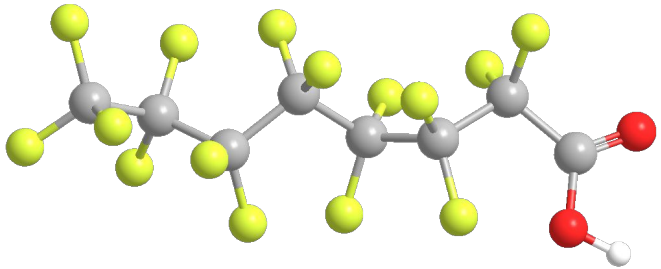
NC STATE UNIVERSITY

IDENTIFYING PFOA SAMPLE LOCATIONS SURROUNDING THE WASHINGTON WORKS CHEMICAL MANUFACTURING PLANT USING REMOTE SENSING



BY: NICHOLAS GROKHOWSKY

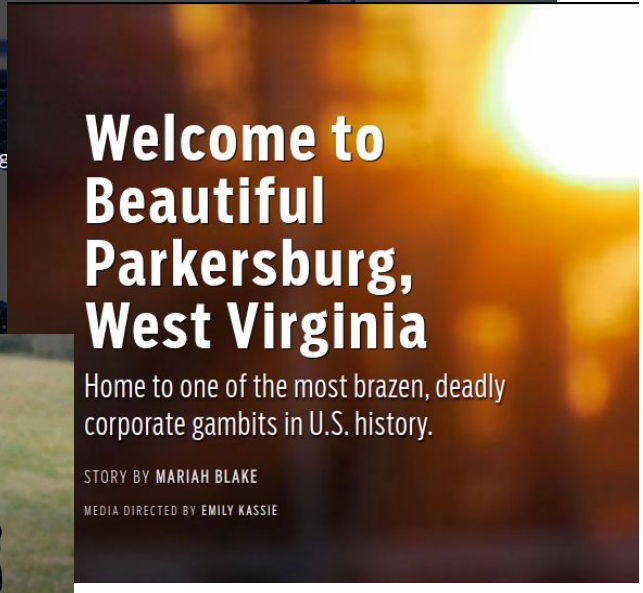
U.S. Department *of* Defense

News Spotlights About

SPOTLIGHT

PFAS: A National Issue That Needs National Solutions

ny industrial and consumer
nclude keeping food from sticking
mattresses more waterproof.



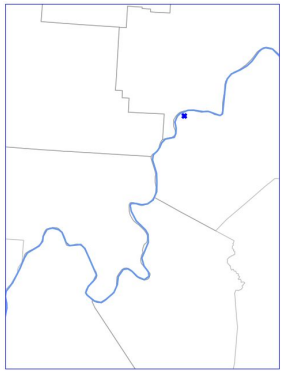
https://highline.huffingtonpost.com/articles/en/welcome-to-beautiful-parkersburg/
https://www.google.com/url?sa=i&url=https%3A%2F%2Fchemtrust.org%2Fdark-waters-and-pfoa-faq%2F&psig=AOvVaw2WP5g-HnODMnxXDQJskZIz&ust=1633444823973000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCODB1KX-sPMCFQAAAAAdAAAAABAN
https://www.nytimes.com/2016/01/10/magazine/the-lawyer-who-became-duponts-worst-nightmare.html
https://www.defense.gov/Spotlights/pfas/

WASHINGTON WORKS PLANT

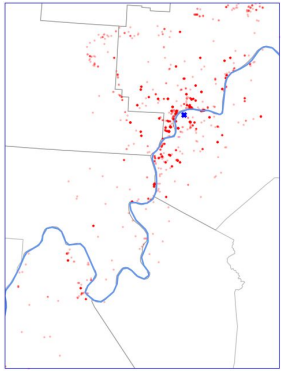
https://www.epa.gov/hwcorrectiveactionsites/hazardous-waste-cleanup-chemours-company-fc-llc-formerly-dupont-washington

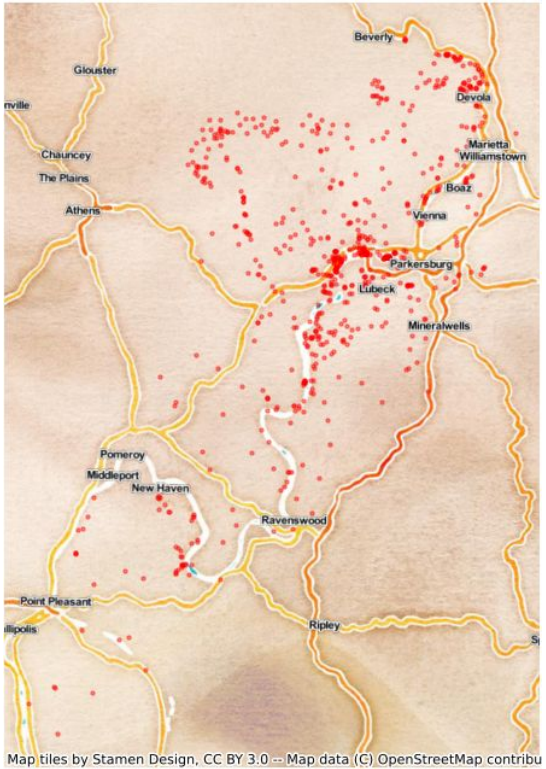
RAW DATA MAPS

WASHINGTON WORKS FACILITY



PFOA OBSERVATIONS





DATA FILTER

RAW DATA SIZE	7,038
RAW DATA UNIQUE LOCATIONS	749
FILTERED DATA SIZE	1,944
PRE-TREATED OBSERVATIONS	1,823

```
84   # Filter out un-useable values
     data = data[data.purpose != "DUP"]
     data = data[data.detected == "Y"]
     data = data[data.result != "NQ"]
     data = data[data.easting != ""]
     data = data[data.northing != ""]
     data = data[data.easting != " "]
     data = data[data.northing != " "]
     data = data.dropna(subset=["easting", "northing"])
```

REMOVE DUPLICATES
REMOVE NON-DETECTED
REMOVE NOT-QUALIFIED

REMOVE EMPTY COORDINATES

DATA FILTER

IDENTIFY PRE-TREATED VALUES:

```
143  nulls = pd.isnull(pd.Series(data["treatment"]))
144  data["treatment"] = data["treatment"].astype(str)
145  data["treatment"] = [ data.loc[i, "treatment"].strip() for i in range(0, len(data["treatment"])) ]
146  for i in range(0, len(nulls)):
147      if nulls[i] == True:
148          subset = data[data.date == data.date[i]]
149          subset = subset[subset.month == data.month[i]]
150          subset = subset[subset.distance == data.distance[i]]
151
152          # Maybe create the condition if another "PT" value is present?
153
154          if subset["result"].max() == data.loc[i, "result"]:
155              data.loc[i, "treatment"] = "PT"
156
```

THIS CONDITION IS NOW INCLUDED

IDENTIFY NULL TREATMENT VALUES

ITERATE EACH NULL VALUE

IF TREATMENT IS NULL

SUBSET ALL VALUES ON OBSERVATION YEAR, MONTH, AND DISTANCE FOR EACH NULL TREATMENT

WHICHEVER VALUE IS THE MAXIMUM FOR THE SUBSET IS LISTED AS A PRE-TREATED OBSERVATION

RAW DATA SUMMARY

UNTREATED GROUNDWATER	1,823
UNIQUE LOCATIONS	522

* According to Theresa Cantu’s analysis

SUMMARY STATISTICS

OBSERVED PFOA

n	1,823
MINIMUM	5.50 ppt
MAXIMUM	66,000 ppt
MEAN	1,259.68 ppt
GEOMETRIC MEAN	321.86 ppt
MEDIAN	320 ppt

date	n	min	max	gMean	mean	median	_1st%	_5th%	_10th%	_25th%	_50th%	_75th%	_90th%	_95th%	_99th%
2001	13	48.50	2,700.00	449.36	912.44	326.00	49.66	54.32	73.36	170.00	326.00	1,500.00	2,246.00	2,514.00	2,662.80
2002	131	56.30	22,700.00	1,083.67	2,930.54	1,210.00	64.63	82.90	99.90	274.00	1,210.00	4,020.00	8,210.00	11,300.00	17,130.00
2004	53	50.50	27,100.00	1,102.12	4,084.09	721.00	52.48	67.74	117.00	219.00	721.00	5,470.00	11,880.00	17,180.00	24,500.00
2005	126	9.90	22,700.00	548.51	2,286.98	486.50	10.95	21.23	46.25	144.75	486.50	2,222.50	7,515.00	10,725.00	16,825.00
2006	93	10.30	14,800.00	436.16	1,543.95	451.00	11.86	31.44	39.20	162.00	451.00	1,330.00	3,292.00	7,540.00	13,972.00
2007	184	13.00	10,000.00	265.92	785.41	320.00	14.66	18.15	30.60	68.25	320.00	972.50	1,800.00	2,585.00	8,423.00
2008	86	20.00	15,000.00	473.50	1,341.73	550.00	25.10	30.00	61.50	155.00	550.00	1,400.00	3,050.00	5,800.00	11,600.00
2009	91	14.00	11,000.00	354.46	919.77	510.00	14.00	26.00	36.00	105.00	510.00	1,400.00	2,000.00	2,450.00	5,150.00
2010	84	19.00	13,000.00	433.98	1,103.19	595.00	21.49	38.15	47.90	130.00	595.00	1,800.00	2,270.00	3,265.00	6,111.00
2011	108	47.00	17,600.00	726.03	1,978.53	855.00	50.07	61.80	80.10	207.50	855.00	2,025.00	4,710.00	10,000.00	15,692.00
2012	84	11.00	66,000.00	528.31	2,456.89	595.00	11.83	25.15	60.80	160.00	595.00	2,000.00	2,770.00	9,455.00	26,990.00
2013	77	6.00	8,100.00	355.25	925.77	590.00	6.30	10.40	34.00	120.00	590.00	1,500.00	1,940.00	2,320.00	5,060.00
2014	58	9.70	7,300.00	373.33	935.06	480.00	12.15	26.50	31.70	130.00	480.00	1,400.00	2,060.00	2,635.00	5,362.00
2015	54	8.30	6,100.00	348.86	823.08	490.00	9.20	15.60	29.70	140.00	490.00	1,275.00	1,770.00	2,405.00	4,351.00
2016	119	5.50	6,900.00	170.72	527.28	180.00	6.61	16.00	23.00	53.00	180.00	620.00	1,440.00	2,110.00	3,946.00
2017	183	10.00	8,600.00	135.63	415.39	96.00	11.64	16.00	25.20	50.00	96.00	355.00	1,316.00	2,070.00	3,600.00
2018	173	10.00	6,400.00	149.06	366.28	140.00	10.72	18.20	30.20	63.00	140.00	340.00	956.00	1,540.00	2,952.00
2019	60	12.00	3,000.00	104.79	285.17	61.50	13.77	28.35	42.60	54.75	61.50	202.50	761.00	1,610.00	2,351.00
2020	46	11.00	2,000.00	67.40	149.91	59.50	12.35	16.00	20.50	35.25	59.50	88.75	305.00	467.50	1,527.50

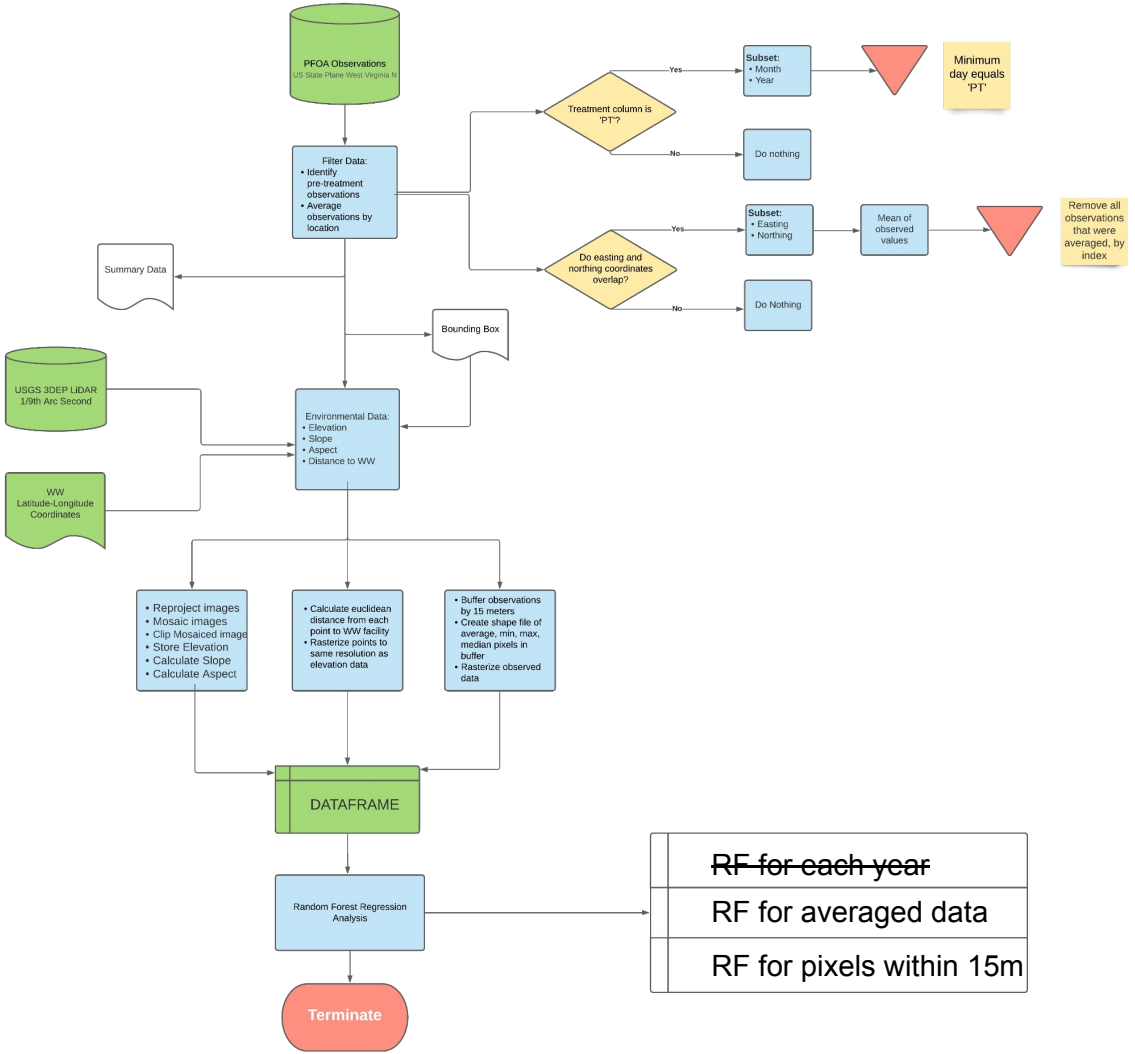
PURPOSE

IN AN EFFORT TO IDENTIFY CONTAMINATED LOCATIONS, SURROUNDING THE WASHINGTON WORKS FACILITY, OUR PURPOSE IS TO LOCATE SAMPLING AREAS WITH PFOA LEVELS HIGHER THAN THE 50 PPT THRESHOLD.

