# TABLE OF CONTENTS

[**TABLE OF CONTENTS**](#_23r46nqirljc) **1**

[**Tree Attribute Visuals**](#_9di09c7jaxaa) **2**

[**Recreated Native Scatter Plots**](#_jd9475rxma54) **5**

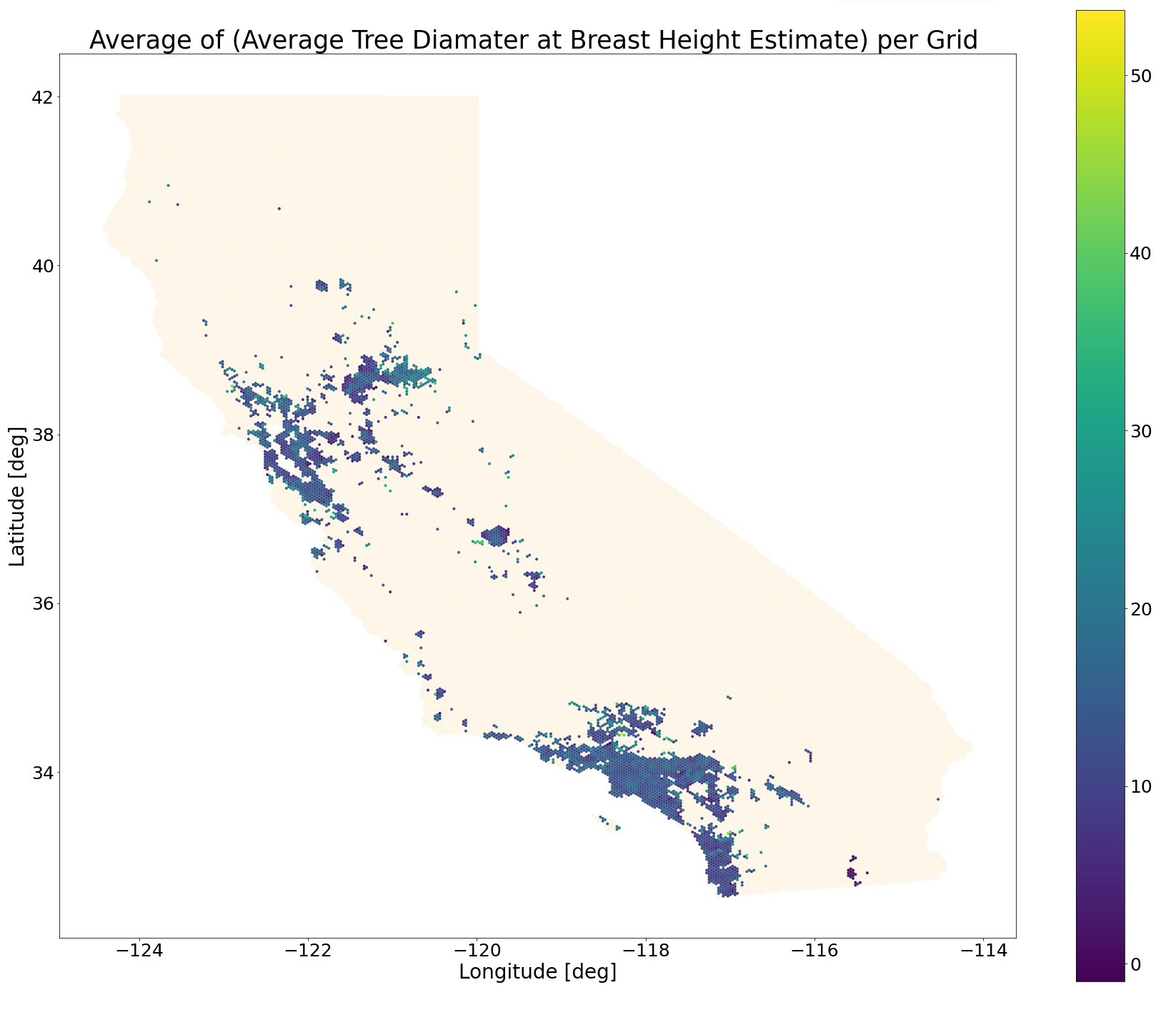
[**Spatial Modeling**](#_hdduzwi138cb) **7**

