NAVAL POSTGRADUATE SCHOOL

OA3802: Computational Methods for Data Analytics

Final Project

Fall 2024

LCDR Alex Bedley, MAJ Jason Stisser, CAPT Gary Tyler, MAJ Conrad Urban

Contents

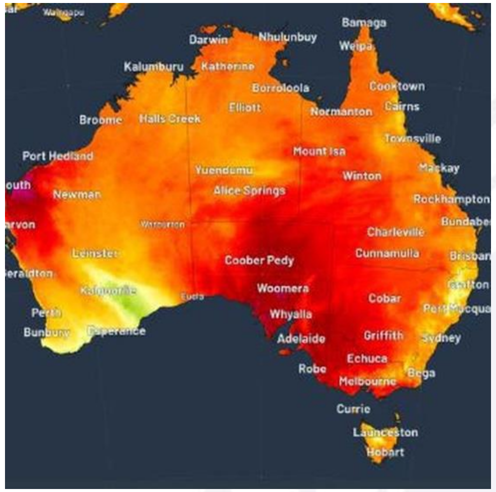
- Background and Problem Description
- Data Description and Source
- APIs, Databases and HPC
- Challenges
- Results - Leveraging Power BI

Problem Statement

This project aims to overcome the challenges of integrating **large-scale satellite vegetation land cover data** into bushfire prediction models for southeastern Australia by processing up to 300GB of complex geospatial data sourced from **AWS S3 (Simple Storage Service)** to inform better environmental and risk assessments.

https://www.istockphoto.com/photo/beautiful-shot-of-a-kangaroo-looking-at-the-camera-while-standing-in-a-dry-grassy-gm1440024914-480124057

+



https://www.9news.com.au/national/victoria-and-south-australia-weather-recordbreaking-heatwave-sweeps-across-australias-southeast/f019c1ea-3c0f-4295-b045-290bf3d38533