FURTHERMORE, WE WOULD LIKE TO IDENTIFY ENVIRONMENTAL CONDITIONS THAT LEAD TO HIGHER PFOA CONCENTRATIONS.

HYPOTHESIS: Environmental variables will explain locations with PFOA limits greater than or equal to 50 ppt and locations with PFOA limits less than 50 ppt

HYPOTHESIS: Environmental variables contribute to the concentration and distribution of PFOA surrounding a known contamination source



METHODOLOGY

DATA INPUTS:

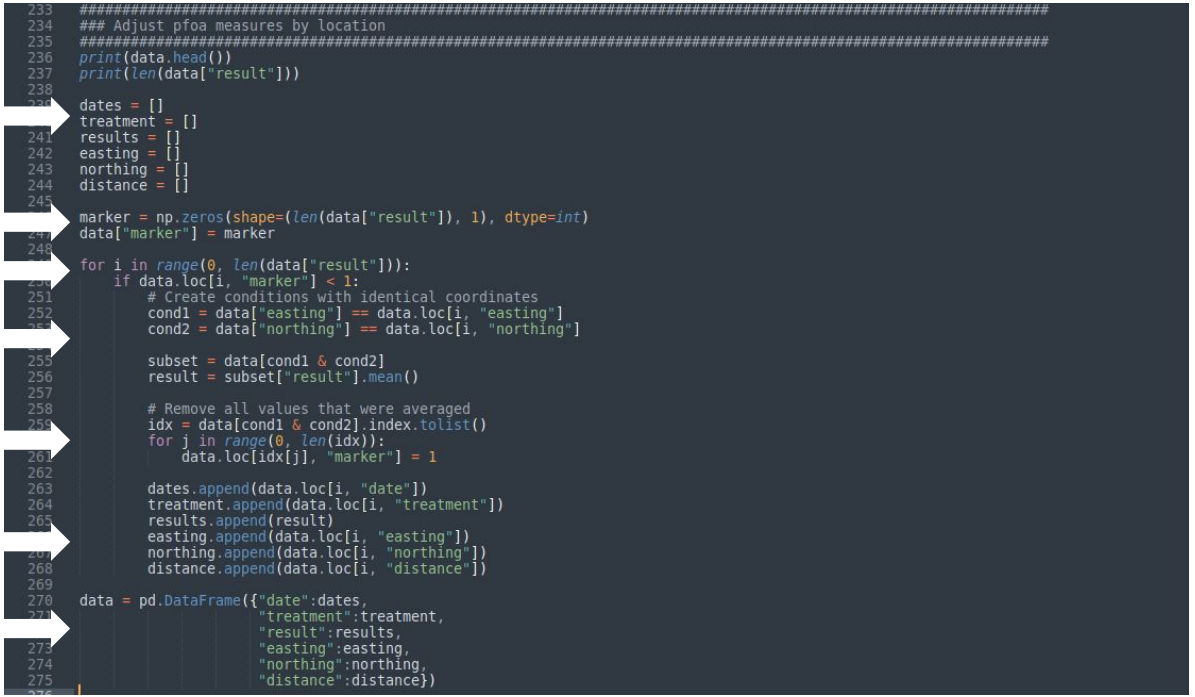
- Observed PFOA from residential wells
- 33 USGS 3DEP LiDAR DEMs at 1/9th Arc Second resolution
- Washington Works latitude-longitude coordinates

OUTPUTS:

- Dataframe with 5 columns
  - Pfoa measure
  - ~~Distance to WW~~
  - Elevation
  - Slope
  - Aspect
- Random Forest Regression & Binary Classification

METHODOLOGY

AVERAGE OVERLAPPING LOCATIONS



```
233  ##########################################################################################################
234  ### Adjust pfoa measures by location
235  ##########################################################################################################
236  print(data.head())
237  print(len(data["result"]))
238
239  dates = []
240  treatment = []
241  results = []
242  easting = []
243  northing = []
244  distance = []
245
246  marker = np.zeros(shape=(len(data["result"]), 1), dtype=int)
247  data["marker"] = marker
248
249  for i in range(0, len(data["result"])):
250      if data.loc[i, "marker"] < 1:
251          # Create conditions with identical coordinates
252          cond1 = data["easting"] == data.loc[i, "easting"]
253          cond2 = data["northing"] == data.loc[i, "northing"]
254
255          subset = data[cond1 & cond2]
256          result = subset["result"].mean()
257
258          # Remove all values that were averaged
259          idx = data[cond1 & cond2].index.tolist()
260          for j in range(0, len(idx)):
261              data.loc[idx[j], "marker"] = 1
262
263          dates.append(data.loc[i, "date"])
264          treatment.append(data.loc[i, "treatment"])
265          results.append(result)
266          easting.append(data.loc[i, "easting"])
267          northing.append(data.loc[i, "northing"])
268          distance.append(data.loc[i, "distance"])
269
270  data = pd.DataFrame({"date":dates,
271                       "treatment":treatment,
272                       "result":results,
273                       "easting":easting,
274                       "northing":northing,
275                       "distance":distance})
```

1. Create empty list variables

2. Add column of 0’s to identify if a row value was already averaged

3. Iterate each row of dataframe

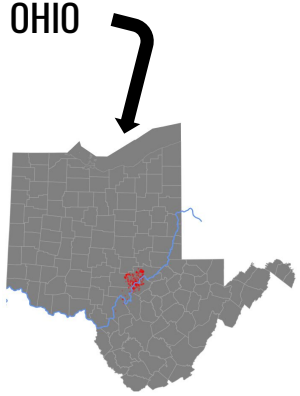
4. Subset dataframe by ‘easting’ and ‘northing’ coordinates

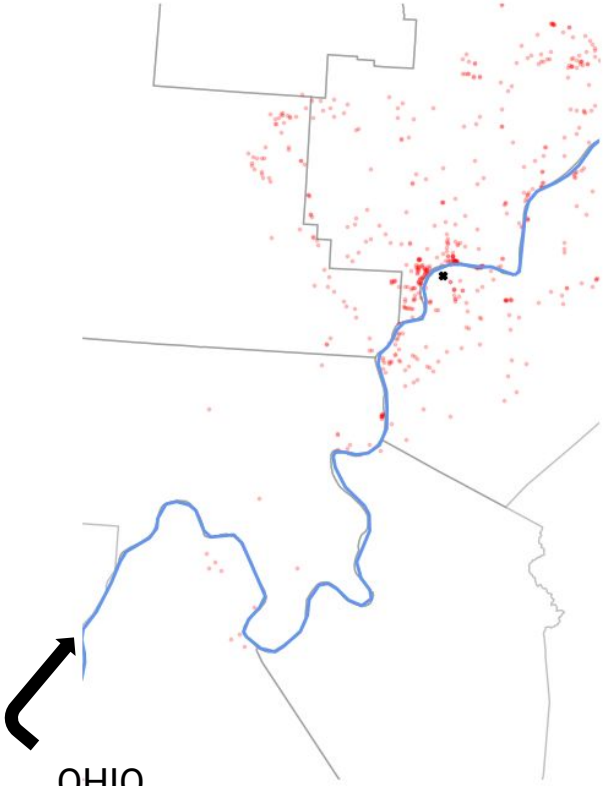
5. Calculate mean value of subset pfoa measures

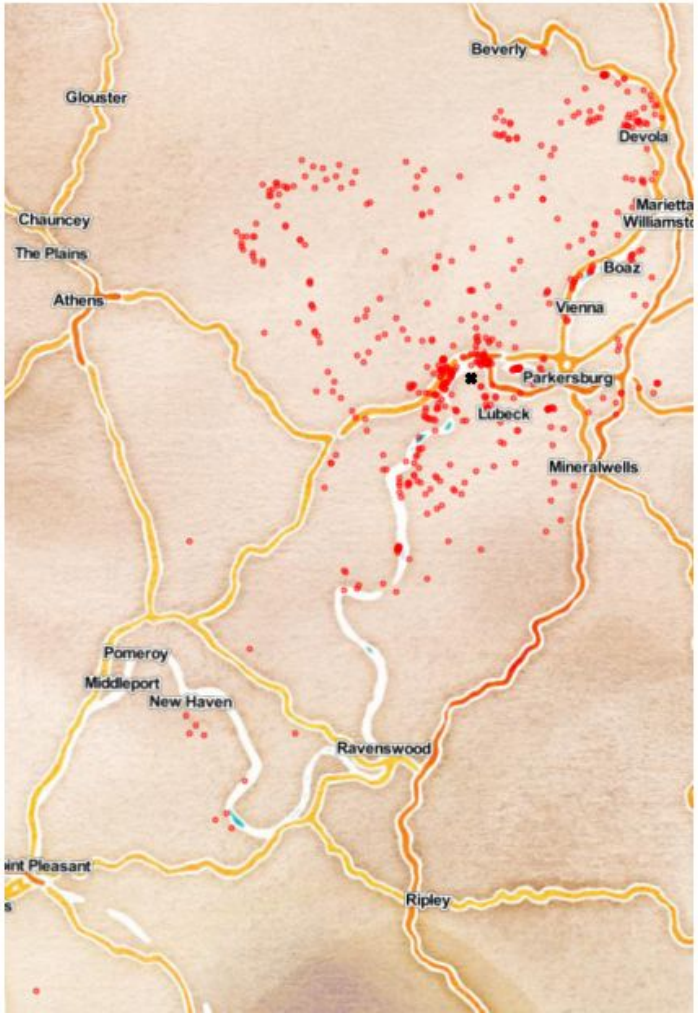
6. Update marker value for all indexes used for mean pfoa value

7. Create pandas dataframe with new, averaged values

SAMPLES USED FOR ANALYSIS







RESULT

UNTREATED GROUNDWATER	522
TREATED GROUNDWATER	NA
FISH TISSUE SAMPLES	NA
UNIQUE LOCATIONS	522

- IDENTIFY UNTREATED GROUNDWATER MEASURES
- REMOVE IMPRECISE AND MISSING COORDINATES
- AVERAGE OVERLAPPING LOCATIONS

SUMMARY STATISTICS

LOCATIONS REPRESENTED BY DATE FIRST OBSERVED

date	n	min	max	gMean	mean	median	_1st%	_5th%	_10th%	_25th%	_50th%	_75th%	_90th%	_95th%	_99th%
2001	13	47.07	2,390.00	426.36	913.03	364.00	48.47	54.07	68.32	113.02	364.00	1,671.67	1,991.75	2,171.00	2,346.20
2002	125	61.00	22,700.00	1,316.68	3,140.35	1,430.00	66.18	127.98	196.33	446.00	1,430.00	4,050.00	8,454.00	11,480.00	18,074.29
2004	9	50.50	742.92	125.86	185.26	93.67	51.47	55.35	60.20	65.94	93.67	213.64	324.18	533.55	701.04
2005	48	9.90	3,483.33	150.06	450.42	119.69	10.28	11.70	19.39	41.44	119.69	644.22	1,190.32	1,560.41	3,014.90
2006	13	10.30	2,704.55	188.20	453.59	254.09	13.73	27.46	39.64	51.96	254.09	434.00	768.26	1,591.22	2,481.88
2007	91	13.00	2,773.33	167.31	446.50	170.00	13.00	15.00	21.00	49.50	170.00	509.23	1,429.78	1,850.00	2,707.33
2009	8	14.00	2,445.33	96.00	399.29	91.00	14.42	16.10	18.20	26.00	91.00	231.25	940.10	1,692.72	2,294.81
2010	2	94.00	1,200.00	335.86	647.00	647.00	105.06	149.30	204.60	370.50	647.00	923.50	1,089.40	1,144.70	1,188.94
2011	5	2,500.00	19,740.00	9,122.20	11,248.00	10,000.00	2,720.00	3,600.00	4,700.00	8,000.00	10,000.00	16,000.00	18,244.00	18,992.00	19,590.40
2016	36	5.50	603.33	64.39	129.79	64.46	5.78	12.83	16.50	24.75	64.46	126.25	381.67	551.67	594.00
2017	74	10.00	706.67	64.32	112.46	68.75	10.00	12.65	16.00	29.75	68.75	127.50	277.00	397.50	679.90
2018	52	10.00	1,100.00	102.26	177.98	117.25	12.04	19.00	21.30	39.50	117.25	216.25	488.00	572.50	845.00
2019	16	40.50	390.00	108.99	149.40	69.13	40.76	41.81	47.75	55.75	69.13	240.00	325.00	352.50	382.50
2020	30	11.00	950.00	64.39	121.81	60.00	11.87	14.45	18.60	25.75	60.00	117.50	293.00	347.50	781.80

SUMMARY STATISTICS

OBSERVED PFOA

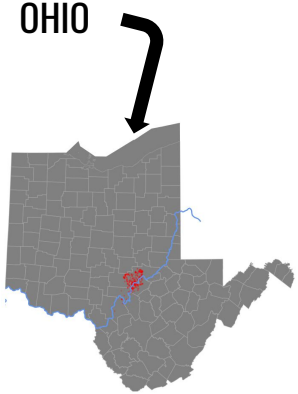
n	522
MINIMUM	5.50 ppt
MAXIMUM	22,700 ppt
MEAN	1,079.03 ppt
GEOMETRIC MEAN	208.34 ppt
MEDIAN	165.70 ppt

