In [84]:

```
# Import libraries
import numpy as np
import rasterio
import pandas as pd
import matplotlib.pyplot as plt
```

In [85]:

```
src = rasterio.open('data/nlcd_2001_phoenix.tif')
src
```

Out[85]:

```
<open DatasetReader name='data/nlcd_2001_phoenix.tif' mode='r'>
```

In [86]:

```
print(f"Number of bands: {src.count}")
print(f"Width: {src.width}")
print(f"Height: {src.height}")
print(f"Data type: {src.dtypes}")
```

```
Number of bands: 1
Width: 8176
Height: 5343
Data type: ('uint8',)
```

In [87]:

```
src.transform
```

Out[87]:

```
Affine(30.000762373473098, 0.0, -1589209.731401832,
       0.0, -29.999261343006733, 1331490.51999102)
```

In [88]:

```
src.bounds
```

Out[88]:

```
BoundingBox(left=-1589209.731401832, bottom=1171204.466635335, right=-1343923.498236316, top=1331490.51999102)
```

In [89]:

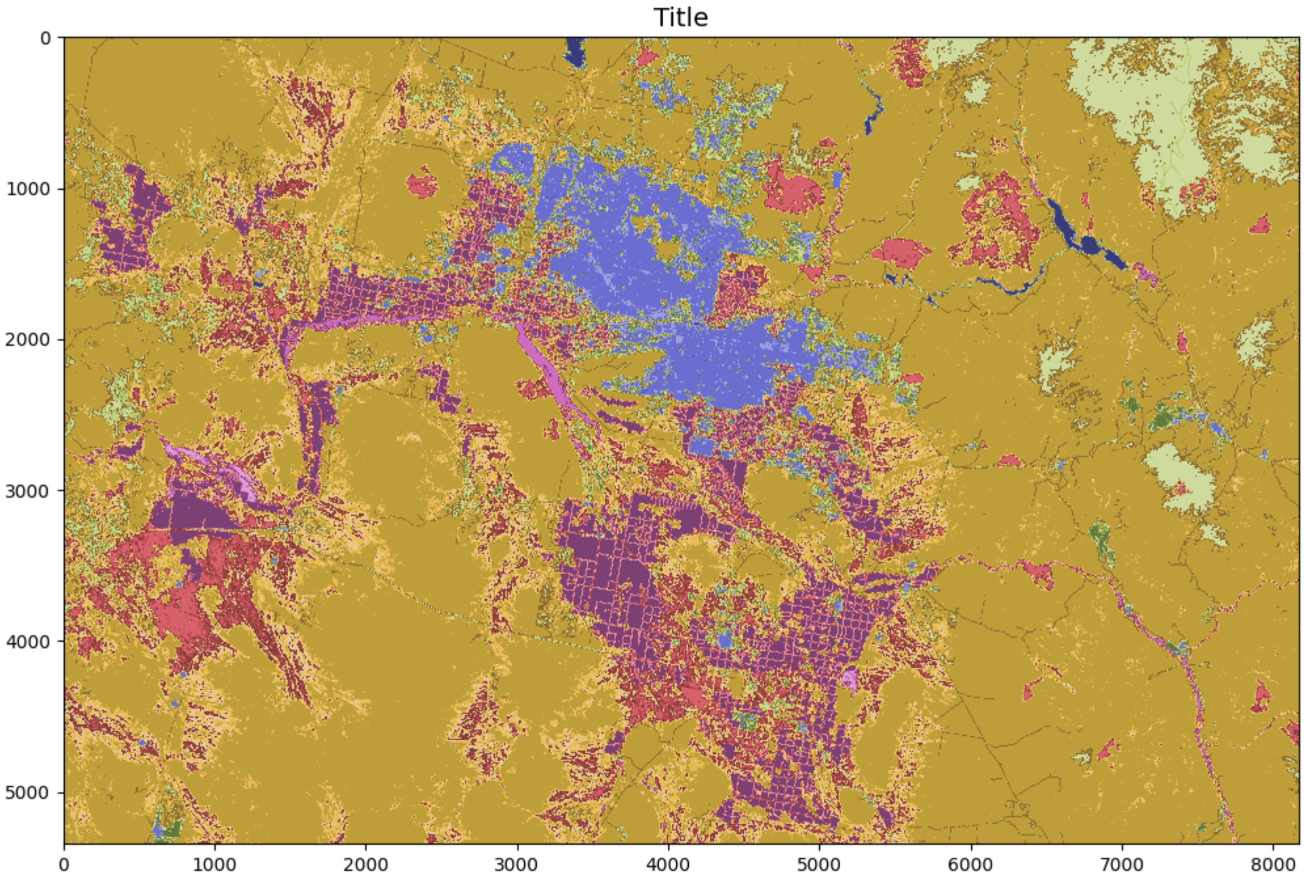
```
nlcd_2001_phoenix= src.read(1)
nlcd_2001_phoenix
```

