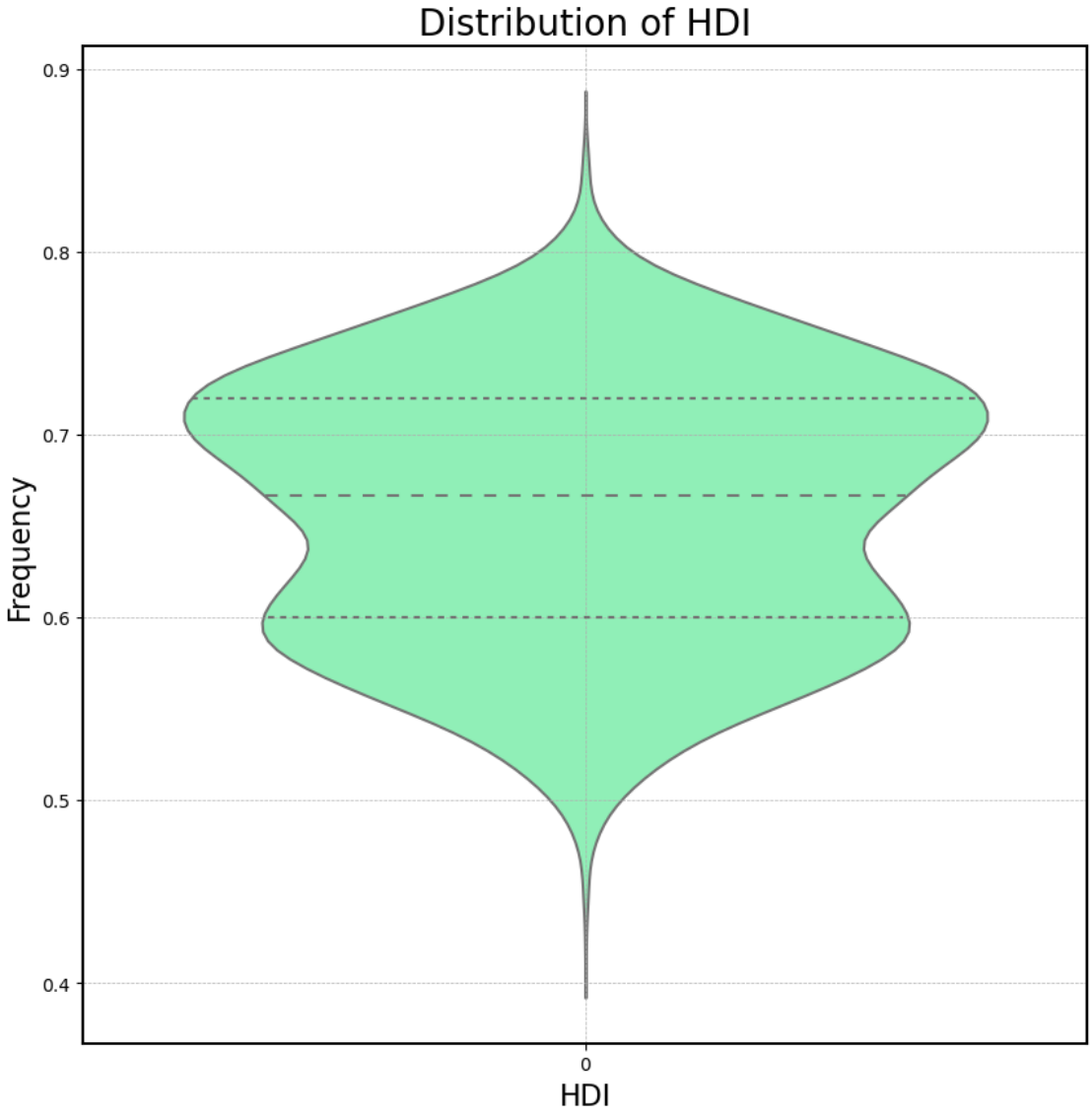


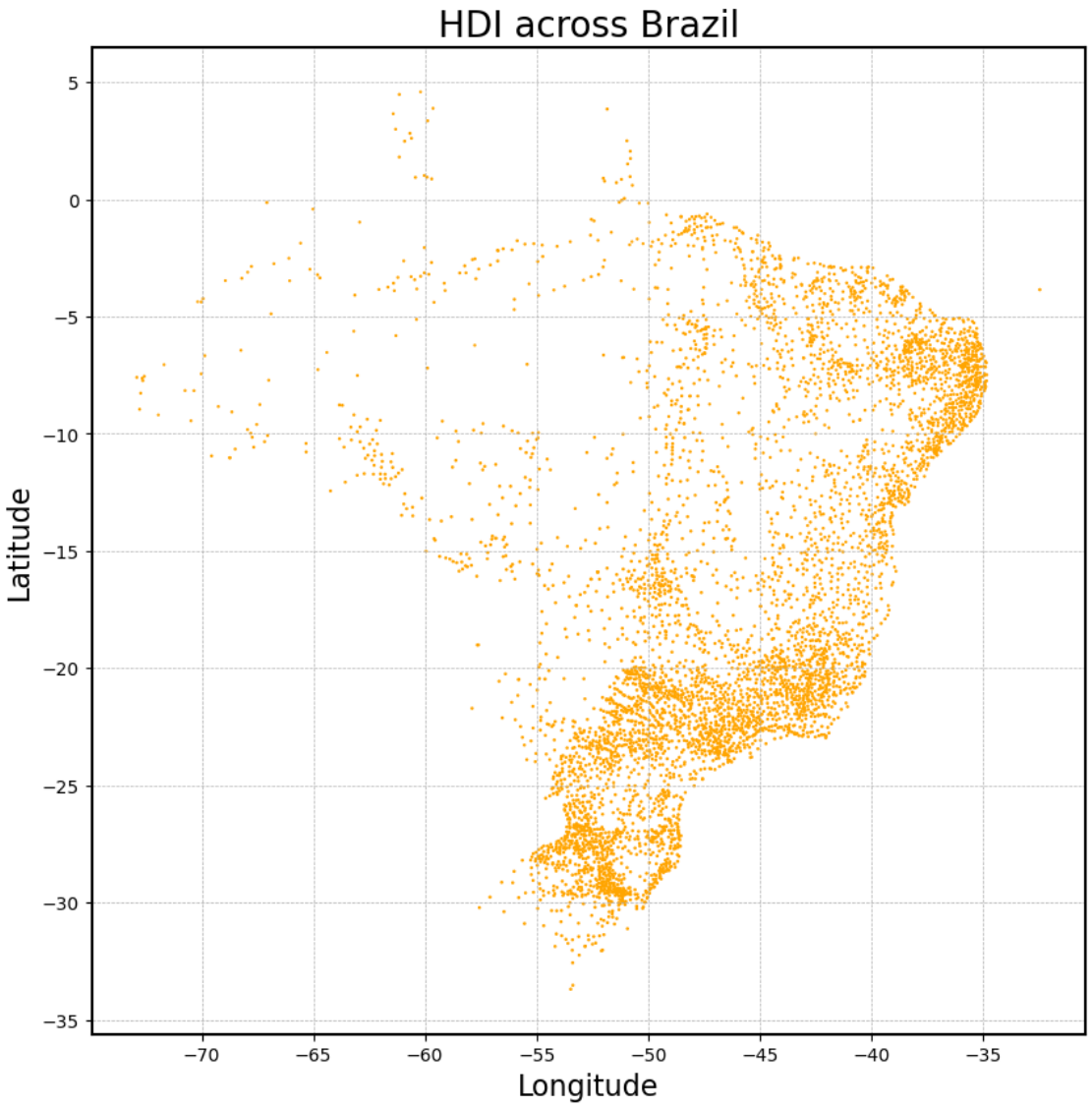


HDI

[34]:
```
feature_summary('HDI',population,df,size = 1)
```

```
<IPython.core.display.HTML object>
```

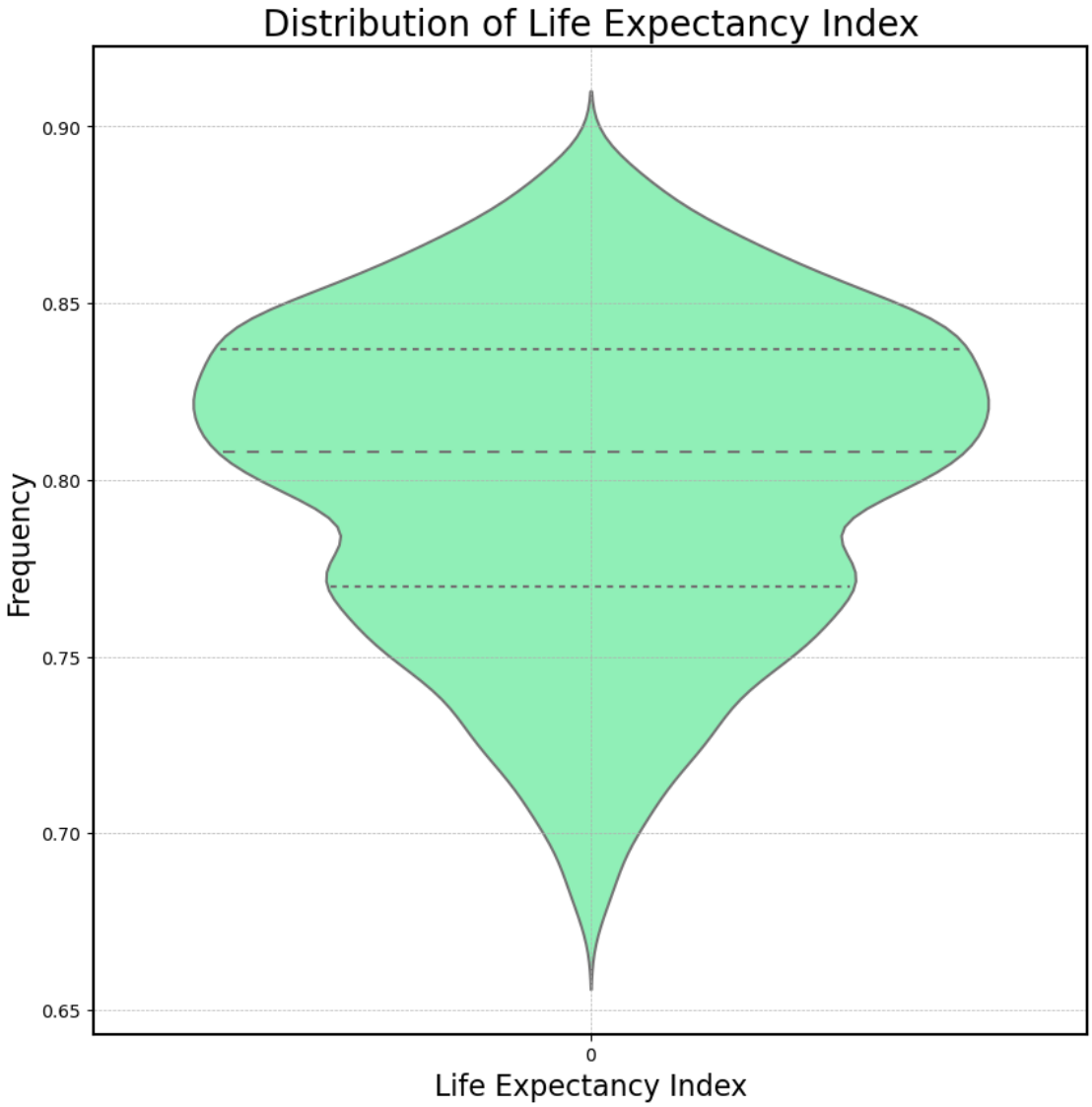

Distribution of HDI
0.9
0.8
0.7
0.6
0.5
0.4
Frequency
0
HDI

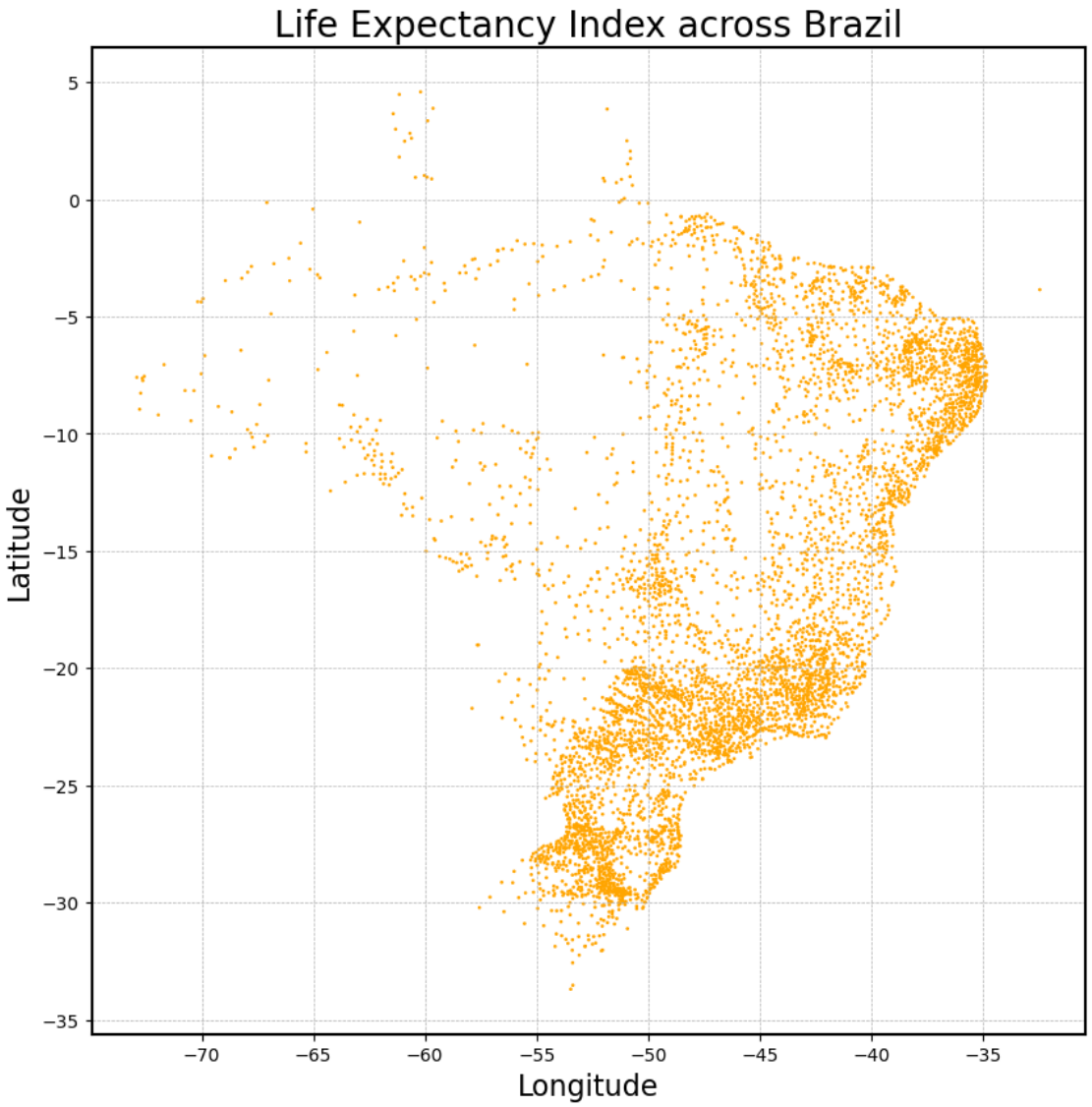


Life Expectancy

[35]:
```
feature_summary('Life Expectancy Index',population,df,size = 1)
```

```
<IPython.core.display.HTML object>
```

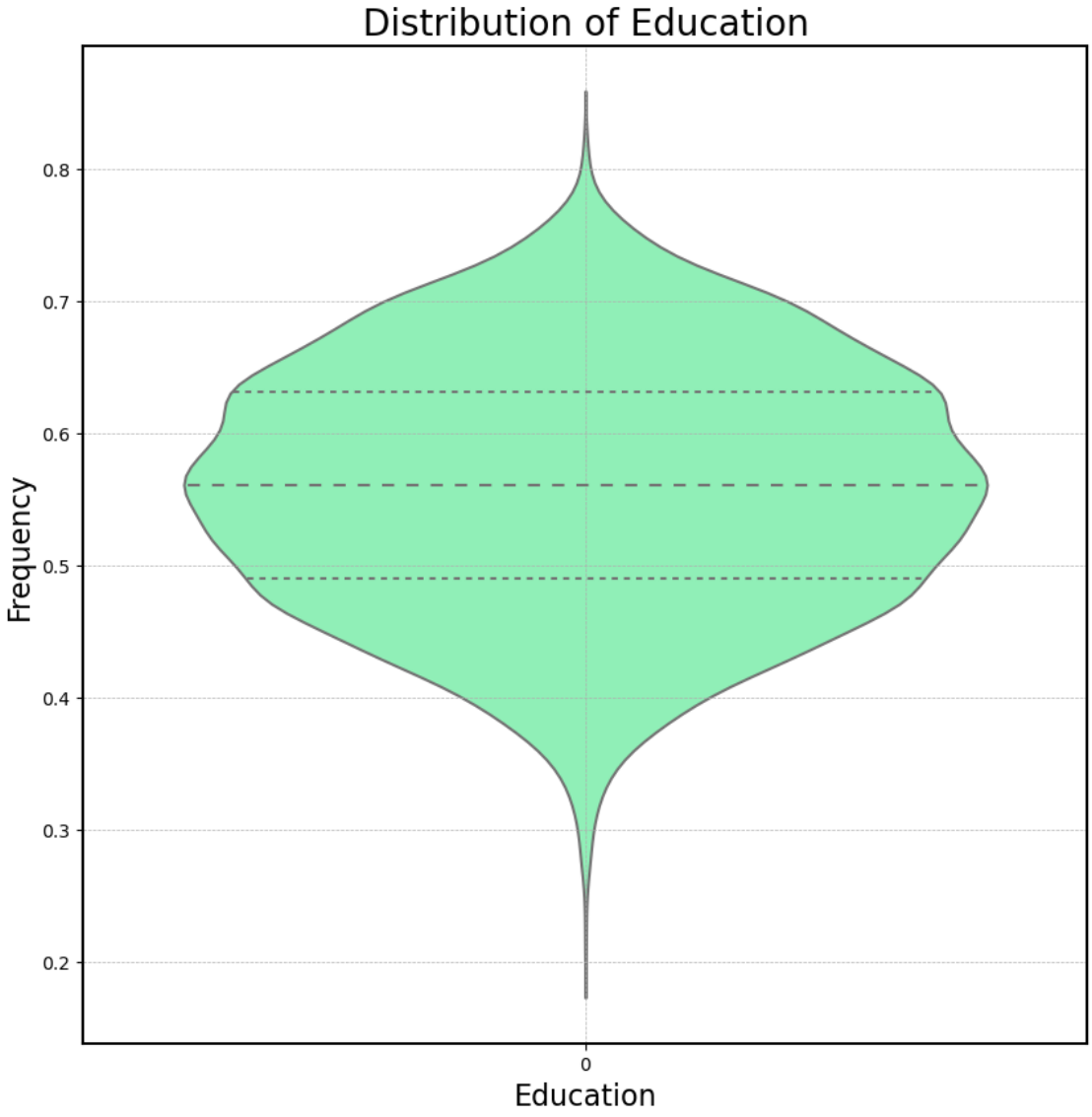

Distribution of Life Expectancy Index
Frequency
0.90
0.85
0.80
0.75
0.70
0.65
0
Life Expectancy Index