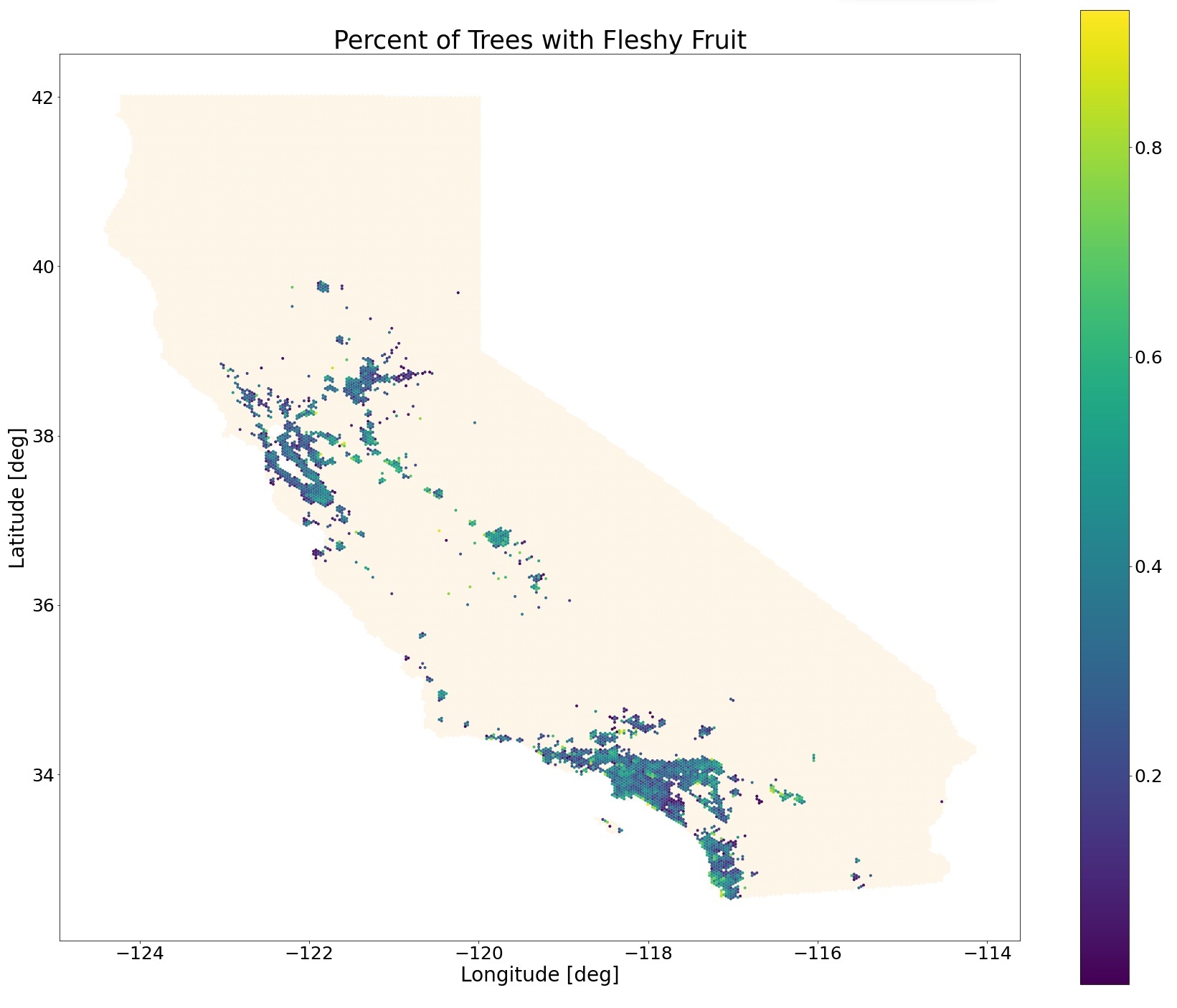
[Data Dependence](#_6t7i33qywvbe) 7