Out[89]:

```
array([[52, 52, 52, ..., 42, 42, 42],
       [52, 52, 52, ..., 42, 42, 42],
       [52, 52, 52, ..., 42, 42, 42],
       ...,
       [52, 52, 52, ..., 52, 52, 52],
       [52, 52, 52, ..., 52, 52, 52],
       [52, 52, 52, ..., 52, 52, 52]], dtype=uint8)
```

```
In [90]: fig, ax = plt.subplots(figsize=(16,8))
         im = ax.imshow(nlcd_2001_phoenix, cmap='tab20b')
         ax.set_title("Title", fontsize=14)
```

```
Out[90]: Text(0.5, 1.0, 'Title')
```



```
In [91]: unique, counts = np.unique(nlcd_2001_phoenix, return_counts=True)
         dict(zip(unique, counts))
```

```
Out[91]: {11: 143014,
          21: 919466,
          22: 809417,
          23: 953533,
          24: 322422,
          31: 706318,
          42: 1497913,
          43: 1603,
          52: 30851488,
          71: 4039637,
          81: 41732,
          82: 3046041,
          90: 289378,
          95: 62406}
```

In [92]:

```
# Count number of land pixels
land_pixels = nlcd_2001_phoenix.size

# Convert to DataFrame
df_2001 = pd.DataFrame(list(zip(unique, counts, (counts/land_pixels)*100)),
                       columns=['lc', 'count_2001', 'fraction_2001'])
df_2001
```

Out[92]:

	lc	count_2001	fraction_2001
0	11	143014	0.327380
1	21	919466	2.104794
2	22	809417	1.852876
3	23	953533	2.182779
4	24	322422	0.738072
5	31	706318	1.616867
6	42	1497913	3.428945
7	43	1603	0.003670
8	52	30851488	70.623634
9	71	4039637	9.247328
10	81	41732	0.095531
11	82	3046041	6.972840
12	90	289378	0.662429
13	95	62406	0.142857

In [93]:

```
open_water = df_2001[df_2001['lc'] == 11]
```

In [94]:

```
open_water
```

Out[94]:

	lc	count_2001	fraction_2001
0	11	143014	0.32738

In [95]:

```
open_water['count_2001'][0]
```

Out[95]:

```
143014
```

In [96]:

```
developed_openspace = df_2001[df_2001['lc'] == 21]
developed_openspace
```

Out[96]:

	lc	count_2001	fraction_2001
1	21	919466	2.104794

In [97]:

```
developed_openspace
```

Out[97]:

	lc	count_2001	fraction_2001
1	21	919466	2.104794

In [98]:

```
open_space_value = developed_openspace['count_2001'][1]
open_space_value
```

Out[98]:

```
919466
```

In [99]:

```
developed_low = df_2001[df_2001['lc'] == 22]
developed_low
```

Out[99]:

	lc	count_2001	fraction_2001
2	22	809417	1.852876

In [100]:

```
low_value = developed_low['count_2001'][2]
low_value
```

Out[100]:

```
809417
```

```
In [101]: developed_med_intensity = df_2001[df_2001['lc'] == 23]
          developed_med_intensity
```

Out[101]:

	lc	count_2001	fraction_2001
3	23	953533	2.182779

```
In [102]: med_value = developed_med_intensity['count_2001'][3]
          med_value

Out[102]: 953533
```

```
In [103]: developed_high_intensity = df_2001[df_2001['lc'] == 24]
          developed_high_intensity
```

Out[103]:

	lc	count_2001	fraction_2001
4	24	322422	0.738072

```
In [104]: high_value = developed_high_intensity['count_2001'][4]
          high_value

Out[104]: 322422
```

```
In [105]: developed_grid_cells = open_space_value + low_value + med_value + high_value
          developed_grid_cells

Out[105]: 3004838
```

```
In [106]: cultivated_crop = df_2001[df_2001['lc'] == 82]
          cultivated_crop
```

Out[106]:

	lc	count_2001	fraction_2001
11	82	3046041	6.97284

```
In [107]: cultivated_crop['count_2001']

Out[107]: 11    3046041
          Name: count_2001, dtype: int64
```

```
In [108]: df_2001
```

Out[108]:

	lc	count_2001	fraction_2001
0	11	143014	0.327380
1	21	919466	2.104794
2	22	809417	1.852876
3	23	953533	2.182779
4	24	322422	0.738072
5	31	706318	1.616867
6	42	1497913	3.428945
7	43	1603	0.003670
8	52	30851488	70.623634
9	71	4039637	9.247328
10	81	41732	0.095531
11	82	3046041	6.972840
12	90	289378	0.662429
13	95	62406	0.142857

```
In [109]: df_2001['count_2001'].sum()

Out[109]: 43684368
```

```
In [110]: nlcd_2001_phoenix[nlcd_2001_phoenix == 21] = 22
```

In [111]:

```
# Check there are no more grid cells classified as normal shrubs
unique, counts = np.unique(nlcd_2001_phoenix, return_counts=True)

# Convert to DataFrame
df_2001 = pd.DataFrame(list(zip(unique, counts, (counts/land_pixels)*100)),
                       columns=['lc', 'count_2001', 'fraction_2001'])
df_2001
```

Out[111]:

	lc	count_2001	fraction_2001
0	11	143014	0.327380
1	22	1728883	3.957670
2	23	953533	2.182779
3	24	322422	0.738072
4	31	706318	1.616867
5	42	1497913	3.428945
6	43	1603	0.003670
7	52	30851488	70.623634
8	71	4039637	9.247328
9	81	41732	0.095531
10	82	3046041	6.972840
11	90	289378	0.662429
12	95	62406	0.142857

In [112]:

```
nlcd_2001_phoenix[nlcd_2001_phoenix == 22] = 23
```

In [113]:

```
# Check there are no more grid cells classified as normal shrubs
unique, counts = np.unique(nlcd_2001_phoenix, return_counts=True)

# Convert to DataFrame
df_2001 = pd.DataFrame(list(zip(unique, counts, (counts/land_pixels)*100)),
                       columns=['lc', 'count_2001', 'fraction_2001'])
df_2001
```

Out[113]:

	lc	count_2001	fraction_2001
0	11	143014	0.327380
1	23	2682416	6.140448
2	24	322422	0.738072
3	31	706318	1.616867
4	42	1497913	3.428945
5	43	1603	0.003670
6	52	30851488	70.623634
7	71	4039637	9.247328
8	81	41732	0.095531
9	82	3046041	6.972840
10	90	289378	0.662429
11	95	62406	0.142857

In [114]:

```
nlcd_2001_phoenix[nlcd_2001_phoenix == 23] = 24
```

In [115]:

```
# Check there are no more grid cells classified as normal shrubs
unique, counts = np.unique(nlcd_2001_phoenix, return_counts=True)

# Convert to DataFrame
df_2001 = pd.DataFrame(list(zip(unique, counts, (counts/land_pixels)*100)),
                       columns=['lc', 'count_2001', 'fraction_2001'])
df_2001
```

Out[115]:

	lc	count_2001	fraction_2001
0	11	143014	0.327380
1	24	3004838	6.878520
2	31	706318	1.616867
3	42	1497913	3.428945
4	43	1603	0.003670
5	52	30851488	70.623634
6	71	4039637	9.247328
7	81	41732	0.095531
8	82	3046041	6.972840
9	90	289378	0.662429
10	95	62406	0.142857