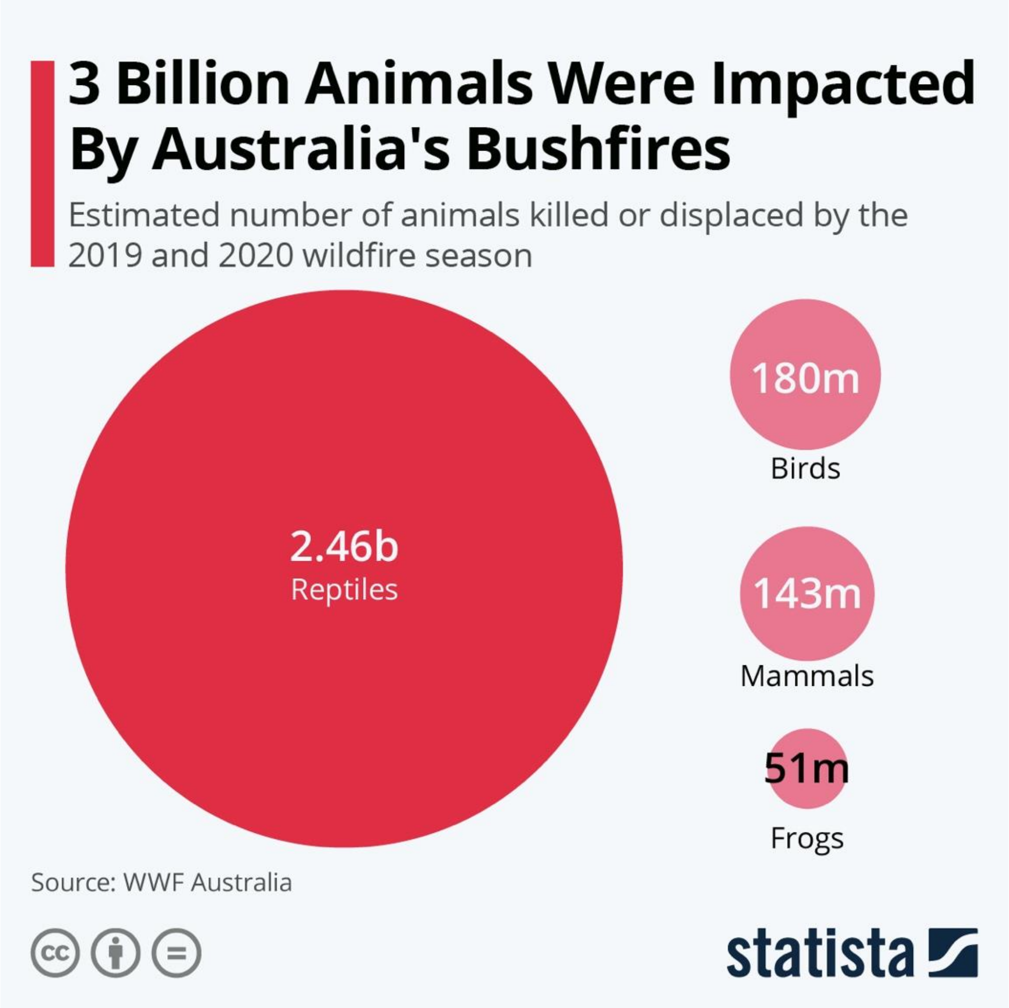
=

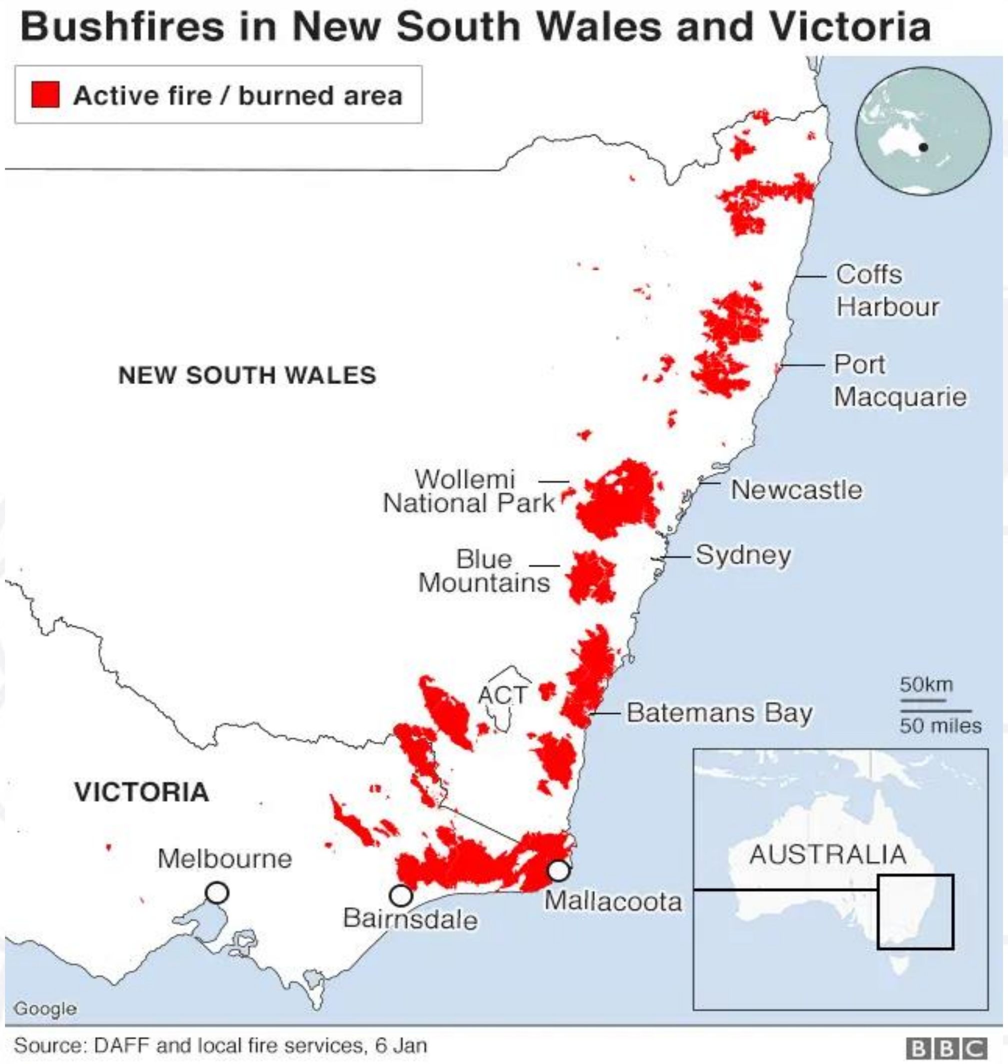
https://abcnews.go.com/International/billion-animals-estimated-dead-australia-wildfires/story?id=68143966

Only YOU can help prevent bushfire!

https://eyekoo.wordpress.com/2020/07/29/firefighter-koala-digital-painting/

Background





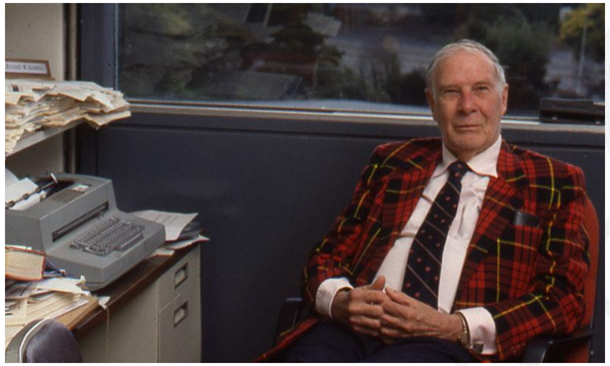
Fires on 06JAN2020 that forced mass evacuations

Tools/Resources Used

SQLite

Amazon S3

Power BI



NPS Hamming HPC

GitHub

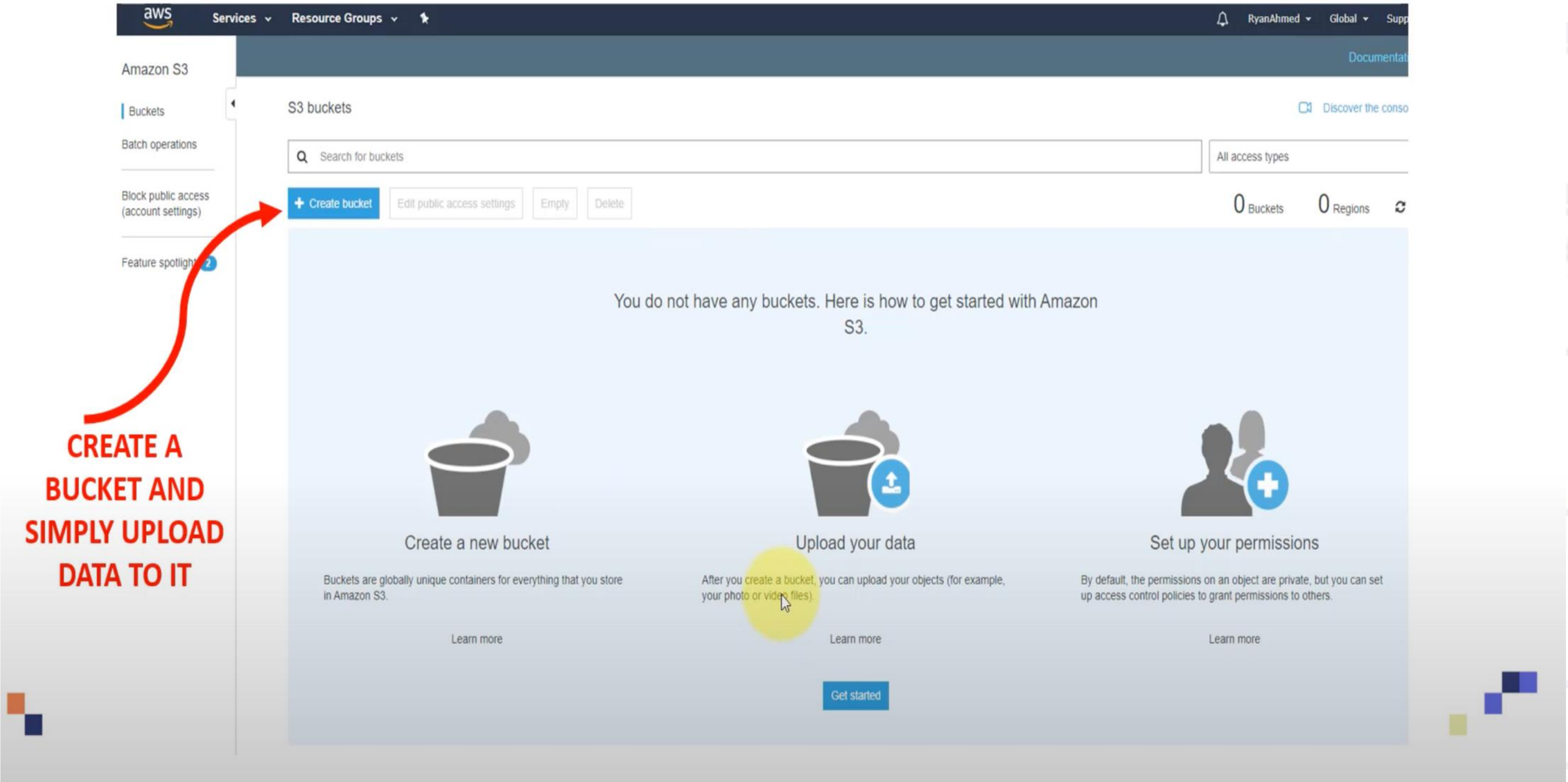
API (not the Duck one)