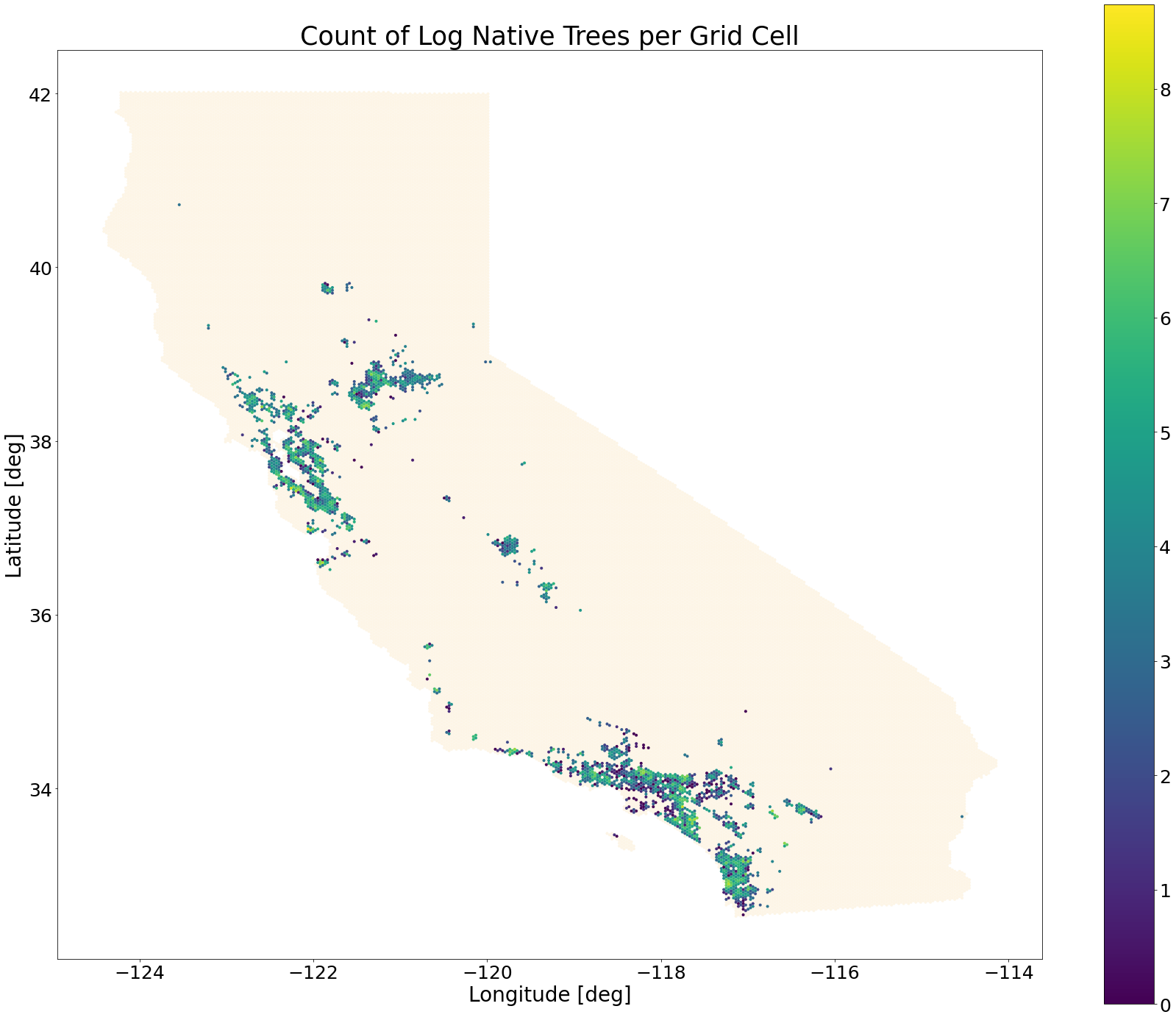
[Modeling Dataset](#_iw9297e0garq) 9

[Model Research](#_31c2o7p0h5xh) 10