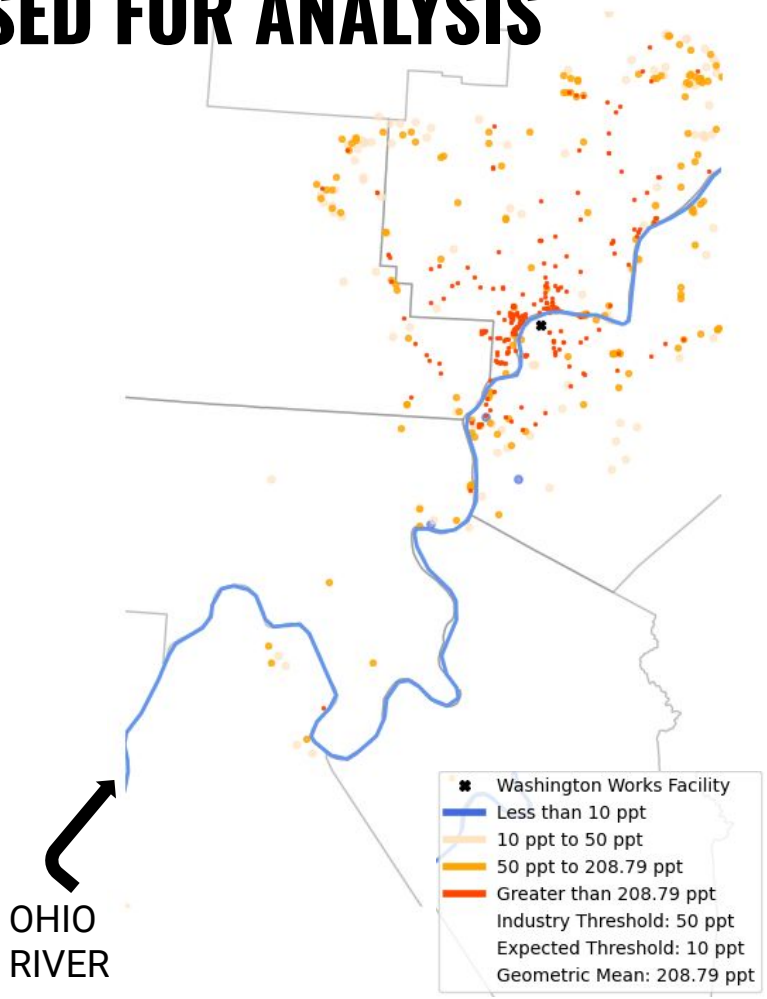
SUMMARY STATISTICS

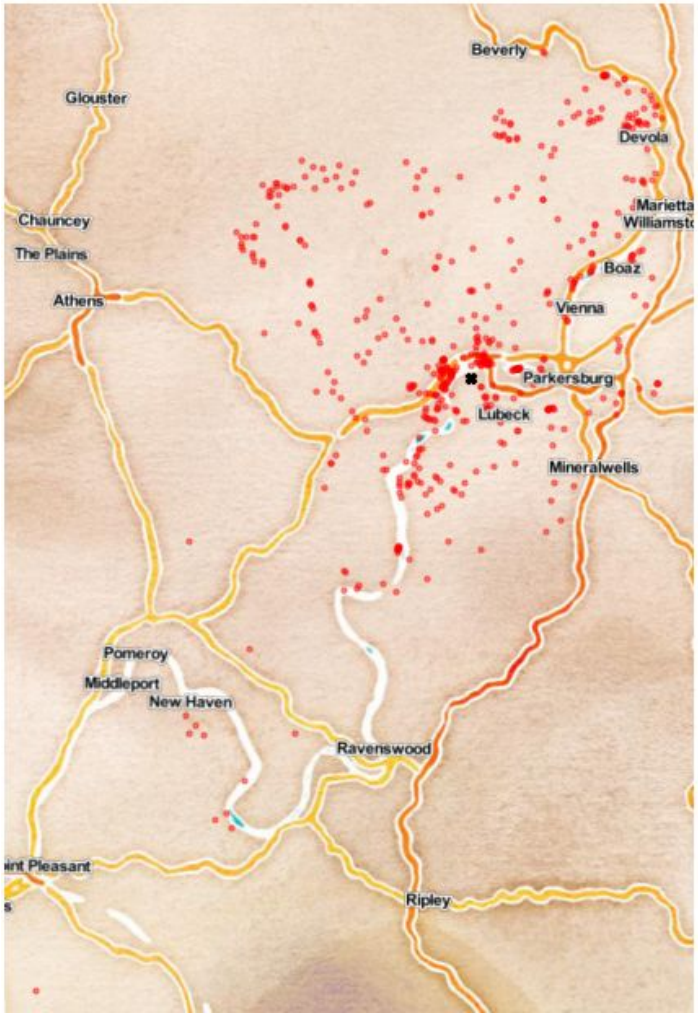
OBSERVED PFOA PERCENTILES

1%	10.06 ppt
5%	15.05 ppt
10%	21.37ppt
25%	54.27 ppt
50%	165.70 ppt
75%	714.00 ppt
90%	2,680.00 ppt
95%	5,832.75 ppt
99%	16,000 ppt

SAMPLES USED FOR ANALYSIS

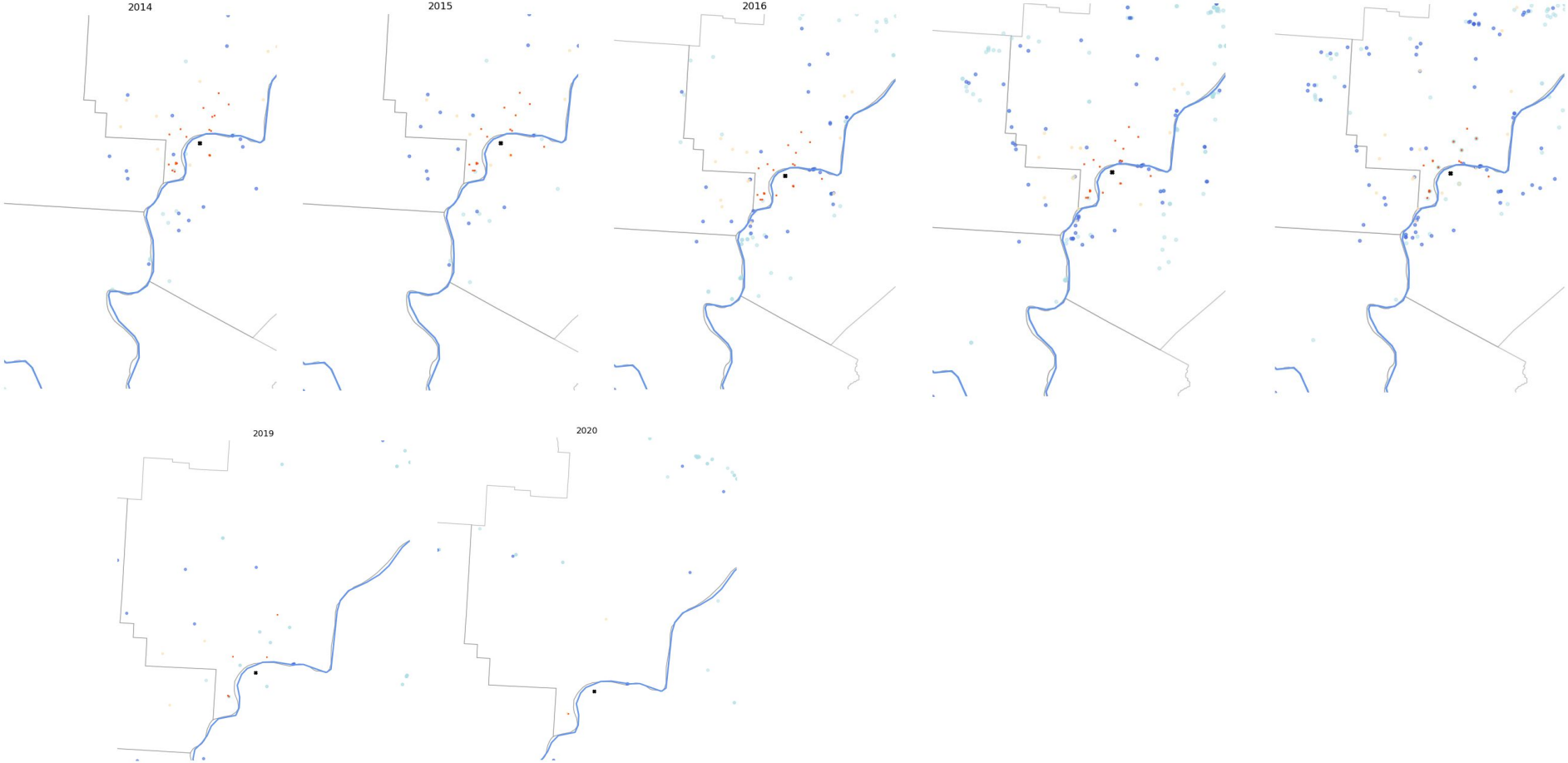






SAMPLES USED FOR ANALYSIS

SAMPLES USED FOR ANALYSIS



```
###########################################################################################################
### Reproject all tiles
###########################################################################################################
# Set destination CRS
dst_crs = rasterio.crs.CRS({"init":"EPSG:32617"})
src_crs = rasterio.crs.CRS({"init":"EPSG:4269"})

# Load all tiles for 2006
inDirectory = "/media/nick/HDD/research/fluorocarbons/data/raster/USGS/img"
outDirectory = "/media/nick/HDD/research/fluorocarbons/data/raster/USGS/img_reproj"
q = os.path.join(inDirectory, "*.img")
inputFiles = glob.glob(q)
q = os.path.join(outDirectory, "*.img")
outputFiles = glob.glob(q)


# Reproject images to EPSG:32617
for i in range(0, len(inputFiles)):

    #  Open raster with Rasterio
    raster = rasterio.open(inputFiles[i])

    # Calculate transformation
    transform, width, height = calculate_default_transform(src_crs, dst_crs, raster.width, raster.height, *raster.bounds)

    # Create meta data for projected raster
    kwargs = raster.meta.copy()
    kwargs.update({
        'crs':dst_crs,
        'transform':transform,
        'width':width,
        'height':height
        })

    # Open destination folder
    destination = rasterio.open(outputFiles[i], 'w', **kwargs)

    reproject(source=rasterio.band(raster, 1),
              destination=rasterio.band(destination, 1),
              src_transform = raster.transform,
              src_crs=src_crs,
              dst_transform = transform,
              dst_crs=rasterio.crs.CRS({"init":"EPSG:32617"}),
              resampling=Resampling.nearest,
              dst_nodata=0
              )
    destination.close()
```

METHODOLOGY

REPROJECT 33 DEMs

1. Set source and destination CRS

2. Load all .img files in input and output directory

3. Iterate each .img file in list

4. Open raster image

5. Calculate transformation values

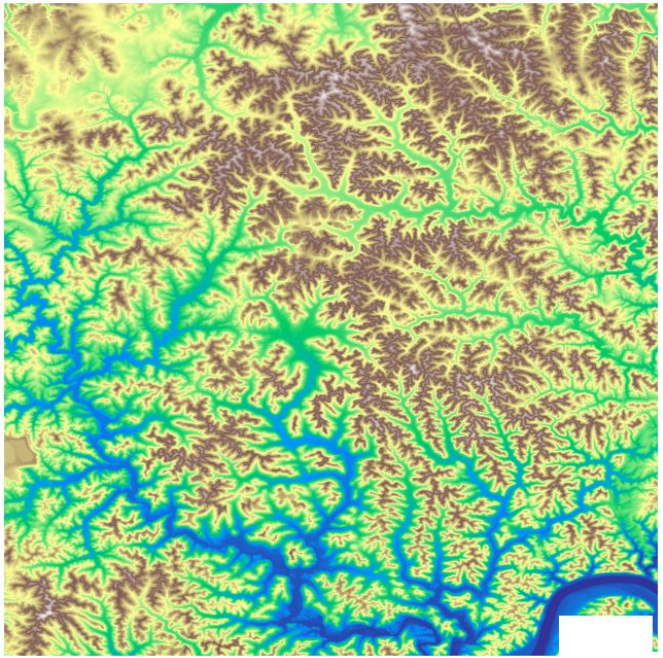
6. Create metadata for reprojected raster

7. Open destination .img as writable object

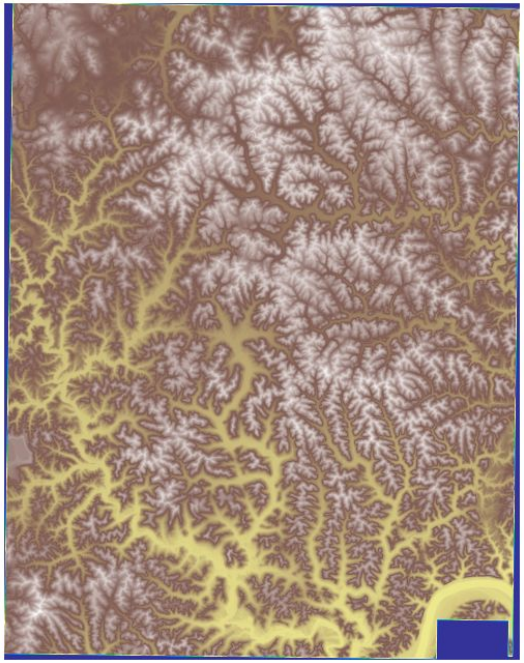
8. Reproject input image to output image location

IMAGE 2 OF 33

EPSG: 4269



EPSG: 32617



RESULT

REPROJECT 33 DEMs

- Each tile was reprojected from NAD83 to US State Plane WV N (meters)
- Displayed with color map equal to “terrain”

```
###########################################################################################################
### Mosaic and clip tiles
###########################################################################################################
# Set destination CRS
dst_crs = rasterio.crs.CRS({"init":"EPSG:32617"})
src_crs = rasterio.crs.CRS({"init":"EPSG:4269"})

# Load all tiles for 2006
inDirectory = "/media/nick/HDD/research/fluorocarbons/data/raster/USGS/img"
outDirectory = "/media/nick/HDD/research/fluorocarbons/data/raster/USGS/img_reproj"
q = os.path.join(inDirectory, "*.img")
inputFiles = glob.glob(q)
q = os.path.join(outDirectory, "*.img")
outputFiles = glob.glob(q)

mosaic_rasterio = []
for file in outputFiles:

    # Open projected raster
    raster = rasterio.open(file)

    # Append reprojected raster image to list
    mosaic_rasterio.append(raster)


# Final merge to mosaic tiles
mosaic, transformation = raster_merge(mosaic_rasterio, indexes=[1], nodata=0)
print(mosaic.shape)
print(type(mosaic))

# Create merged dataset
mosaic_ds = create_dataset(mosaic[0], rasterio.crs.CRS({"init":"EPSG:32617"}), transformation)

# Crop dataset image
polygon = Polygon([(minx, miny), (minx, maxy), (maxx, maxy), (maxx, miny), (minx, miny)])
mosaic, transformation = mask(mosaic_ds, [polygon], crop=True)
print(mosaic.shape)
print(type(mosaic))

# Save new dataset
new_dataset = rasterio.open(
    "/media/nick/HDD/research/fluorocarbons/data/raster/USGS/img_reproj/mosaic.img",
    'w',
    driver='GTiff',
    height=mosaic.shape[1],
    width=mosaic.shape[2],
    count=1,
    dtype=mosaic.dtype,
    crs=dst_crs,
    transform=transformation,
    nodata= 0
)

new_dataset.write(mosaic)
new_dataset.close()
```

METHODOLOGY

MOSAIC DEMs AND CLIP TO EXTENT

1. Set source and destination CRS

2. Load all .img files in input and output directory

3. Iterate each .img file in list

4. Append reprojected raster to list

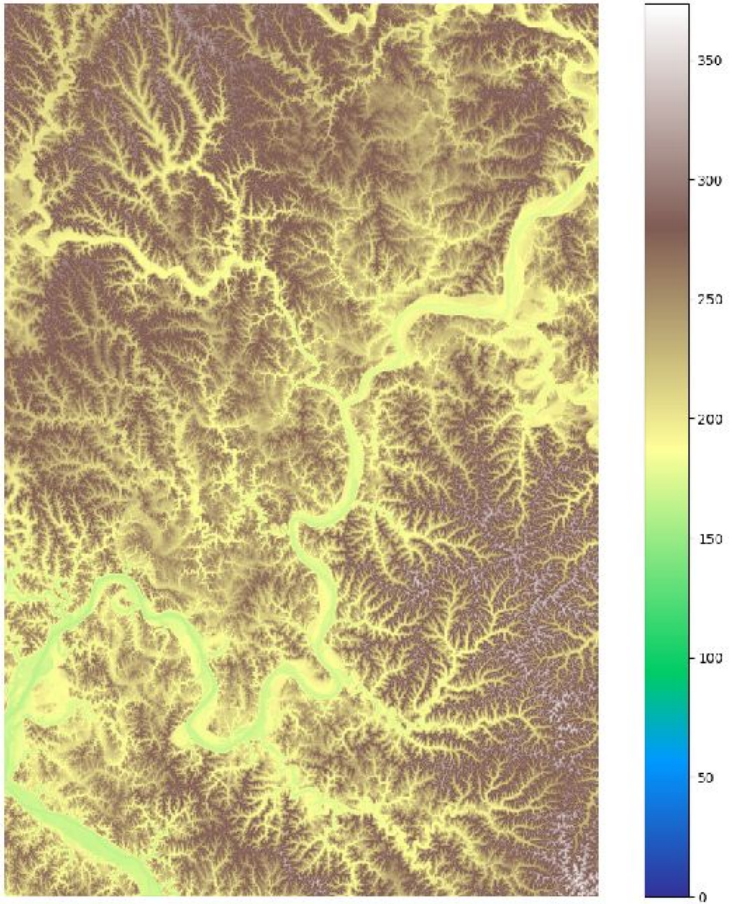
5. Merge reprojected raster images

6. Store mosaic image as a memory mapped data file

7. Create extent (polygon) and mask the mosaic image

8. Save clipped mosaic file to the hard disk

ELEVATION MAP

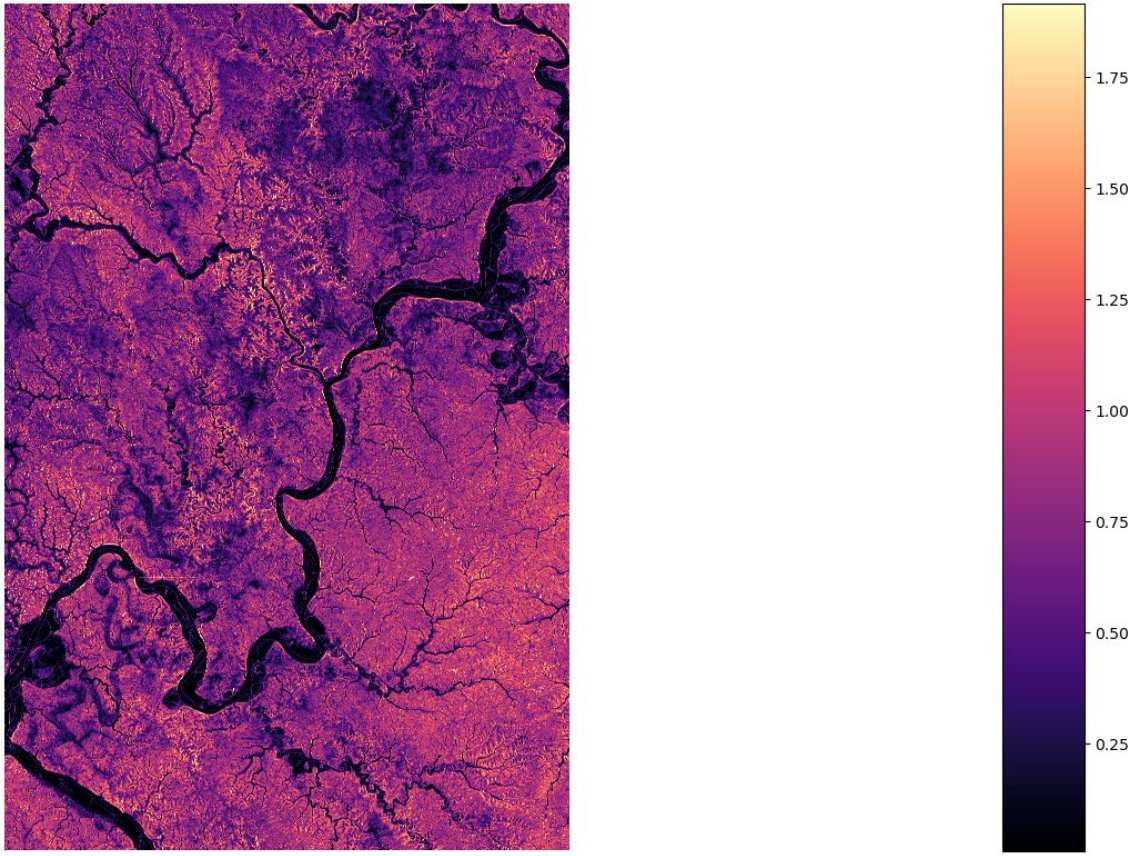


RESULT

MOSAIC DEMs AND CLIP TO EXTENT

- Displayed with color map equal to “terrain”
- Colorbar shows elevation in meters

SLOPE MAP

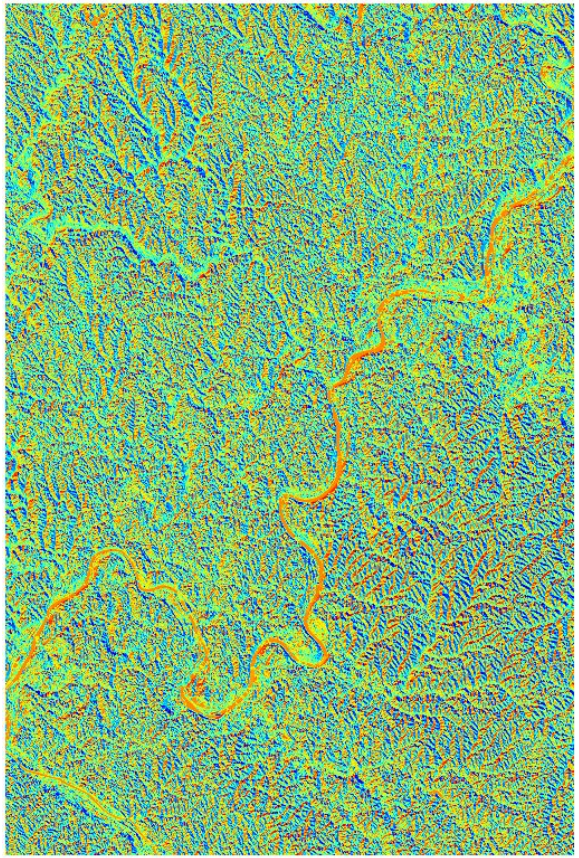


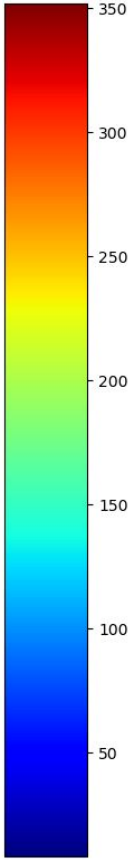
RESULT

MOSAIC DEMs AND CLIP TO EXTENT

- Slope was calculated using Rich DEM rise_run method
- Displayed with color map equal to “magma”
- Colorbar shows rise over run proportion

ASPECT MAP





RESULT

MOSAIC DEMs AND CLIP TO EXTENT

- Aspect was calculated using Rich DEM aspect method
- Displayed with color map equal to “jet”
- Colorbar shows aspect in degrees

```
850  ##########################################################################################################
851  ### Open memory mapped arrays for elevation, slope, & aspect and add them to a dataframe
852  ##########################################################################################################
853  # Create dataset with pfoa and distance values
854  print("So far so good...")
855
856  print("...pfoa...")
     pfoa = rasterio.open("/media/nick/HDD/research/fluorocarbons/data/raster/pfoa.img")
     pfoa = pfoa.read(1)
859  pfoa = np.hstack(pfoa)
860  pfoa = da.from_array(pfoa, chunks=50000)
861
862  print("...distance...")
     dist = rasterio.open("/media/nick/HDD/research/fluorocarbons/data/raster/distance.img")
     dist = dist.read(1)
865  dist = np.hstack(dist)
866  #dist = np.ravel(dist)
867  dist = da.from_array(dist, chunks=50000)
868
869  print("...elevation...")
     # Open memory mapped distance data
     elevation = np.memmap("/media/nick/HDD/research/fluorocarbons/data/temp/elevation.dat", dtype=np.float32, shape=(nrows, ncols), mode='r')
872  elevation = np.hstack(elevation)
873  #elevation = np.ravel(elevation)
874  elevation = da.from_array(elevation, chunks=50000)
875
     print("...slope...")
877  # Open memory mapped distance data
878  slope = np.memmap("/media/nick/HDD/research/fluorocarbons/data/temp/slope.dat", dtype=np.float32, shape=(nrows, ncols), mode='r')
879  slope = np.hstack(slope)
880  #slope = np.ravel(slope)
881  slope = da.from_array(slope, chunks=50000)
882
883  print("...aspect...")
884  # Open memory mapped distance data
     aspect = np.memmap("/media/nick/HDD/research/fluorocarbons/data/temp/aspect.dat", dtype=np.float32, shape=(nrows, ncols), mode='r')
886  aspect = np.hstack(aspect)
887  #aspect = np.ravel(aspect)
888  aspect = da.from_array(aspect, chunks=50000)
889
890  # Combine each array to dask dataframe
891  print("Combining data...")
892
893  # Iterate remaining dataframes and add to dask dataframe
     #dask_data = dd.concat([dd.from_dask_array(a) for a in arr], axis=1)
895  dask_data = dd.from_dask_array(pfoa, columns=["pfoa"])
896  dask_data["distance"] = dist
897  dask_data["elevation"] = elevation
898  dask_data["slope"] = slope
899  dask_data["aspect"] = aspect
900  del pfoa
901  del dist
902  del elevation
903  del slope
904  del aspect
905
     # Remove all 0 value observations from dataframe
907  print("Remove all zero values...")
908  dask_data = dask_data[(dask_data.pfoa > 0)]
909  data = dask_data.compute()
910  del dask_data
```

METHODOLOGY

CREATE DATAFRAME FOR ANALYSIS

1. Load rasterized pfoa values
2. Load rasterized distance values
3. Load elevation raster image
4. Load slope raster image
5. Load aspect raster image
6. Convert all Numpy Arrays to Dask Arrays
7. Create Dask dataframe from Dask Arrays
8. Remove all rows where PFOA values equal 0 and convert back to Pandas DataFrame

PROOF OF CONCEPT

CONSOLE OUTPUT:

```
Regression analysis...

Train MSE:  2.677772174831958
Test MSE:  10.754401893338665
Predicted MSE:  5.111612608239929

MSE train: 2.678, test: 10.754
R-squared train: 0.531, test: -0.172
R-squared: 0.244

Feature Importance:

aspect       0.362831
elevation    0.349725
slope        0.287444
```

RANDOM FOREST ANALYSIS

REGRESSION ANALYSIS PER PIXEL

- MEAN OBSERVED: 1,079 ppt
- RMSE: 3,280 ppt
- VARIATION EXPLAINED: disagreement
- THIS MODEL DOES NOT FIT THE DATA

PROOF OF CONCEPT

CONSOLE OUTPUT:

```
Classification Report:

              precision    recall  f1-score   support

       above       0.77      0.93      0.84       121
       below       0.20      0.06      0.09        36

    accuracy                           0.73       157
   macro avg       0.48      0.49      0.47       157
weighted avg       0.64      0.73      0.67       157

Cohen's Kappa:  -0.014149492463857438
```

RANDOM FOREST ANALYSIS

BINARY CLASSIFICATION ANALYSIS PER PIXEL

- CLASSIFICATION THRESHOLD: 50 ppt
- ACCURACY: 73%
- PRECISION ABOVE 50 ppt: 77%
- PRECISION BELOW 50 ppt: 20%

PROOF OF CONCEPT

CONSOLE OUTPUT:

```
Regression analysis...

Train MSE:  1.0283453215162885
Test MSE:  8.482567433458698
Predicted MSE:  3.2703240026943696

MSE train: 1.028, test: 8.483
R-squared train: 0.835, test: -0.079
R-squared: 0.515
smin        0.228544
smax        0.154970
amin        0.120226
smedian     0.079174
smean       0.072015
amax        0.064734
amedian     0.060516
emin        0.049533
amean       0.046567
emean       0.043690
emax        0.042435
emedian     0.037596
```

RANDOM FOREST ANALYSIS

REGRESSION ANALYSIS 15 METER BUFFER

- MEAN OBSERVED: 1,079 ppt
- RMSE: 2,913 ppt
- VARIATION EXPLAINED: disagreement
- THIS MODEL DOES NOT FIT THE DATA

PROOF OF CONCEPT

CONSOLE OUTPUT:

```
              precision    recall  f1-score   support

       above       0.79      0.93      0.86       122
       below       0.38      0.14      0.21        35

    accuracy                           0.76       157
   macro avg       0.59      0.54      0.53       157
weighted avg       0.70      0.76      0.71       157

0.09960760639903421
smax        0.101676
emin        0.094315
emean       0.089415
emax        0.086322
emedian     0.084827
smin        0.081855
amax        0.080871
smedian     0.079876
amean       0.079804
smean       0.076882
amin        0.075748
amedian     0.068409
```

RANDOM FOREST ANALYSIS

BINARY CLASSIFICATION ANALYSIS 15 METER BUFFER

- CLASSIFICATION THRESHOLD: 50 ppt
- ACCURACY: 76%
- PRECISION ABOVE 50 ppt: 79%
- PRECISION BELOW 50 ppt: 38%

NEXT STEPS

- DATA NEEDS MORE QA
  - ARE THERE TOO MANY POINTS CLOSE TO THE FACILITY?
  - WHY ARE HIGHER PFOA MEASURES MISSING COORDINATES?
- MORE VARIABLES NEED TO BE INCLUDED
- OTHER STATS/ML MODELS NEED TO BE INVESTIGATED

VARIABLES:

- FLOW ACCUMULATION
- WETNESS INDEX (FLOW (ACCUMULATION * GRID CELL SIZE) / SLOPE
- SOIL MOISTURE INDEX
- VEGETATION INDEX
- LAND CLASSIFICATION
- ATMOSPHERE
- AQUIFER LOCATION
- WATER FLOW

ANALYSIS:

- BINARY CLASSIFICATION
- NEURAL NETWORK
- CLUSTER DETECTION

THANK YOU!