Wildfires in Florida & Colorado

Justin Nichols[1]

Colorado School of Mines
Computer Science Department

Abstract

This paper focuses on a small spatial database derived from a larger one that identifies wildfires in the United States from 1992-2015. The small piece of this database consists of two tables with information about the level one eco-regions as well as various features on each fire such as their size and cause. The reduced database is solely on Florida and Colorado. The paper will explore the different features mentioned, summary statistics from the tables, as well as provide visualizations to better interpret what is sparking all the fuss about wildfires.

[1] jnichols2@mymail.mines.edu

Contents

List of Figures

List of Tables

1 Database Introduction

The database used in this paper consists of two tables, which have been shrunken down from containing 1.8 million observations to less than 200,000 in each. The original fires data comes from a spatial database on wildfires in the United States from 1992 to 2015. It is the third update to an original publication to support the Fire Program Analysis system [1]. We see that the owner to this data has waived all their rights, making it public for anyone to use. Since wildfires can impact so many people, this database provides an interesting look into what seems like an overgrowing issue—something we expect everyone to be curious about.

Due to the information being given over many years, we are able to see if this worsening trend is actually occurring. However, because each area of the United States has its unique ways of functioning, we also see some differences in ways the data is reported. More specifically, some of the less important variables tend to be forgotten in some places. Another interesting find is that sometimes it's noticeable that someone forgot to use 24-hour time. This makes it difficult to get an idea of how long it took to contain a fire.

2 A Glance at the Tables

To begin, the data was cleaned, exported to `.csv` files, and then uploaded to the database (see Appendix B.1 for details). Since there are some entries that aren't complete, we have to make sure the database will recognize this. This led to some necessary modifications to the tables before importing. For example, some of the should be `NULL` values weren't being interpreted as they were blank spaces. Once the files were cleaned they were imported into tables that were structured according to their type of entry values. Lets examine the two tables.

One table will provide information about wildfires in Colorado and Florida from 1992-2015. Below, we get a glance at what is seen in some of the rows of this table.

Table 1: Look into the `fires` Table

fpa_id	fire_name	discover_time	stat_cause_code	stat_cause_descr	cont_time	fire_size	fire_size_class
FS-1418940	HATCH	1300	1	Lightning	1530	0.5	B
FS-1418976	TROUT CREEK II	1100	1	Lightning	1218	0.2	A
FS-1418978	MT ELBERT	1241	4	Campfire	1506	0.1	A
FS-1419150	8GN	1907	1	Lightning	2130	80	C
FS-1419344	BEAVER	1428	9	Miscellaneous	1736	2.5	B
FS-1419573	LAZY CHAIR	1908	1	Lightning	1831	0.25	A
FS-1419815	REDWATER	1228	9	Miscellaneous	1301	0.1	A
FS-1420852	MYSTERY	110	9	Miscellaneous	1200	0.1	A
FS-1420865	MOONLIGHT	1115	5	Debris Burning	1700	0.1	A
FS-1420912	HOWDY	1515	5	Debris Burning	1800	1.7	B
FS-1420913	MUD PRAIRIE	1603	8	Children	1721	0.4	B
FS-1420914	SHERRILLS MILL	2000	7	Arson	1212	1.25	B

latitude	longitude	state	county	disc_date	cont_date	season	human
39.2922222	-105.18306	CO	colorado,douglas	2005-06-14	2005-06-14	summer (JJA)	0
38.9133333	-105.98361	CO	colorado,park	2005-05-30	2005-05-31	spring (MAM)	0
39.1002778	-106.3675	CO	colorado,lake	2005-06-21	2005-06-21	summer (JJA)	1
37.345	-102.80583	CO	colorado,baca	2005-07-07	2005-07-07	summer (JJA)	0
38.8911111	-105.43194	CO	colorado,park	2005-06-13	2005-06-13	summer (JJA)	1
39.3975	-105.24611	CO	colorado,jefferson	2005-06-24	2005-06-25	summer (JJA)	0
29.1880556	-81.888611	FL	florida,marion	2005-02-09	2005-02-09	winter (DJF)	1
30.2677778	-84.400278	FL	florida,wakulla	2005-01-22	2005-01-23	winter (DJF)	1
30.2847222	-84.366667	FL	florida,wakulla	2005-01-23	2005-01-23	winter (DJF)	1
30.2	-84.383333	FL	florida,wakulla	2005-01-23	2005-01-23	winter (DJF)	1
29.0925	-81.891111	FL	florida,marion	2005-01-25	2005-01-25	winter (DJF)	1
29.1680556	-81.896111	FL	florida,marion	2005-01-30	2005-01-31	winter (DJF)	1