Australian Government
Geoscience Australia

Digital Earth
AUSTRALIA

OPEN DATA CUBE

AWS S3 (Simple Storage Solutions)



Database Comparison

Amazon Redshift

- Pay-as-you-go service ($)
- Fastest cloud data warehousing service for large datasets.
- Columnar storage and data compression.
- Queries are run against redshift storage or data stored in S3. (Amazon Redshift Spectrum)
- Uses machine learning to optimize performance.

SQLite

- Free and open-source
- Serverless, local storage
- Simple, lightweight option
- Can be stood up on NPS HPC
- Row-based storage (slower than columnar)

Story So Far

- Current Fire prediction models:
  - Short-Term Focus: Current fire prediction models forecast only 1-2 days ahead.
  - Specialized Tools: Software is highly specialized. Requires specific training and expertise
  - Resource Intensive: Computationally expensive to run.

- Links to my Thesis
  - Use machine learning and statistical techniques to develop a lightweight, user-friendly model for predicting fire risk 6-12 months ahead.
  - Aim is to provide early insights for forward planning, despite lower accuracy compared to short-term models.
  - **Something to bridge the gap** between “Summer is Bushfire Season” and specific, detailed simulation of individual fires

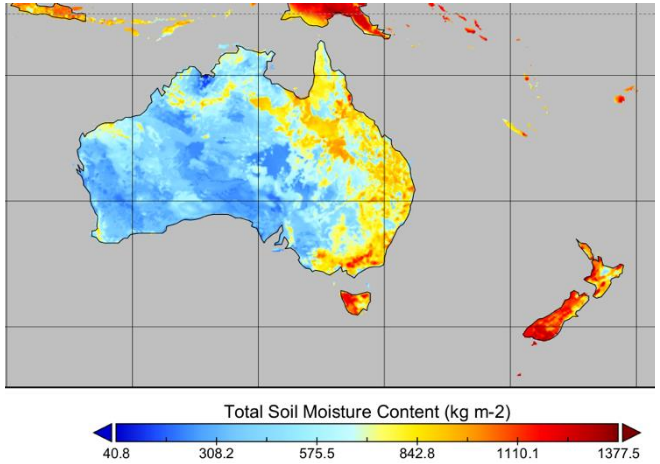
Issues

- **Method:** Split the area of interest into 10x10 km grid squares, find data relevant to fire occurrence in each square, for each time step. Do some stats (TS/ML/etc). Area and timeframe of study ~ 3M Rows (One for each Grid square/Timestep)

- **Data Availability**: Most open-source data (e.g., weather, climate indices) are user-friendly, gridded, and easy to integrate via spatial joins and filtering (standard Pandas/GeoPandas tools).

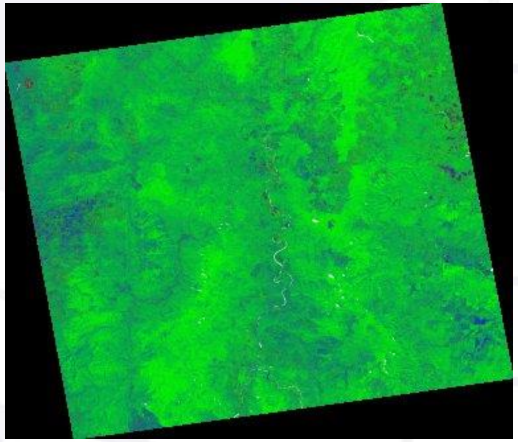
- **Challenge:** Land Cover (Vegetation) data, critical for understanding vegetation types, is less accessible and harder to incorporate. *The curse of the .TIFF file*

NETCDF – 10km/monthly Resolution. One file shows all of Oceania (300KB)



VS

TIFF – 30m^2 annual (or daily) resolution. Each file shows 100km^2 (4-10MB)