# Tree Attribute Visuals

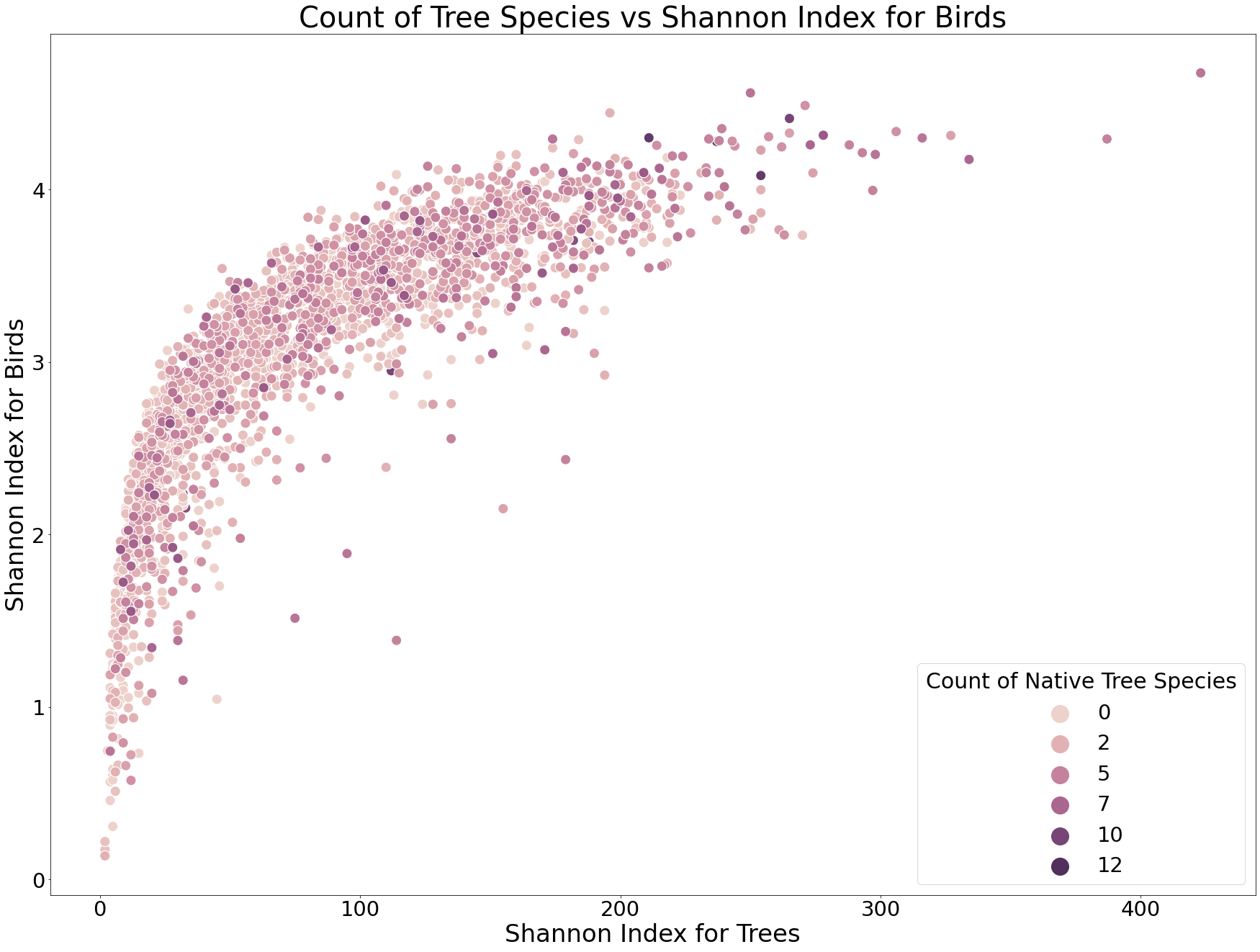






# Recreated Native Scatter Plots

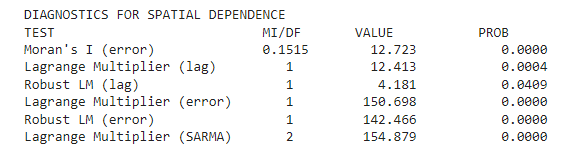
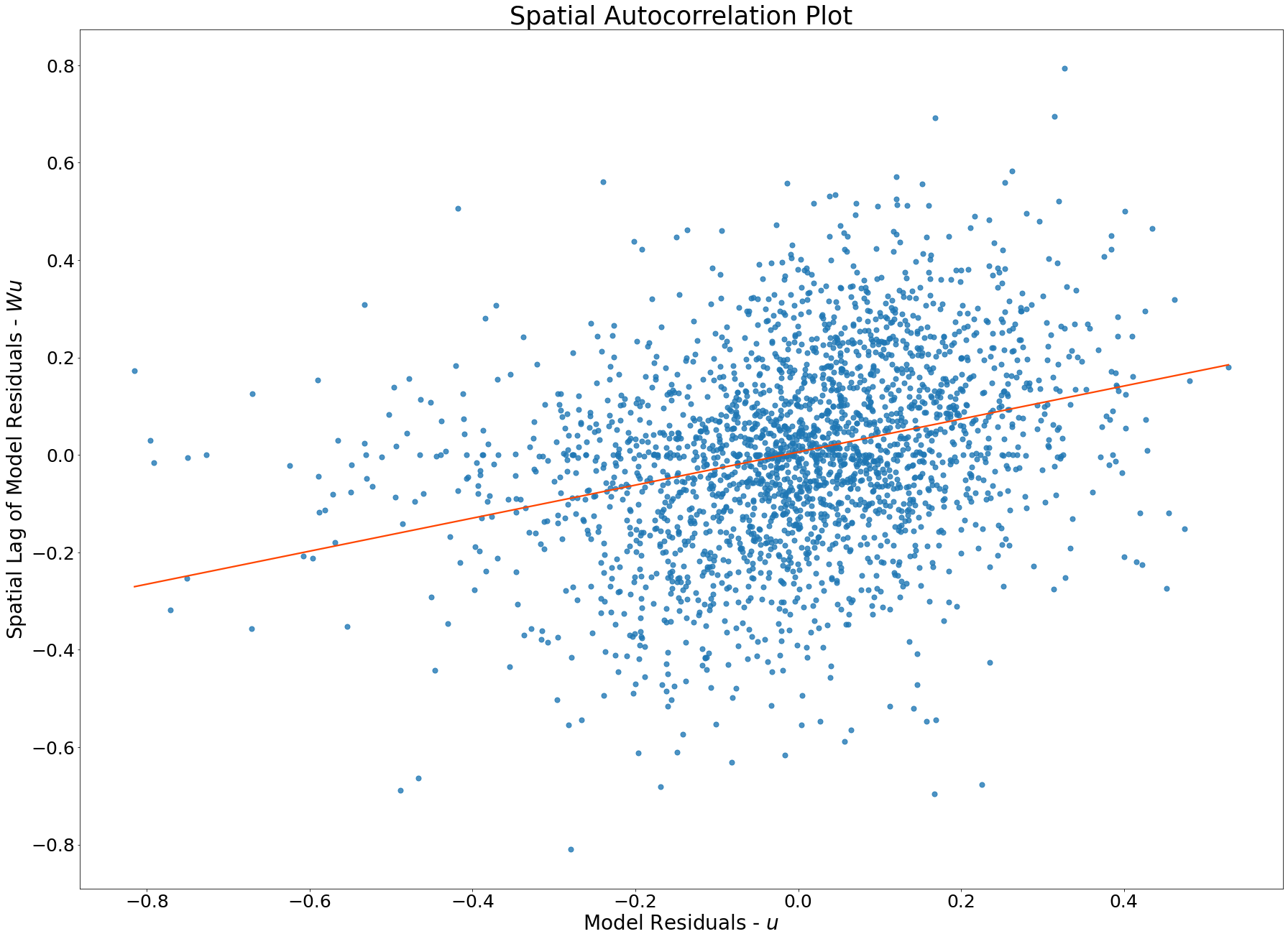


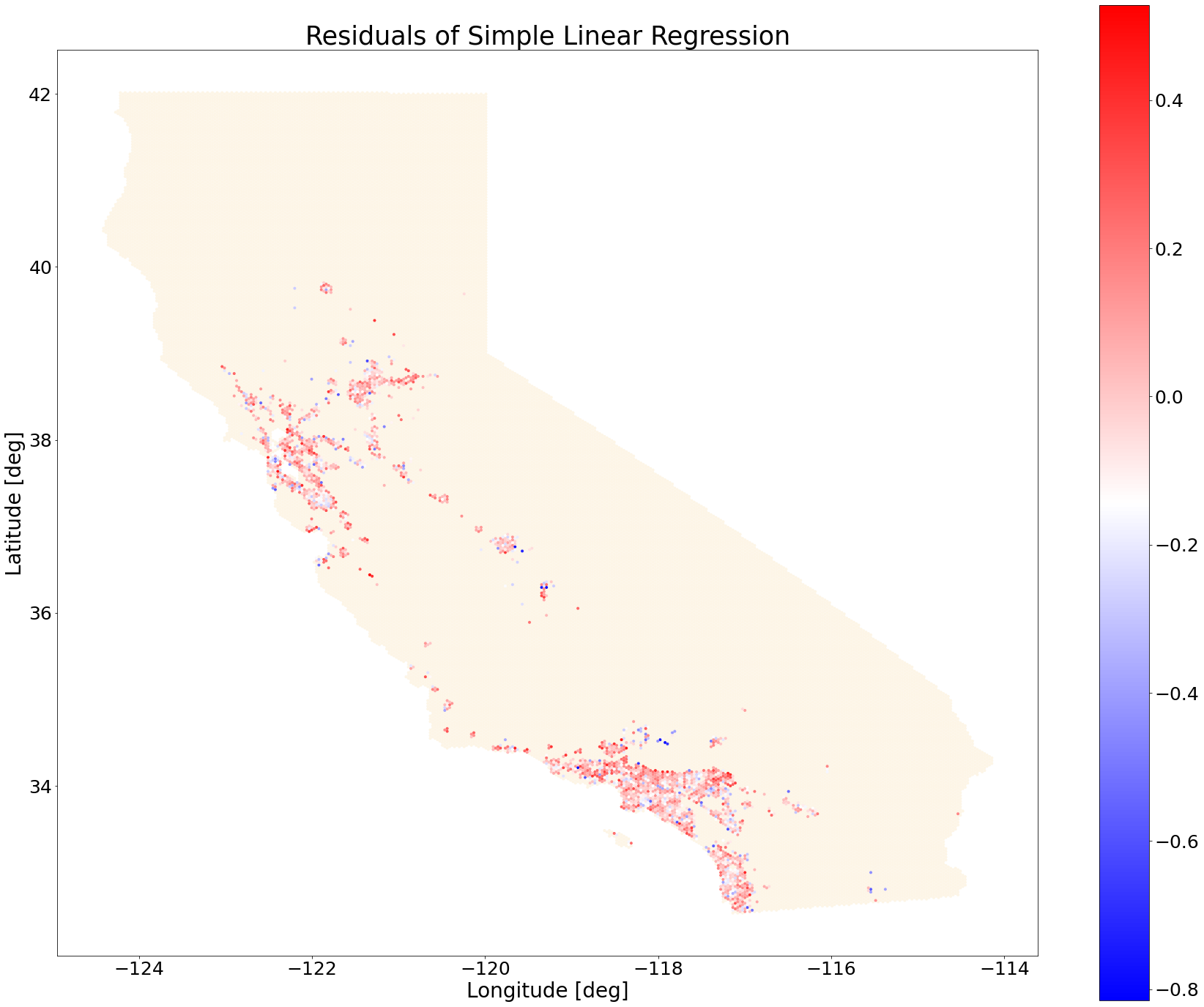


# Spatial Modeling

## Data Dependence

Spatial AutoCorrelation is important and worth understanding and considering

* The measure of spatial autocorrelation is Moran’s I
* 
* Moran’s test indicates spatial dependence
* Additionally, spatial autocorrelation plot shows a strong correlation



## Modeling Dataset

%%bigquery --pro ject urban-forest-269703 df\_10km

select \* except(BirdCounts, TreeCounts, LogBirdCounts, BirdSpeciesCounts,

PercentNativeTrees, EvenBirds, LoveTrees, LoveBirds, avg\_beta\_richness\_tree,

avg\_beta\_richness\_bird, avg\_beta\_turnover\_bird, avg\_beta\_turnover\_tree)

FROM `urban-forest-269703.treebirdagg.10km-stats`

df\_10km['PercentNativeTrees'] = df\_10km['CountNativeTrees'] / np.exp(df\_10km['LogTreeCounts'])

df\_10km['PercentNativeTreeSpecies'] = df\_10km['CountNativeTreeSpecies'] / df\_10km['TreeSpeciesCounts']

df\_10km['PercentNativeRegion'] = df\_10km['nativeTree'] / np.exp(df\_10km['LogTreeCounts'])

df\_10km['PercentNativeRegionSpecies'] = df\_10km['nativeSpecies'] / df\_10km['TreeSpeciesCounts']

df\_10km['LogPopulation'] = np.log(df\_10km['Population'] + 1)

df\_10km['pct\_fleshy\_fruit'] = df\_10km['cnt\_fleshy\_fruit'] / (

df\_10km['cnt\_fleshy\_fruit'] + df\_10km['cnt\_dehiscent'] + df\_10km['cnt\_indehiscent']

)

df\_10km['pct\_dehiscent'] = df\_10km['cnt\_dehiscent'] / (

df\_10km['cnt\_fleshy\_fruit'] + df\_10km['cnt\_dehiscent'] + df\_10km['cnt\_indehiscent']

)

df\_10km['pct\_indehiscent'] = df\_10km['cnt\_indehiscent'] / (

df\_10km['cnt\_fleshy\_fruit'] + df\_10km['cnt\_dehiscent'] + df\_10km['cnt\_indehiscent']

)

df\_10km['geometry'] = df\_10km['geom'].apply(wkt.loads)

y = df\_10km['ShannonBird']

X = df\_10km.drop(['ID', 'geom', 'ShannonBird', 'geometry', 'pct\_partly\_deciduous',

'pct\_perfect\_flowers', 'CountNativeTrees', 'CountNativeTreeSpecies',

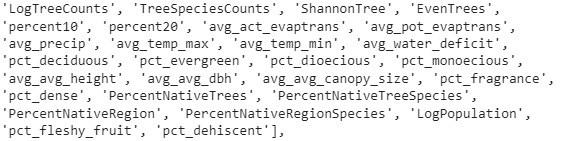
'nativeTree', 'nativeSpecies', 'cnt\_fleshy\_fruit', 'cnt\_dehiscent',

'cnt\_indehiscent', 'pct\_indehiscent', 'Population'], axis=1)

X\_norm=(X-X.mean())/X.std()

X\_geo = pd.concat([X\_norm, df\_10km['geometry']], axis=1)

X\_gdf = gpd.GeoDataFrame(X\_geo, geometry='geometry')



These are our chosen predictor variables:

## Model Research

Given the spatial autocorrelation in our data, we must consider model options other than simple Linear Regression. There are two general categories of spatial models in the literature. The first are **spatial dependence models**. For spatial dependence models, the outcomes observed at locations that are nearby geographically may be dependent on each other. In this case, the independence assumption is definitely violated. The general method with spatial dependence models is a spatial lag matrix, which represents weights for the k nearest neighbors of any hexagon. We have two types of spatial dependence models to account for this.

<https://web.pdx.edu/~crkl/WISE/SEAUG/papers/anselin01_CTE14.pdf>, page 7

<https://geographicdata.science/book/notebooks/11_regression.html#spatial-dependence>

The first one is a **spatial lag model**, where the spatial lag matrix is used to obtain the weighted average of the outcome variable for nearby hexagons, which is added as a predictor variable. In this case, the existence and strength of a spatial interaction becomes of interest in our model. Adding the outcome variable as a predictor is troublesome because it violates the exogeneity assumption that the independent variables X are not dependent on the dependent variable. For this reason, a two-stage least squares estimator is used instead of linear regression

The second model is a **spatial error model**, where the spatial lag matrix is used to obtain the weighted average of the error for nearby hexagons, which is included in the random error term of the regression model. This model treats the spatial correlation as a nuisance due to the spatial nature of the data. Basically, the spatial correlation is treated as a biasing influence that needs to be corrected for rather than a variable of interest. Once again, we can’t use a simple linear regression due to adding the spatial error, which violates the heteroscedasticity assumption. Therefore, a general method of moments estimator that accounts for heteroscedasticity is used.

The next type of model is a **geographically weighted model. Geographically weighted regression (GWR)** is a spatial statistical technique that recognizes that traditional ‘global’ regression models may be limited when spatial processes vary with spatial context. GWR captures process spatial heterogeneity by allowing effects to vary over space. To do this, GWR calibrates an ensemble of local linear models at any number of locations using ‘borrowed’ nearby data. This provides a surface of location-specific parameter estimates for each relationship in the model that is allowed to vary spatially, as well as a single bandwidth parameter that provides intuition about the geographic scale of the processes.

<https://www.mdpi.com/2220-9964/8/6/269>

**Geographically Weighted Random Forests: (**Deemed not Successful)<https://www.tandfonline.com/doi/full/10.1080/10106049.2019.1595177>

* This version was designed to be similar to GWR
* While it does have predictive capabilities I see it would be useful for showing the effect of predictors in areas.
  + This can already be done by GWR so maybe just to see if they have the same conclusions
* Can also show R^2 spatially, to see where the models are most accurate/explained.

Using this Implementation : [SpatialML: Spatial Machine Learning](https://rdrr.io/cran/SpatialML/)

**OUTPUT**

* Importance (For each predictor prints)
  + %IncMSE : analogous to Effect Size, reflects the mean increase in MSE that the predictor contributes, divided by a measure of variability
  + IncNodePurity : measure from the lost function (higher results mean more useful)
* Residuals OOB IQR
* Residuals Predicted IQR
* %IncMSE for each predictor (Min, Max, Mean, Sd)
* %IncNodePurity for each predictor (Min, Max, Mean, Sd)
* **LGofFit**
  + y
  + LM\_yfitOOB
  + LM\_ResOOB
  + LM\_yfitPred
  + LM\_ResPred
  + LM\_MSR
  + LM\_Rsq100
* Local.Pc.IncMSE : a numeric data frame with the local feature importance (IncMSE) for each predictor in each local random forest model
* Local.IncNodePurity : a numeric data frame with the local IncNodePurity for each predictor in each local random forest model

<http://www.biomedware.com/files/documentation/spacestat/Statistics/Multivariate_Modeling/Regression/About_Geographically_Weighted_Regression.htm>

* “Due to the focus on local relationships, and the strong influence of how "local" is defined on GWR results (i.e., through choice of neighbor relationships and bandwidths), GWR is often thought of as a tool for exploring patterns in your data and generating hypothesis for further testing, rather than as a tool for testing a priori hypotheses”

<https://onlinelibrary.wiley.com/doi/full/10.1046/j.1466-822X.2003.00322.x>

<https://onlinelibrary.wiley.com/doi/abs/10.1111/gcb.12834>