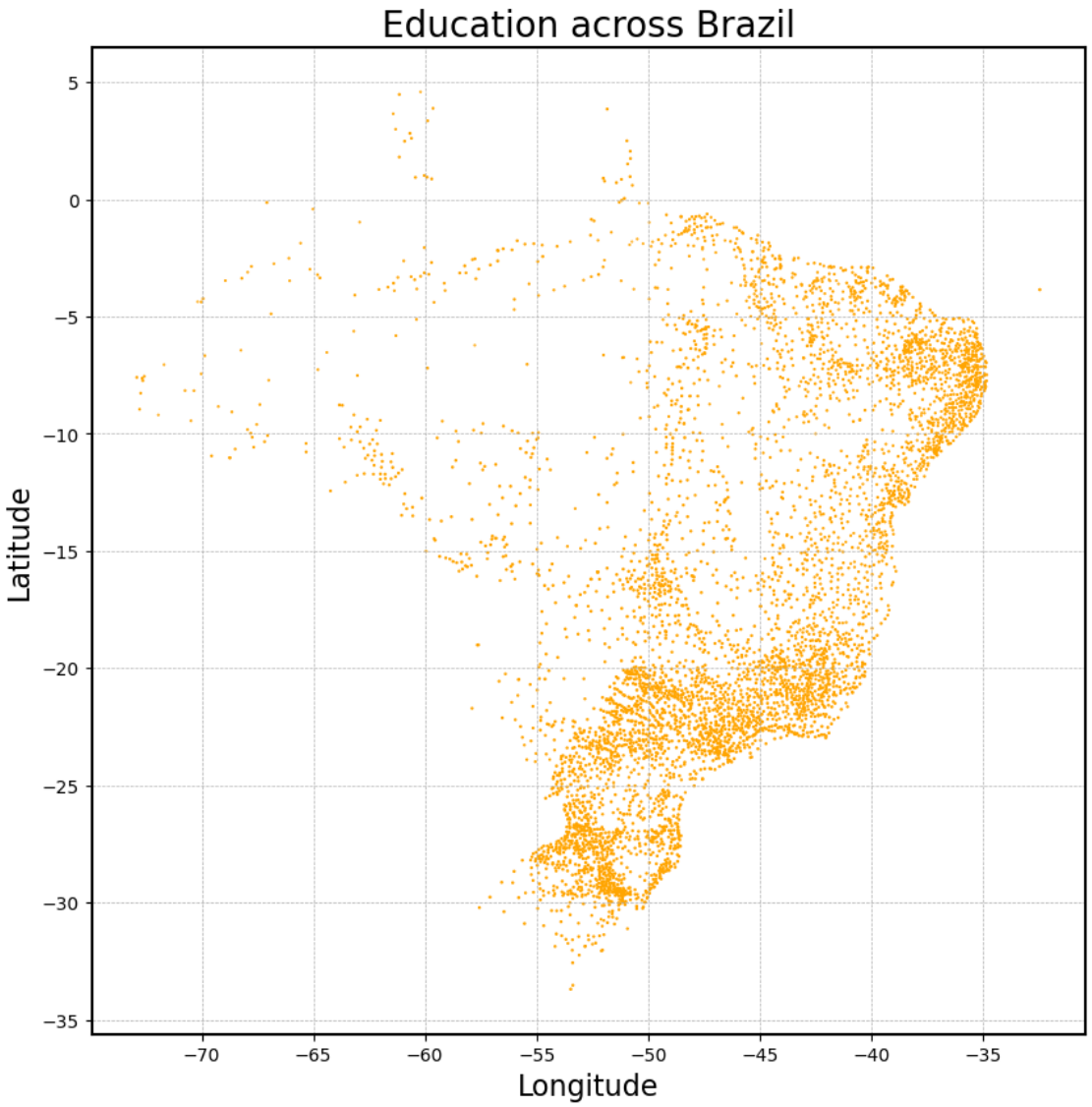


Education Index

[36]:
```
feature_summary('Education',population,df,size = 1)
```

```
<IPython.core.display.HTML object>
```


Distribution of Education
Frequency
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
Education



2.1.3 Public and Private Wealth

[41]:
```
df['MUN_EXPENDIT'] = df['MUN_EXPENDIT'].fillna(df['MUN_EXPENDIT'].mean())
```

[42]:
```
df['MUN_EXPENDIT_PC'] = df['MUN_EXPENDIT']/(df[population['Population␣
↪Size']]*1000)
```

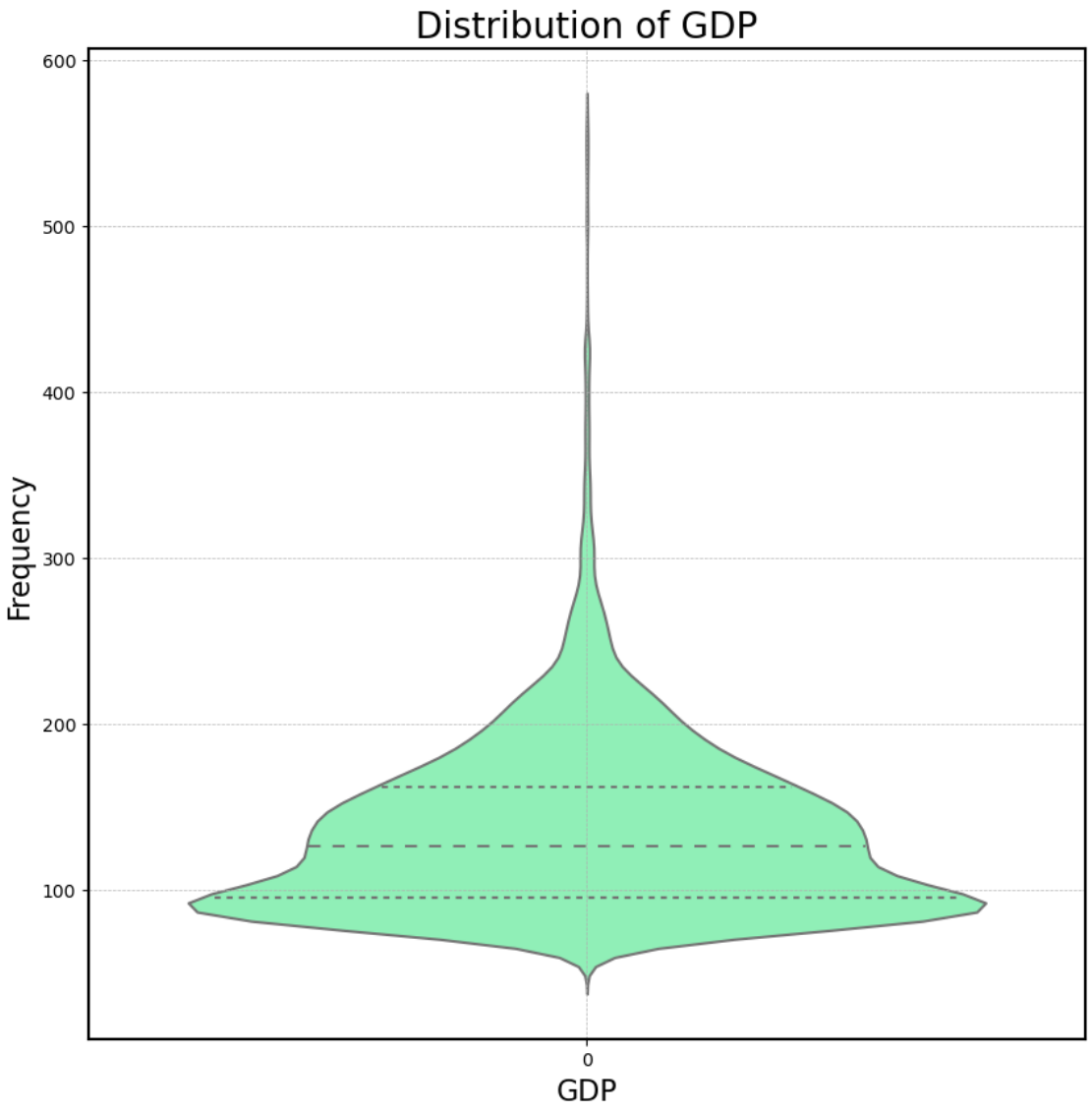
[43]:
```
wealth = {'GDP':'GDP_CAPITA',
          'Municipal Expenditures':'MUN_EXPENDIT_PC'
          }
```

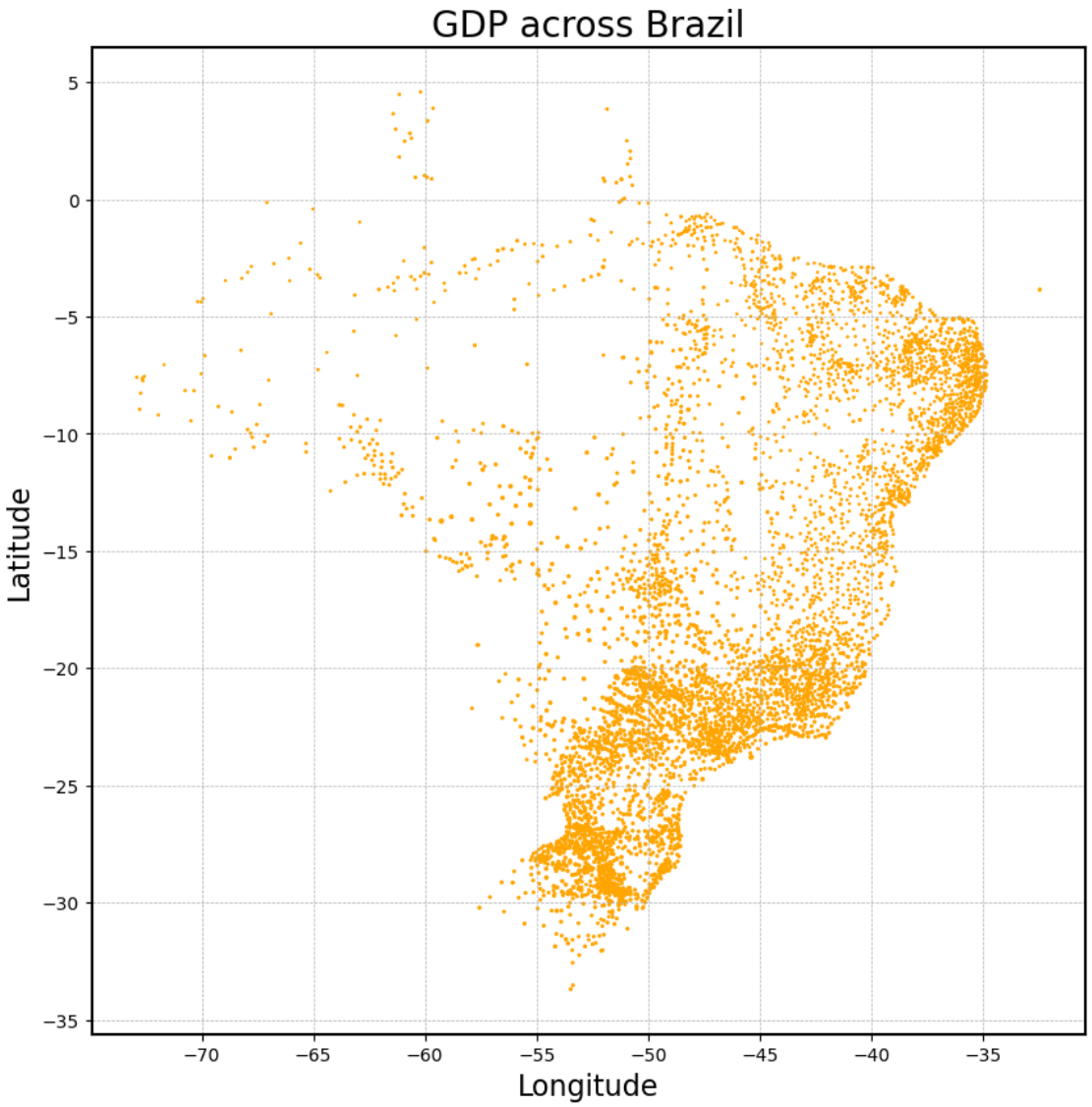
GDP

[44]:
```
df['GDP_CAPITA'] = np.sqrt(df['GDP_CAPITA'])
```

[128]:
```
feature_summary('GDP',wealth,df,size = 100)
```

```
<IPython.core.display.HTML object>
```



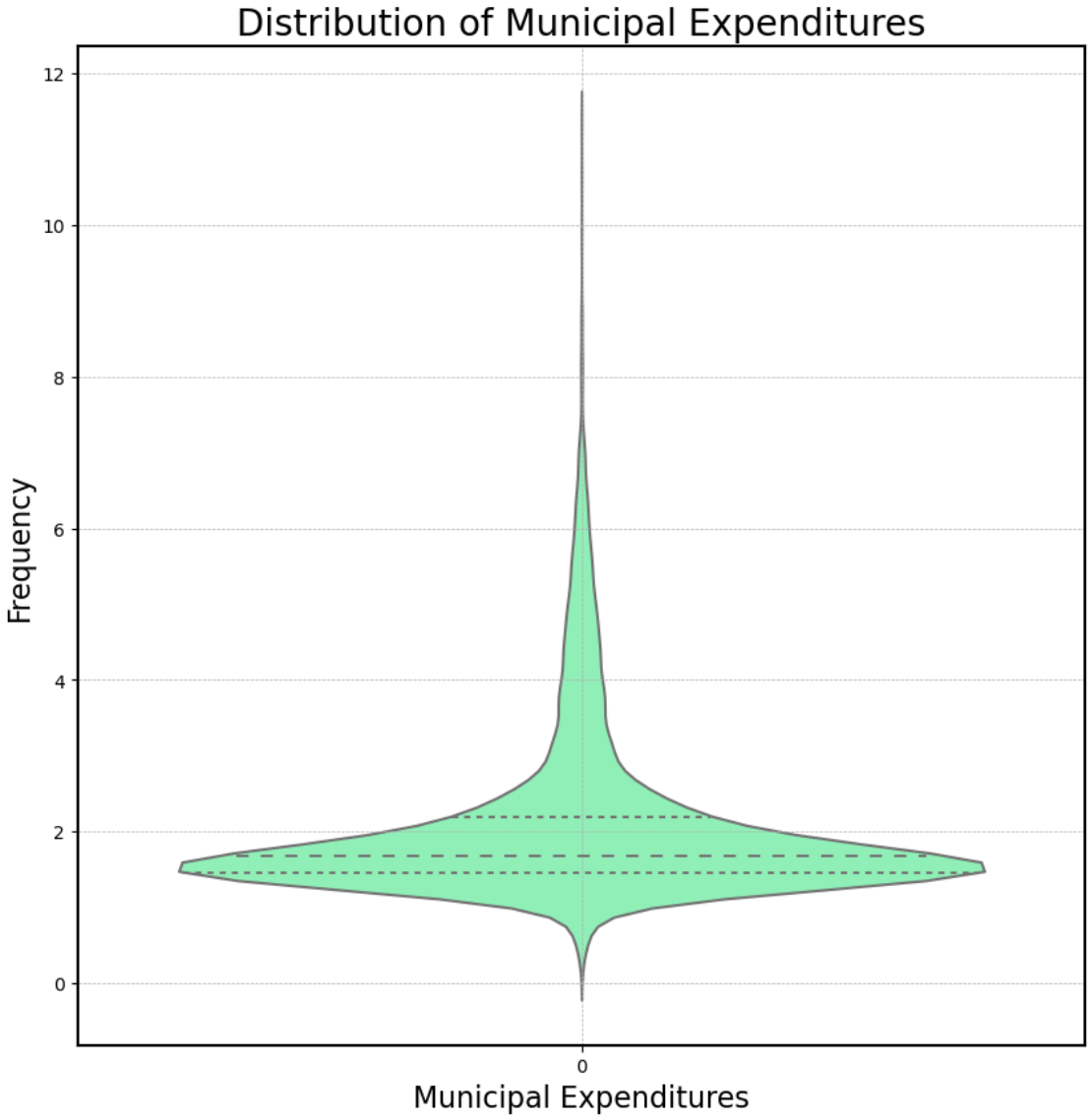


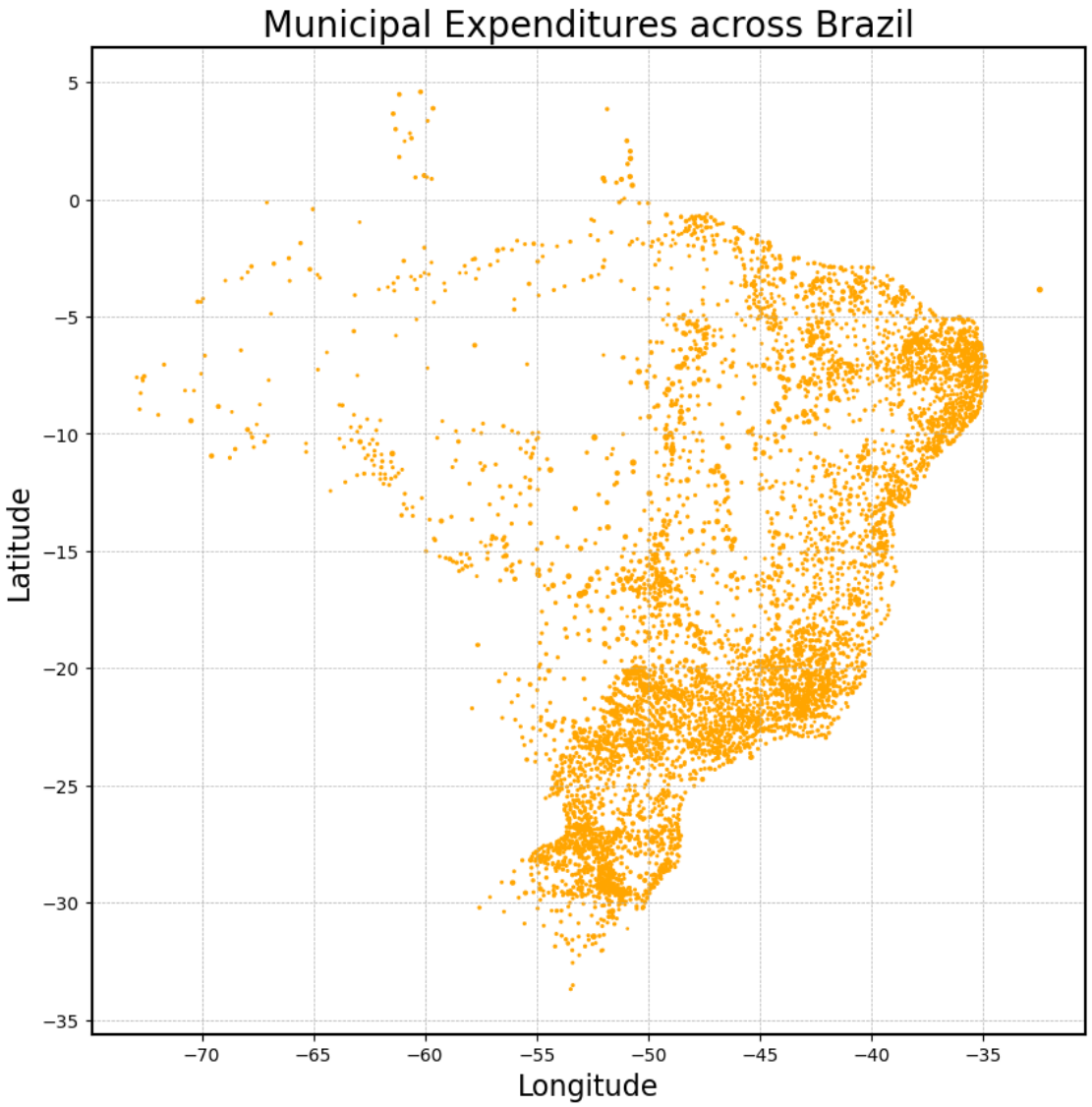
Municipal Expenditure Per Capita

[46]:
```
df['MUN_EXPENDIT_PC'] = np.sqrt(df['MUN_EXPENDIT_PC'])
```

[47]:
```
feature_summary('Municipal Expenditures',wealth,df,size = 1)
```

```
<IPython.core.display.HTML object>
```


Distribution of Municipal Expenditures
Frequency
12
10
8
6
4
2
0
0
Municipal Expenditures



2.1.4 Industry Features

[48]:
```
primary = ['COMP_A','COMP_B']
secondary = ['COMP_D','COMP_E','COMP_F']
tertiary =␣
 ↪['COMP_G','COMP_H','COMP_I','COMP_J','COMP_K','COMP_L','COMP_M','COMP_N','COMP_O','COMP_P',
```

[49]:
```
df['PRIMARY_PROPORTION'] = df[primary].sum(axis = 1)/df['COMP_TOT']
df['SECONDARY_PROPORTION'] = df[secondary].sum(axis = 1)/df['COMP_TOT']
df['TERTIARY_PROPORTION'] = df[tertiary].sum(axis = 1)/df['COMP_TOT']
```