The other table, `regions`, has information about the different level one eco-regions at each fire location during this time period. This was done using polygon projections based on the longitude and latitude values relative to mapping file with information about the regions. The table includes this information for each observation from the `fires` table (the table exclusively including Colorado and Florida), as well as for Alabama, Connecticut, Delaware, and Wyoming. Here, we get a chance to see what this table looks like.

Table 2: Look into the `regions` Table

fpa_id	state	eco_id	eco_region
FS-1429869	AL	8	EASTERN TEMPERATE FORESTS
FS-1430963	AL	8	EASTERN TEMPERATE FORESTS
FS-1418940	CO	6	NORTHWESTERN FORESTED MOUNTAINS
FS-1418976	CO	6	NORTHWESTERN FORESTED MOUNTAINS
W-239843	CT	5	NORTHERN FORESTS
FWS-2002CTSMR5713	CT	8	EASTERN TEMPERATE FORESTS
FWS-1992DEBHR5082	DE	8	EASTERN TEMPERATE FORESTS
FWS-1992DEPHR5046	DE	8	EASTERN TEMPERATE FORESTS
FS-1419815	FL	8	EASTERN TEMPERATE FORESTS
FS-1420852	FL	8	EASTERN TEMPERATE FORESTS
FS-1418933	WY	6	NORTHWESTERN FORESTED MOUNTAINS
FS-1418965	WY	6	NORTHWESTERN FORESTED MOUNTAINS

As we see, there are 16 columns in the `fires` table and 4 in the `regions` table (see Appendix A for an explanation on each of these). Comparing the two tables, we see there are two columns that each have in common, `fpa_id` and `state`. In order to factor the regions into analysis of fires in Florida and Colorado, we will merge these two tables by the primary key in both tables `fpa_id`. Since this is a primary key, it is unique, and therefore we know we get a one-to-one relationship between these two tables. This will eventually allow the data to be separated by state and region. The following query demonstrates this join in action:

```sql
SELECT f.*, r.eco_id, r.eco_region
FROM fires as f, regions as r
WHERE f.fpa_id = r.fpa_id;
```

The search over each value for `fpa_id` is compared and each, then if they match, we return back everything in the `fires` tables and then just the information relating to the regions from the table. We limit its return to those columns because the `regions` other two columns exists in the `fires` table.

2.1 Summary Statistics

From the previous section, we saw what will be the basis for many queries along the way as it provides another way to further divide the data for analyzing. Since the focus point of this paper is Florida and Colorado, let's start by getting an observation count for each as follows:

```sql
SELECT state, COUNT(*)
FROM fires
GROUP BY state;
```

state	count
FL	87110
CO	31044

We combine these two query approaches in order to get an idea of how many observations are in each of the eco regions within each state.

```
SELECT f.state, r.eco_region, COUNT(*)
FROM fires as f, regions as r
WHERE f.fpa_id = r.fpa_id
GROUP BY f.state, r.eco_region;
```

state	eco_region	count
CO	GREAT PLAINS	4573
CO	NORTH AMERICAN DESERTS	14633
CO	NORTHWESTERN FORESTED MOUNTAINS	11838
FL	EASTERN TEMPERATE FORESTS	79687
FL	TROPICAL WET FORESTS	7423

One important summary to take note of in any database is the number of `NULL` values in each column. In analysis, this can at times cause negative impacts in results. Therefore, it is a good idea to identify where these are prior to further investigation. We tally up the `NULL` values for the `fires` table below by returning only columns where they exist and provide the total number in each.

```
SELECT key as column_name, count(*) as null_values
FROM fires as f
CROSS JOIN jsonb_each_text(to_jsonb(f))
WHERE value IS NULL
GROUP BY key;
```

column_name	null_values
cont_date	68495
cont_time	69281
county	2105
discovery_time	66738
fire_name	68451

The same could be done for the `regions` table, however there aren't any `NULL` values located in the table. With an idea of what lies within the tables, we now dive into some ways to visualize the spatial data over time by stratifying it different ways.

2.2 Visualizations

Now that we have an idea of how to divide the data into regions and/or states, we will now use Python in order to visually see what the data looks like. First, we will read the data in as two data frames—one for Florida and the other for Colorado. These will be the foundation for analyzing fires in these two drastically different locations. Let's first begin by looking at the average `fire_size` by the day of the year. We will look at this in terms of each state, so that we can determine if there are differences. We will use the columns `fire_size` and `disc_date` in order to produce this visual. However, we will look to extract the day of the year from our datetime object. The main query for this is

```
SELECT DATE_PART('doy', disc_date) as day_of_year, avg(fire_size)
FROM fires
WHERE state=%s
GROUP BY day_of_year
ORDER BY day_of_year
```

In order to get a visual representation, we communicate this query to the database through Python in order to get the data into a Pandas dataframe. Above, we let the `%s` be a our prepared query in order to determine which date we want. Therefore, when doing this directly from the database we would actually replace with either "'FL'" or "'CO'". This modular approach makes accessing say the original database with all of the states much more fluid.

With our query defined we are then able to use Python to explore a visualization of this fire size over time. We plot the average fire size by each day of the year and color them by state (see Appendix B.2 for the code). We display this with the day of the year on the x-axis and the average fire size for each day on the y-axis. The color red indicates Florida, while blue represents Colorado.

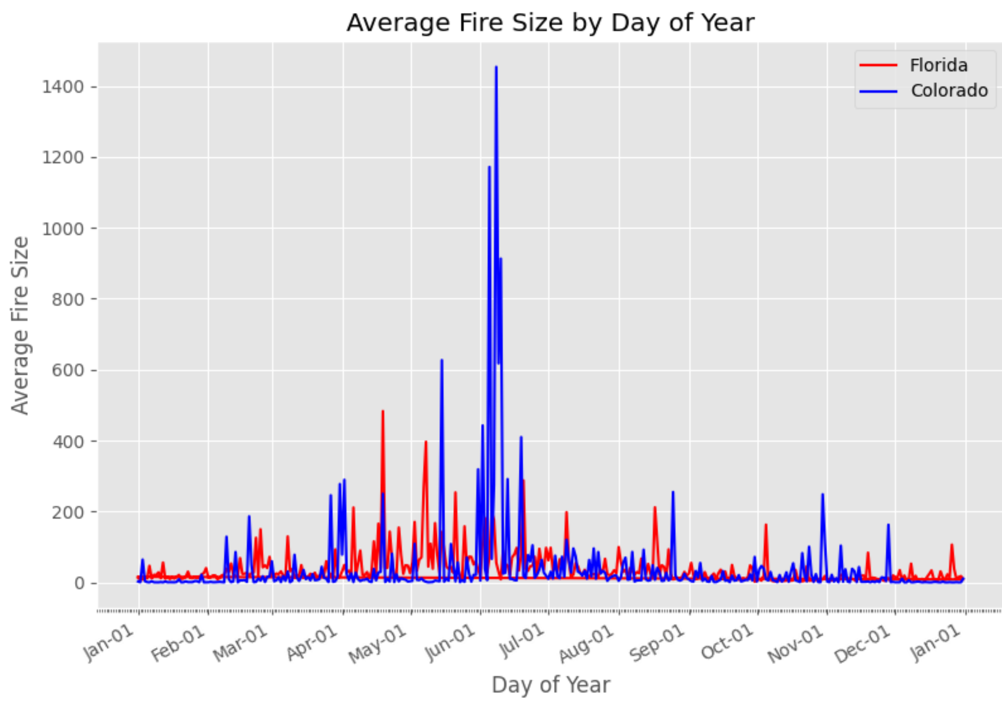

Figure 1: Average Fire Size per Day in Colorado and Florida 1992-2015

We see heavy spikes for Colorado during the summer months. This most likely relates to more people normally camping, as well as it being much more dry during these months. We will explore this regional relationship later. However, in Florida we see smaller spikes, but mostly minimal.

Since each fire is given a classficiation A-G, we will plot a box chart of each of these for the two states. Each bin will represent the number of fires in that class for a given state. Again, we will query these two states separately, this time focusing on the count in each `fire_size_class`. Our main query is

```
SELECT fire_size_class, count(*)
FROM fires
WHERE state=%s
GROUP BY fire_size_class
ORDER BY fire_size_class
```

Once again, we would fill the `%s` with a string representing the state if doing this directly in the database. This will return a table where each row represents a class of fire and the number of fires in that class. Since we have the where clause, we will return just the state identified in that line.

Creating a box plot of these results gives us an idea of the distribution of the fires in each state. We will overlap the two state in order to better compare their similarities and differences. From this plot (see Appendix B.3 for detailed code), we are able to identify which state has the most fires of each class, and which state dominates in each class. This result follows

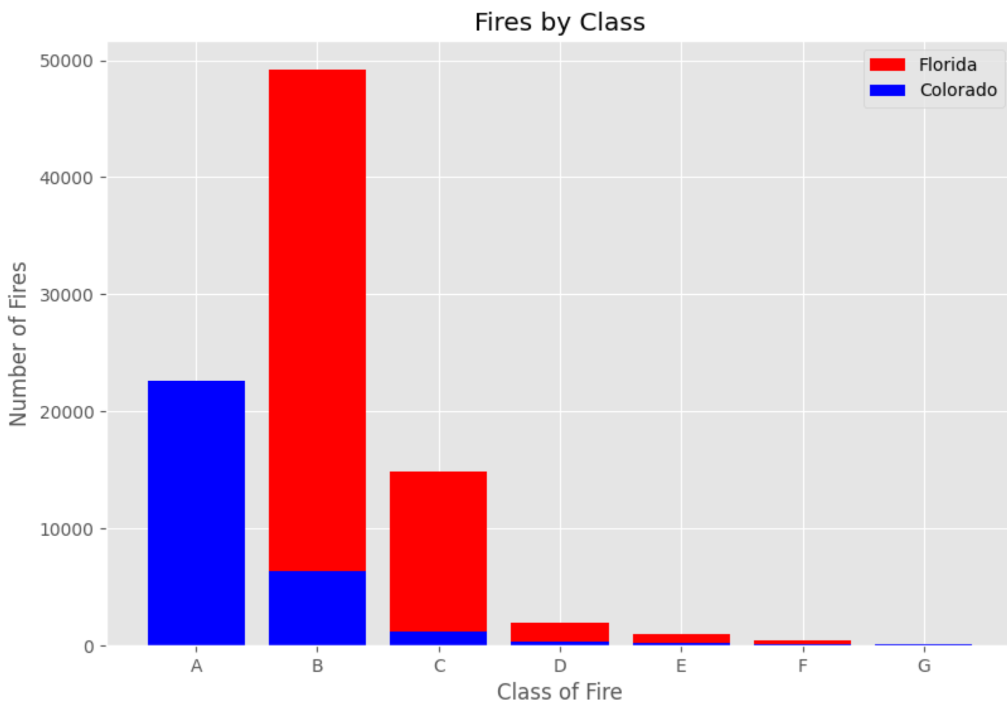

Figure 2: Number of Fires in Colorado and Florida by Class

This tells us that Colorado has significantly more A and G level fires, but Florida has more of every other class. Therefore, combining the information from this plot with the one previously we would think that Colorado is having less, but size of those small number of fires are much larger.

Next, we want to begin incorporating the regions in each state. So, rather than combining the plots of average fire size, we will now separate them by state. Then, within each plot of a state, we will plot the average fire size again, but this time by the region it appears in. This is communicated as follows

```
SELECT DATE_PART('doy', disc_date) as day_of_year, avg(fire_size), eco_region
FROM fires as f, regions as r
WHERE f.state=%s
AND f.fpa_id = r.fpa_id
GROUP BY day_of_year, eco_region
ORDER BY day_of_year
```

The query will behave the same way as our previous once, except before finding the aggregate average, it will merge the `fires` table with the provided `eco_region` located in the `regions` table. Again, it is a prepared query in order for each state to be handled separately with Python.

The plots below are the result of plotting the average fire size of an eco region each day of the year. The left figure displays the two level one eco-regions found in Florida, while the right-hand figure is the three regions found in Colorado (see the code in Appendix B.4 for more details).

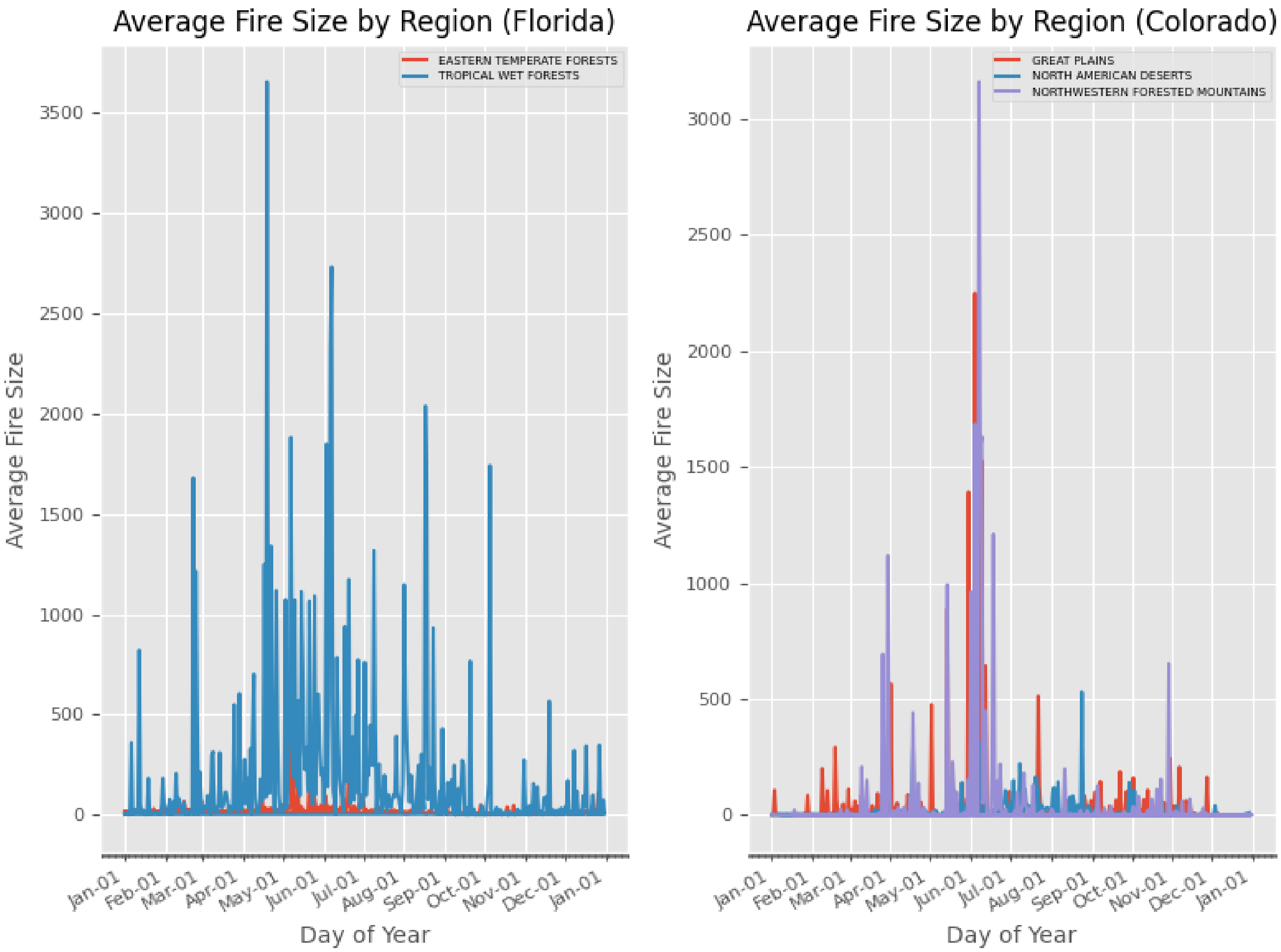

Figure 3: Average Fire Size by Region in Colorado and Florida

We see that the TROPICAL WET FORESTS tend to have drastically larger fires than its counterpart. It would be interesting to further see how the containment is handled within each of these areas. However, since the reporting times are tainted, it was too difficult to get an accurate result. Colorado has relatively similar spikes and fire sizes for each region, but still has clear order to which stands out more than another. One similarity throughout both of the plot presented here in this paper are the jumps in the summers. This should be identified as the source of issue, but it also amounts to more activities in the summer outside.

3 Takeaways

The most noticeable conclusion from working within this database to gather results was how much less time it took to do analysis. By separating the objective into two tasks, getting the required data, and then using Python to work with it reduced the time significantly. Rather than dealing with an entire data set each run time, the program is able to subset down to just the necessary pieces.

Appendix

A Variable Information

A.1 Fires Data

Variable Name	Description	Example
`fpa_id`	Unique identifier that contains information necessary to track back to the original record in the source dataset	FS-1418940
`fire_name`	Name of the incident, from the fire report (primary) or ICS-209 report	HATCH
`discovery_time`	Time of day that the fire was discovered or confirmed to exist (HHMM)	1300
`stat_cause_descr`	Description of the cause of fire	Lightning
`stat_cause_code`	Numerical encoding of `stat_cause_descr`	1
`cont_time`	Time of day that the fire was declared contained or otherwise controlled (HHMM)	1530
`fire_size`	Estimate of acres within the final perimeter of the fire	0.5
`fire_size_class`	Code for fire size based on the number of acres within the final fire perimeter expenditures	B
`latitude`	Latitude for location of fire	39.29222222
`longitude`	Longitude for location of fire	-105.18305556
`state`	2 letter state code for location of fire	CO
`county`	County name in form of "state, county"	colorado,douglas
`disc_date`	Date a fire was first discovered (YYYY-MM-DD)	2005-06-14
`cont_date`	Date a fire was reported contained or controlled (YYYY-MM-DD)	2005-06-14
`season`	Season in which the fire occurred (winter, spring, summer, fall)	summer (JJA)
`human`	Binary variable deciding whether a fire was human-caused	0

A.2 Regions Data

Variable Name	Description	Example
`fpa_id`	Unique identifier that contains information necessary to track back to the original record in the source dataset	FS-1418933
`state`	2 letter state code for location of fire	WY
`eco_region`	Level one eco-region	NORTHWESTERN FORESTED MOUNTAINS
`eco_id`	Numerical encoding of `eco_region`	6

B Code

B.1 Preprocessing

Data Preprocessing (R)

```r
rm(list=ls())
library(tidyverse)
library(reshape2)

load('../data/my_fires.rda')
fires_with_regions <- readRDS('../data/fires_clean.rds')

my_data <- my_data %>%
  filter(STATE != 'IN')
my_fires <- my_data %>%
  select(-c('FIRE_CODE', 'OWNER_CODE', 'OWNER_DESCR'))
my_fires$date <- as.Date(my_fires$date)
my_fires$cont_date <- as.Date(my_fires$cont_date)

# work around to replace blank cells with common NA
ix <- which(my_fires$CONT_TIME=='')
my_fires$CONT_TIME[ix] <-1
my_fires$CONT_TIME[ix] <- NA

regions <- fires_with_regions %>%
  filter(STATE %in% c('FL', 'CO', 'WY', 'AL', 'DE', 'CT')) %>%
  select(c('FPA_ID', 'eco1', 'STATE')) %>%
  mutate(colsplit(eco1, ' ', c('eco_id', 'eco_region'))) %>%
  select(-eco1)
```

Importing Data into Database (SQL)

```sql
DROP TABLE IF EXISTS fires, regions;

CREATE TABLE regions(
   fpa_id text PRIMARY KEY,
   state char(2),
   eco_id integer,
   eco_region text
);

CREATE TABLE fires(
   fpa_id text PRIMARY KEY,
   fire_name text,
   discovery_time integer,
   stat_cause_code integer,
   stat_cause_descr text,
   cont_time integer,
   fire_size decimal,
   fire_size_class char(1),
   latitude decimal,
   longitude decimal,
   state char(2),
   county text,
   disc_date date,
   cont_date date,
   season text,
   human integer
);

\COPY regions FROM '../data/eco_regions.csv' DELIMITER ',' CSV HEADER
\COPY fires FROM '../data/fires.csv' DELIMITER ',' CSV QUOTE '"' HEADER NULL 'NA'
```

B.2 Average Fire Size by Day of the Year

Function for Average Fire Size—state only (Python)

```python
def get_fires(cursor, state_name):
    query = f"""SELECT DATE_PART('doy', disc_date) as day_of_year, avg(fire_size)
                FROM fires
                WHERE state=%s
                GROUP BY day_of_year
                ORDER BY day_of_year"""
    try:
        cursor.execute(query, (state_name, ))
        table = [row for row in cursor.fetchall()]
        columns = [col[0] for col in cursor.description]
        df = pd.DataFrame(table, columns=columns)
        # results = df.describe(include='all').iloc[0:2, :]
        # results.loc['na_count'] =  df.isna().sum()
    except pg8000.Error as e:
        messagebox.showerror('Database error', e)

    df['day_of_year'] = pd.to_datetime(df['day_of_year'], format='%j').dt.strftime('%b-%d')

    return df

def plot_fires(cursor):
    fl_fires = get_fires(cursor, 'FL')
    co_fires = get_fires(cursor, 'CO')

    days = mdates.DayLocator()
    months = mdates.MonthLocator()
    month_fmt = mdates.DateFormatter('%b-%d')

    fig1 = plt.figure(figsize=(8,6))
    ax1 = fig1.add_axes([0.1,0.15,0.9,0.75])
    ax1.plot(fl_fires['day_of_year'], fl_fires['avg'], label='Florida', color='red')
    ax1.plot(co_fires['day_of_year'], co_fires['avg'], label='Colorado', color='blue')
    ax1.set_xlabel('Date')
    ax1.set_ylabel('Average Fire Size')
    ax1.set_title('Average Fire Size by Day of Year')
    ax1.legend(loc=0)
    ax1.xaxis.set_major_locator(months)
    ax1.xaxis.set_major_formatter(month_fmt)
    ax1.xaxis.set_minor_locator(days)
    ax1.format_xdata = mdates.DateFormatter('%b-%d')

    fig1.autofmt_xdate()
    fig1.savefig('../figures/fires.png')
    plt.show()
```

B.3 Number of Fires by Class

Function for Binning Fires by State and Class (Python)

```
def get_fires_class(cursor, state_name):
    query = f"""SELECT fire_size_class, count(*)
                FROM fires
                WHERE state=%s
                GROUP BY fire_size_class
                ORDER BY fire_size_class"""
    try:
        cursor.execute(query, (state_name, ))
        table = [row for row in cursor.fetchall()]
        columns = [col[0] for col in cursor.description]
        df = pd.DataFrame(table, columns=columns)
        # results = df.describe(include='all').iloc[0:2, :]
        # results.loc['na_count'] =  df.isna().sum()
    except pg8000.Error as e:
        messagebox.showerror('Database error', e)

    return df

def plot_fires_class(cursor):
    fl_fires_class = get_fires_class(cursor, 'FL')
    co_fires_class = get_fires_class(cursor, 'CO')

    fig2 = plt.figure(figsize=(8,6))
    ax2 = fig2.add_axes([0.1,0.1,0.9,0.8])
    ax2.bar(fl_fires_class['fire_size_class'], fl_fires_class['count'], label='Florida', color='red')
    ax2.bar(co_fires_class['fire_size_class'], co_fires_class['count'], label='Colorado', color='blue')
    ax2.set_xlabel('Class of Fire')
    ax2.set_ylabel('Number of Fires')
    ax2.set_title('Fires by Class')
    ax2.legend(loc=0)

    fig2.savefig('../figures/fires_class.png')
    plt.show()
```

B.4 Average Fire Size by Region

Function for Average Fire Size—state and region (Python)

```
def get_fires_regions(cursor, state_name):
    query = f"""SELECT DATE_PART('doy', disc_date) as day_of_year, avg(fire_size), eco_region
                FROM fires as f, regions as r
                WHERE f.state=%s
                AND f.fpa_id = r.fpa_id
                GROUP BY day_of_year, eco_region
                ORDER BY day_of_year"""
    try:
        cursor.execute(query, (state_name, ))
        table = [row for row in cursor.fetchall()]
        columns = [col[0] for col in cursor.description]
        df = pd.DataFrame(table, columns=columns)
        # results = df.describe(include='all').iloc[0:2, :]
        # results.loc['na_count'] =  df.isna().sum()
    except pg8000.Error as e:
        messagebox.showerror('Database error', e)

    df['day_of_year'] = pd.to_datetime(df['day_of_year'], format='%j').dt.strftime('%b-%d')

    return df

def plot_fires_regions(cursor):
    fl_fires_regions = get_fires_regions(cursor, 'FL')
    co_fires_regions = get_fires_regions(cursor, 'CO')

    days = mdates.DayLocator()
    months = mdates.MonthLocator()
    month_fmt = mdates.DateFormatter('%b-%d')

    fig3, ax3 = plt.subplots(figsize=(8, 6), nrows=1, ncols=2, dpi=100)

    ax3[0].set_title('Average Fire Size by Region (Florida)', fontsize=12)
    ax3[0].set_xlabel('Day of Year', fontsize=10)
    ax3[0].set_ylabel('Average Fire Size', fontsize=10)
    for region in fl_fires_regions['eco_region'].unique():
        ax3[0].plot(fl_fires_regions.loc[fl_fires_regions['eco_region']==region, 'day_of_year'],
                    fl_fires_regions.loc[fl_fires_regions['eco_region']==region, 'avg'], label=region)

    ax3[0].legend(loc=0, prop={'size': 5})
    ax3[0].xaxis.set_major_locator(months)
    ax3[0].xaxis.set_major_formatter(month_fmt)
    ax3[0].xaxis.set_minor_locator(days)
    ax3[0].format_xdata = mdates.DateFormatter('%b-%d')
    ax3[0].tick_params(labelsize=8)
    ax3[0].set_frame_on(True)

    ax3[1].set_title('Average Fire Size by Region (Colorado)', fontsize=12)
    ax3[1].set_xlabel('Day of Year', fontsize=10)
    ax3[1].set_ylabel('Average Fire Size', fontsize=10)

    for region in co_fires_regions['eco_region'].unique():
        ax3[1].plot(co_fires_regions.loc[co_fires_regions['eco_region']==region, 'day_of_year'],
                    co_fires_regions.loc[co_fires_regions['eco_region']==region, 'avg'], label=region)

    ax3[1].legend(loc=0, prop={'size': 5})
    ax3[1].xaxis.set_major_locator(months)
    ax3[1].xaxis.set_major_formatter(month_fmt)
    ax3[1].xaxis.set_minor_locator(days)
    ax3[1].format_xdata = mdates.DateFormatter('%b-%d')
    ax3[1].tick_params(labelsize=8)
    fig3.autofmt_xdate()
    ax3[1].set_frame_on(True)

    plt.tight_layout()

    fig3.savefig('../figures/fires_regions.png')
    plt.show()
```

References

[1] Short Karen C. *Spatial wildfire occurrence data for the United States, 1992-2015 [FPA_FOD_20170508] (4th Edition)*. 2017. DOI: 10.2737/RDS-2013-0009.4.