In [35]:
```
DATA_DIR = '/data/tmp/arogya/afg_updated/'
PREDICTIONS_DIR = f'{DATA_DIR}/outputs/predictions'
POPPY_COUNTS_FILE = '../../data/inputs/poppy_1994-2020.csv'
```

In [104…
```
import os
import pandas as pd
import numpy as np
pred_dirs = os.listdir(PREDICTIONS_DIR)
```

In [48]:
```
def get_poppy_pixels(filepath, year, district):
    counts = pd.read_csv(filepath)
#     return counts
    return counts[counts['distid']==district][f'X{year}'].iloc[0]
```

In [49]:
```
get_poppy_pixels(POPPY_COUNTS_FILE, '2019', 2305)
```

Out[49]:
```
1857.0
```

In [54]:
```
all_results = None
for run in pred_dirs:
    results = f"{PREDICTIONS_DIR}/{run}/{run}.csv"
    results = pd.read_csv(results)
    results['model'] = run.split("_")[0]
    results['n'] = run.split("_")[1]
    results['dataset'] = run.split("_")[2] +"_" +dir.split("_")[3]
    results['district'] = run.split("_")[4]
    results['unodc_ha'] = get_poppy_pixels(POPPY_COUNTS_FILE, '2019', int(run.split("_")
    if all_results is None:
        all_results = results
    else:
        all_results = pd.concat([all_results, results], axis=0)

#     print(dir)
```

In [96]:
```
datasets = ["diff_bands", "pre_bands"]
models = ["kmeans", "gmm"]
ns = [3,4,5]

corrs = None
for model in models:
    for dataset in datasets:
        for n in ns:
            df = all_results[all_results['model']==model]
            df = df[df['n']==str(n)]
            df = df[df['dataset']== dataset]
            df = df[['clust', 'clustering_ha', 'unodc_ha']].groupby('clust').corr()
            df = df.reset_index()
            df = df[df['level_1'] != 'unodc_ha']
            df = df.drop(['level_1', 'clustering_ha'], axis =1)
            df.columns = ['clust', 'pearson_correlation']
            df['model'] = model
            df['n'] = n
            df['dataset'] = dataset

            if df is None:
                corrs = df
            else:
                corrs = pd.concat([df, corrs], axis=0)
```

In [99]:
```
corrs.sort_values('pearson_correlation', ascending=False)
```

Out[99]:

	clust	pearson_correlation	model	n	dataset
4	2.0	0.938810	gmm	3	diff_bands
0	0.0	0.932809	kmeans	5	diff_bands
2	1.0	0.928945	gmm	5	diff_bands
0	0.0	0.927670	gmm	4	pre_bands
2	1.0	0.927276	kmeans	4	pre_bands
2	1.0	0.921378	kmeans	5	pre_bands
4	2.0	0.920846	gmm	5	pre_bands
0	0.0	0.916571	kmeans	3	pre_bands
2	1.0	0.916365	kmeans	4	diff_bands
0	0.0	0.915271	gmm	5	pre_bands
0	0.0	0.914517	kmeans	3	diff_bands
4	2.0	0.898531	gmm	3	pre_bands
4	2.0	0.894333	gmm	4	pre_bands
6	3.0	0.887577	gmm	4	diff_bands
8	4.0	0.881561	kmeans	5	diff_bands
8	4.0	0.869490	gmm	5	pre_bands
6	3.0	0.867396	kmeans	5	pre_bands
0	0.0	0.866659	gmm	3	pre_bands
4	2.0	0.866255	kmeans	4	diff_bands
2	1.0	0.837989	gmm	3	diff_bands
0	0.0	0.831256	gmm	5	diff_bands
0	0.0	0.830782	gmm	4	diff_bands
6	3.0	0.822872	gmm	5	diff_bands
6	3.0	0.800723	kmeans	4	pre_bands
8	4.0	0.792191	kmeans	5	pre_bands
6	3.0	0.778957	gmm	5	pre_bands
2	1.0	0.771889	kmeans	3	diff_bands
0	0.0	0.768832	kmeans	4	pre_bands
6	3.0	0.764168	kmeans	5	diff_bands
6	3.0	0.750607	gmm	4	pre_bands
0	0.0	0.742178	kmeans	4	diff_bands
0	0.0	0.722096	kmeans	5	pre_bands
4	2.0	0.720932	kmeans	3	pre_bands
2	1.0	0.718002	kmeans	3	pre_bands
4	2.0	0.700438	kmeans	5	diff_bands
4	2.0	0.696918	kmeans	3	diff_bands
6	3.0	0.662073	kmeans	4	diff_bands
2	1.0	0.650113	kmeans	5	diff_bands
2	1.0	0.561643	gmm	3	pre_bands