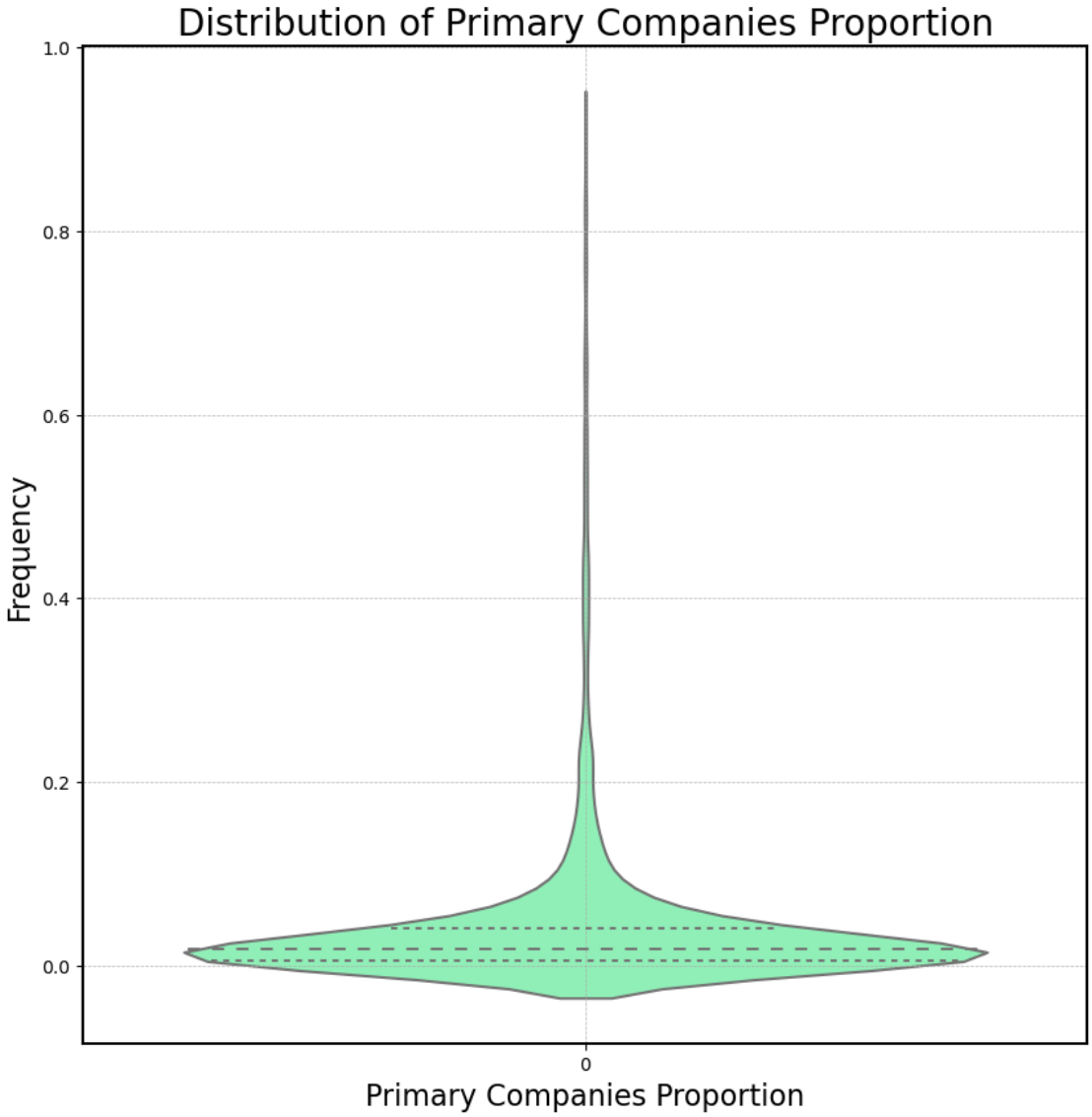
```
df['GVA_AGROPEC_PC'] = np.sqrt(df['GVA_AGROPEC']/(df[population['Population␣
 ↪Size']])) #all are per person, dont need to be mult by 100 as they are␣
 ↪already per 1000 dollars
df['GVA_INDUSTRY_PC'] = np.sqrt(df['GVA_INDUSTRY']/(df[population['Population␣
 ↪Size']]))
df['GVA_SERVICES_PC'] = np.sqrt(df['GVA_SERVICES']/(df[population['Population␣
 ↪Size']]))
```

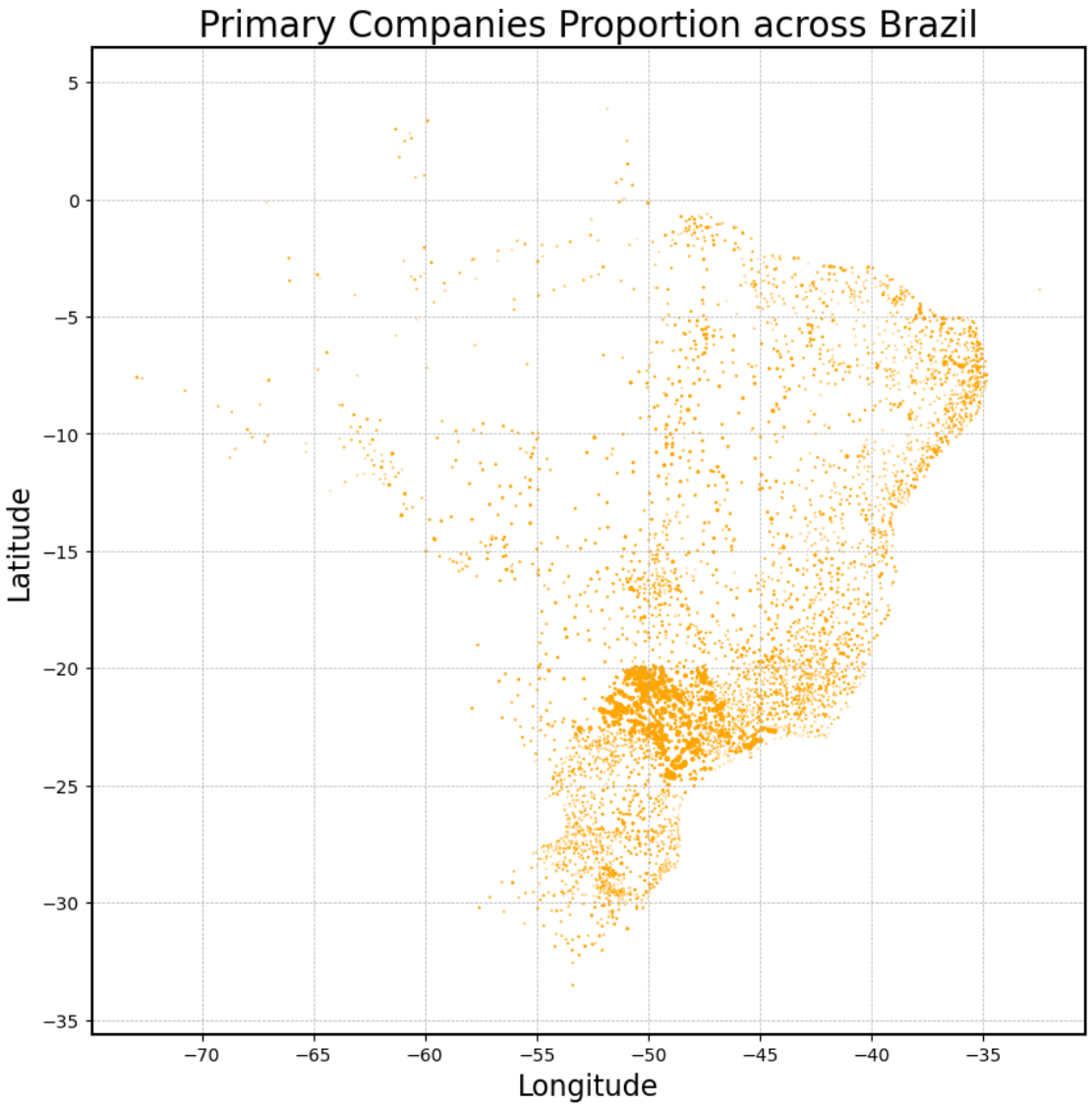
[50]:
```
industry = {'Primary Companies Proportion':'PRIMARY_PROPORTION',
            'Secondary Companies Proportion':'SECONDARY_PROPORTION',
             'Tertiary Companies Proportion':'TERTIARY_PROPORTION',
            'GVA Primary':'GVA_AGROPEC_PC',
            'GVA Secondary':'GVA_INDUSTRY_PC',
            'GVA Tertiary':'GVA_SERVICES_PC'


            }
```

Primary Proportion

[51]:
```
feature_summary('Primary Companies Proportion',industry,df,size = 0.1)
```

```
<IPython.core.display.HTML object>
```

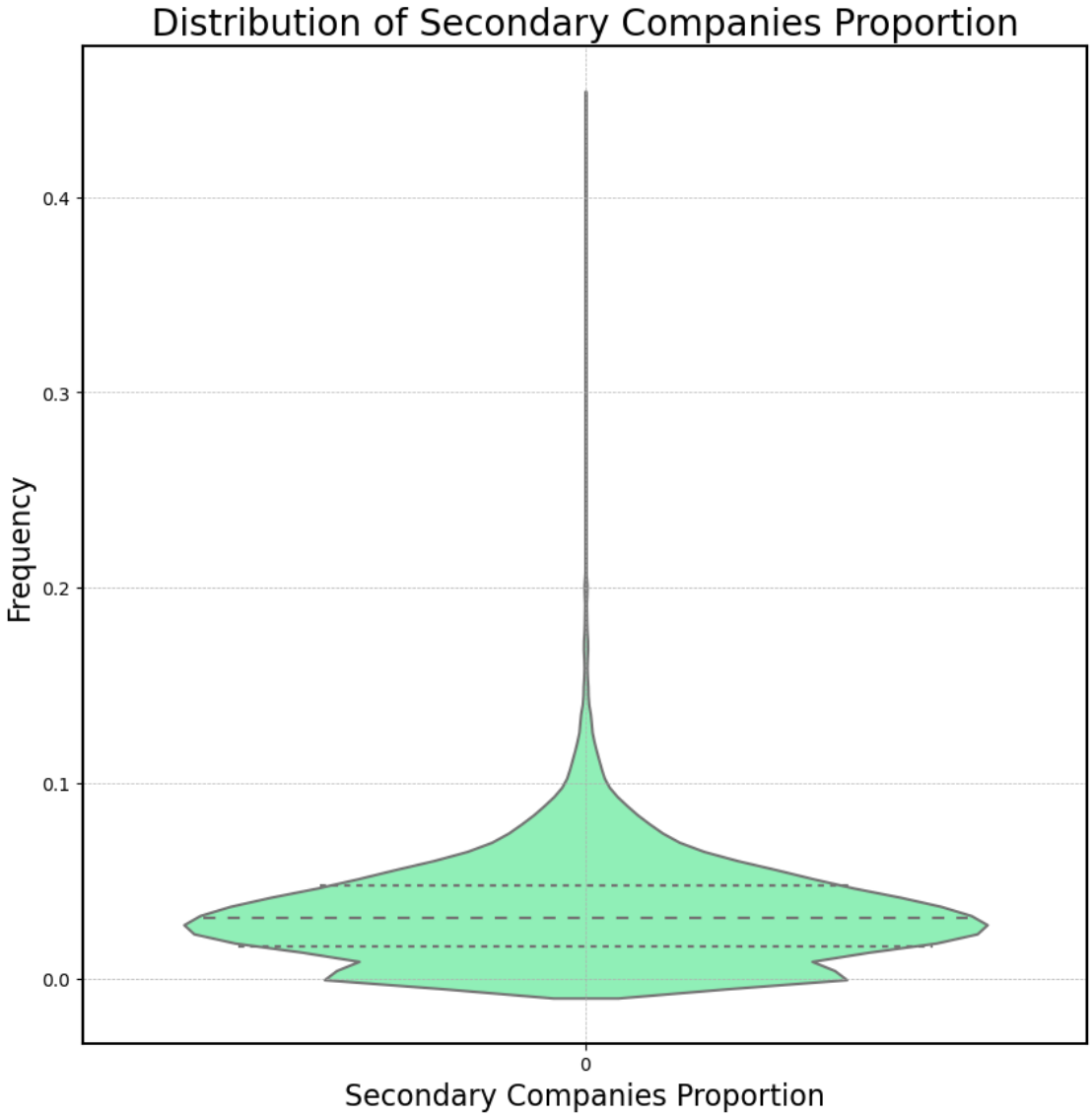

Distribution of Primary Companies Proportion
Frequency
1.0
0.8
0.6
0.4
0.2
0.0
0
Primary Companies Proportion

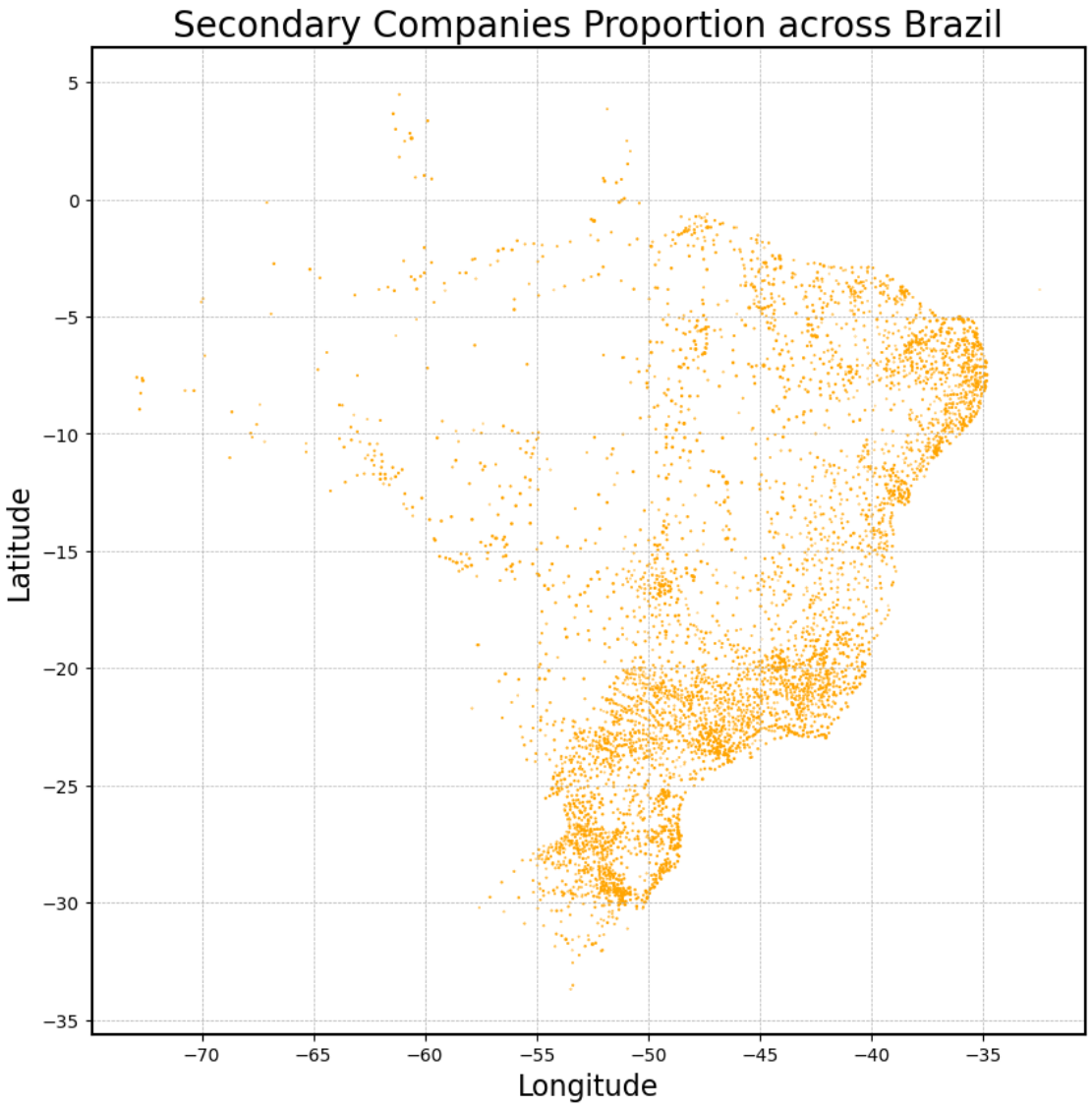


Secondary Proportion

[52]:
```
feature_summary('Secondary Companies Proportion',industry,df,size = 0.1)
```

```
<IPython.core.display.HTML object>
```

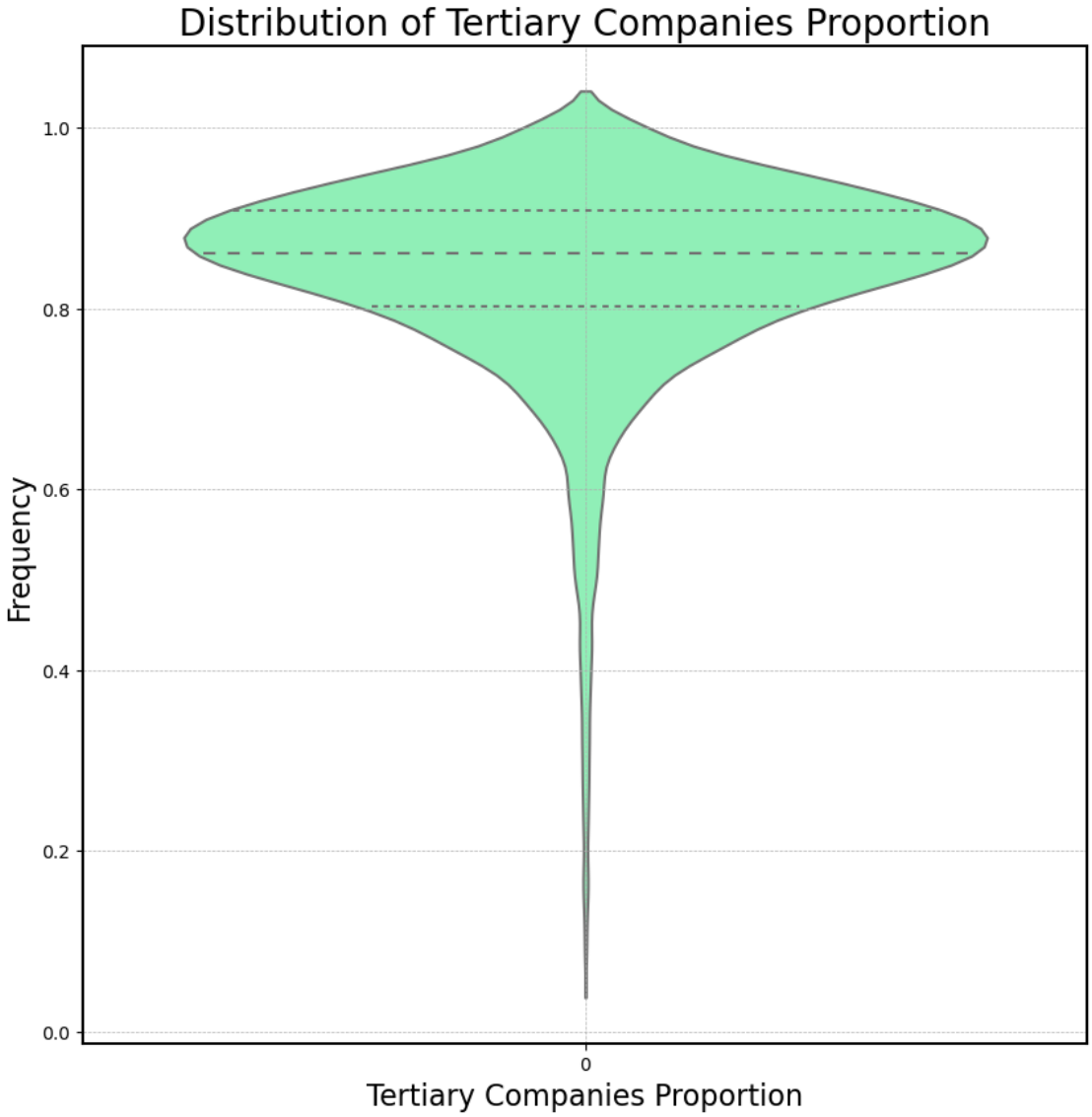

Distribution of Secondary Companies Proportion
Frequency
0.4
0.3
0.2
0.1
0.0
0
Secondary Companies Proportion

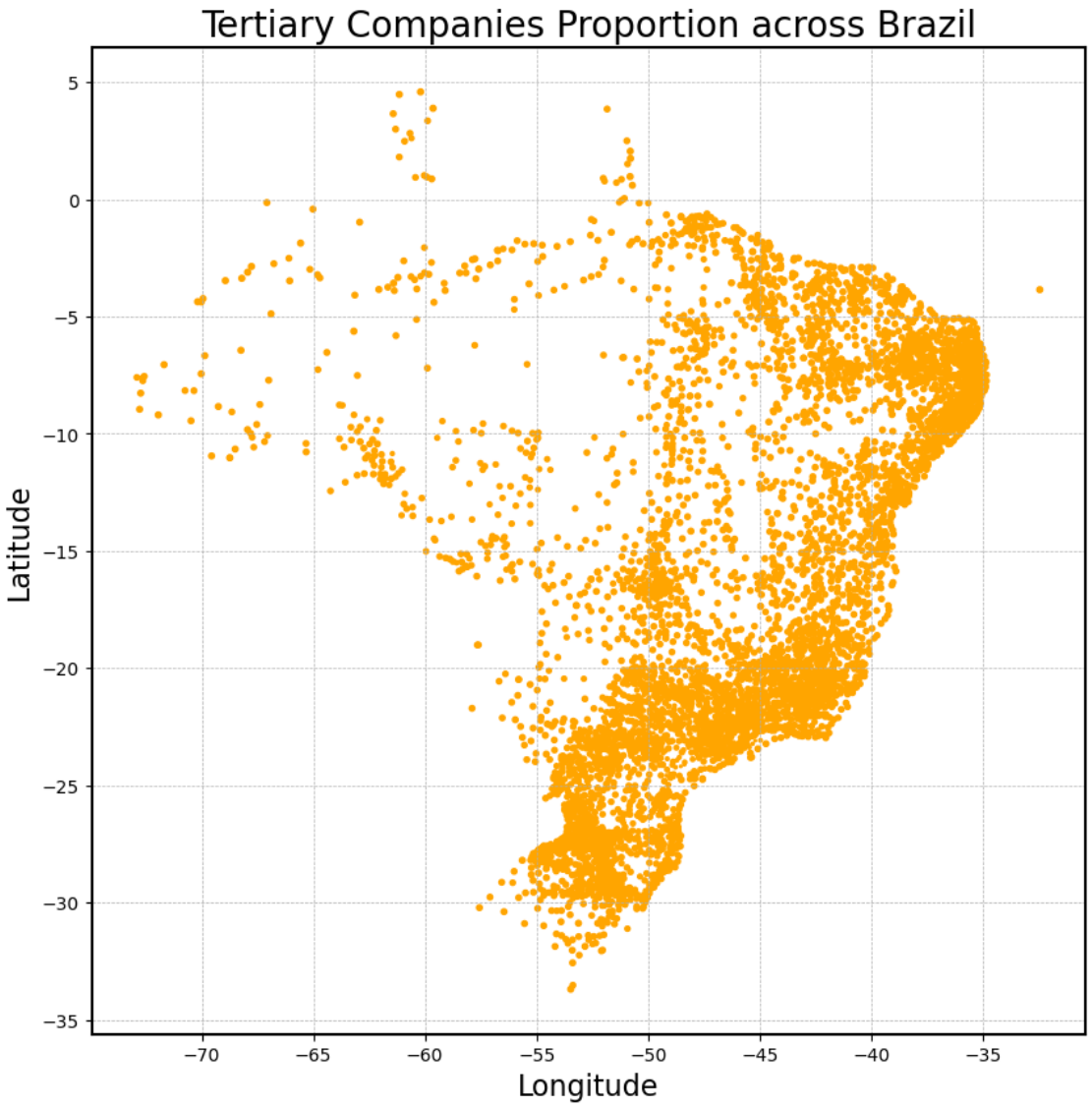


Tertiary Proportion

[130]:
```
feature_summary('Tertiary Companies Proportion',industry,df,size = 0.1)
```

```
<IPython.core.display.HTML object>
```