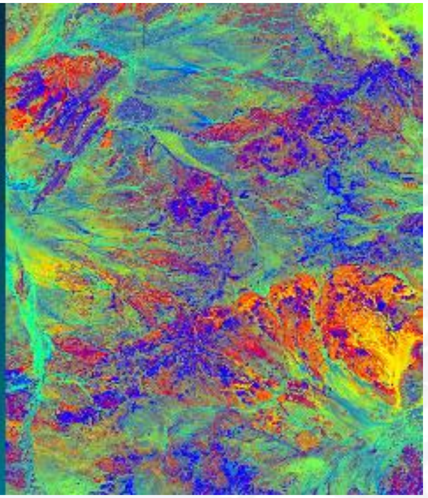


Data

DEA Fractional Cover Percentiles (Landsat)

Geoscience Australia Landsat Fractional Cover Percentiles Collection 3

Version:	4.0.0 (Latest)
Product types:	Derivative, Raster
Time span:	1987 – Present
Update frequency:	Yearly
Product ID:	ga_ls_fc_pc_cyear_3



- Digital Earth Australia has vast data repositories of satellite imagery – preprocessed for specific uses
- We are interested in Fractional Cover Percentiles
- Represent the 10th, 50th, and 90th percentiles of green vegetation, non-green vegetation, and bare soil cover each year.
- Naming conventions:
  - BS = Bare Soil
  - PV = Photosynthetic Vegetation (i.e. Green)
  - NPV = Non-Photosynthetic Vegetation (i.e. dry or dead)

- Hypothesis – Lots of previously green vegetation that is now dry increases risk of bushfire

Why is this a challenge?

- TIFF files are very large
  - 4-10MB each (x 30 000 files)
  - ~300GB for Area of Interest.
- Spatial joins across **multiple image files** – uses up **Memory** very quickly
- Takes **time.** Sequentially joining over 3M rows takes ~forever
- Converting **CRS** projections is problematic (Buffers and Error)
- Local **storage** of this much data (my surface only has 250GB total)

Solution?

Enlist the help of your Comp 3 Team!

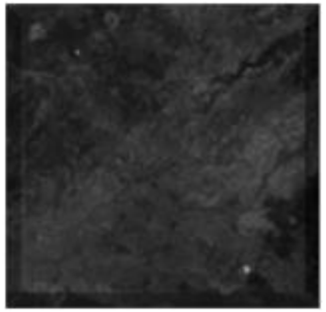
Die Wunder der deutschen Ingenieurskunst

Data

Raw Data

Australian Government
Geoscience Australia
Digital Earth AUSTRALIA

- Massive Data Set
- Curated by Australian Government
- TIFF files
- Different API's



Tasks:

1. Get the data correct
2. Convert the data

Size: ~300 GB

OPEN DATA CUBE

ODC
–
AWS

Amazon S3

SQL Database

```
$sqlite3 fire_trun.db "SELECT COUNT(*) FROM ground_data;"
18048

$sqlite3 fire_trun.db "PRAGMA table_info(ground_data);" | wc -l
22

$du -h fire_trun.db
3.7M
```

SQL Lite DB

- Created a base from data tiles of southeast Australia
- Truncated set on narrow region
- Range 1987 - 2018

Tasks:

1. Create database from original file
2. Add nine (9) columns for new predictors
3. Import data via ODC API
4. Establish db on truncated set

Size: 500MB (truncated to 3.7MB)

API Walkthrough

- Take current data (from Gary) from csv to database – commit
- (allows us to pull just from the grids we are interested in)
- (build reference frame we want to call)
- Searching the API with pystac for datasets – STAC (SpatioTemporal Asset Catalog)
- Using the ODC API to retrieve data

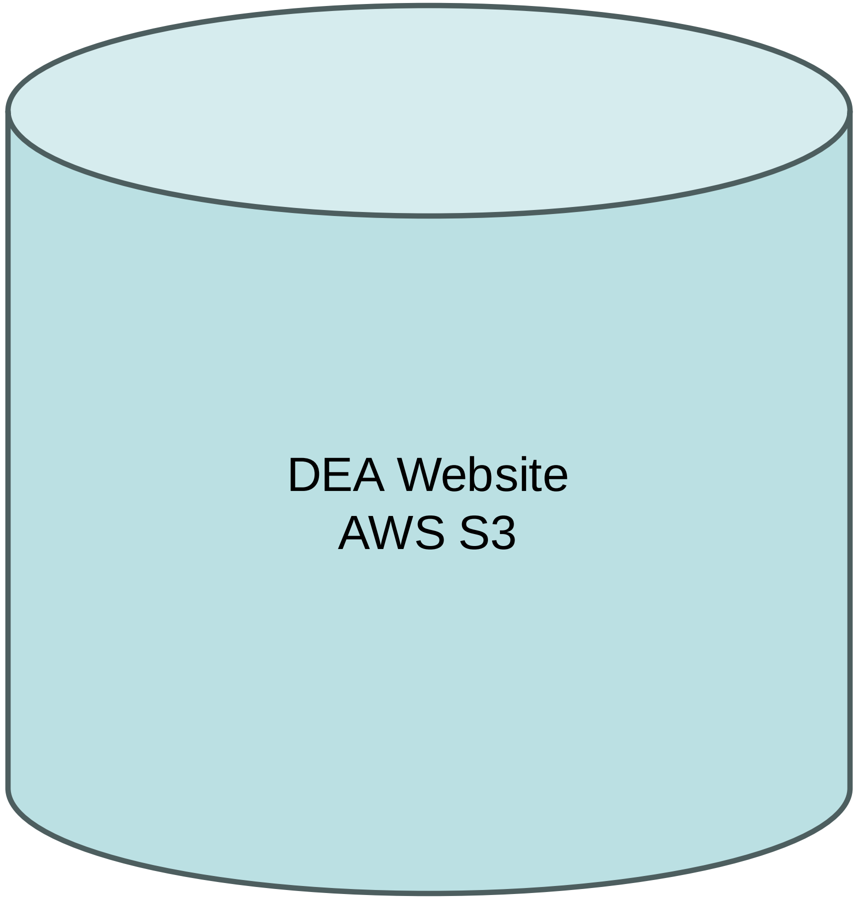
“The Open Data Cube (ODC) is a free, open-source software package that simplifies the management and analysis of large amounts of satellite imagery and other Earth observation data. It allows users to easily access, process, and analyze decades of geographical data to track changes on Earth's surface over time. ODC is designed to help scientists, researchers, and government agencies make better-informed decisions in areas such as environmental issues, land use, and resource management.”

ODC Partners. Source: opendatacube.org

Data

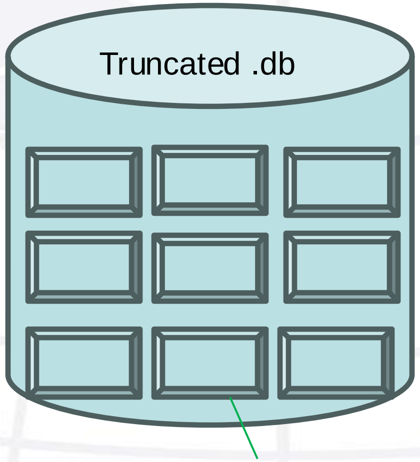
DEA | **Hamming** | **Usable**

We began with a massive data set on an open-source site, pulled the relevant data into a SQLite database via Hamming HPC, then called a usable chunk of data (i.e. a set date and area range) as a .csv.



<~300 GB>

<550 MB>



<3.7 MB>

Challenges I

Issue: *Pulling Data*
Discussion: Data files are large
Solution: Used API to pull data

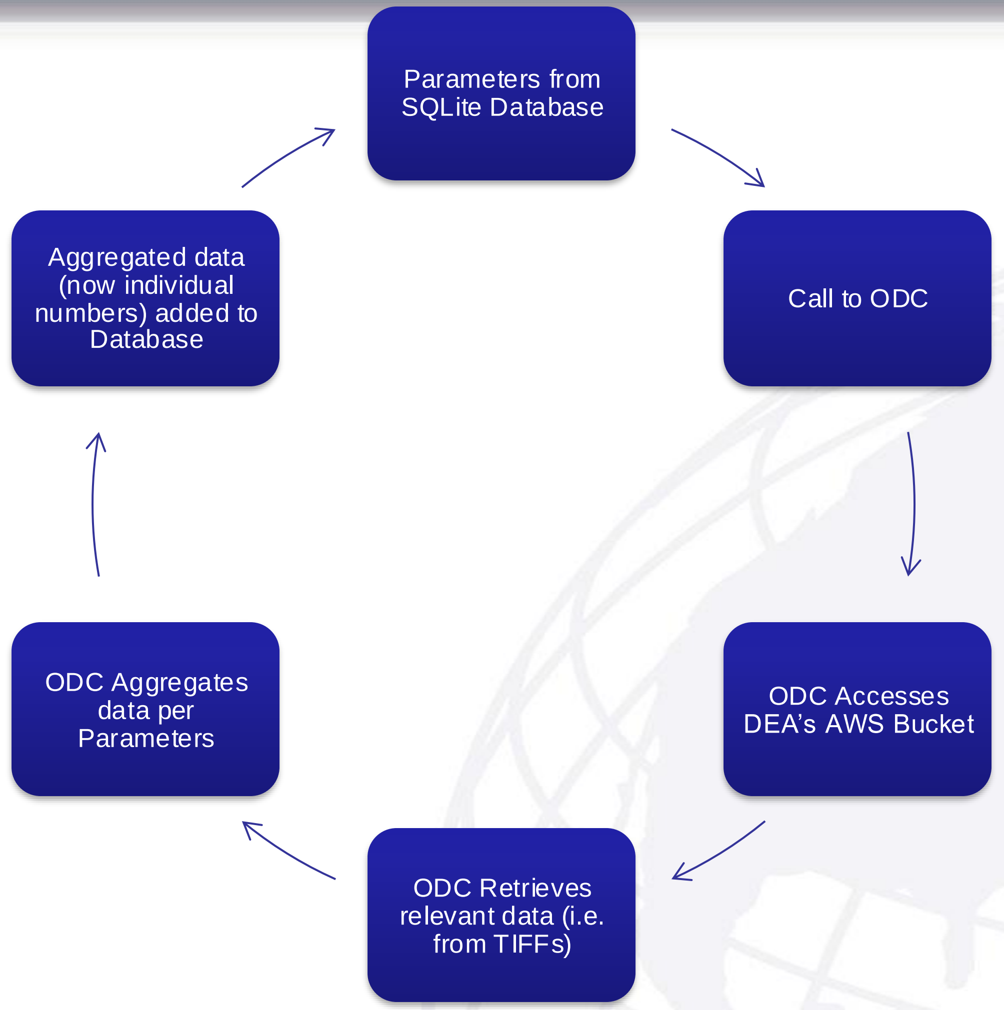
Issue: *Pulling Data*
Discussion: Dimensions for data don't match those of the project
Solution: With API, able to pull based on Lat/Lon, making it easier to convert

Issue: *Data Consistency*
Discussion: How can we all work on the same Dataset
Solution: Set up a group on Hamming and store the data in a SQL database

Issue: *Processing Data*
Discussion: Even with refinement, this is a lot to pull and process
Solution:
- Leverage Hamming HPC to run Python script for calling the API and importing the data to the SQL server
- Parallelize the code
- Add error handling

https://www.istockphoto.com/photos/pulling-hair-out-computer

Code Review



- Supports both sequential and parallel processing modes
- Currently setup to initialize from csv but can handle database/parquet etc

ODC Specifics

```
# Initialize STAC catalog and configure AWS connection
catalog = pystac_client.Client.open("https://explorer.dea.ga.gov.au/stac")
odc.stac.configure_rio(
    cloud_defaults=True,
    aws={"aws_unsigned": True},
)
```

Establish connection to AWS Server – The STAC Catalog contains all publicly availably satellite data within DEA's Bucket