In [116]:

```
developed_perc = df_2001[df_2001['lc'] == 24]
```

In [117]:

```
developed_perc['fraction_2001'][1]
```

Out[117]:

```
6.87852002345553
```

In [118]:

```
openwater_perc = df_2001[df_2001['lc'] == 11]
```

In [119]:

```
openwater_perc
```

Out[119]:

	lc	count_2001	fraction_2001
0	11	143014	0.32738

In [120]:

```
openwater_perc['fraction_2001'][0]
```

Out[120]:

```
0.32738026563644
```

In [121]:

```
most_common_perc = df_2001['fraction_2001'].max()
most_common_perc
```

Out[121]:

```
70.62363360733524
```

In [122]:

```
most_com_class = df_2001[df_2001['fraction_2001'] == most_common_perc]
```

In [123]:

```
most_com_class
```

Out[123]:

	lc	count_2001	fraction_2001
5	52	30851488	70.623634

In [124]:

```
most_com_class['lc'][5]
```

Out[124]:

```
52
```

In [125]:

```
abc = rasterio.open('data/nlcd_2019_phoenix.tif')
abc
```

Out[125]:

```
<open DatasetReader name='data/nlcd_2019_phoenix.tif' mode='r'>
```

In [126]:

```
print(f"Number of bands: {abc.count}")
print(f"Width: {abc.width}")
print(f"Height: {abc.height}")
print(f"Data type: {abc.dtypes}")
```

```
Number of bands: 1
Width: 8176
Height: 5343
Data type: ('uint8',)
```

```
In [127]: nlcd_2019 = abc.read(1)
          nlcd_2019
```

```
Out[127]: array([[52, 52, 52, ..., 42, 42, 42],
                 [52, 52, 52, ..., 42, 42, 42],
                 [52, 52, 52, ..., 42, 42, 42],
                 ...,
                 [52, 52, 52, ..., 52, 52, 52],
                 [52, 52, 52, ..., 52, 52, 52],
                 [52, 52, 52, ..., 52, 52, 52]], dtype=uint8)
```

```
In [128]: unique, counts = np.unique(nlcd_2019, return_counts=True)
          dict(zip(unique, counts))
```

```
Out[128]: {11: 155179,
           21: 1016781,
           22: 1070489,
           23: 1367277,
           24: 459333,
           31: 685585,
           41: 8,
           42: 1245304,
           43: 1120,
           52: 30186997,
           71: 4284805,
           81: 37785,
           82: 2822333,
           90: 281933,
           95: 69439}
```

```
In [129]: # Count number of land pixels
          land_pixels = nlcd_2019.size

          # Convert to DataFrame
          df_2019 = pd.DataFrame(list(zip(unique, counts, (counts/land_pixels)*100)),
                                 columns=['lc', 'count_2019', 'fraction_2019'])
          df_2019
```

Out[129]:

	lc	count_2019	fraction_2019
0	11	155179	0.355228
1	21	1016781	2.327563
2	22	1070489	2.450508
3	23	1367277	3.129900
4	24	459333	1.051481
5	31	685585	1.569406
6	41	8	0.000018
7	42	1245304	2.850686
8	43	1120	0.002564
9	52	30186997	69.102515
10	71	4284805	9.808554
11	81	37785	0.086495
12	82	2822333	6.460739
13	90	281933	0.645386
14	95	69439	0.158956

```
In [130]: df_2019['count_2019'].sum()
```

```
Out[130]: 43684368
```

```
In [131]: nlcd_2019[nlcd_2019 == 21] = 22
```

In [132]:

```
# Check there are no more grid cells classified as normal shrubs
unique, counts = np.unique(nlcd_2019, return_counts=True)

# Convert to DataFrame
df_2019 = pd.DataFrame(list(zip(unique, counts, (counts/land_pixels)*100)),
                       columns=['lc', 'count_2019', 'fraction_2019'])
df_2019
```

Out[132]:

	lc	count_2019	fraction_2019
0	11	155179	0.355228
1	22	2087270	4.778071
2	23	1367277	3.129900
3	24	459333	1.051481
4	31	685585	1.569406
5	41	8	0.000018
6	42	1245304	2.850686
7	43	1120	0.002564
8	52	30186997	69.102515
9	71	4284805	9.808554
10	81	37785	0.086495
11	82	2822333	6.460739
12	90	281933	0.645386
13	95	69439	0.158956

In [133]:

```
nlcd_2019[nlcd_2019 == 22] = 23
```

In [134]:

```
# Check there are no more grid cells classified as normal shrubs
unique, counts = np.unique(nlcd_2019, return_counts=True)

# Convert to DataFrame
df_2019 = pd.DataFrame(list(zip(unique, counts, (counts/land_pixels)*100)),
                       columns=['lc', 'count_2019', 'fraction_2019'])
df_2019
```

Out[134]:

	lc	count_2019	fraction_2019
0	11	155179	0.355228
1	23	3454547	7.907971
2	24	459333	1.051481
3	31	685585	1.569406
4	41	8	0.000018
5	42	1245304	2.850686
6	43	1120	0.002564
7	52	30186997	69.102515
8	71	4284805	9.808554
9	81	37785	0.086495
10	82	2822333	6.460739
11	90	281933	0.645386
12	95	69439	0.158956

In [164]:

```
nlcd_2019[nlcd_2019 == 23] = 24
```

In [165]:

```
# Check there are no more grid cells classified as normal shrubs
unique, counts = np.unique(nlcd_2019, return_counts=True)

# Convert to DataFrame
df_2019 = pd.DataFrame(list(zip(unique, counts, (counts/land_pixels)*100)),
                       columns=['lc', 'count_2019', 'fraction_2019'])
df_2019
```

Out[165]:

	lc	count_2019	fraction_2019
0	11	155179	0.355228
1	24	3913880	8.959452
2	31	685585	1.569406
3	41	8	0.000018
4	42	1245304	2.850686
5	43	1120	0.002564
6	52	30186997	69.102515
7	71	4284805	9.808554
8	81	37785	0.086495
9	82	2822333	6.460739
10	90	281933	0.645386
11	95	69439	0.158956

In [166]:

```
# Check there are no more grid cells classified as normal shrubs
unique, counts = np.unique(nlcd_2001_phoenix, return_counts=True)

# Convert to DataFrame
df_2001 = pd.DataFrame(list(zip(unique, counts, (counts/land_pixels)*100)),
                       columns=['lc', 'count_2001', 'fraction_2001'])
df_2001
```

Out[166]:

	lc	count_2001	fraction_2001
0	11	143014	0.327380
1	24	3004838	6.878520
2	31	706318	1.616867
3	42	1497913	3.428945
4	43	1603	0.003670
5	52	30851488	70.623634
6	71	4039637	9.247328
7	81	41732	0.095531
8	82	3046041	6.972840
9	90	289378	0.662429
10	95	62406	0.142857

In [167]:

```
df_2001
```

Out[167]:

	lc	count_2001	fraction_2001
0	11	143014	0.327380
1	24	3004838	6.878520
2	31	706318	1.616867
3	42	1497913	3.428945
4	43	1603	0.003670
5	52	30851488	70.623634
6	71	4039637	9.247328
7	81	41732	0.095531
8	82	3046041	6.972840
9	90	289378	0.662429
10	95	62406	0.142857

In [168]:
```
df = pd.merge(df_2001, df_2019, on=['lc'])
df
```

Out[168]:

	lc	count_2001	fraction_2001	count_2019	fraction_2019
0	11	143014	0.327380	155179	0.355228
1	24	3004838	6.878520	3913880	8.959452
2	31	706318	1.616867	685585	1.569406
3	42	1497913	3.428945	1245304	2.850686
4	43	1603	0.003670	1120	0.002564
5	52	30851488	70.623634	30186997	69.102515
6	71	4039637	9.247328	4284805	9.808554
7	81	41732	0.095531	37785	0.086495
8	82	3046041	6.972840	2822333	6.460739
9	90	289378	0.662429	281933	0.645386
10	95	62406	0.142857	69439	0.158956

In [169]:
```
df['change'] = (df['count_2019'] - df['count_2001'])
df['change_percent'] = (((df['count_2019'] - df['count_2001']) / df['count_2001']) * 100)
df
```

Out[169]:

	lc	count_2001	fraction_2001	count_2019	fraction_2019	change	change_percent
0	11	143014	0.327380	155179	0.355228	12165	8.506160
1	24	3004838	6.878520	3913880	8.959452	909042	30.252613
2	31	706318	1.616867	685585	1.569406	-20733	-2.935363
3	42	1497913	3.428945	1245304	2.850686	-252609	-16.864064
4	43	1603	0.003670	1120	0.002564	-483	-30.131004
5	52	30851488	70.623634	30186997	69.102515	-664491	-2.153838
6	71	4039637	9.247328	4284805	9.808554	245168	6.069060
7	81	41732	0.095531	37785	0.086495	-3947	-9.457970
8	82	3046041	6.972840	2822333	6.460739	-223708	-7.344222
9	90	289378	0.662429	281933	0.645386	-7445	-2.572760
10	95	62406	0.142857	69439	0.158956	7033	11.269750

In [170]:
```
dev_land_change = (df[df['lc'] == 24])
```

In [171]:
```
dev_land_change
```

Out[171]:

	lc	count_2001	fraction_2001	count_2019	fraction_2019	change	change_percent
1	24	3004838	6.87852	3913880	8.959452	909042	30.252613

In [172]:
```
dev_land_change['change_percent']
```

Out[172]:
```
1    30.252613
Name: change_percent, dtype: float64
```

In [173]:
```
cult_crop_change = (df[df['lc'] == 82])
```

In [174]:
```
cult_crop_change
```

Out[174]:

	lc	count_2001	fraction_2001	count_2019	fraction_2019	change	change_percent
8	82	3046041	6.97284	2822333	6.460739	-223708	-7.344222

In [175]:
```
cult_crop_change['change_percent']
```

Out[175]:
```
8   -7.344222
Name: change_percent, dtype: float64
```

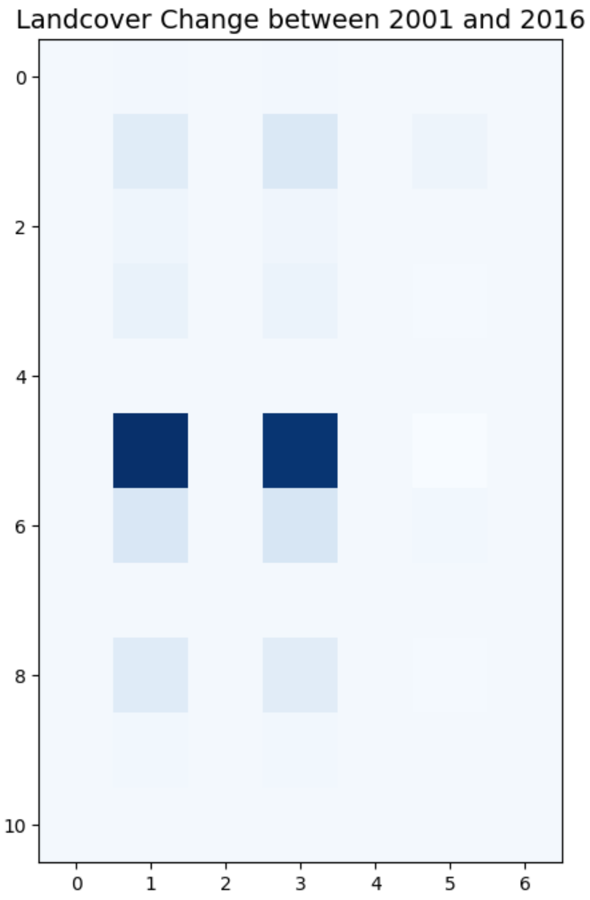
In [186]:
```
mask = nlcd_2019[nlcd_2001 == 24]
unique, counts = np.unique(mask, return_counts=True)
change_df = pd.DataFrame(list(zip(unique, counts, (counts/mask.shape[0])*100)),
                 columns=['lc', 'count', 'fraction'])
change_df
```

Out[186]:

	lc	count	fraction
0	11	8	0.002481
1	24	322344	99.975808
2	52	51	0.015818
3	82	19	0.005893

In [188]:
```
fig, ax = plt.subplots(figsize=(16,8))
im = ax.imshow(df.astype(int), cmap='Blues')
ax.set_title("Landcover Change between 2001 and 2016", fontsize=14)
```

Out[188]:
```
Text(0.5, 1.0, 'Landcover Change between 2001 and 2016')
```



In [ ]: