Statistical Modeling of Wildfire Dynamics in Southeast Asia: The Role of Rainfall and Drought

Individual Report
Case Studies in Statistics
March 24, 2023

Qichen Huang
The University of British Columbia
qhuang20@student.ubc.ca

Abstract

In past studies, ecologists found that nearly 30.5% of continental Southeast Asia (SEA)'s land have recurrent fires occurring every year within a fifteen year time period, with the most serious cases in Laos, Cambodia, and Burma.[1] These forest fires can cause major economic, social, ecological, and environmental damage, thus, studying the different characteristics of these wildfires can assist in the development of prevention strategies to mitigate the damage caused by the wildfire across the world as well as understanding how the global climate can contribute to these fire activities. Therefore, this research aims to find the similar regions in SEA exhibiting similar fire patterns as well as describing the relationships between rainfall patterns and fire patterns. The methodology employed in our research includes a clustering algorithm for finding areas of similar fire patterns, as well as generalized additive models and random forest regression for determining the most important rainfall variables influencing the degree of fire activity. Our results show that drought severity and percentage of natural vegetation are the most important rainfall factors that affect the fire patterns in the region. Additionally, results also show various areas within SEA having similar fire patterns, with a total of 16 different clustered areas.

Introduction

The negative impact that wildfires have on the environment, especially in today's geographical climate, should be monitored and mitigated as largely as possible. In order to address this problem, it may be helpful to identify the source and characteristics of wildfires in a certain area. Findings may yield possible directions in which preventative measures and policies can be taken. Our project investigates the relationships between climate factors – in particular, rainfall measurements – and fire activity in continental SEA. We also investigate whether fire activity varies across the continent in any special patterns, which may indicate special geographical characteristics of particular areas. These objectives are summarized into two research questions: 1) how do fire patterns vary across continental SEA, and 2) how do rainfall patterns influence variations in fire activity. By using generalized additive models and the Random Forest method, we find that among the different rainfall pattern measures, the measures of drought severity and percentage of natural vegetation have an important effect on fire activity. Using spatially constrained clustering methods to form geographical clusters based on fire patterns, we also find that fire patterns do vary across continental SEA in a way that could be attributed to special geographical characteristics. Details and reasonings for our chosen methods of statistical analysis can be found later in the report. This report includes an overview of our data and empirical data analysis, an explanation of our statistical analysis methods, the details and results of our analysis, concluding remarks, and further discussions.

Data Description

The data was formed by taking the average of satellite data collected over 20 years. Each average is one observation and covers an area of 0.25° resolutions, which converts approximately to a 26 x 26 kilometer square. The dataset contains 2498 rows and 16 columns. No missing cells in the dataset were identified

Below is a table containing the code names of each variable as found in the dataset and their respective definitions.

Table 1.0: Variable Definitions

Variable name	Variable Code	Definition
x	x	The x-coordinate of an observation.
y	y	The y-coordinate of an observation.
Country identifier	country_id	An integer that represents the country of an observation.
Country name	country_name	The name of the country that a gridcell is located in.

[1] https://www.nature.com/articles/s41598-019-43940-x

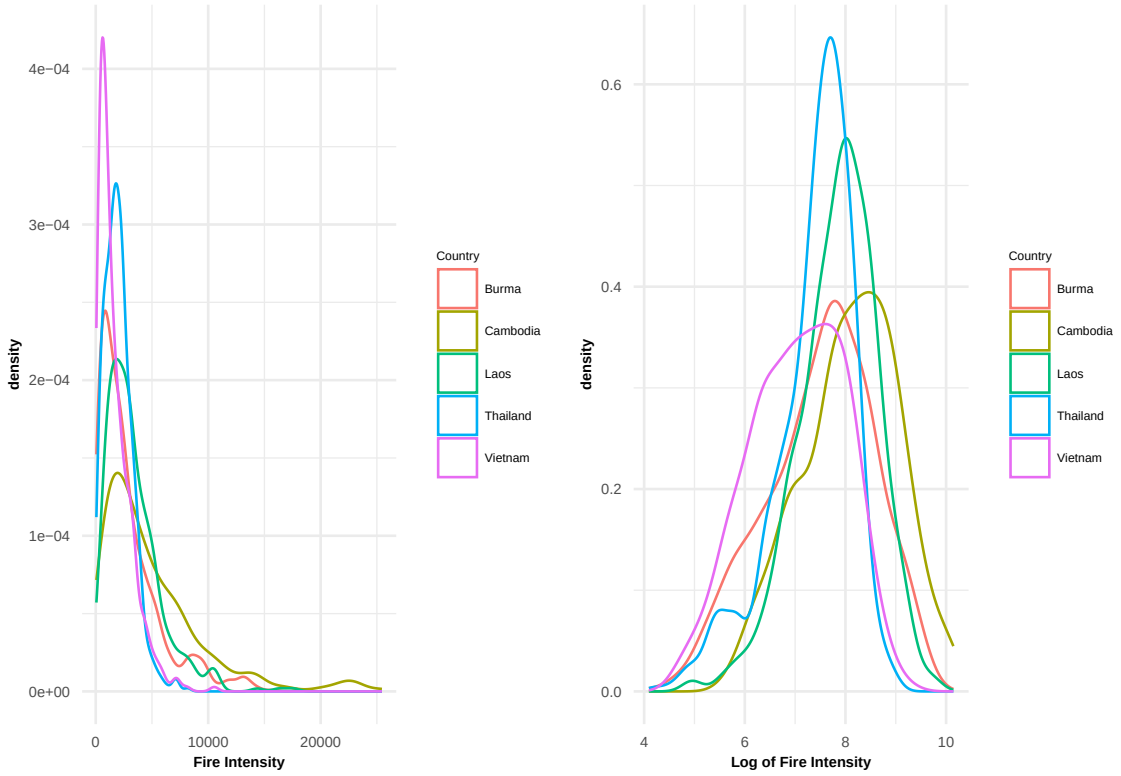
Variable name	Variable Code	Definition
Land type identifier	igbpLC_ID	The Annual International Geosphere-Biosphere Programme (IGBP) classification of land cover, coded as integers.
Simplified land type identifier	simpLD_ID	Simplified land cover classification, coded as integers.
Land type name	igbp_type	The name of the IGBP land cover type that the gridcell is located in.
Simplified land type name	simpLC_type	The name of the simplified land cover type that the gridcell is located in.
Average annual precipitation	MAP	Mean Annual Precipitation (units: millimeters); the measure of how much annual rain occurs in each gridcell.
Drought severity	drought_severity	The severity of the drought, measured by the Maximum Climatological Water Deficit (MCWD).
Seasonality of rainfall distributions	seasonality_index	An indication of how rainfall is distributed throughout the year, calculated with Feng's Seasonality Index.
Percent of natural vegetation	pctnatveg	Percent Natural Vegetation, indicating a rough estimate for amount of human activity in the gridcell.
Fire intensity	fire_intensity	Characterised by the Maximum Fire Radiative Power (FRP), a measure of how much energy is released when biomass is burned. An indicator of how intense the fire was at the given gridcell.
Burned area	burned_area	Total (summed) area burned in a gridcell (units: km^2).
Fire size	fire_size	The average size of the fire perimeter for fires in each gridcell (units: km^2).
Fire frequency	number_of_fires	The average number of fires that occurred in the gricell for the 20 year study period.

Exploratory Data Analysis

Transformations performed to the data include: normalization of the explanatory variables (drought severity, seasonality index, percentage of natural vegetation) and taking the natural log of the response variables (fire intensity, burned area, fire size, number of fires).

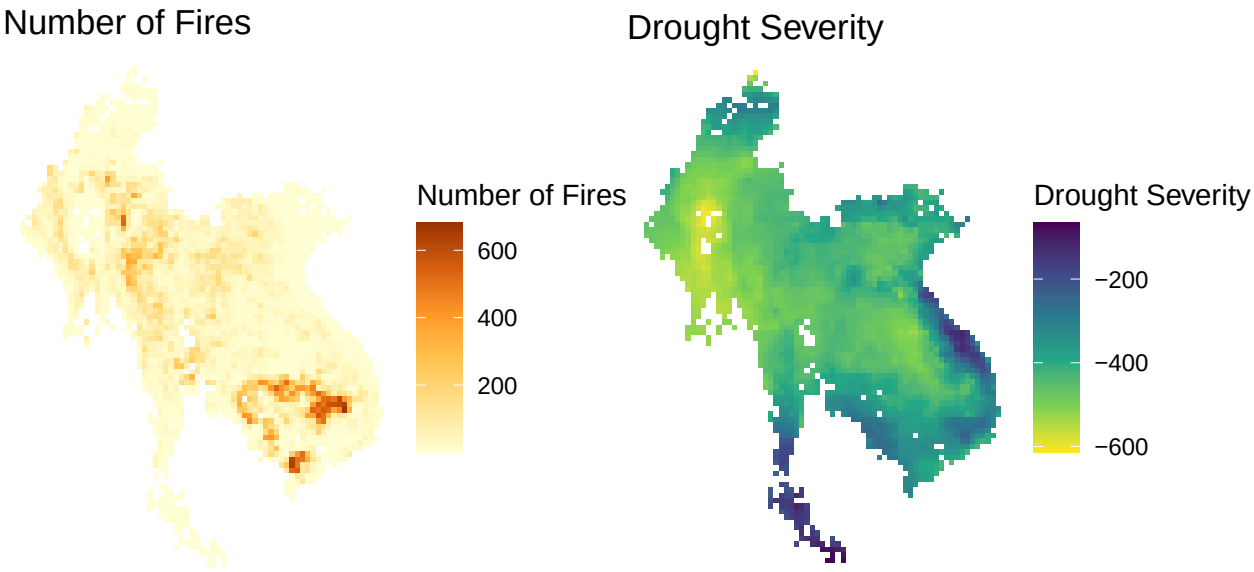
The following plots are an example of why it was necessary to take the natural log of the response variables, as the data is otherwise heavily right-skewed.

Figure 1.0: Fire Intensity Before and After Log Transformation



Raster plots were used visually inspect any patterns among the fire pattern and rainfall patterns. Per Figure 2.0, we found a clear inverted U-shape in the lower region of SEA. We also found that drought severity is the lowest in the coasts of SEA.

Figure 2.0: Raster Plots for Number of Fires and Drought Severity



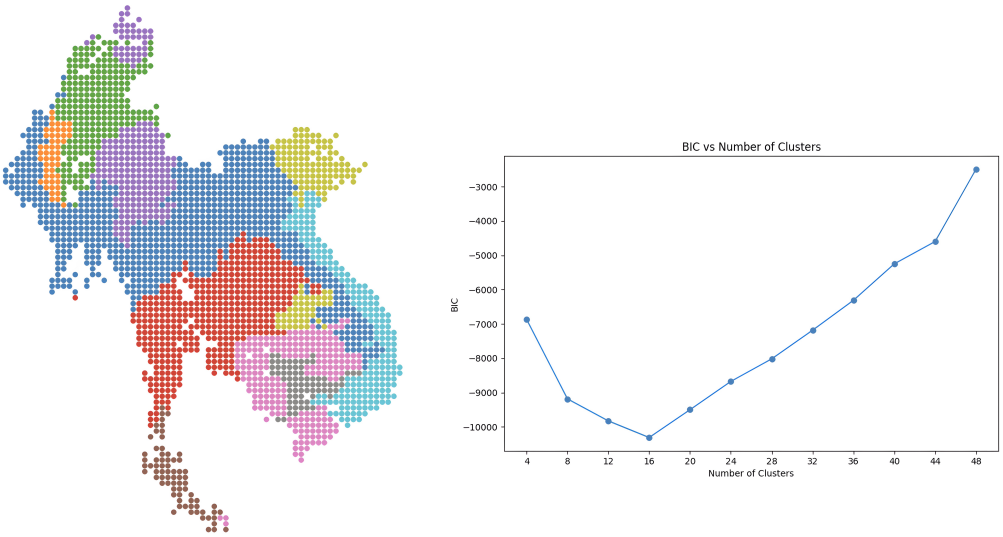
Statistical Methods and Results

This section includes all the statistical methods that we have performed and their analyses.

Regionalization

We selected the SKATER (Spatial 'K'luster Analysis by Tree Edge Removal) algorithm as our spatially constrained clustering technique to aggregate similar (x, y) coordinates. We opted for SKATER as it guarantees that the clustered points are geographically adjacent – a frequent necessity for regionalization within the field of ecology.

Figure 3.0: Regionalization of South East Asia



We selected k=16 as the optimal number of clusters, as it exhibited the lowest Bayesian Information Criterion (BIC) score. We used the BIC score as it penalizes using a higher k that leads to overfitting while still optimizing for better clustering.

From the observed BIC score, our clusters demonstrate that there is variability in fire and rainfall patterns across SEA. Further, we can visually inspect that the identified clusters effectively capture the patterns we saw from our EDA from Figure 2.0 (e.g. inverted U-shape in fire patterns in lower SEA).

Generalized Additive Models

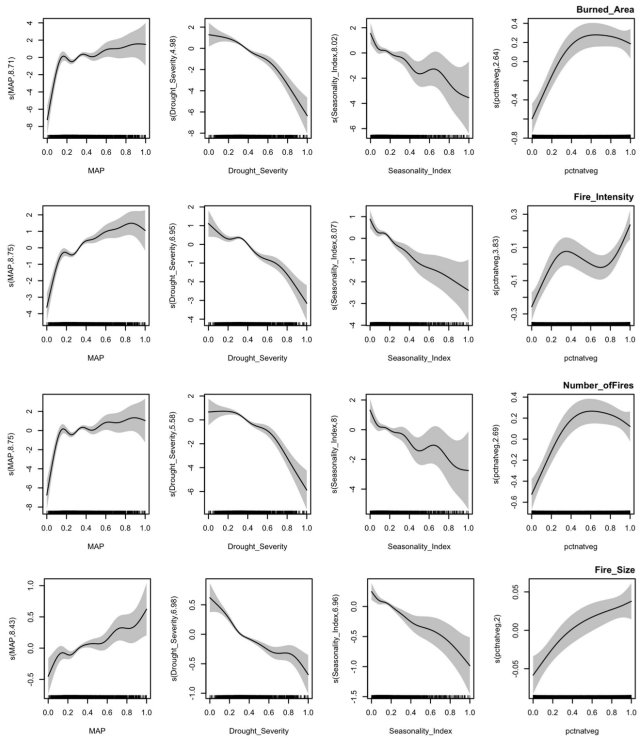
GAMs offer several advantages over traditional linear regression models. For instance, they enable researchers to identify and understand complex, non-linear patterns in data, which are often found in ecological systems. Additionally, GAMs provide flexibility in choosing the type of smooth function applied to each predictor variable, allowing for a better fit to the data.

The model equation is as follows, where s_i is the smooth function.

$$\log(y) = s_1(\text{MAP}) + s_2(\text{drought severity}) + s_3(\text{seasonablity index}) + s_4(\text{pctnatveg})$$

In our study, we use Generalized Additive Models (GAM) to explore the relationships between rainfall patterns and fire activity within the ecological context of Southeast Asia. GAMs allow us to represent the complex interplay between predictor variables, such as rainfall and drought severity, and the response variable, which is fire activity in this case, as a sum of smooth functions. This approach is particularly beneficial in ecological research because it can effectively capture the non-linear relationships that frequently exist between environmental factors and the ecological processes they influence.

Figure 4.0: Generalized Additive Model Results



In our analysis, we used the cubic regression spline as the smooth function. This choice allowed us to account for potential non-linear relationships between the predictor variables and the response variable while minimizing the risk of overfitting. The use of GAMs, with their ability to capture non-linear relationships, helps us gain a deeper understanding of the intricate connections between climate factors and fire activity in Southeast Asia.

Our analysis reveals several key findings about the relationships between climate factors and fire activity in Southeast Asia. We observe that mean annual precipitation has a positive correlation with all dependent variables, which suggests that higher precipitation levels are associated with increased fire activity. In contrast, drought severity and seasonality index are negatively correlated with all dependent variables, indicating that more severe drought conditions and higher seasonality result in reduced fire activity.

The influence of the percentage of natural vegetation on fire activity shows a more complex pattern. For both the number of fires and the burned area, the relationship is modeled by an upside-down U-shape, which implies that there is an optimal level of vegetation for these variables. The effect of the percentage of natural vegetation on fire size is positively linear, meaning that as vegetation increases, so does the size of fires. Finally, the impact of the percentage of natural vegetation on fire intensity is modeled by a cubic function with a constant, indicating a more intricate relationship between these factors.

Figure 5.0: Generalized Additive Model F-stat Results

To further understand the importance of each explanatory variable in our model, we employed the F-stat to analyze the significance of each smooth function for the fire activity response variables. Our findings suggest that drought severity is the most crucial factor influencing fire size, while the percentage of natural vegetation plays the most significant role in determining both the burned area and the number of fires.

Random Forest Regression

We also use Random Forest regression to analyze how the accuracy of each model changes when an explanatory variable is excluded. This method helps to identify which, if any, explanatory variable is especially important in our model.

Random Forest regression is constructed in such a way that the predictions made by each estimator is averaged as the final prediction of the model. Throughout the model training process, we performed hyperparameter tuning on the random forest model in order to yield the best possible prediction results as well as to provide meaningful parameters to our model rather than using trivial parameters. Table 2.0 shows the results of our hyperparameter tuning. Note that during this training phase, we performed a 3-fold cross-validation training method repeated 2 times.

Table 2.0: Results for Hyperparameter Tuning on Random Forest Regression

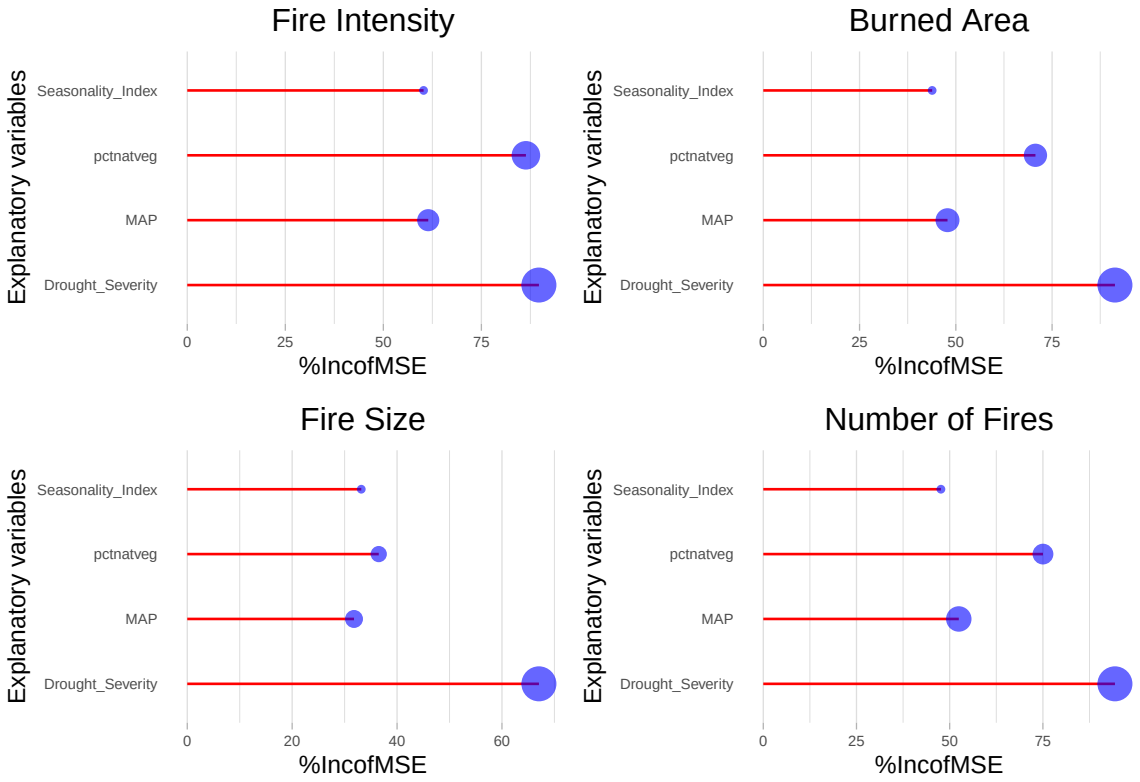
	# of estimators/trees	# of variables tried each split	MSE	% of Variance Explained
Fire Intensity	500	2	0.66	32.66
Burned Area	500	3	2.90	33.09
Fire Size	500	2	0.09	25.53
Number of Fires	500	2	2.36	36.64

From the results shown above, our best number of estimators/trees for each four random forest models is 500 trees. We also see that for the majority of the models, the number of variables tried at each split is 2, with the exception of the model on burned area. We can also see that all four models have a relatively low mean squared error (MSE) during the training, with also a relatively low percentage of variance explained. As this

will be discussed in the further discussions part of the paper, this low variance explained is largely due to the nature of our aggregated data.

After obtaining the set of hyperparameters (ie. number of trees), we then use these learned values to fit the full random forest regression model to obtain our variable importance in each model. Figure 4.0 shows the the variable importance plots for each model, measured in percentage of increase in mean squared error (%IncMSE). The higher the value for %IncMSE, the more significant the explanatory variable is to the overall model.

Figure 6.0: Random Forest Regression Variable Importance Plots



The Random Forest models are represented thus: A for fire intensity, B for burned area, C for fire size, and D for number of fires. As we can see in all four plots, drought severity and percentage of natural vegetation are the most important explanatory variables for each of the four fire activity models. This is consistent with the ecological context that these two rainfall factors can have a significant impact on the degree of forest fires in the area.

Conclusions

Our work so far allows us to confidently answer the question of how rainfall patterns influence variations in fire activity. Through our GAM analysis, we find evidence of mean annual precipitation having a positive effect on all fire activity variables; that is, as mean annual precipitation increases, all measures of fire activity increase. There is evidence of drought severity having a negative effect on all fire activity variables; that is, as drought becomes less severe, all measures of fire activity decrease. There is evidence of the seasonality index having a negative effect on all fire activity variables; that is, as the seasonality index increases, all measures of fire activity decrease. There is evidence of the effect of percentage of natural vegetation on fire activity variables but the type of effect varies per measure. From our RF analysis, drought severity has the greatest explanatory significance to the fire pattern data. We'd also like to note that the direction of the effect of mean annual precipitation, drought severity and seasonality index on all four fire activity variables is agreed

upon by our GAM and RF models. That drought severity and percentage of natural vegetation are the more important explanatory variables in accounting for the variation in fire activity is also agreed upon by our GAM and RF models.

With regards to our regionalization work, we find no results of significance. At our specified size of region, fire patterns do not vary according to any geographical pattern across continental Southeast Asia.

Further Discussions

It may be worthwhile to explore the relationships, if any, between the rainfall variables themselves; the extent of our analysis only explores relationships between rainfall pattern and fire activity variables. Secondly, it may be possible to achieve better clustering with regards to our regionalization analysis by making each region smaller; we may have found insignificant results due to regions being too distinct to be clustered together. Lastly, it may be noted that we do not comment on the predictability of our models due to the nature of our data. Our analysis does not account for any trends that may plausibly affect the data over the next 20 years (ie. the effects of climate change). In order to generate models with more predictive power, we suggest a time series analysis on the raw, that is, not aggregated, data over 20 years; that way, any trend or seasonal affects may be properly detected and dealt with.

Appendix

Regionalization Code (in Python)

Code for Importing Python Libraries

```
import pandas as pd
import numpy as np
import geopandas as gpd
import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import pairwise as skm
from shapely.geometry import Point
from libpysal.weights import Queen
import spopt

# VRC, BIC, Gap Statistic Stuff
from sklearn.cluster import KMeans
from sklearn.mixture import GaussianMixture
from scipy.spatial.distance import cdist
from scipy.stats import uniform

# Multiprocessing stuff
import concurrent.futures
import json
```

Code for Transforming Data

```
def transform_df(df):
    # Select all the right columns
    X_y = df[[
        'x',
        'y',
        'MAP',
        'Drought_Severity',
        'Seasonality_Index',
        'pctnatveg',
        'Fire_Intensity',
        'Burned_Area',
        'Fire_Size',
        'Number_ofFires'
    ]]

    # Scale All
    scaler = MinMaxScaler()
    scaled_columns = [
        'MAP',
        'Drought_Severity',
        'Seasonality_Index',
        'pctnatveg'
    ]
    X_y_scaled = X_y.copy()
    X_y_scaled[scaled_columns] = scaler.fit_transform(X_y[scaled_columns])
```

```
    # Apply log transformation to the remaining columns
    log_transform_columns = [
        'Fire_Intensity',
        'Burned_Area',
        'Fire_Size',
        'Number_ofFires'
    ]

    X_y_scaled[log_transform_columns] = np.log(X_y[log_transform_columns])
    return X_y_scaled
```

Code for Converting Scaled X, y into Geopandas DataFrame

```
def convert_to_gdf(X_y_scaled):
    # Convert X_y_scaled to geopands dataframe
    geometry = [Point(xy) for xy in zip(X_y_scaled.x, X_y_scaled.y)]
    clean_gdf = gpd.GeoDataFrame(X_y_scaled, geometry=geometry)

    # Drop the x, y columns
    clean_gdf = clean_gdf.drop(['x', 'y'], axis=1)
    return clean_gdf
```

Code for Running SKATER Algorithm

```
def runSkater(gdf, attrs, n_clusters=4, floor=5, trace=True, islands="ignore"):
    # Calculate Queen contiguity weights
    w = Queen.from_dataframe(gdf)

    # Spanning Forest kwds
    spanning_forest_kwds = dict(
        dissimilarity=skm.manhattan_distances,
        affinity=None,
        reduction=np.sum,
        center=np.mean,
        verbose=2
    )

    # Parameters
    n_clusters = n_clusters # number of clusters
    floor = floor # min number of spatial objects in each region
    trace = trace
    islands = islands

    # Instantiate the SKATER model
    model = spopt.region.Skater(
        gdf,
        w,
        attrs,
        n_clusters=n_clusters,
        floor=floor,
        trace=trace,
        islands=islands,
```

```
        spanning_forest_kwds=spanning_forest_kwds
    )

    # Fit the model to the data
    model.solve()

    # Assign clusters
    gdf["cluster"] = model.labels_

    return gdf, model
```

Code for Computing Regionalization Metrics (i.e. VRC, BIC, Gap Statistic)

```
"""
Compute VRC
"""
def compute_vrc(X, labels, n_clusters):
    N, D = X.shape
    cluster_centroids = np.array([X[labels == k].mean(axis=0) for k in range(n_clusters)])
    global_centroid = X.mean(axis=0)
    BGSS = np.sum(
      [
        np.sum((cluster_centroids[k] - global_centroid) ** 2) * len(X[labels == k])
        for k in range(n_clusters)
      ]
    )
    WGSS = np.sum(
      [
        np.sum((X[labels == k] - cluster_centroids[k]) ** 2)
        for k in range(n_clusters)
      ]
    )
    VRC = (BGSS / (n_clusters - 1)) / (WGSS / (N - n_clusters))
    return VRC

"""
Compute BIC
"""
def compute_bic(X, labels, n_clusters):
    N, D = X.shape
    gmm = GaussianMixture(n_components=n_clusters, random_state=42)
    gmm.fit(X)
    LL = gmm.score(X) * N
    P = (n_clusters * (D + 1) * (D + 2)) / 2
    BIC = -2 * LL + P * np.log(N)
    return BIC

"""
Compute Gap Statistic
"""
def compute_gap_statistic(X, labels, n_clusters, B=10):
    N, D = X.shape
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
```

```
    kmeans.fit(X)
    WGSS = kmeans.inertia_
    WGSS_ref_list = []
    for b in range(B):
        X_ref = uniform.rvs(np.min(X, axis=0), np.ptp(X, axis=0), size=(N, D))
        kmeans_ref = KMeans(n_clusters=n_clusters, random_state=42)
        kmeans_ref.fit(X_ref)
        WGSS_ref_list.append(kmeans_ref.inertia_)
    E_log_WGSS_ref = np.mean(np.log(WGSS_ref_list))
    Gap = E_log_WGSS_ref - np.log(WGSS)
    return Gap
```

Code for Running SKATER algorithm for multiple n_cluster value and computing metrics

```
def run_experiment(n_cluster, clean_gdf, attrs):
    print("Handling n_cluster = ", n_cluster, "...")
    gdf = clean_gdf.copy()
    gdf, model = runSkater(gdf, attrs, n_clusters=n_cluster)

    # Compute bSStSSRatio
    ratio = computebSStSSRatio(gdf, attrs)

    # Compute WSS
    WSS = computeWSS(gdf, attrs)

    # Compute BIC
    BIC = compute_bic(gdf[list(attrs)].values, gdf.cluster.values, n_cluster)

    # Compute VRC
    VRC = compute_vrc(gdf[list(attrs)].values, gdf.cluster.values, n_cluster)

    # Compute Gap Statistic
    Gap = compute_gap_statistic(gdf[list(attrs)].values, gdf.cluster.values, n_cluster)

    return n_cluster, ratio, WSS, model, BIC, VRC, Gap
```

Code for Running SKATER algorithm for multiple n_cluster values across multiple processors

```
def main():
    # Read in the data
    df = pd.read_csv('../../data/Stat450allvariables_df.csv', index_col=0)

    # Transform (log + scale)
    df = transform_df(df)

    # Convert to geopanads
    clean_gdf = convert_to_gdf(df)

    # Set Up for the Experiments
    attrs = np.array([
        "MAP",
        "Drought_Severity",
```

```
        "Seasonality_Index",
        "pctnatveg",
        "Fire_Intensity",
        "Burned_Area",
        "Fire_Size",
        "Number_ofFires"
    ])

    # Sanity Check, as we increase n_clusters, bSStSSRatio should decrease
    # k_to_ratio = {}
    # k_to_WSS = {}
    k_to_models = {}
    k_to_vrc = {}
    k_to_bic = {}
    k_to_gap = {}

    # Define the range of n_clusters
    max_cluster = 49
    n_clusters_range = range(4, max_cluster, 4)

    # Use a process pool to run the experiments in parallel
    with concurrent.futures.ProcessPoolExecutor() as executor:
        futures = [
          executor.submit(
            run_experiment,
            n_cluster,
            clean_gdf,
            attrs
            ) for n_cluster in n_clusters_range
          ]

        for future in concurrent.futures.as_completed(futures):
            n_cluster, _, _, model, BIC, VRC, Gap = future.result()
            k_to_bic[n_cluster] = BIC
            k_to_vrc[n_cluster] = VRC
            k_to_gap[n_cluster] = Gap
            k_to_models[n_cluster] = model

    with open('k_to_bic.json', 'w') as f:
        json.dump(k_to_bic, f)

    with open('k_to_vrc.json', 'w') as f:
        json.dump(k_to_vrc, f)

    with open('k_to_gap.json', 'w') as f:
        json.dump(k_to_gap, f)

    return
```

GAM Code

Code for Plotting both the model and F-stat for GAM

```
### GAM summary/plot fit and cv accuracy
Accuracy_fit$y <- ynames; Accuracy_fit$Modelling <- "Fit"
Accuracy_cv$y <- ynames; Accuracy_cv$Modelling <- "CV"
Accuracy <- bind_rows(Accuracy_fit, Accuracy_cv)
ga <- ggplot(Accuracy, aes(x=y, y=Rsquared, group=Modelling, fill=Modelling))+
  geom_bar(position="dodge", stat="identity") +
  labs(y=bquote(R^2)) +
  scale_fill_manual(values=c("#D984A0","#53D1C4")) +
  theme(panel.background=element_rect(fill="white",color="grey35"),
        panel.grid= element_line(color="grey95"))
ggsave(filename="Accuracy.jpg", plot=ga, width=6, height=3, units="in", dpi=300)
```

Code for GAM Hyperparameter Tuning and F-Stat Calculation

```
############################## Use gmcv package ################################
xnames <- c("MAP", "Drought_Severity", "Seasonality_Index", "pctnatveg")
ynames <- c("Fire_Intensity", "Burned_Area", "Fire_Size", "Number_ofFires")
#### for all y ####
Accuracy_fit <- data.frame()
Accuracy_cv <- data.frame()
Varible_Fva <- data.frame()
for(yn in ynames) {
  # print(paste("Y variable:", yn))
  fit_data <- X_y_scaled
  #fit_data <- cbind(log(raw_data[,yn]),Xscale) #raw_data[,c(yn,xnames)]
  # change the fire variable's name to y
  colnames(fit_data)[colnames(fit_data)==yn] <- "y"
  ### GAM using mgcv package
  set.seed(1)

  formula_gam <- as.formula(paste("y ~", paste(paste("s(",xnames,")", sep=""),
                                               collapse="+")))
  model_gam <- gam(formula_gam, data=fit_data, method="GCV.Cp")
  cvg <- CVgam(formula_gam, data=fit_data, nfold=3, method="GCV.Cp")
  Accuracy_fit <- bind_rows(Accuracy_fit,
                            postResample(pred=model_gam$fitted.values,
                                         obs=fit_data$y))
  Accuracy_cv <- bind_rows(Accuracy_cv, postResample(pred=cvg$fitted,
                                                     obs=fit_data$y))
  anov_gam <- anova.gam(model_gam) #summary.gam(model_gam)
  Varible_Fva <- bind_rows(Varible_Fva, anov_gam$s.table[, "F"])

  ### plot smooth terms for each model
  jpeg(file=paste(yn,".jpg", sep=""), width=7, height=2, units="in",res=300)
  par(mfrow=c(1,4),mar=c(5,5,2,1), cex=0.5)
  plot(model_gam, shade=TRUE, scale=0) #pages=1,
  title(yn,adj = 1)
  dev.off()
}
```

Code for GAM F-stat Plot

```
### GAM summary/plot smoother importance via ANOVA test F-value
Varible_Fva$y <- ynames
Varible_Fva_long <- gather(Varible_Fva, "Smooth", "F_stat", names(Varible_Fva)[1:4])
gv <- ggplot(Varible_Fva_long, aes(x=y, y= Smooth, fill=F_stat)) +
  geom_tile() +
  scale_fill_gradient(low="#BBD7D4",high="#007E79")

ggsave(filename="./Smooth_Importance.jpg", plot=gv, width=6, height=3, units="in", dpi=300)
```

Random Forest Regression Code

Code for Random Forest Regression Hyperparameter Tuning

```
# HYPERPARAMTER TUNING
################################ Use caret packages ##########################
#xnames <- c("MAP", "Drought_Severity", "Seasonality_Index", "pctnatveg")
ynames <- c("Fire_Intensity", "Burned_Area", "Fire_Size", "Number_ofFires")
## min-max scale for X variables
#Xscale <- apply(raw_data[,xnames], MARGIN =2, FUN = function(x, na.rm = TRUE)
#{ return((x- min(x)) /(max(x)-min(x)))})
yn <- ynames[1] # change index from 1 to 4 to change response var
#print(paste("Y variable:", yn))
#fit_data <- cbind(log(raw_data[,yn]),Xscale) #raw_data[,c(yn,xnames)]
colnames(fit_data)[colnames(X_y_scaled)==yn] <- "y" # change the fire variable's name to y
formula <- as.formula(paste("y ~ ", paste(xnames, collapse= "+")))
### k-fold cross-validation
set.seed(1)
repeat_cv <- trainControl(method='repeatedcv', number=3, repeats=2)
train_index <- createDataPartition(y=fit_data$y, p=0.7, list=FALSE)
training_set <- fit_data[train_index, ]
testing_set <- fit_data[-train_index, ]
print("-----------------------------------------------")
##### Random Forest
Rforest <- train(formula, data=training_set, method="rf",
                 trControl=repeat_cv, metric='Rsquared')
print("Rforest-----------")
print(Rforest$finalModel) # print(Rforest)
print(Rforest)
# ##### GAM
Rgam <- train(formula, data=training_set, method="gam",
              trControl=repeat_cv, metric='Rsquared')
print("Rgam-----------")
print(Rgam$finalModel) # print(Rforest)
print(Rgam)
```

Code for Data Manipulation

Code for Selecting X, y from Raw Data

```
#Data transformations
X_y <- raw_data %>%
  dplyr::select(
```

```
    MAP,
    Drought_Severity,
    Seasonality_Index,
    pctnatveg,
    Fire_Intensity,
    Burned_Area,
    Fire_Size,
    Number_ofFires
)
```