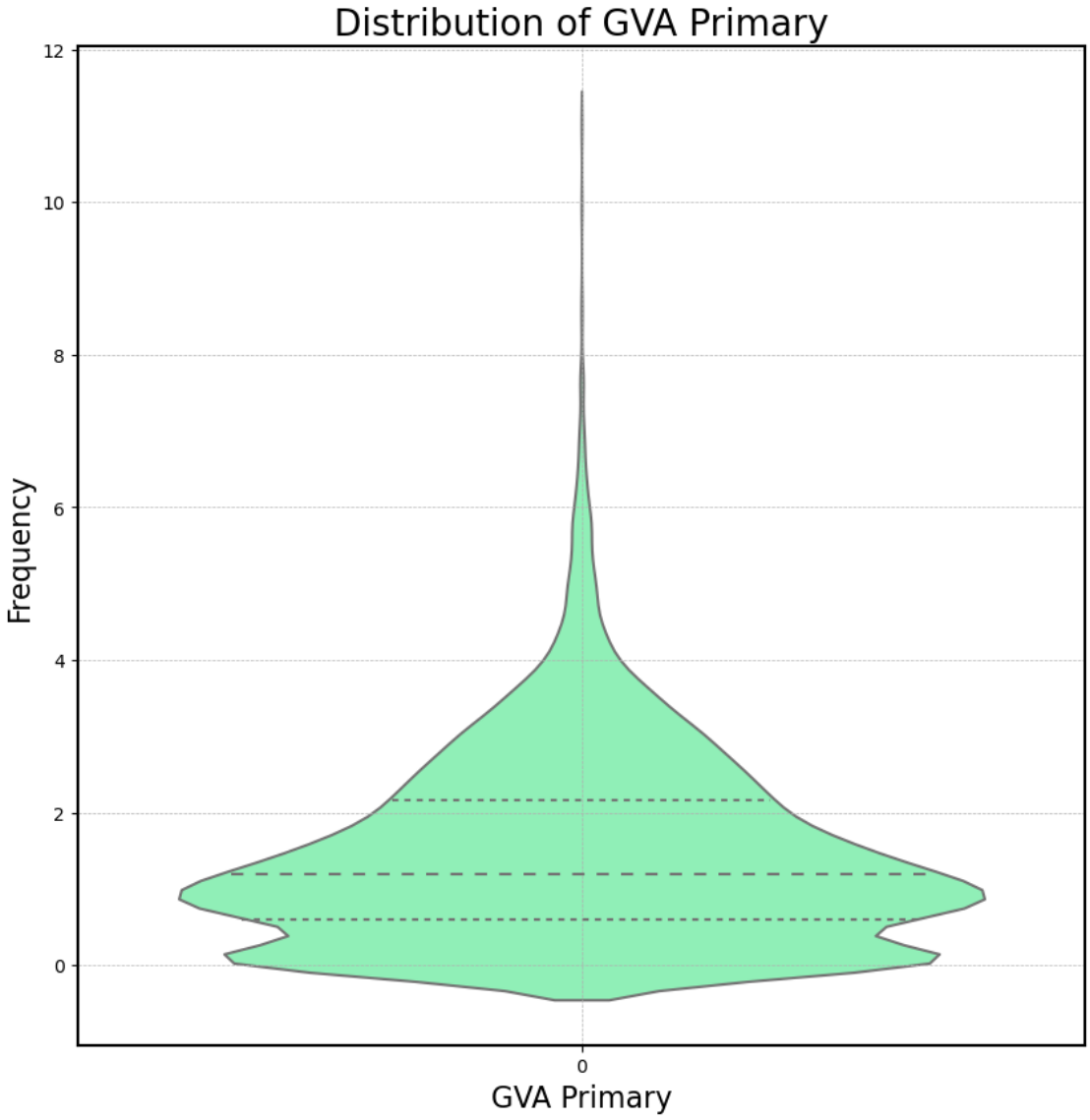

Distribution of Tertiary Companies Proportion
Frequency
1.0
0.8
0.6
0.4
0.2
0.0
0
Tertiary Companies Proportion

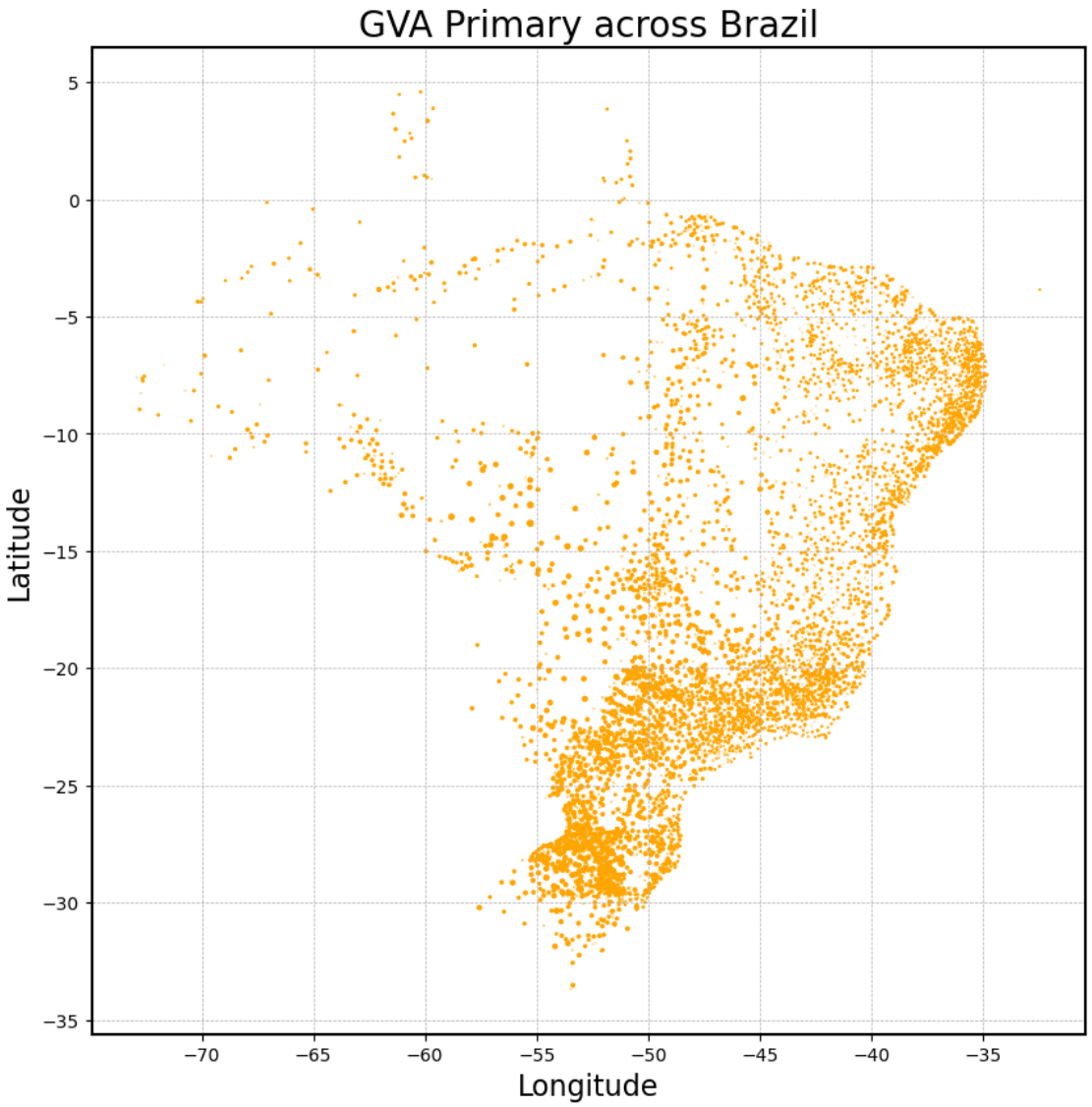


GVA Primary

[54]:
```
feature_summary('GVA Primary',industry,df,size = 1)
```

```
<IPython.core.display.HTML object>
```

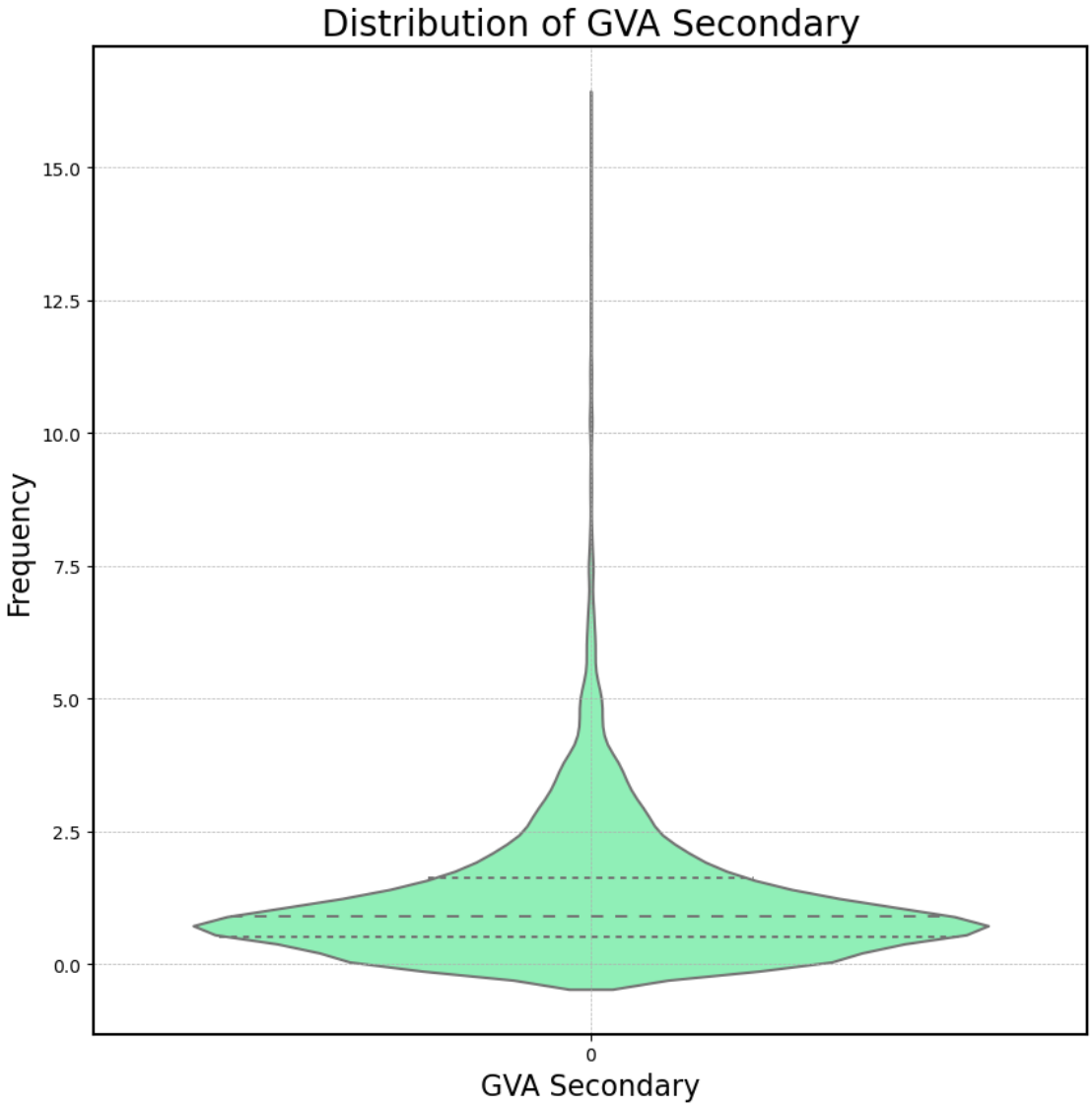

Distribution of GVA Primary
Frequency
12
10
8
6
4
2
0
0
GVA Primary

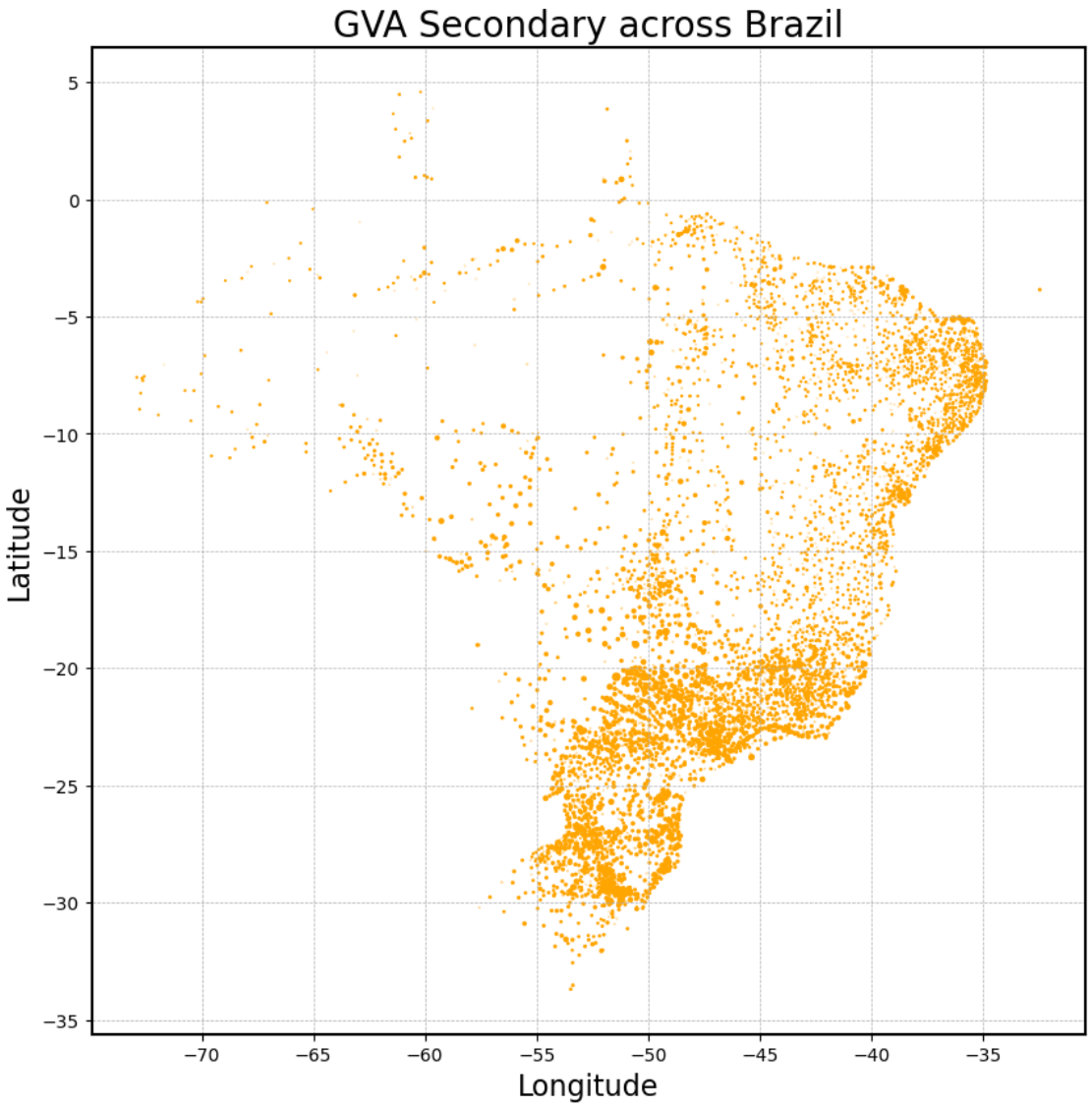


GVA Secondary

[55]:
```
feature_summary('GVA Secondary',industry,df,size = 1)
```

```
<IPython.core.display.HTML object>
```

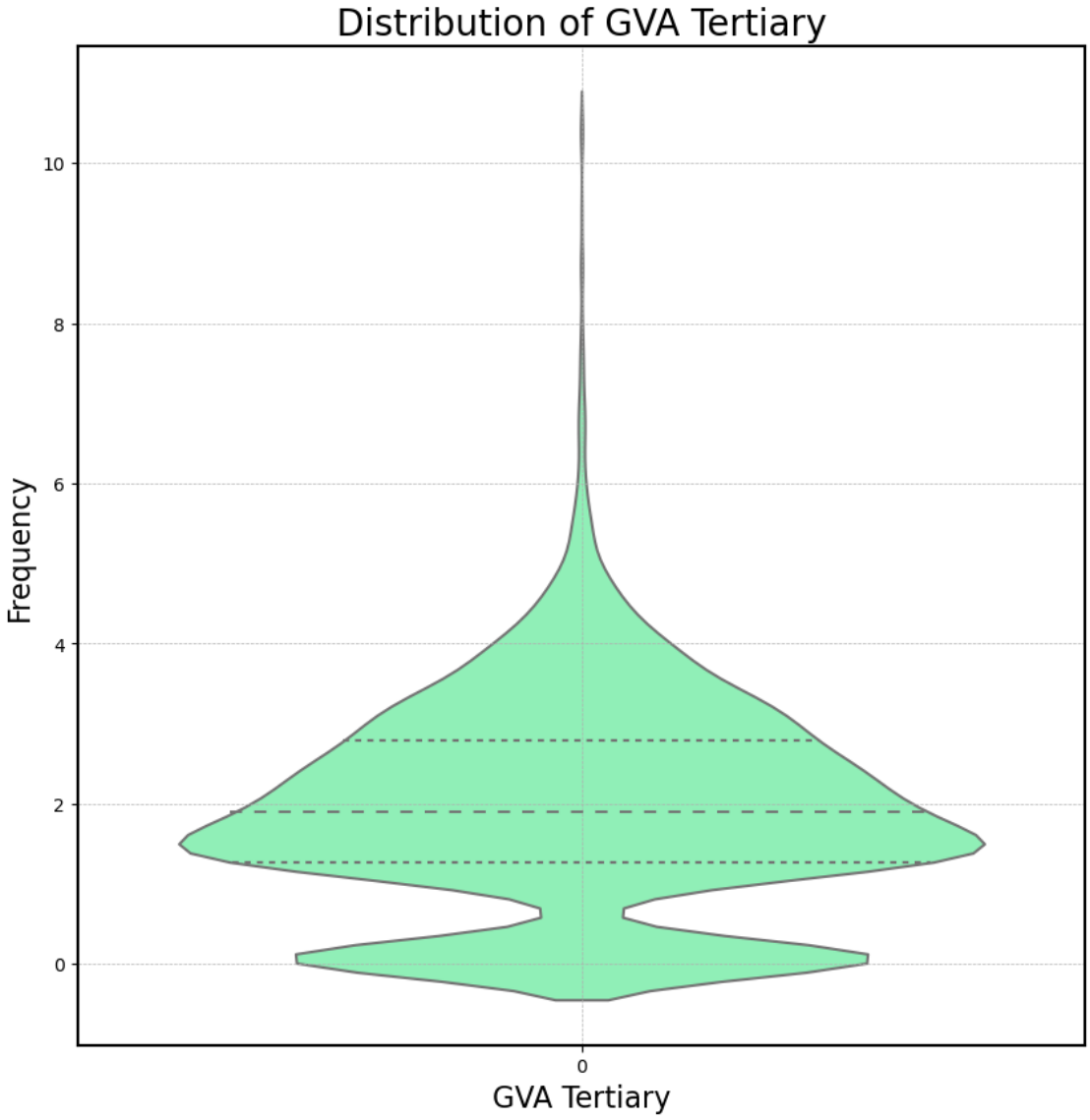

Distribution of GVA Secondary
Frequency
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
GVA Secondary