4	2.0	0.550680	gmm	4	diff_bands
0	0.0	0.519208	gmm	3	diff_bands
2	1.0	0.496216	gmm	4	pre_bands
4	2.0	0.493390	kmeans	4	pre_bands
4	2.0	0.478239	kmeans	5	pre_bands
4	2.0	0.434769	gmm	5	diff_bands
2	1.0	0.207858	gmm	5	pre_bands
2	1.0	-0.023217	gmm	4	diff_bands
8	4.0	-0.023958	gmm	5	diff_bands

In [105…
```
total_prod = all_results.groupby(['district']).max('unodc_ha').sum()['unodc_ha']
dist_index = all_results.groupby('district').apply(lambda x: np.max(x['unodc_ha'])/total
```

In [106…
```
total_prod
```

Out[106]:
```
80587.0
```

In [115…
```
indices = pd.DataFrame(dist_index)
indices.columns = ['yield_index']
```

In [120…
```
all_results = pd.merge(all_results, indices, on='district')
```

In [136…
```
all_results['yield_corrected_se'] = (((all_results['clustering_ha'] - all_results['unodc
```

In [137…
```
all_results.groupby(['model', 'n', 'dataset', 'clust']).sum()['yield_corrected_se'].rese
```

Out[137]:

	model	n	dataset	clust	yield_corrected_se
15	gmm	5	diff_bands	1.0	5.759050e+06
38	kmeans	5	diff_bands	0.0	6.002431e+06
34	kmeans	4	pre_bands	0.0	7.680648e+06
44	kmeans	5	pre_bands	1.0	7.778827e+06
35	kmeans	4	pre_bands	1.0	7.905353e+06
10	gmm	4	pre_bands	0.0	9.180176e+06
31	kmeans	4	diff_bands	1.0	9.246428e+06
23	gmm	5	pre_bands	4.0	9.856746e+06
46	kmeans	5	pre_bands	3.0	1.222376e+07
19	gmm	5	pre_bands	0.0	1.318496e+07
25	kmeans	3	diff_bands	1.0	1.324935e+07
12	gmm	4	pre_bands	2.0	1.367780e+07
43	kmeans	5	pre_bands	0.0	1.591301e+07
24	kmeans	3	diff_bands	0.0	1.637102e+07
5	gmm	3	pre_bands	2.0	1.647622e+07
21	gmm	5	pre_bands	2.0	1.687615e+07
40	kmeans	5	diff_bands	2.0	1.811781e+07
30	kmeans	4	diff_bands	0.0	2.072288e+07

41	kmeans	5	diff_bands	3.0	2.110958e+07
28	kmeans	3	pre_bands	1.0	2.409270e+07
1	gmm	3	diff_bands	1.0	2.512654e+07
33	kmeans	4	diff_bands	3.0	2.593608e+07
32	kmeans	4	diff_bands	2.0	2.661342e+07
29	kmeans	3	pre_bands	2.0	2.666057e+07
39	kmeans	5	diff_bands	1.0	2.669079e+07
11	gmm	4	pre_bands	1.0	2.709750e+07
37	kmeans	4	pre_bands	3.0	2.805791e+07
47	kmeans	5	pre_bands	4.0	2.840935e+07
13	gmm	4	pre_bands	3.0	2.845762e+07
36	kmeans	4	pre_bands	2.0	2.915300e+07
45	kmeans	5	pre_bands	2.0	3.202392e+07
27	kmeans	3	pre_bands	0.0	3.241239e+07
42	kmeans	5	diff_bands	4.0	3.340544e+07
2	gmm	3	diff_bands	2.0	3.346943e+07
4	gmm	3	pre_bands	1.0	3.399401e+07
17	gmm	5	diff_bands	3.0	3.587112e+07
3	gmm	3	pre_bands	0.0	3.769231e+07
22	gmm	5	pre_bands	3.0	3.975068e+07
20	gmm	5	pre_bands	1.0	4.554954e+07
9	gmm	4	diff_bands	3.0	4.756753e+07
8	gmm	4	diff_bands	2.0	5.338955e+07
16	gmm	5	diff_bands	2.0	6.466623e+07
18	gmm	5	diff_bands	4.0	8.831185e+07
7	gmm	4	diff_bands	1.0	8.871236e+07
0	gmm	3	diff_bands	0.0	9.225264e+07
6	gmm	4	diff_bands	0.0	9.421877e+07
14	gmm	5	diff_bands	0.0	9.427790e+07
26	kmeans	3	diff_bands	2.0	1.012442e+08