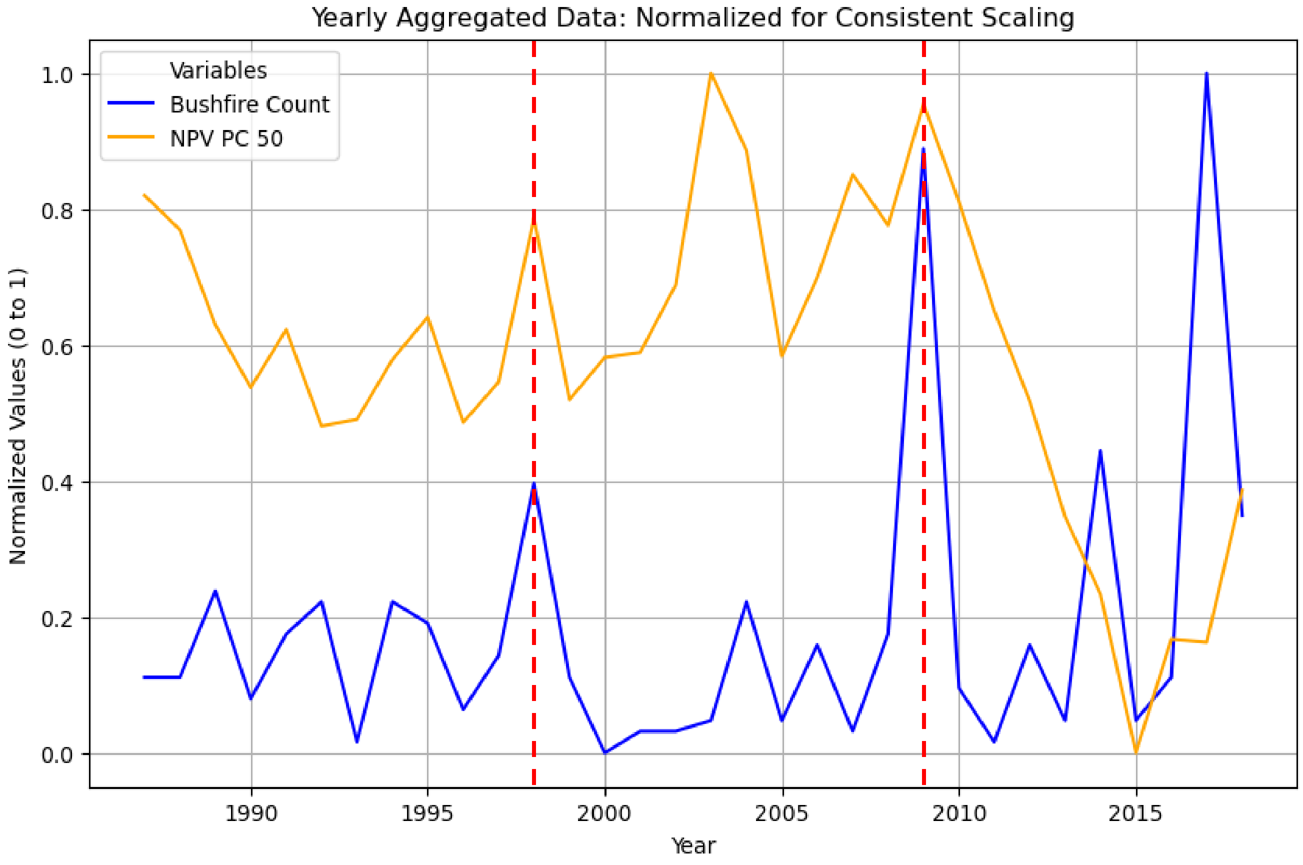
```
try:
    # Calculate bounding box for the tile
    bbox = [data[2] - 0.05, data[1] - 0.05, data[2] + 0.05, data[1] + 0.05]
    start_date = f"{data[0][:4]}-01-01"
    end_date = f"{data[0][:4]}-12-31"

    # Query STAC catalog
    query = catalog.search(
        bbox=bbox,
        collections=["ga_ls_fc_pc_cyear_3"],
        datetime=f"{start_date}/{end_date}",
    )

    items = list(query.items())
    if not items:
        print(f"No data found for {start_date}-{end_date}, Location: {data[1]}, {data[2]}")
        return None

    # Load and process data
    ds = odc.stac.load(
        items=items,
        crs="EPSG:3577",
        lat=(bbox[1], bbox[3]),
        lon=(bbox[0], bbox[2]),
        time=(start_date, end_date)
    )

    # Calculate means for all percentiles
    return {
        'time': data[0],
        'grid_id': data[3],
        'bs_pc_10': float(ds.bs_pc_10.mean().values),
        'bs_pc_50': float(ds.bs_pc_50.mean().values),
```

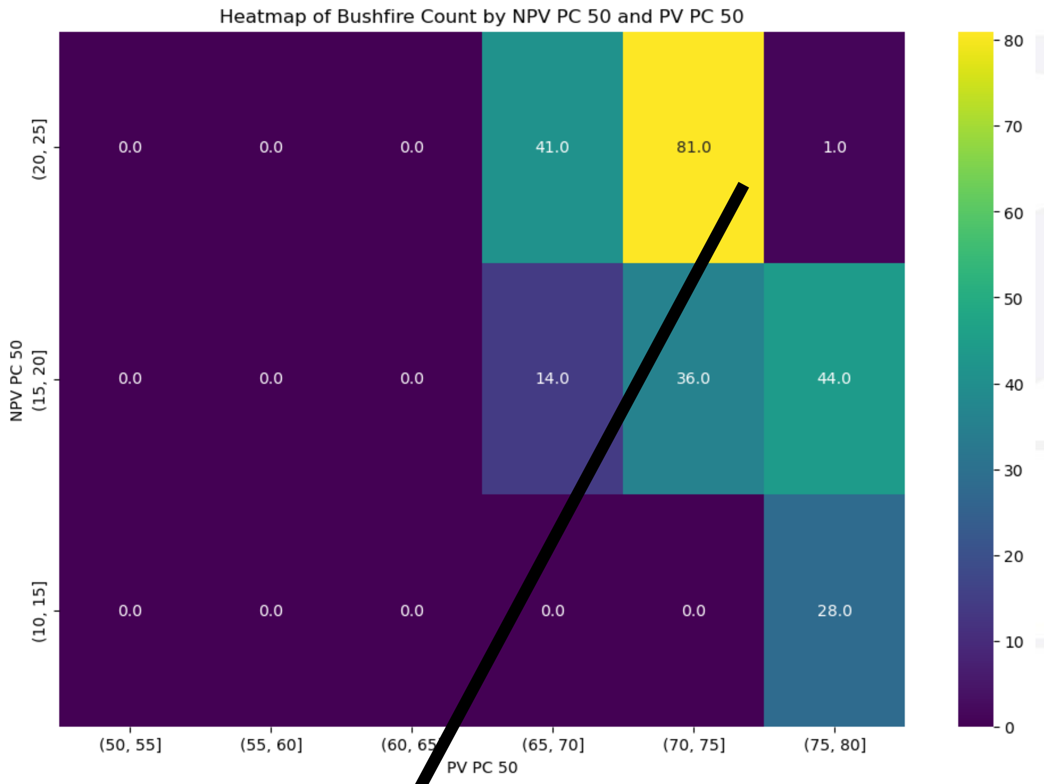
Define time and space of interest (i.e. what row to update)

Look up the data of interest

Load and aggregate the data, ready to update database

Results





Sweet Spot for Fuel Load

The Power of PowerBI

Example PowerBI Dashboard

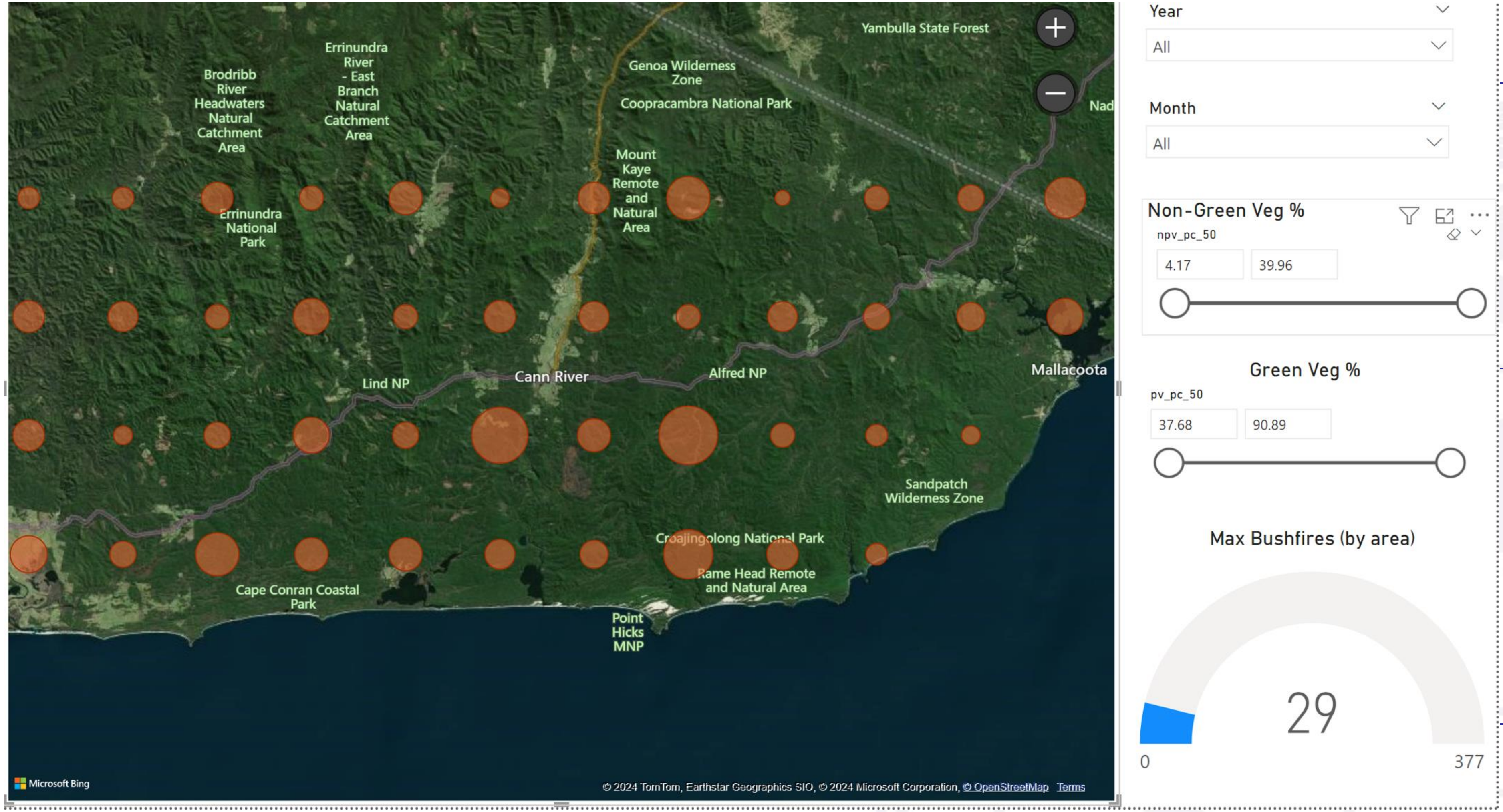
Drag and Drop Graphs

Data Direct to Maps

"Slicers" to Drill Down

Tile Cards to Display Summary

Adding Vegetation Coverage



Modify slicers based on how you want to see the data

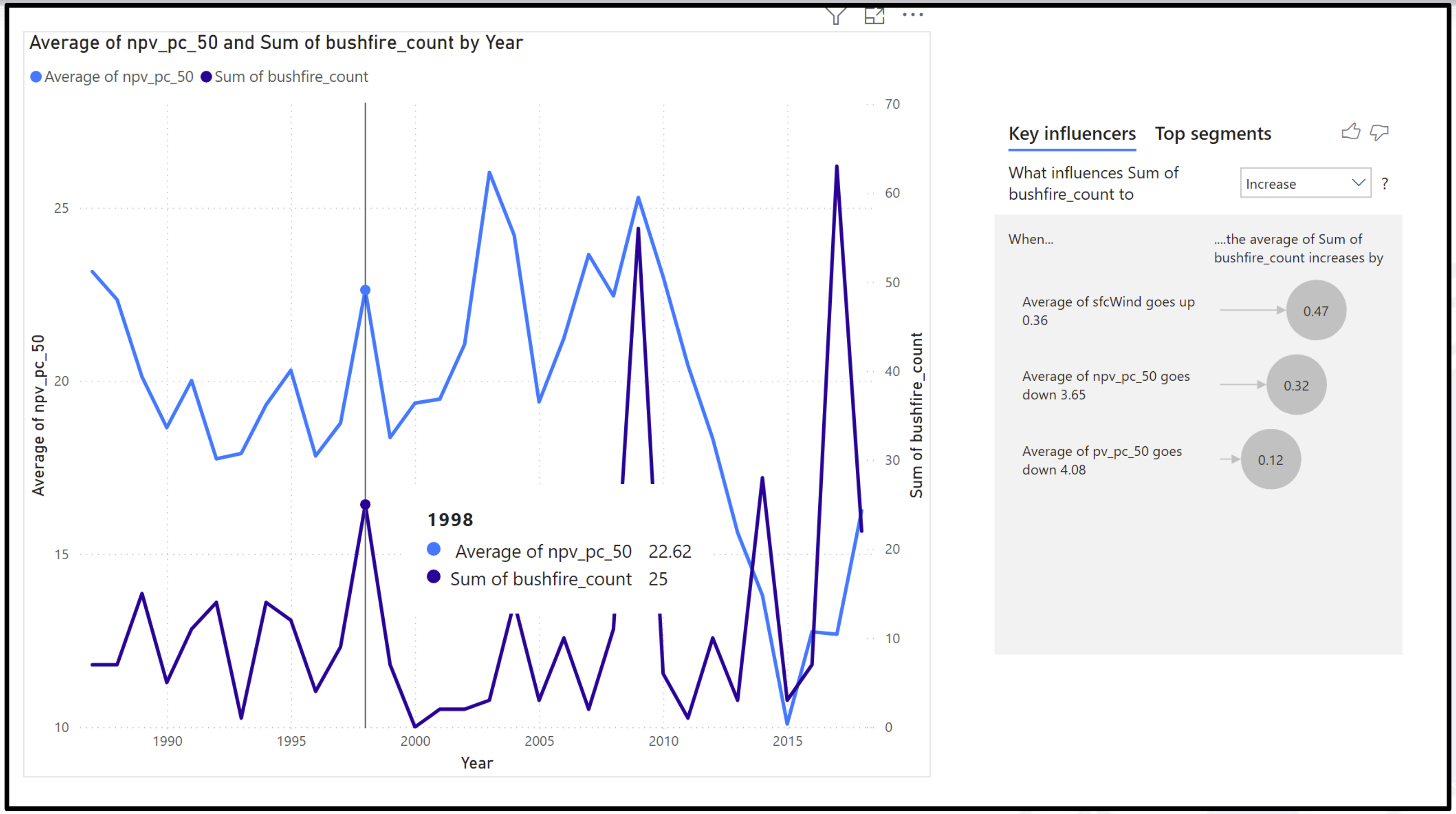
Easily add new filters to analyze data with the complete information

CAUTION: beware dependencies and data types!

Additional visualization tools available

NAVAL POSTGRADUATE SCHOOL

Remember Our Python Chart?



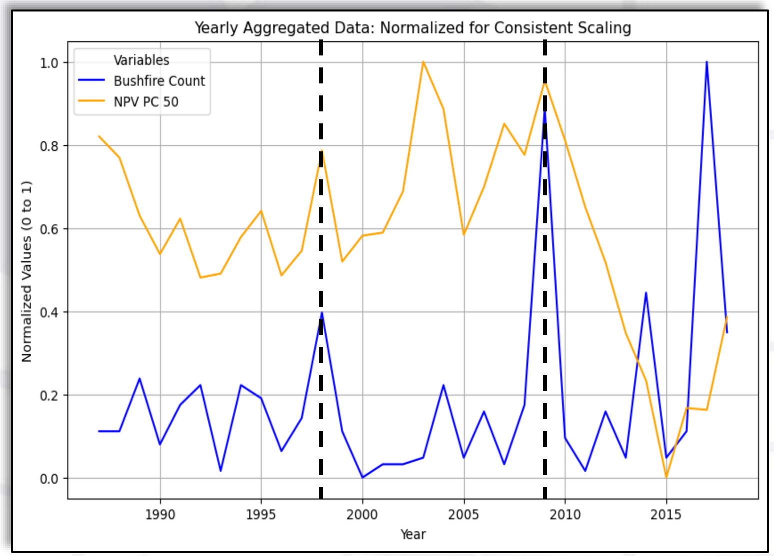
Single Tab in PowerBI Project

Strengths

- Customizable
- GUI
- Integration
- Data Sources

Weaknesses

- Desktop Application
- Platform Restrictions
- Sensitivity to Data



Python

Future Work

- Extend API data retrieval method to full dataset
- Incorporate Fractional Cover Percentiles into predictive models
- Predict some fires
- Explore cloud resources, including tutorials, on DEA and National Computational Infrastructure (NCI). Free for approved academic purposes.

Conclusion/Check on Learning

- Background and Problem Description
- Data Description and Source
- APIs, Databases and HPC
- Challenges
- Results - Leveraging Power BI

https://in.pinterest.com/pin/618682067553345972/

NAVAL
POSTGRADUATE
SCHOOL

Questions