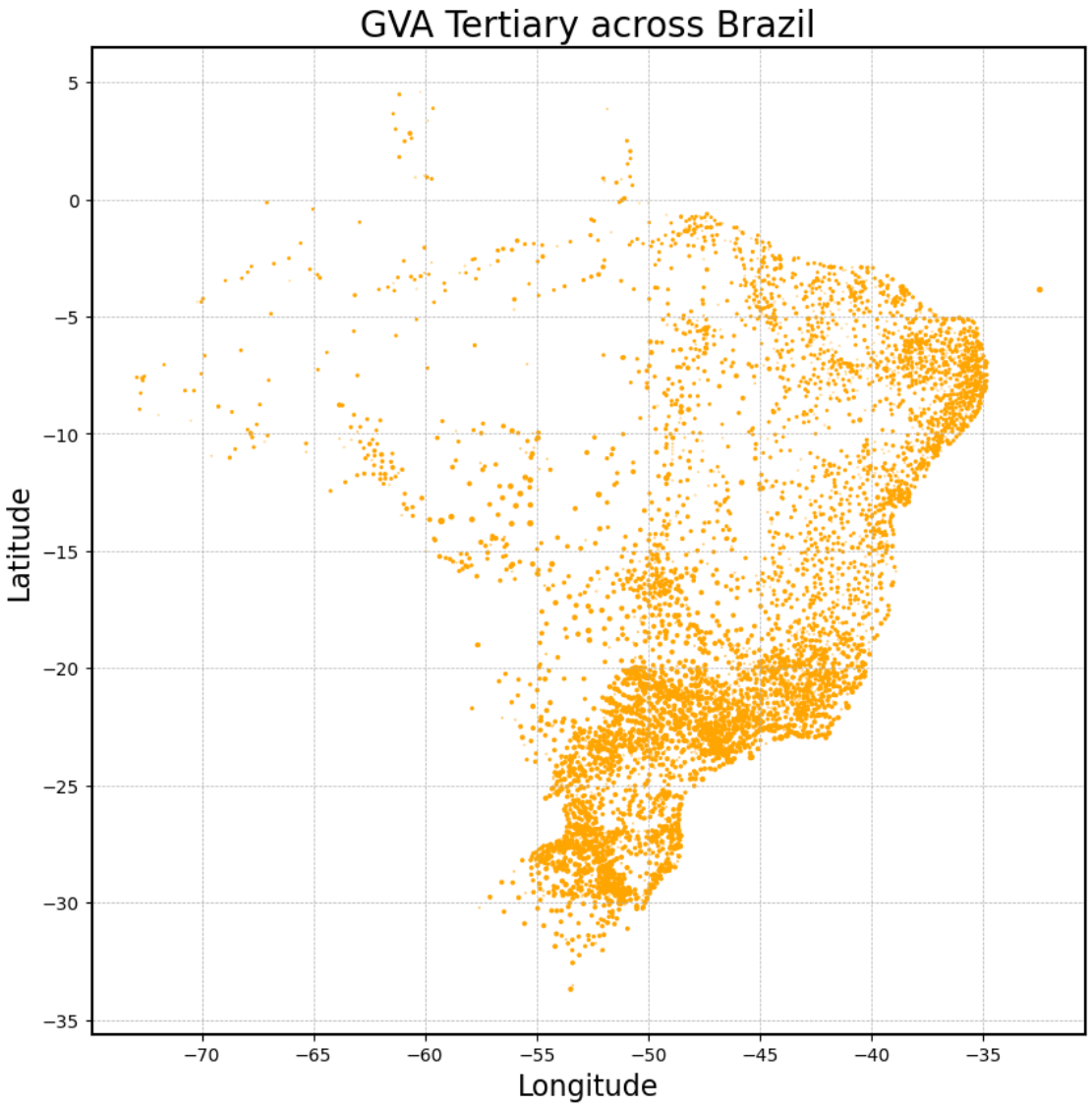


GVA Tertiary

[56]:
```
feature_summary('GVA Tertiary',industry,df,size = 1)
```

```
<IPython.core.display.HTML object>
```


Distribution of GVA Tertiary
Frequency
10
8
6
4
2
0
0
GVA Tertiary



[57]:
```
feature_variables = {'geographic':geographic,
             'population':population,
              'wealth':wealth,
             'industry':industry}
```

3 Data Preparation

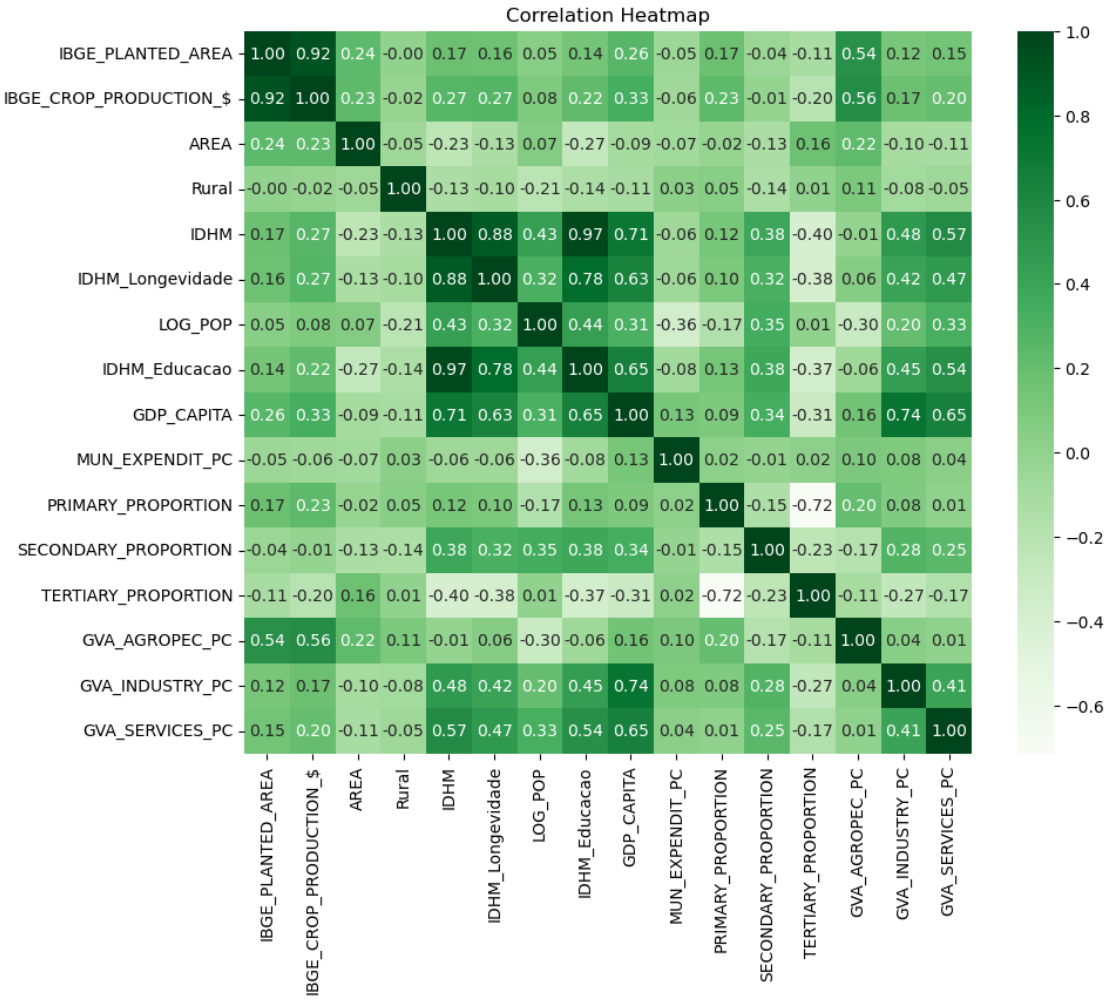
[58]:
```
all_features = {inner_k: inner_v for outer_k, inner_dict in feature_variables.
  ↪items() for inner_k, inner_v in inner_dict.items()}
```

[59]:
```
del all_features['Population Size']
```

[60]:
```
full_df = pd.merge(df_y,df,left_on = 'Name',right_on = 'CITY',how = 'inner')
full_df.drop(['POP_GDP'],inplace = True,axis = 1)
```

[65]:
```
X_raw = full_df[all_features.values()]
```

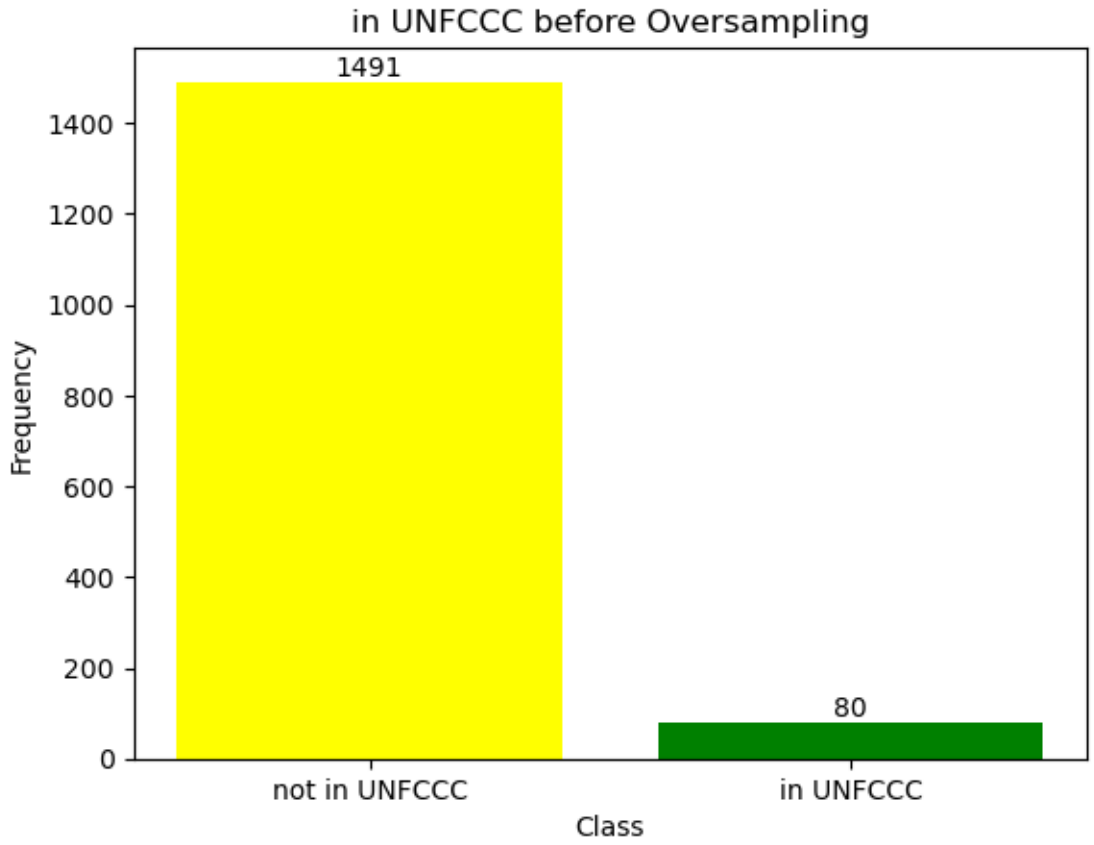
[132]:
```
corr_mtx = X_raw.corr() #correlation matrix between all variables in the dataset

plt.figure(figsize=(10, 8)) #setting a plt figure

sns.heatmap(corr_mtx, cmap='Greens', annot=True, fmt=".2f") #applying a heatmap␣
↪to the correlation matrix and rounding to 2dp
plt.title('Correlation Heatmap')
plt.show()
```



[66]:
```
y_feature = 'In UNFCC'
y_raw = full_df[targets_variables[y_feature]]
```

Before Oversampling

[67]:
```
counts = y_raw.value_counts()

plt.bar(['not in UNFCCC', 'in UNFCCC'], counts, color=['yellow', 'green'])

for i, count in enumerate(counts):
    plt.text(i, count, str(count), ha='center', va='bottom')


plt.title('in UNFCCC before Oversampling')
plt.xlabel('Class')
plt.ylabel('Frequency')

plt.show()
```



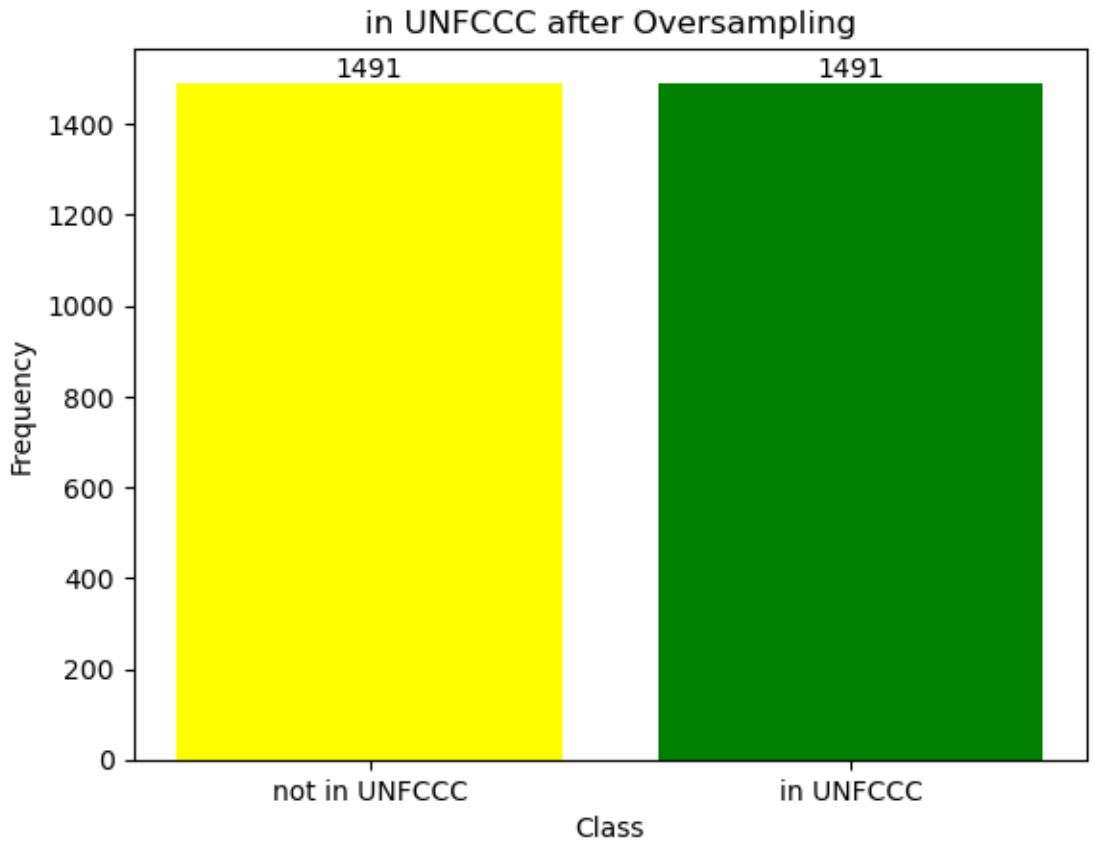
Oversampling and Standardising

[68]:
```
ros = RandomOverSampler(random_state=42)
```

```
# Resample the training data
X_train_resampled, y_train_resampled = ros.fit_resample(X_raw, y_raw)



X = X_train_resampled
y = y_train_resampled


non_binary_columns = [col for col in X_raw.columns if X_raw[col].nunique() > 2]


X[non_binary_columns] = (X[non_binary_columns]-X[non_binary_columns].mean())/
 ↪X[non_binary_columns].std()
```

After Oversampling

[69]:
```
counts = y.value_counts()

plt.bar(['not in UNFCCC', 'in UNFCCC'], counts, color=['yellow', 'green'])

for i, count in enumerate(counts):
    plt.text(i, count, str(count), ha='center', va='bottom')


plt.title('in UNFCCC after Oversampling')
plt.xlabel('Class')
plt.ylabel('Frequency')

plt.show()
```



Missing Values

[70]:
```
na_count = X.isnull().sum().sort_values(ascending = False).reset_index()␣
  ↪#sorting from high to low and using reset_index() to turn series into DF
na_count.columns = ['Feature',"Count of NA's"]
na_count.set_index('Feature',inplace = True)
print(na_count)
```

```
                      Count of NA's
Feature
IBGE_PLANTED_AREA                 0
IBGE_CROP_PRODUCTION_$            0
AREA                              0
Rural                             0
IDHM                              0
IDHM_Longevidade                  0
LOG_POP                           0
IDHM_Educacao                     0
GDP_CAPITA                        0
MUN_EXPENDIT_PC                   0
PRIMARY_PROPORTION                0
```

```
SECONDARY_PROPORTION                0
TERTIARY_PROPORTION                 0
GVA_AGROPEC_PC                      0
GVA_INDUSTRY_PC                     0
GVA_SERVICES_PC                     0
```

Maybe take this bit out

4 Modelling i)

[71]:
```
X.insert(0,'intercept',1)
```

[72]:
```
model = sm.OLS(y, X).fit()

print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  inUNFCC   R-squared:                       0.343
Model:                              OLS   Adj. R-squared:                  0.339
Method:                   Least Squares   F-statistic:                     96.56
Date:                  Wed, 24 Apr 2024   Prob (F-statistic):          1.60e-255
Time:                          15:21:09   Log-Likelihood:                -1538.9
No. Observations:                  2982   AIC:                             3112.
Df Residuals:                      2965   BIC:                             3214.
Df Model:                            16
Covariance Type:            nonrobust
==============================================================================
==========
                             coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----------
intercept                    0.5127      0.008     64.491      0.000       0.497
0.528
IBGE_PLANTED_AREA           -0.0611      0.016     -3.878      0.000      -0.092
-0.030
IBGE_CROP_PRODUCTION_$       0.0385      0.016      2.439      0.015       0.008
0.069
AREA                        -0.0054      0.009     -0.599      0.550      -0.023
0.012
Rural                       -0.1144      0.025     -4.565      0.000      -0.163
-0.065
IDHM                         0.0078      0.060      0.129      0.897      -0.110
0.126
IDHM_Longevidade            -0.0436      0.021     -2.041      0.041      -0.085
-0.002
LOG_POP                      0.2011      0.013     15.370      0.000       0.175
```

```
0.227
IDHM_Educacao              0.0336      0.045      0.743      0.457      -0.055
0.122
GDP_CAPITA                 0.0909      0.017      5.345      0.000       0.058
0.124
MUN_EXPENDIT_PC            0.0150      0.008      1.765      0.078      -0.002
0.032
PRIMARY_PROPORTION         0.0112      0.012      0.968      0.333      -0.011
0.034
SECONDARY_PROPORTION      -0.0103      0.010     -1.059      0.290      -0.029
0.009
TERTIARY_PROPORTION        0.0411      0.012      3.351      0.001       0.017
0.065
GVA_AGROPEC_PC            -0.0414      0.010     -4.073      0.000      -0.061
-0.021
GVA_INDUSTRY_PC           -0.0485      0.011     -4.515      0.000      -0.070
-0.027
GVA_SERVICES_PC            0.0249      0.012      2.020      0.043       0.001
0.049
==============================================================================
Omnibus:                      246.777   Durbin-Watson:                   0.686
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              114.592
Skew:                           0.294   Prob(JB):                     1.31e-25
Kurtosis:                       2.241   Cond. No.                         22.9
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

4.1 Logistic Regression

[73]:
```
fit = sm.Logit(y,X).fit()
```

```
Optimization terminated successfully.
         Current function value: 0.482665
         Iterations 7
```

[74]:
```
print(fit.summary())
print(fit.aic)
print(fit.bic)
```

```
                           Logit Regression Results
==============================================================================
Dep. Variable:                  inUNFCC   No. Observations:                 2982
Model:                            Logit   Df Residuals:                     2965
Method:                             MLE   Df Model:                           16
Date:                  Wed, 24 Apr 2024   Pseudo R-squ.:                  0.3037
```

```
Time:                        15:21:10   Log-Likelihood:                -1439.3
converged:                       True   LL-Null:                       -2067.0
Covariance Type:            nonrobust   LLR p-value:                1.984e-257
================================================================================
==========
                              coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
----------
intercept                   0.2684      0.052      5.197      0.000       0.167
0.370
IBGE_PLANTED_AREA          -0.3476      0.097     -3.579      0.000      -0.538
-0.157
IBGE_CROP_PRODUCTION_$      0.0822      0.097      0.846      0.398      -0.108
0.273
AREA                       -0.0915      0.055     -1.652      0.099      -0.200
0.017
Rural                      -0.6605      0.162     -4.068      0.000      -0.979
-0.342
IDHM                        0.3114      0.368      0.845      0.398      -0.411
1.034
IDHM_Longevidade           -0.2958      0.129     -2.298      0.022      -0.548
-0.043
LOG_POP                     1.5054      0.104     14.437      0.000       1.301
1.710
IDHM_Educacao              -0.1887      0.276     -0.682      0.495      -0.731
0.353
GDP_CAPITA                  0.4969      0.111      4.460      0.000       0.279
0.715
MUN_EXPENDIT_PC             0.1560      0.053      2.943      0.003       0.052
0.260
PRIMARY_PROPORTION          0.1753      0.078      2.261      0.024       0.023
0.327
SECONDARY_PROPORTION       -0.1273      0.061     -2.088      0.037      -0.247
-0.008
TERTIARY_PROPORTION         0.3151      0.078      4.043      0.000       0.162
0.468
GVA_AGROPEC_PC             -0.1263      0.063     -2.006      0.045      -0.250
-0.003
GVA_INDUSTRY_PC            -0.3083      0.071     -4.355      0.000      -0.447
-0.170
GVA_SERVICES_PC             0.2387      0.074      3.209      0.001       0.093
0.384
================================================================================
==========
2912.6111033881725
3014.6170448086923
```

So the statistically significant variables are:

[75]:
```
coefficients = fit.params
p_values = fit.pvalues

alpha = 0.05

# Find statistically significant variables
significant_indices = p_values < alpha

# Extract significant variables and their coefficients
significant_variables = X.columns[significant_indices]
significant_coefficients = coefficients[significant_indices]

# Create DataFrame with significant variables and coefficients
significant_df = pd.DataFrame(significant_coefficients,␣
 ↪index=significant_variables, columns=['Coefficients'])

# Display DataFrame
print('DataFrame with Significant Variables and Coefficients:')
print(significant_df)
```

```
DataFrame with Significant Variables and Coefficients:
                      Coefficients
intercept                 0.268421
IBGE_PLANTED_AREA        -0.347611
Rural                    -0.660542
IDHM_Longevidade         -0.295827
LOG_POP                   1.505440
GDP_CAPITA                0.496878
MUN_EXPENDIT_PC           0.155984
PRIMARY_PROPORTION        0.175265
SECONDARY_PROPORTION     -0.127309
TERTIARY_PROPORTION       0.315126
GVA_AGROPEC_PC           -0.126274
GVA_INDUSTRY_PC          -0.308278
GVA_SERVICES_PC           0.238708
```

4.2 Tree Based Model

[76]:
```
rf = RandomForestRegressor(n_estimators = 10, random_state = 101)
X.drop(['intercept'],axis = 1,inplace = True)
rf.fit(X,y)
```

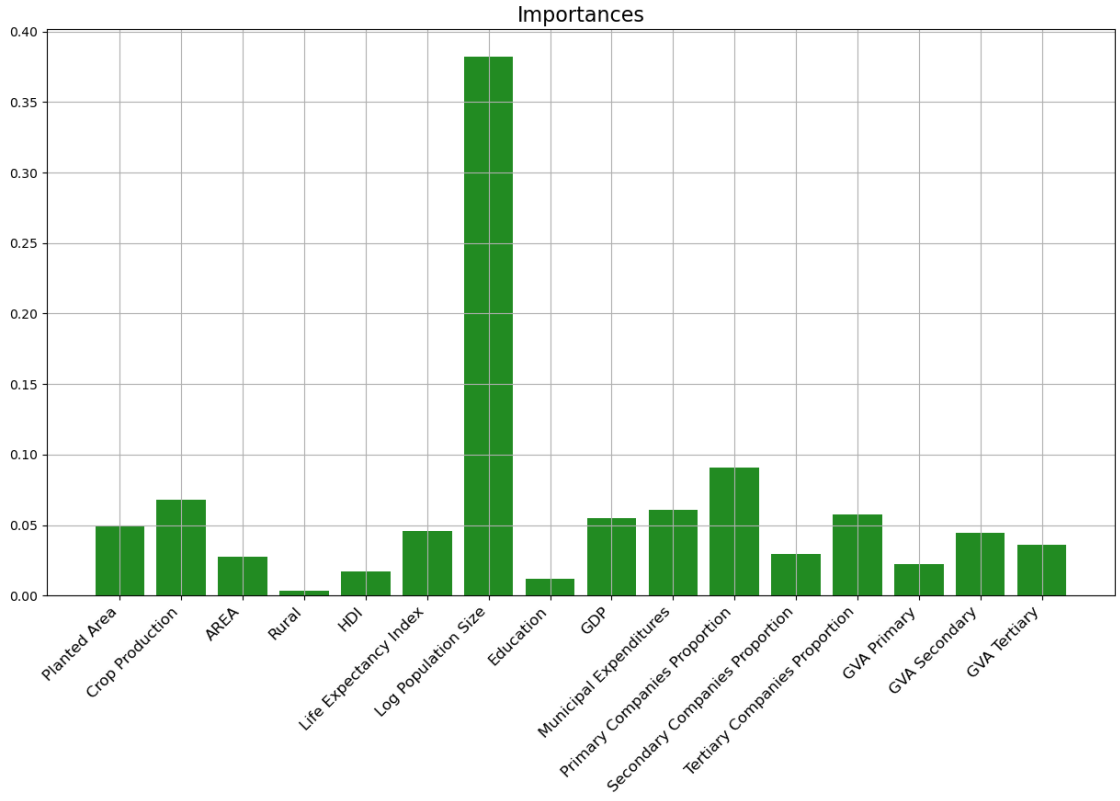
[76]:
```
RandomForestRegressor(n_estimators=10, random_state=101)
```

Forming a dictionary for naming the variables correctly

[77]:
```
var_map = {value:key for key,value in all_features.items()}
```

[78]:
```
feature_importances = pd.DataFrame(rf.feature_importances_, index=X.columns,␣
 ↪columns=['Importance'])
sorted_features = feature_importances.sort_values(by='Importance',␣
 ↪ascending=False)
print(sorted_features)

feature_importances = rf.feature_importances_

plt.figure(figsize=(15, 8))
plt.bar(range(len(feature_importances)), feature_importances, align='center')
plt.xticks(range(len(feature_importances)), [var_map[i] for i in X.columns],␣
 ↪rotation=45, ha='right', fontsize=12)  # Increase font size to 12
plt.title('Importances', fontsize=16)  # Increase title font size to 16
plt.grid()
plt.show()
```

```
                       Importance
LOG_POP                  0.382535
PRIMARY_PROPORTION       0.090686
IBGE_CROP_PRODUCTION_$   0.067968
MUN_EXPENDIT_PC          0.060845
TERTIARY_PROPORTION      0.057193
GDP_CAPITA               0.054635
IBGE_PLANTED_AREA        0.048953
IDHM_Longevidade         0.046019
GVA_INDUSTRY_PC          0.044481
GVA_SERVICES_PC          0.035938
SECONDARY_PROPORTION     0.029281
AREA                     0.027537
GVA_AGROPEC_PC           0.022159
IDHM                     0.016953
IDHM_Educacao            0.011757
Rural                    0.003059
```



4.3 Clustering

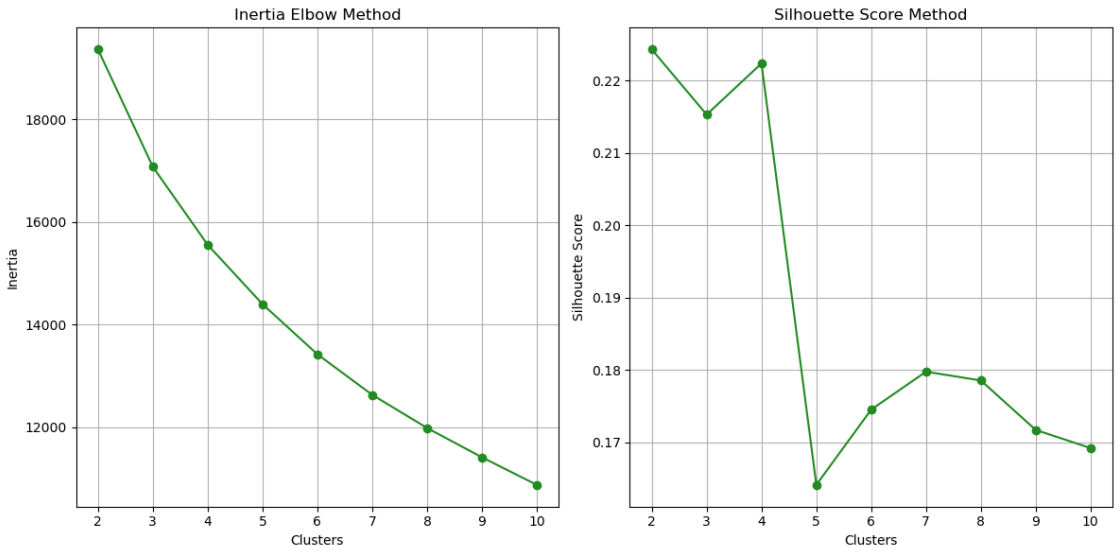
4.3.1 K-Means

Using the not balanced data for accuracy

[79]:
```
X_clustering = (X_raw - X_raw.mean())/X_raw.std()
clustering_df = pd.concat([X_clustering,y_raw],axis = 1)
```

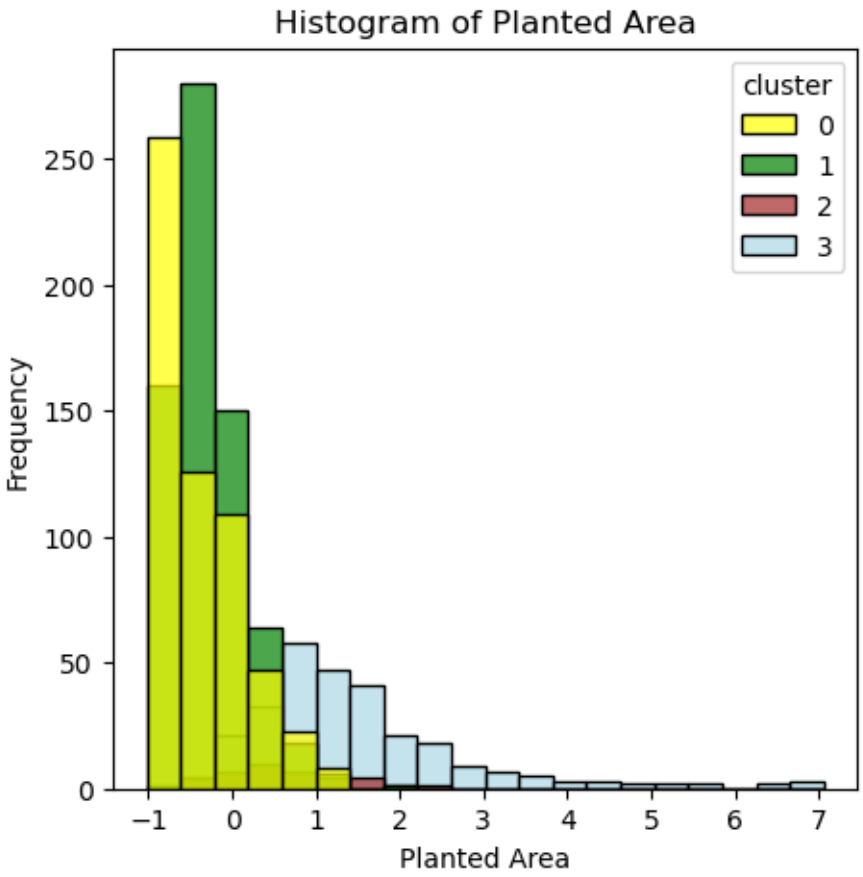
[81]:
```
#Number of clusters
clusters = range(2, 11)

#Lists to store the performance metrics
inertias = []
silhouette_scores = []

#Calculating the performance metrics for all number of clusters
for k in clusters:

    # fitting model
    km = KMeans(n_clusters=k, n_init = 10) #n_init is the number of times the␣
 ↪fitting was run
```

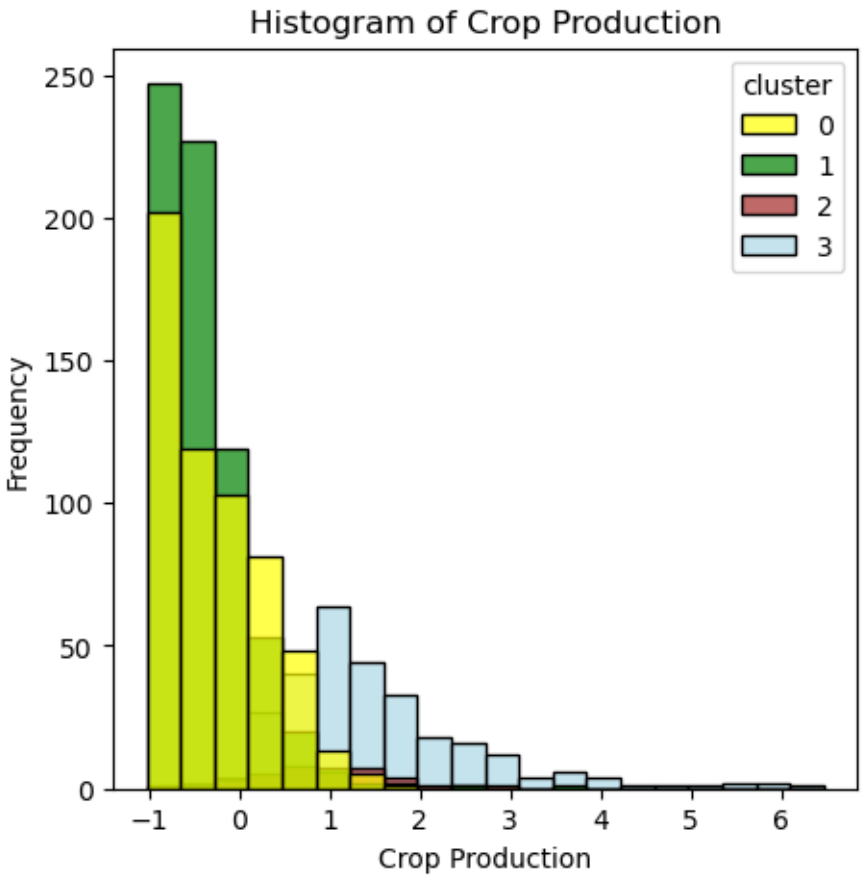
```
    km.fit(clustering_df)


    inertias.append(km.inertia_)

    if k > 1:
        silhouette_scores.append(silhouette_score(clustering_df, km.labels_))

#plot inertia (elbow method)
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(clusters, inertias, marker='o')
plt.title('Inertia Elbow Method')
plt.xlabel('Clusters')
plt.ylabel('Inertia')
plt.grid()

#plot the silhouette scores
plt.subplot(1, 2, 2)
plt.plot(clusters, silhouette_scores, marker='o')
plt.title('Silhouette Score Method')
plt.xlabel('Clusters')
plt.ylabel('Silhouette Score')
plt.grid()

plt.tight_layout()
plt.show()
```

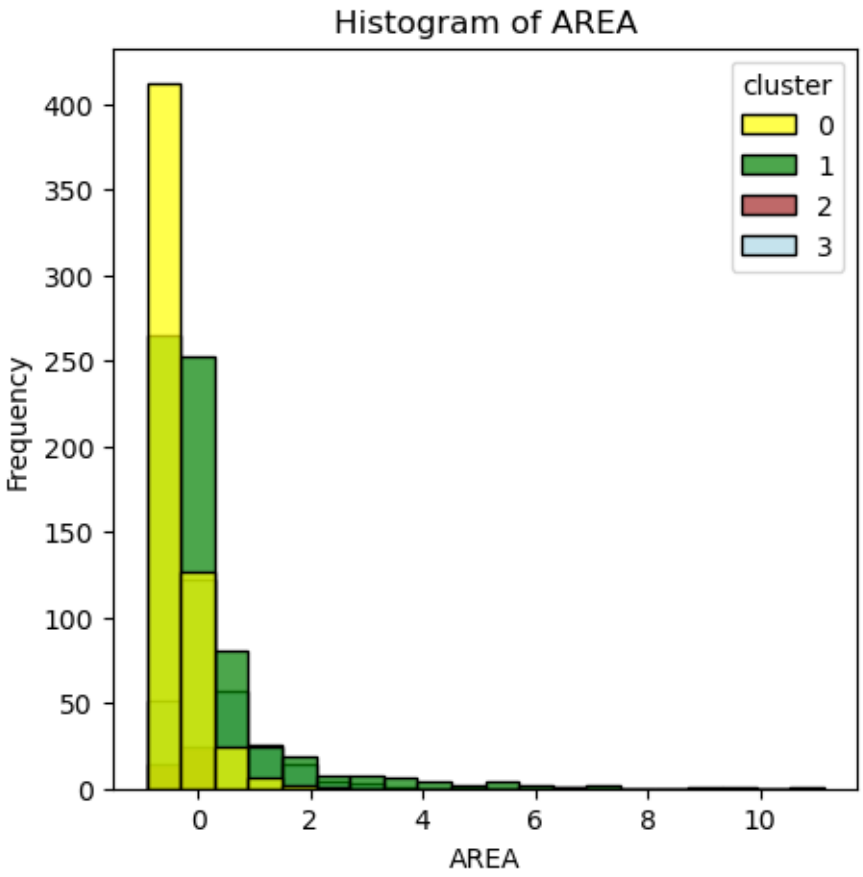


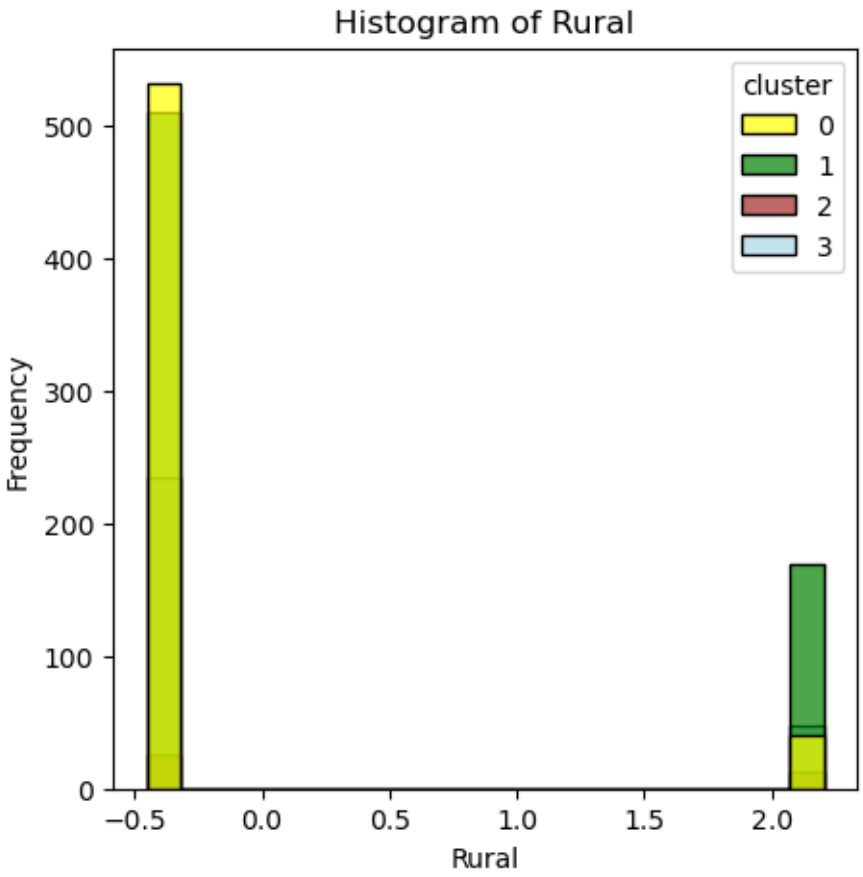
[82]:
```
var_map['inUNFCC'] = 'inUNFCC'
```

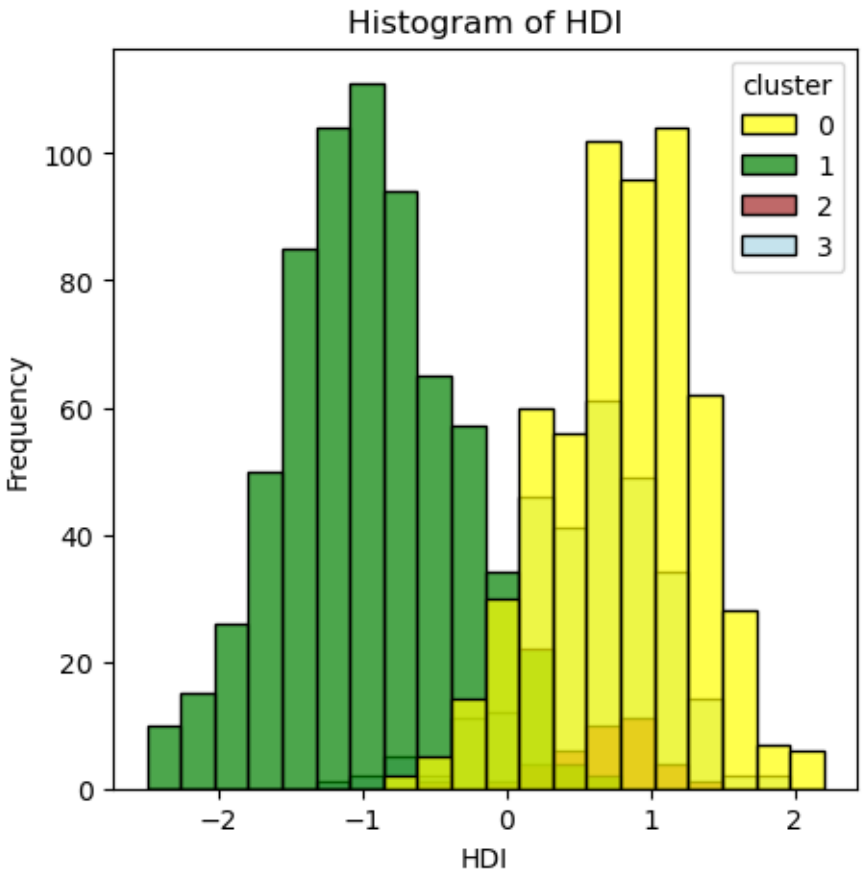
[83]:
```
# Define the optimal number of clusters
optimal_clusters = 4

# Fit model with the optimal number of clusters
km = KMeans(n_clusters=optimal_clusters, n_init=25)
labels = km.fit_predict(clustering_df)

# Create a DataFrame with data and cluster labels
with_clusters = clustering_df.copy()
with_clusters['cluster'] = labels

num_histograms = 3

variables = with_clusters.columns[:-1]  # Leaving out the cluster column

palette = ['yellow','green','brown','lightblue'] #'orange'


for variable in variables:
    plt.figure(figsize=(5, 5))
    sns.histplot(data=with_clusters, x=variable, bins=20, hue='cluster',␣
 ↪kde=False,
                 alpha=0.7, palette=palette)
    plt.title(f'Histogram of {var_map[variable]}')
    plt.xlabel(var_map[variable])
    plt.ylabel('Frequency')
    plt.show()
```

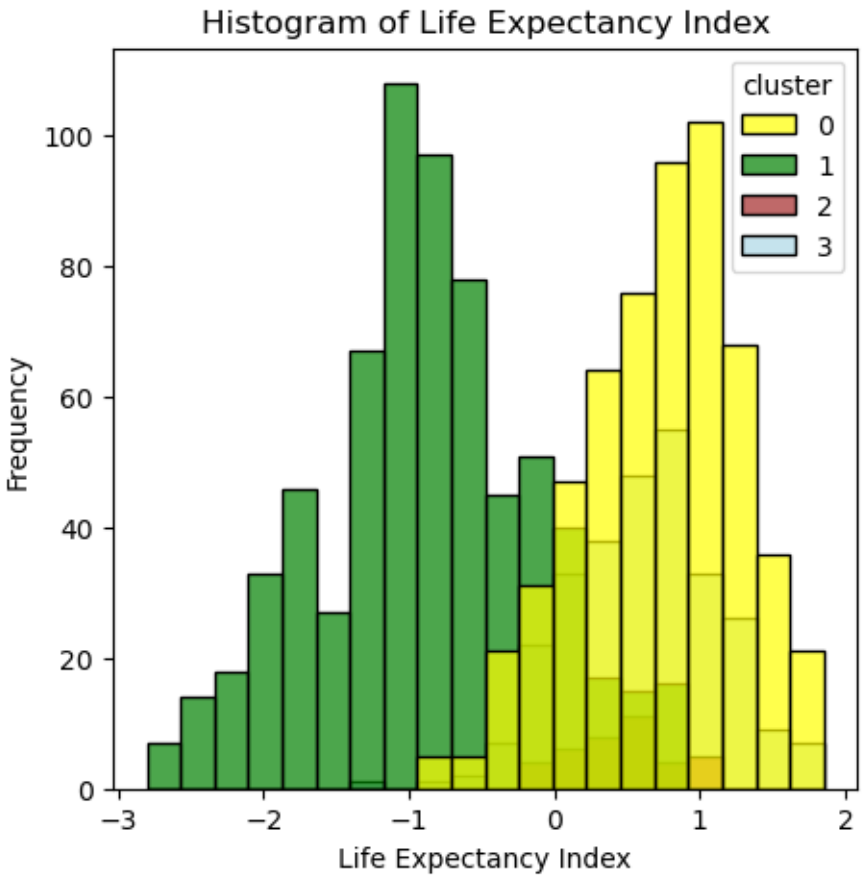

Histogram of Planted Area
cluster
0
1
2
3
Frequency
250
200
150
100
50
0
−1
0
1
2
3
4
5
6
7
Planted Area

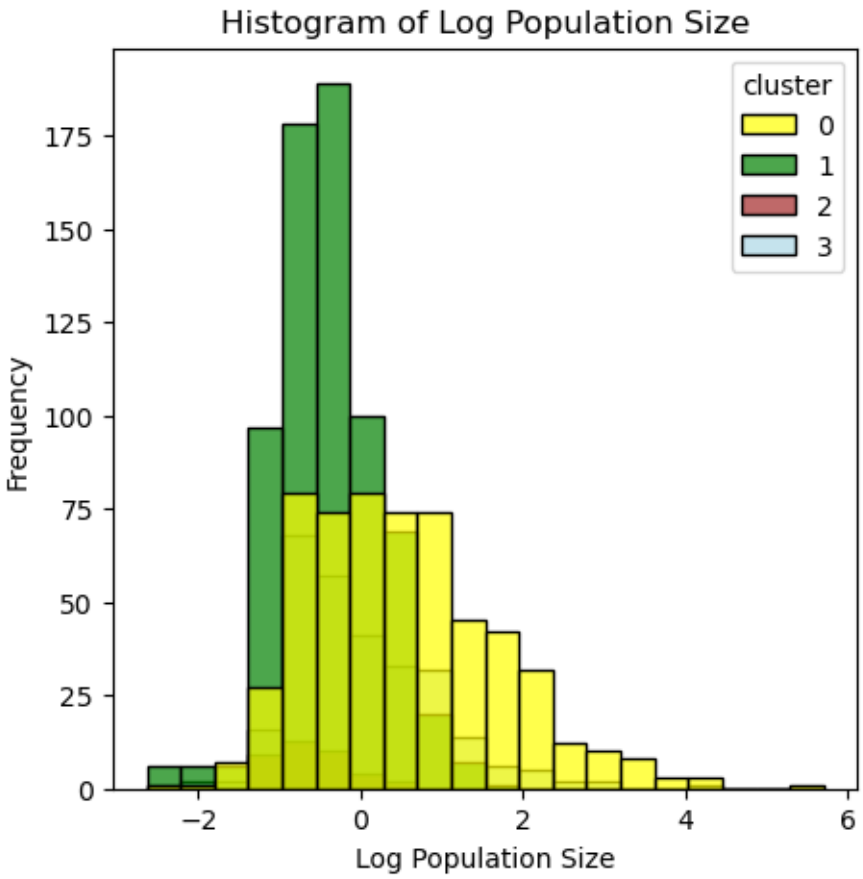

Histogram of Crop Production
cluster
0
1
2
3
Frequency
250
200
150
100
50
0
−1
0
1
2
3
4
5
6
Crop Production

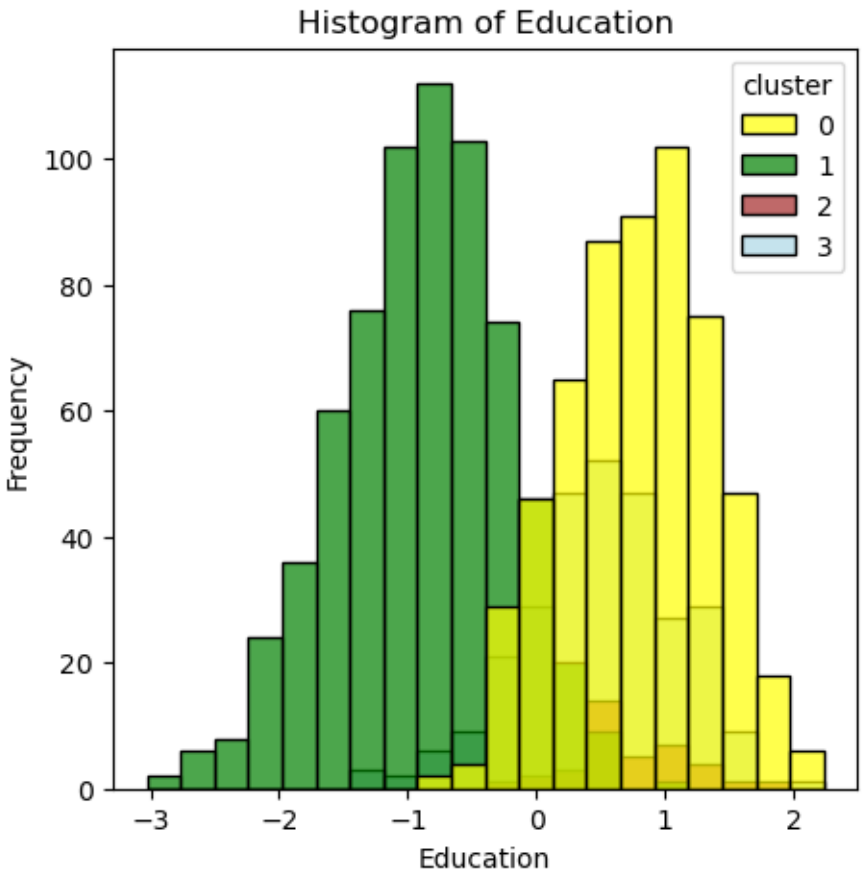

Histogram of AREA
cluster
0
1
2
3
Frequency
400
350
300
250
200
150
100
50
0
0
2
4
6
8
10
AREA

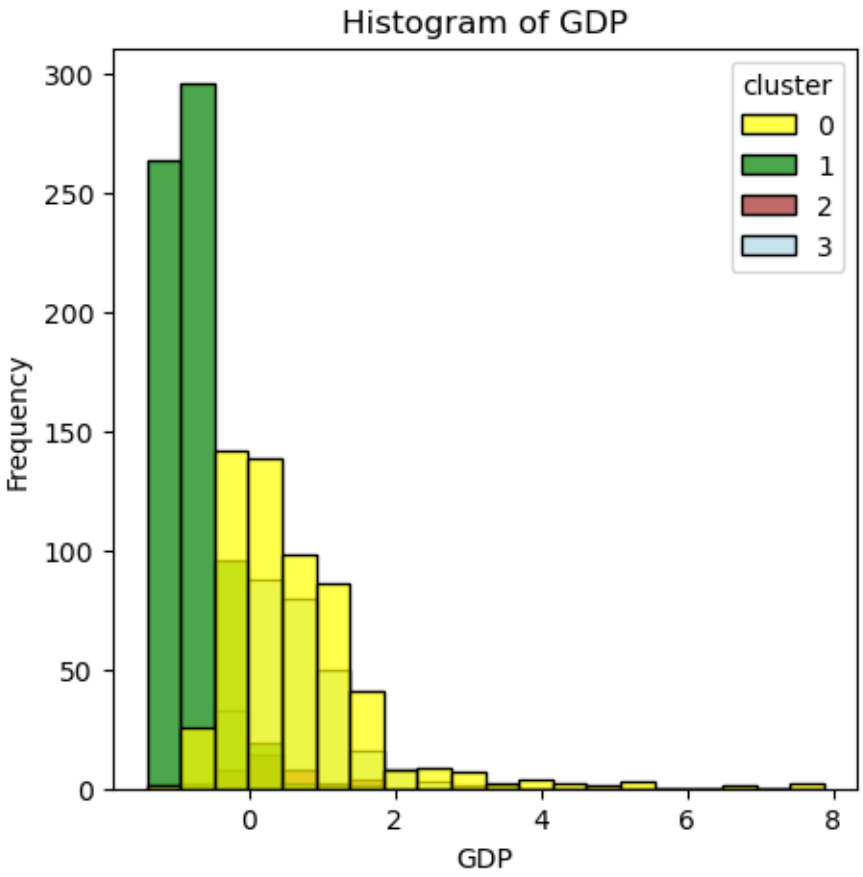

Histogram of Rural
cluster
0
1
2
3
Frequency
500
400
300
200
100
0
−0.5
0.0
0.5
1.0
1.5
2.0
Rural

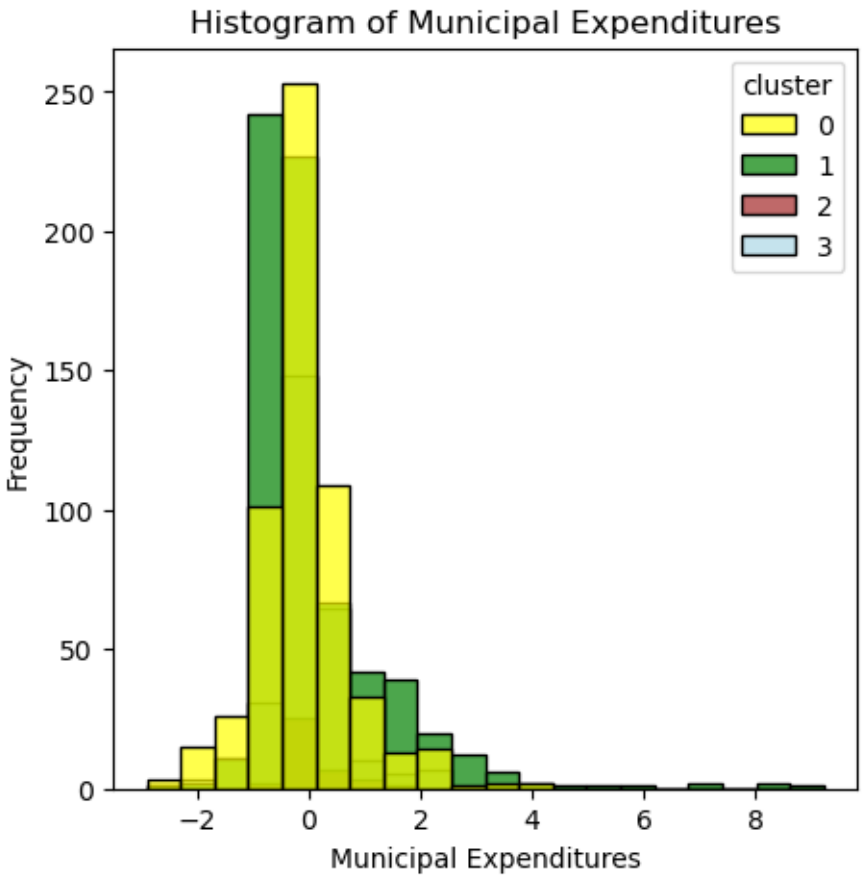

Histogram of HDI
cluster
0
1
2
3
Frequency
HDI
0
20
40
60
80
100
−2
−1
0
1
2

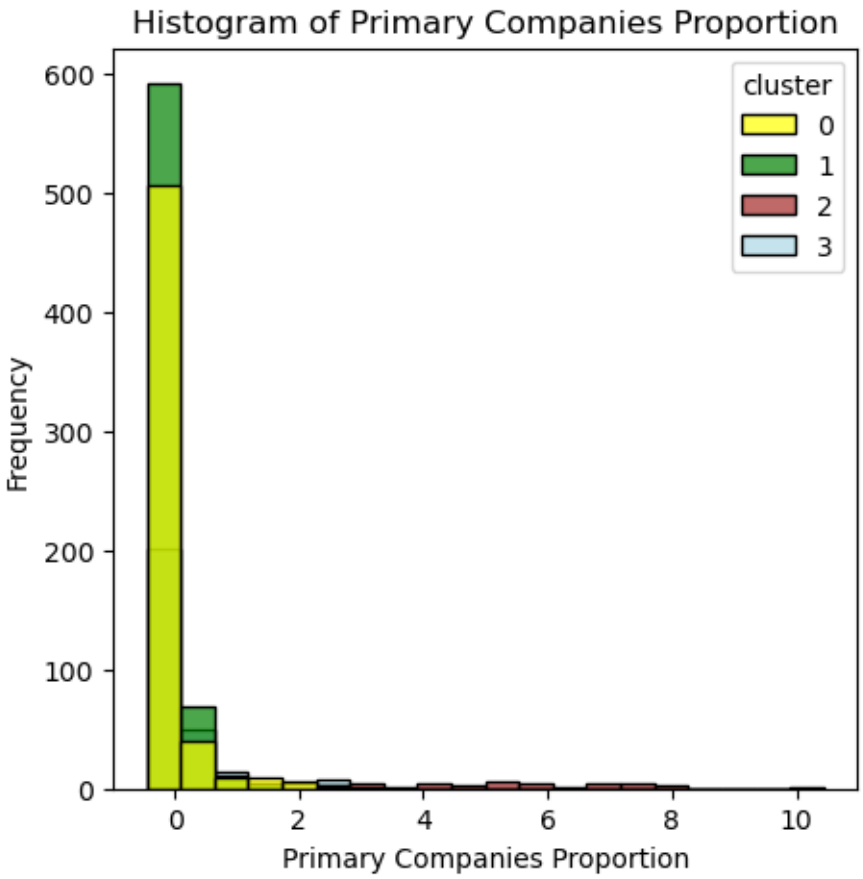

Histogram of Life Expectancy Index
cluster
0
1
2
3
Frequency
Life Expectancy Index
0
20
40
60
80
100
−3
−2
−1
0
1
2

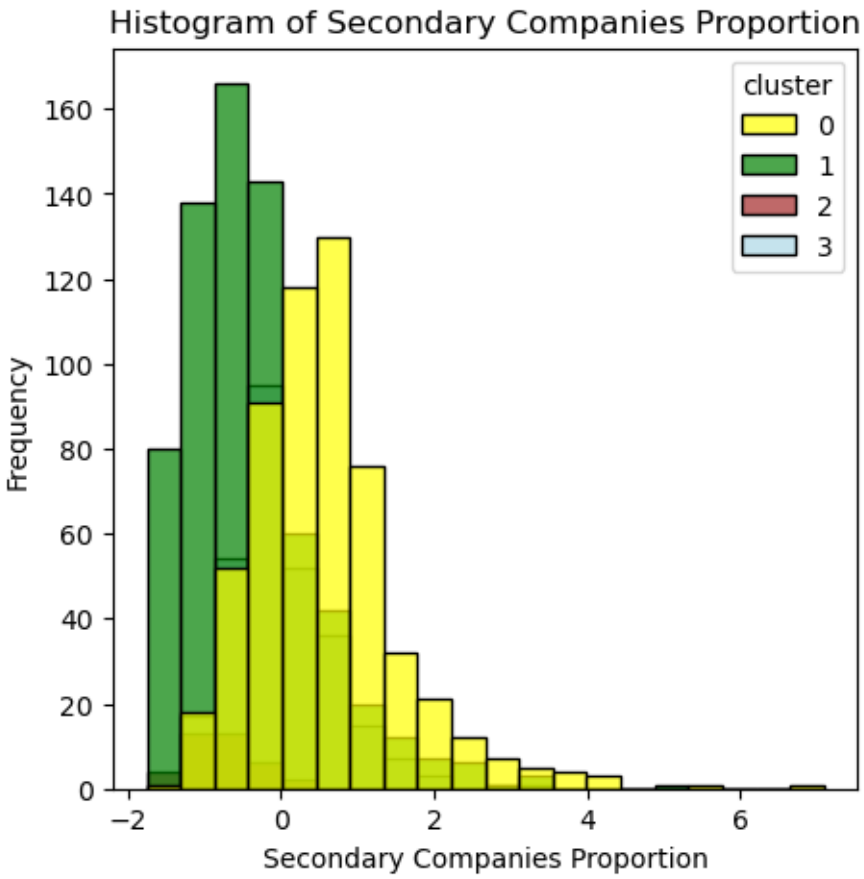

Histogram of Log Population Size
cluster
0
1
2
3
Frequency
175
150
125
100
75
50
25
0
−2
0
2
4
6
Log Population Size

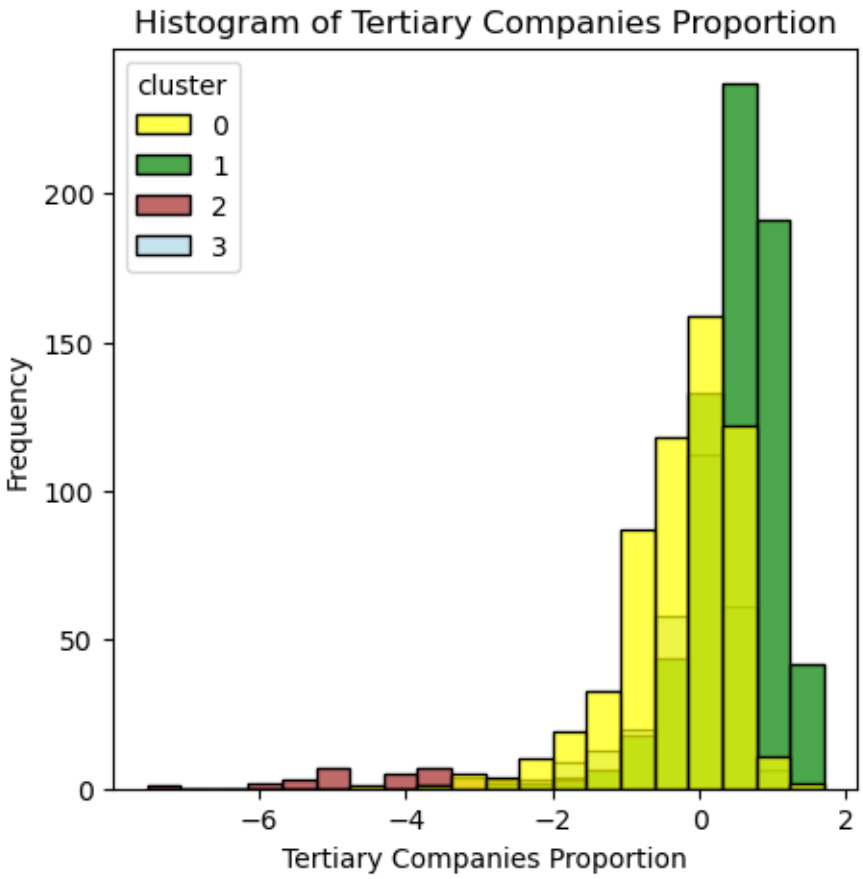

Histogram of Education
cluster
0
1
2
3
Frequency
100
80
60
40
20
0
−3
−2
−1
0
1
2
Education

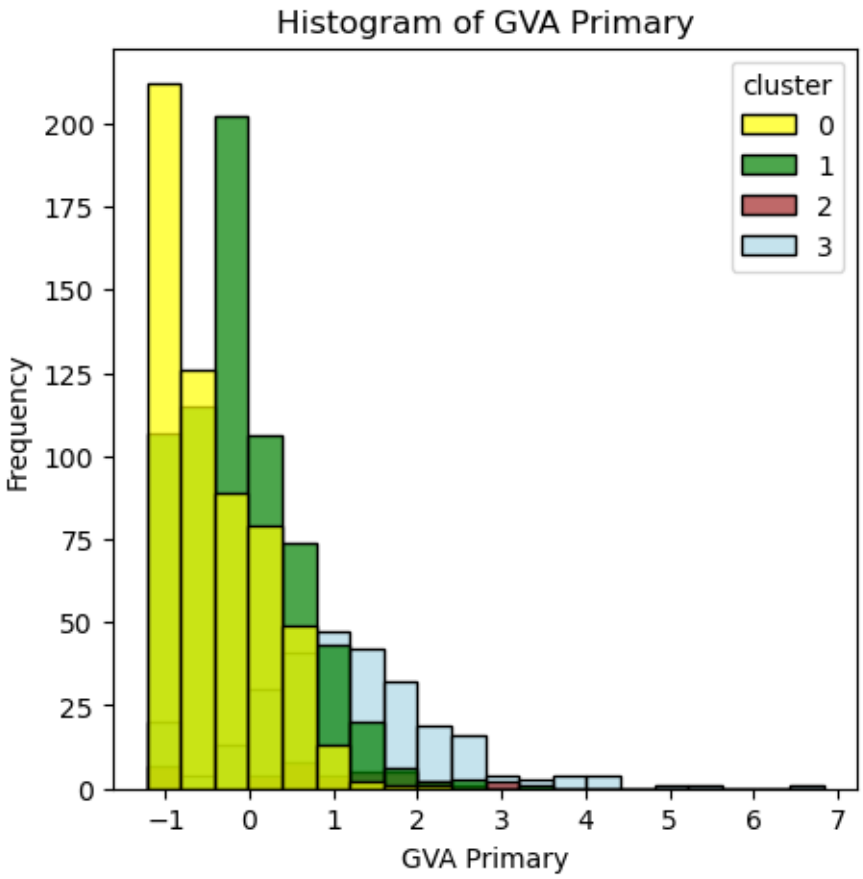

Histogram of GDP
cluster
0
1
2
3
Frequency
GDP
300
250
200
150
100
50
0
0
2
4
6
8

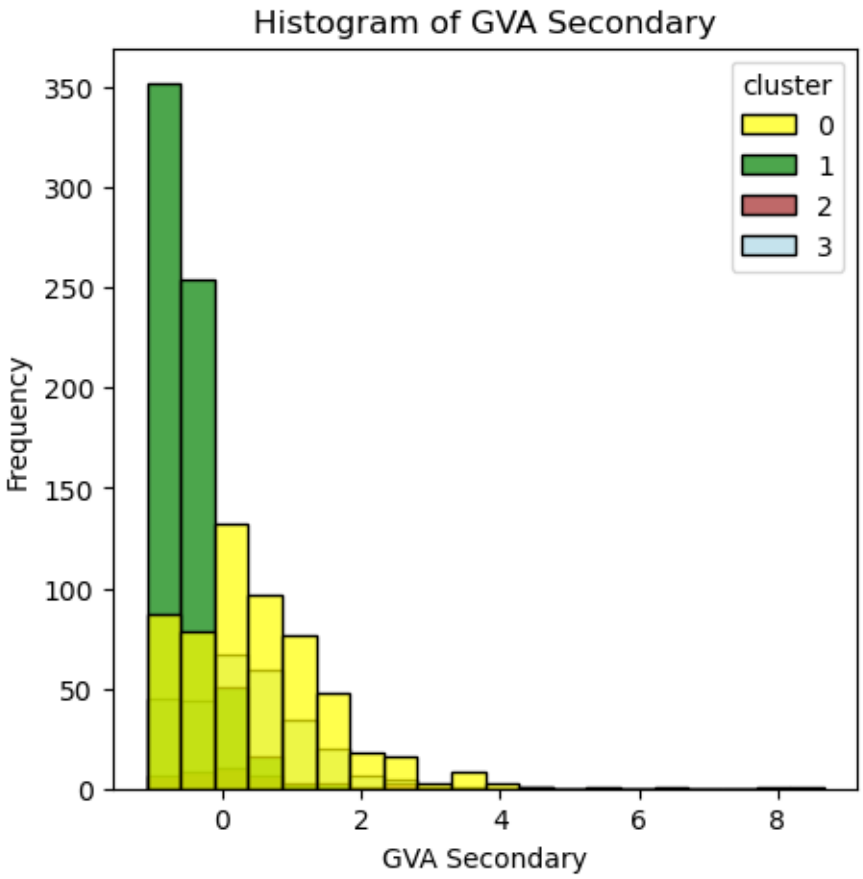

Histogram of Municipal Expenditures
cluster
0
1
2
3
Frequency
250
200
150
100
50
0
−2
0
2
4
6
8
Municipal Expenditures

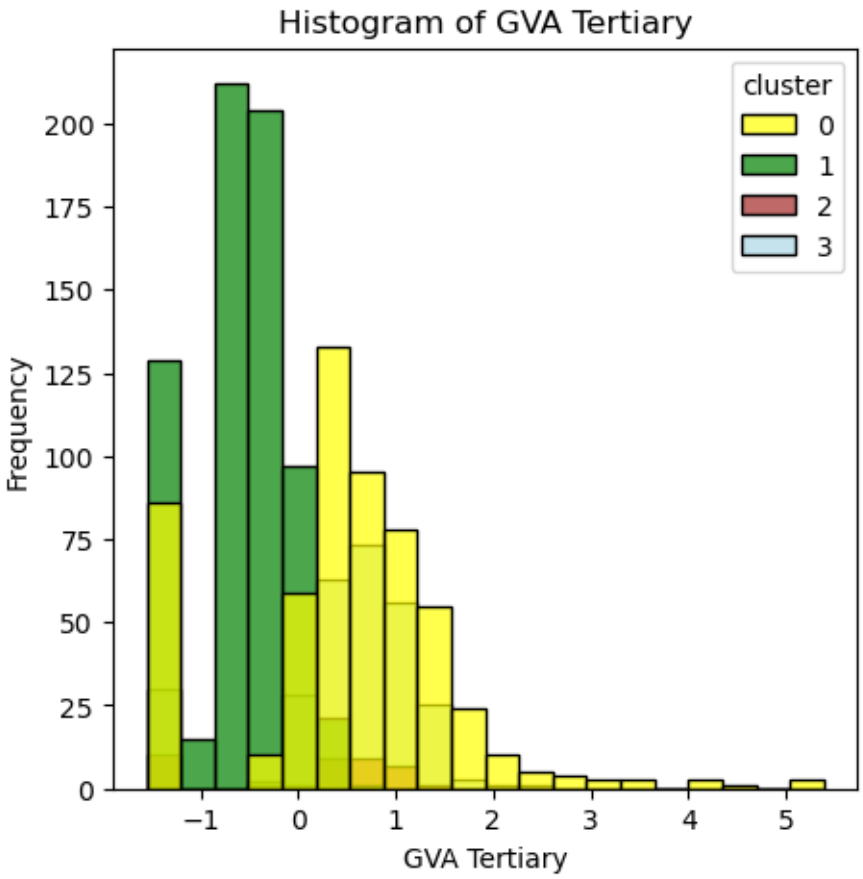

Histogram of Primary Companies Proportion
cluster
0
1
2
3
Frequency
600
500
400
300
200
100
0
0
2
4
6
8
10
Primary Companies Proportion

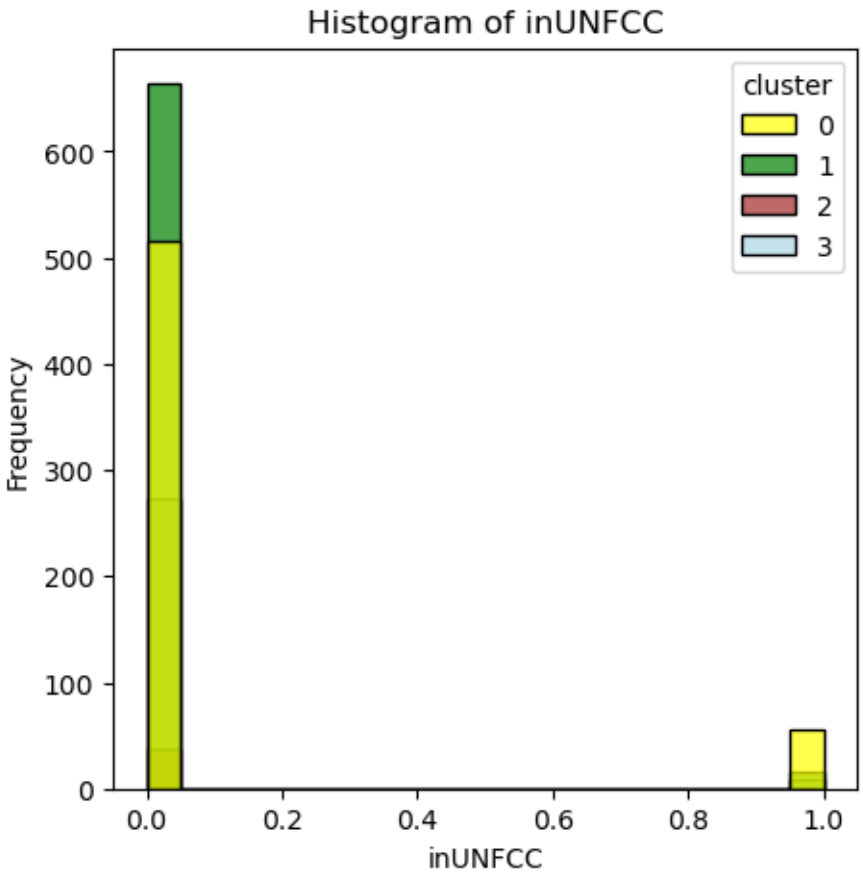

Histogram of Secondary Companies Proportion
cluster
0
1
2
3
Frequency
160
140
120
100
80
60
40
20
0
−2
0
2
4
6
Secondary Companies Proportion


Histogram of Tertiary Companies Proportion
cluster
0
1
2
3
Frequency
200
150
100
50
0
−6
−4
−2
0
2
Tertiary Companies Proportion


Histogram of GVA Primary
cluster
0
1
2
3
Frequency
200
175
150
125
100
75
50
25
0
−1
0
1
2
3
4
5
6
7
GVA Primary


Histogram of GVA Secondary
cluster
0
1
2
3
Frequency
350
300
250
200
150
100
50
0
0
2
4
6
8
GVA Secondary


Histogram of GVA Tertiary
cluster
0
1
2
3
Frequency
200
175
150
125
100
75
50
25
0
−1
0
1
2
3
4
5
GVA Tertiary



[84]:
```
with_clusters['cluster'].value_counts()
```

[84]:
```
1    679
0    572
3    282
2     38
Name: cluster, dtype: int64
```

[134]:
```
with_clusters.loc[with_clusters['inUNFCC'] == 1,'cluster'].
 ↪value_counts(normalize=True)
```

[134]:
```
0    0.7000
1    0.1875
3    0.1125
Name: cluster, dtype: float64
```

[86]:
```
for i in range(optimal_clusters):
    mean = with_clusters.loc[with_clusters['cluster'] == i,'inUNFCC'].mean()
```

```
    print(f'The percentage of cities in cluster {i} signed up to UNFCC is␣
 ↪{mean}')
```

```
The percentage of cities in cluster 0 signed up to UNFCC is 0.0979020979020979
The percentage of cities in cluster 1 signed up to UNFCC is 0.022091310751104567
The percentage of cities in cluster 2 signed up to UNFCC is 0.0
The percentage of cities in cluster 3 signed up to UNFCC is 0.031914893617021274
```

[ ]:
```
for i in range(optimal_clusters):
    mean = with_clusters.loc[with_clusters['cluster'] == i,'inUNFCC'].mean()
    print(f'The percentage of cities in cluster {i} signed up to UNFCC is␣
 ↪{mean}')
```

5 Modelling ii)

[91]:
```
inUNFCC_data = full_df[full_df['inUNFCC'] == 1]
features = inUNFCC_data.loc[:,all_features.values()]
```

Removing any targets that are zero or one

[92]:
```
all_targets = inUNFCC_data.iloc[:,3:22]
columns_to_drop = [col for col in all_targets.columns if all_targets[col].
 ↪nunique() == 1]
targets = all_targets.drop(columns =␣
 ↪columns_to_drop+['actorProperties_population'])
targets = targets.astype(float)
```

[94]:
```
for i in targets.columns:
    print(i)
```

```
hasCommitments
hasEmissionInventory
hasInitiativeParticipations
hasActionsUndertaken
hasImpact
hasMitigations
hasAdaptations
hasRiskAssessments
hasClimateActionPlans
actorProperties_commitmentsCount
actorProperties_initiativeParticipationsCount
actorProperties_actionsUndertakenCount
actorProperties_mitigationsCount
actorProperties_adaptationsCount
actorProperties_climateActionPlanCount
```

[95]:
```
X = features.insert(0,'intercept',1)
```

[98]:
```
features.drop(['Rural'],axis = 1, inplace = True)
```

Selecting just the classifiers

[102]:
```
binary_targets = [col for col in targets.columns if targets[col].nunique() <= 2]
```

Loop to fit and evaluate models

[103]:
```
for y in binary_targets:

    # Fit logistic regression model
    model = sm.Logit(targets[y], features).fit(disp=False)

    # Get coefficients and p-values
    coefficients = model.params
    p_values = model.pvalues

    # Filter significant coefficients (p < 0.1)
    significant_coefficients = coefficients[p_values < 0.1]

    # Create DataFrame with significant coefficients and p-values
    result_df = pd.DataFrame({'Coefficient': significant_coefficients,␣
 ↪'P-value': p_values[p_values < 0.1]})

    # Printmodel information
    print(y)
    print(result_df)

    print("Pseudo R-squared:", model.prsquared)
    print("AIC:", model.aic)
    print("BIC:", model.bic)
    print('\n\n\n')
```

```
hasCommitments
                Coefficient   P-value
LOG_POP            0.862861  0.068046
GVA_INDUSTRY_PC    0.873829  0.046393
Pseudo R-squared: 0.35876028723584774
AIC: 86.70173249592509
BIC: 124.81415865070718



hasEmissionInventory
           Coefficient   P-value
intercept   -52.220211  0.084638
```

```
LOG_POP       1.466759  0.042160
Pseudo R-squared: 0.4306906971876815
AIC: 64.03712760232597
BIC: 102.14955375710807
```

```
hasInitiativeParticipations
         Coefficient   P-value
LOG_POP     0.973587  0.031767
Pseudo R-squared: 0.20848539304404312
AIC: 95.37213241719535
BIC: 133.48455857197746
```

```
hasActionsUndertaken
               Coefficient   P-value
IDHM            102.785551  0.002930
IDHM_Educacao   -65.553406  0.002613
Pseudo R-squared: 0.23192133903227108
AIC: 112.47535635948094
BIC: 150.58778251426304
```

```
hasImpact
Empty DataFrame
Columns: [Coefficient, P-value]
Index: []
Pseudo R-squared: 0.2718431377297107
AIC: 63.03514798910694
BIC: 101.14757414388905
```

```
hasMitigations
                 Coefficient   P-value
GDP_CAPITA         -0.062832  0.026369
GVA_INDUSTRY_PC     0.715719  0.035459
GVA_SERVICES_PC     2.276106  0.026432
Pseudo R-squared: 0.3386483684202599
AIC: 100.51038258034953
BIC: 138.62280873513163
```

```
hasAdaptations
               Coefficient   P-value
IDHM             92.079915  0.006475
IDHM_Educacao   -61.684940  0.004102
Pseudo R-squared: 0.2357497708827455
AIC: 113.63618970691135
BIC: 151.74861586169345
```

```
hasRiskAssessments
                     Coefficient   P-value
intercept             -81.972858  0.034999
TERTIARY_PROPORTION    54.183615  0.033492
GVA_SERVICES_PC        -0.656703  0.048452
Pseudo R-squared: 0.4245155831486165
AIC: 70.92199886637523
BIC: 109.03442502115733
```

```
hasClimateActionPlans
                  Coefficient  P-value
GVA_INDUSTRY_PC     0.942783  0.07397
Pseudo R-squared: 0.30445648505489764
AIC: 76.5592078720931
BIC: 114.6716340268752
```

dataGenerator

April 28, 2024

1 generateData Class

For my analysis I used a .py file of this class but I have used a notebook to convert it to pdf

[ ]:
```
import pandas as pd
from unidecode import unidecode
from difflib import SequenceMatcher

class generateData:

    def __init__(self,country,manual_matching = False):
        '''Class to handle all data pipes including the levenshtein distance␣
↪algorithm,
        see report for more description on this'''


        self.all_cities_fp = f'brazil_cities.xlsx'
        self.UNFCC_cities_fp = f'Cities_UNFCCC.csv'
        self.matching = f'matched_cities.xlsx'

        self.country = country
        self.manual_matching = manual_matching


        self.UNFCCData = self.generateUNFCCData()
        self.AllCityData = self.generateAllCityData()
        self.datasets = self.loadDataSets()



    def generateUNFCCData(self) -> pd.DataFrame:
        full_data = pd.read_csv(self.UNFCC_cities_fp)
        data = full_data[full_data['country'] == self.country].reset_index(drop␣
↪= True)
        data['Date'] = pd.to_datetime(data['Date'])
        idx = data.groupby('id')['Date'].idxmax()
        data = data.loc[idx].reset_index(drop = True)
        data = data[data['actorProperties_population']>1000]
```

```
        data['ASCII'] = [unidecode(city) for city in data['organizationName']]
        data = data.drop_duplicates(subset = 'ASCII')
        print(len(data))

        return data

    def generateAllCityData(self) -> pd.DataFrame:
        all_cities = pd.read_excel(self.all_cities_fp)
        cities = all_cities[all_cities['Country name EN'] == self.country].
↪reset_index(drop = True)
        cities['Alternate Names'] = cities['Alternate Names'].fillna('none')
        cities['comb'] = [city_names.split(', ') for city_names in␣
↪(cities['ASCII Name'] + ', ' + cities['Alternate Names'])]
        cities = cities.explode('comb')
        cities['ASCII'] = [unidecode(city) for city in cities['comb']]
        cities = cities.drop_duplicates(subset = ['ASCII'])

        return cities

    def calcuateSimilarity(self,df1,df2,df1_col,df2_col,similarity_min):
        matched_cities = []
        for city1 in df1[df1_col].unique():
            for city2 in df2[df2_col].unique():
                similarity = SequenceMatcher(None, city1.lower(),city2.lower()).
↪ratio()
                if similarity >= similarity_min:
                    print(f'{city1} -> {city2} with similarity {similarity}')
                    matched_cities.append([city1,city2,similarity])
        match_data = pd.DataFrame(matched_cities,columns =␣
↪['city1','city2','similarity'])

        return match_data

    def matchDataSets(self,df1,df2,df1_col,df2_col,similarity_min = 0.85):
        match_data = self.
↪calcuateSimilarity(df1,df2,df1_col,df2_col,similarity_min)
        data = match_data.loc[match_data.groupby('city1')['similarity'].
↪idxmax(), ['city1', 'city2']]


        #remove any that arent matched
        filtered_data = pd.merge(df2,data,left_on = df2_col, right_on =␣
↪'city2',how = 'inner')

        full_data = pd.merge(filtered_data,df1,left_on = 'city1',right_on =␣
↪df1_col,how = 'right')
```

```
        return full_data

    def inUNFCCData(self) -> pd.DataFrame:

        data = self.matchDataSets(self.AllCityData,self.
↪UNFCCData,'ASCII','ASCII',1)

        data['inUNFCC'] = data['city1'].map(lambda x: 0 if pd.isnull(x) else 1)
        data = data.drop_duplicates(subset = ['ASCII Name','inUNFCC'])
        data = data.sort_values('inUNFCC').drop_duplicates(subset=['Name'],␣
↪keep='last').reset_index(drop = True)


        cols_to_drop = ['Feature Class','Feature Code','Country Code',\
                'Country name EN','Country Code 2','Admin1 Code',\
                'Admin2 Code','Admin3 Code','Admin4 Code','Timezone',\
                'LABEL EN','comb','ASCII_y']
        final_data = data.iloc[:,5:].drop(columns = cols_to_drop)

        return final_data


    def loadDataSets(self):
        brazil_cities = pd.read_csv('data/BRAZIL_CITIES.csv',sep = ';')\
        .sort_values('IBGE_RES_POP')\
        .drop_duplicates(subset=['CITY'], keep='last').reset_index(drop = True)

        return {'brazil_cities':brazil_cities}
```