In [139…

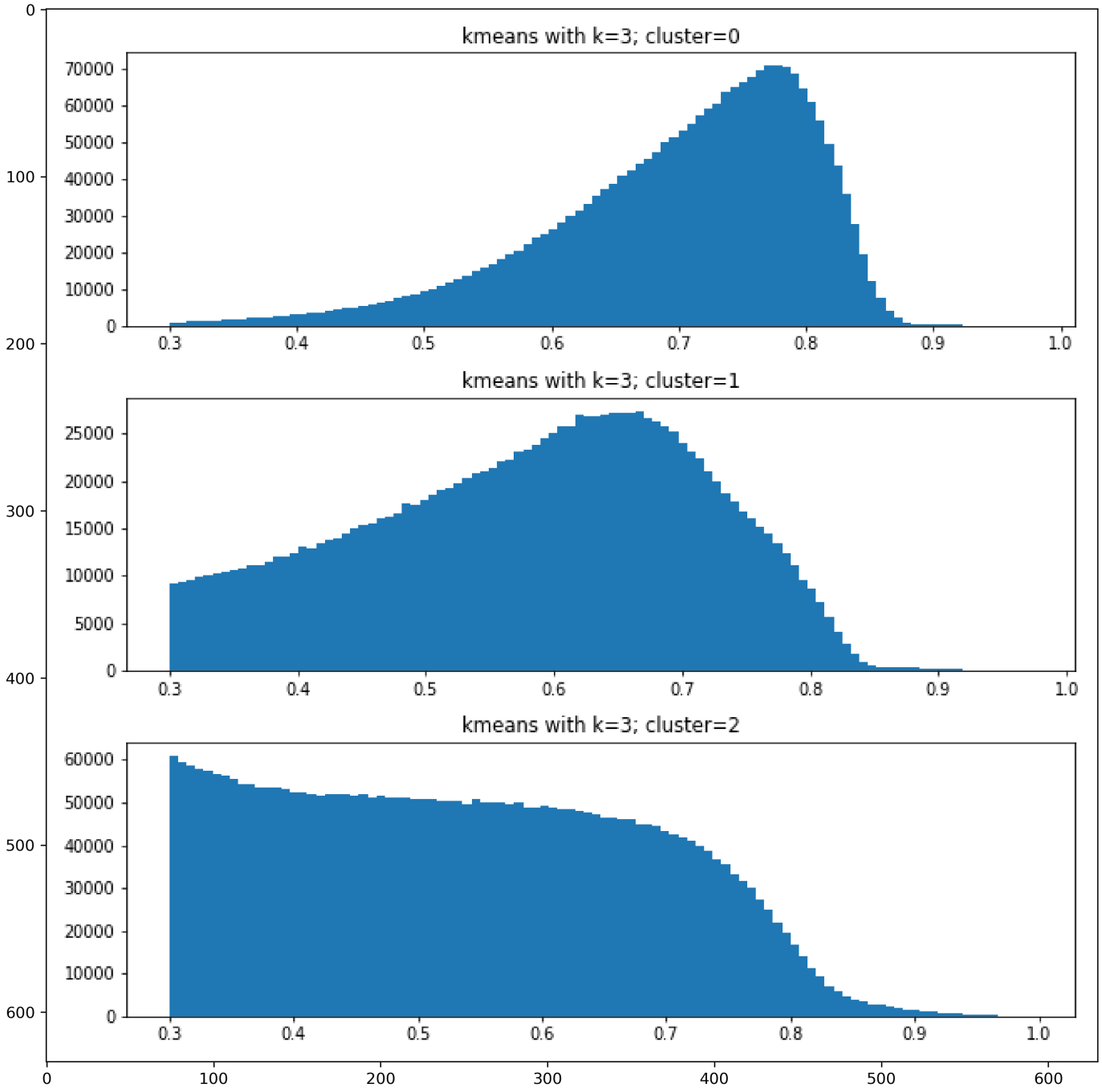
```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg


# nrows = len(districts['full'])//3
fig, ax = plt.subplots(1,1,dpi=300, figsize=(10,10))
# ax = ax.flatten()

ax.imshow(mpimg.imread(f"{PREDICTIONS_DIR}/kmeans_3_diff_bands_2308/kmeans_3_diff_bands_

# for i, dist in enumerate(districts['full']):
#     ax[i].imshow(mpimg.imread(f"{PREDICTIONS_DIR}/ndvi_{dist}_gmm_5.png"))
#     ax[i].set_title(dist)
#     ax[i].set_xticks([])
```

```
#     ax[i].set_yticks([])
plt.tight_layout()
```



In [ ]: