In [54]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import numpy as np
import rasterio
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from matplotlib.colors import LinearSegmentedColormap
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import DBSCAN
from sklearn.metrics import silhouette_score
warnings.filterwarnings("ignore", category=UserWarning)
```

In [15]:
```
pt = pd.read_csv("data/PresentTopSoil.csv")
ps = pd.read_csv("data/PresentSubSoil.csv")
```

In [16]:
```
pt
```

Out[16]:

	Location	LandUse	Latitude	Longitude	Stock	SOCT	pH	TN	Clay	SB
0	Ajmiriganj	Irrigated Boro-Fallow	24.501500	91.376917	351.12	44.63	4.10	0.15	55.44	1.5
1	Balaganj	Jute-T Aman-Fallow/Rabi Crops	24.431333	91.227111	148.71	15.90	5.39	0.12	49.63	2.0
2	Goainghat	Rabi Crops-Fallow	25.022900	91.197500	149.65	19.37	5.86	0.11	67.56	1.3
3	Hakaluki	Boro-Fallow	24.301139	91.534306	109.14	16.70	4.34	0.11	33.56	1.6
4	Kanairghat	Grassland	24.282694	91.244306	151.46	18.37	4.02	0.13	35.96	1.6
5	Phagu	Boro-Fallow	24.902556	91.005250	98.34	14.08	5.25	0.10	48.88	1.6
6	Sarail	Irrigated Boro	24.103722	91.144583	96.25	13.10	4.19	0.12	37.27	1.8
7	Sulla	Boro-Fallow	24.592194	91.300389	225.03	24.45	4.31	0.13	31.69	1.8
8	Terchibari	Boro-Fallow	24.810611	91.380500	128.83	19.17	5.24	0.13	72.20	1.2

In [17]:
```
biomass_agb = pd.read_csv("geodata/biomass_agb.csv")
biomass_nd = pd.read_csv("geodata/biomass_ndvi.csv")
hydro_veg = pd.read_csv("geodata/Hydro_Veg.csv")
```

```python
indices = pd.read_csv("geodata/indices_1985_2025.csv")
lulc = pd.read_csv("geodata/LULCAreaCover.csv")
ndvic = pd.read_csv("geodata/ndvi_changes.csv")
```

In [18]:
```python
biomass_agb.head(), biomass_nd.head()
```

Out[18]:
```
(   Unnamed: 0  year  mean_agb_Mg_per_ha  total_agb_Mg
 0           0  1985                 NaN           NaN
 1           1  1986                 NaN           NaN
 2           2  1987                 NaN           NaN
 3           3  1988                 NaN           NaN
 4           4  1989                 NaN           NaN,
    Unnamed: 0  year  mean_ndvi  biomass_t_ha_yr
 0           0  1985        NaN              NaN
 1           1  1986        NaN              NaN
 2           2  1987        NaN              NaN
 3           3  1988   0.097779         0.385825
 4           4  1989   0.119175         0.333500)
```

In [19]:
```python
hydro_veg.head(), indices.head()
```

Out[19]:
```
(   Unnamed: 0    Location  Water Area Change (m²)  Flooded Area Change (m
²)  \
 0           0  Ajmiriganj           -1.801432e+09                       0.
0
 1           1    Balaganj           -1.801236e+09                       0.
0
 2           2   Goainghat           -1.801432e+09                       0.
0
 3           3    Hakaluki           -1.801432e+09                       0.
0
 4           4  Kanairghat           -1.801432e+09                       0.
0

    Flood Area Change (m²)  Urban Area Change (m²)  \
 0            -453631400.0             674840200.0
 1            -453561100.0             674877600.0
 2            -453631400.0             674916100.0
 3            -453631400.0             674816400.0
 4            -453624200.0             674876900.0

    Vegetation Area Change (m²)  SOC%_1985  SOC%_2025  Stock_1985  ...  \
 0                 -243894900.0      2.630      2.630      351.12  ...
 1                 -243896300.0      0.960      0.960      148.71  ...
 2                 -243895900.0      1.100      1.100      149.65  ...
 3                 -243893000.0      1.325      1.325      109.14  ...
 4                 -243895900.0      1.220      1.220      151.46  ...

    pH_1985  pH_2025  TN_1985  TN_2025  SBD_1985  SBD_2025  CEC_1985  CEC_2
025  \
 0     4.10     4.10     0.15     0.15      1.56      1.56    198.75    19
8.75
 1     5.39     5.39     0.12     0.12      2.05      2.05    206.90    20
6.90
 2     5.86     5.86     0.11     0.11      1.37      1.37    271.50    27
1.50
 3     4.34     4.34     0.11     0.11      1.66      1.66    167.48    16
7.48
 4     4.02     4.02     0.13     0.13      1.69      1.69    180.39    18
0.39

    Clay_1985  Clay_2025
 0      55.44      55.44
 1      49.63      49.63
 2      67.56      67.56
 3      33.56      33.56
 4      35.96      35.96

 [5 rows x 21 columns],
    system:index     mean_bui     mean_lst    mean_ndvi    mean_ndwi    yea
r  \
 0             0 -9999.000000 -9999.000000 -9999.000000 -9999.000000  1985.
0
 1             1 -9999.000000 -9999.000000 -9999.000000 -9999.000000  1986.
0
 2             2 -9999.000000 -9999.000000 -9999.000000 -9999.000000  1987.
0
```

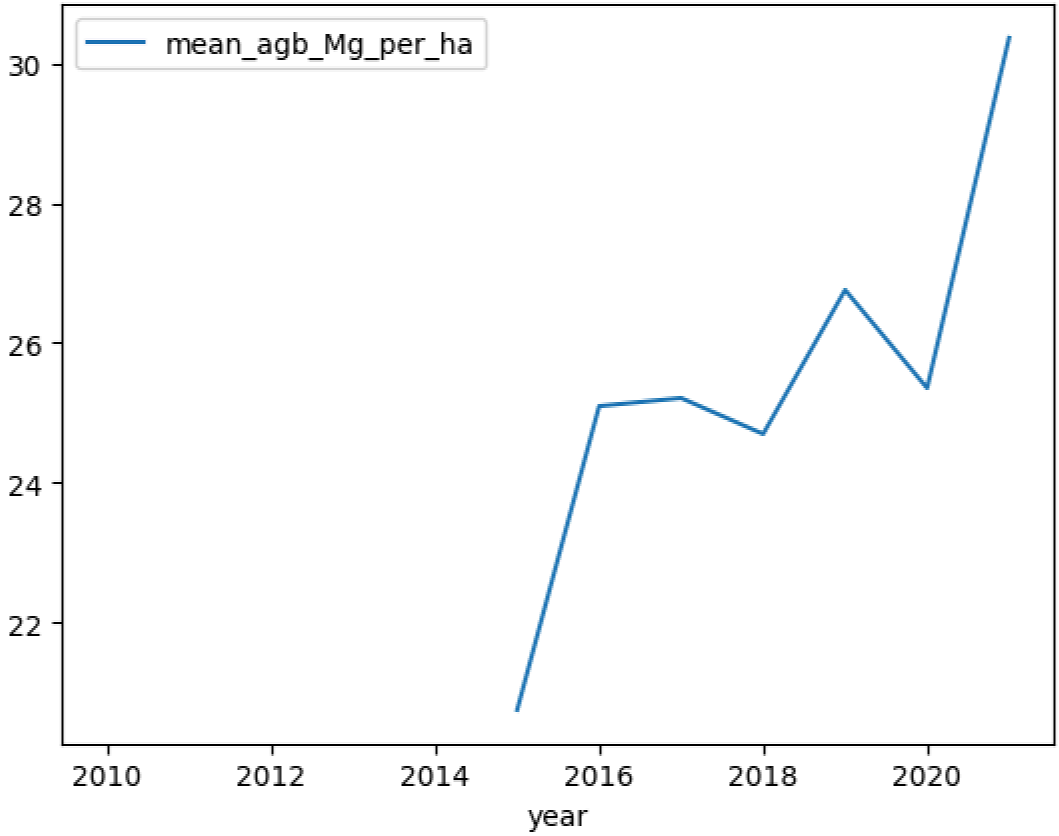
```
 3              3  4096.382285  4096.382285     0.097779    -0.019772  1988.
0
 4              4  4076.217308  4076.217308     0.119175    -0.046192  1989.
0

                                       .geo
 0  {"type":"MultiPoint","coordinates":[]}
 1  {"type":"MultiPoint","coordinates":[]}
 2  {"type":"MultiPoint","coordinates":[]}
 3  {"type":"MultiPoint","coordinates":[]}
 4  {"type":"MultiPoint","coordinates":[]}  )
```

In [20]:
```
lulc.head(), ndvic.head()
```

Out[20]:
```
(   Unnamed: 0  Year  Water Area (m²)  Flood Area (m²)  Flooded Area (m²)  \
 0           0  2017     4.141525e+09      604154200.0                0.0
 1           1  2018     2.427318e+09      237302100.0                0.0
 2           2  2019     2.083190e+09      257137400.0                0.0
 3           3  2020     2.841814e+09      179889000.0                0.0
 4           4  2021     1.534033e+09      225442200.0                0.0

    Vegetation Area (m²)  Urban Area (m²)
 0          1.120824e+09     8.822642e+08
 1          9.827860e+08     1.114500e+09
 2          9.597035e+08     1.260649e+09
 3          9.010560e+08     1.341468e+09
 4          8.630248e+08     1.532445e+09  ,
    Unnamed: 0  year  mean_ndvi
 0           3  1988   0.097771
 1           4  1989   0.119169
 2           5  1990   0.091069
 3           6  1991   0.079623
 4           7  1992   0.080583)
```

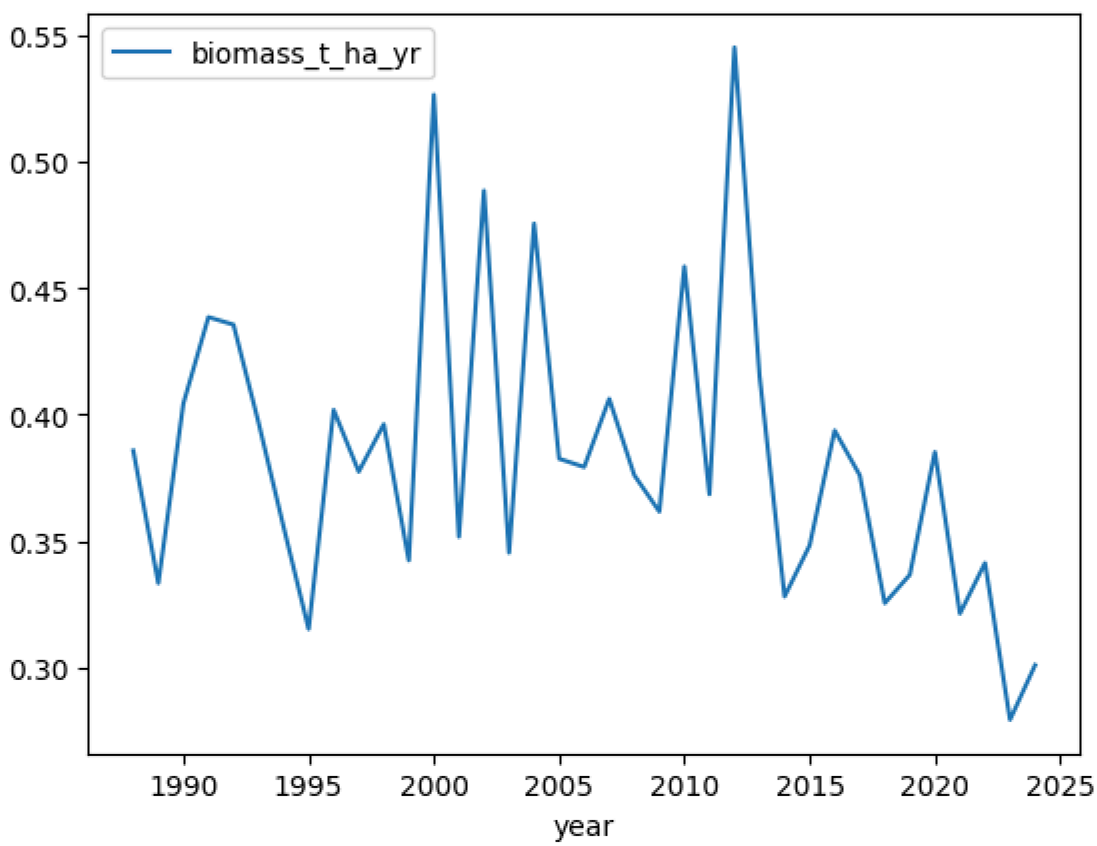
In [21]:
```
biomass_agb.plot(x = "year", y = "mean_agb_Mg_per_ha")
```

Out[21]:
```
<Axes: xlabel='year'>
```



In [22]:
```
biomass_nd.plot(x = "year", y = "biomass_t_ha_yr")
```

Out[22]: <Axes: xlabel='year'>



In [23]:
```
hydro_veg
```

Out[23]:

	Unnamed: 0	Location	Water Area Change (m²)	Flooded Area Change (m²)	Flood Area Change (m²)	Urban Area Change (m²)	Veg Area (
0	0	Ajmiriganj	-1.801432e+09	0.0	-453631400.0	674840200.0	-24389
1	1	Balaganj	-1.801236e+09	0.0	-453561100.0	674877600.0	-24389
2	2	Goainghat	-1.801432e+09	0.0	-453631400.0	674916100.0	-24389
3	3	Hakaluki	-1.801432e+09	0.0	-453631400.0	674816400.0	-24389
4	4	Kanairghat	-1.801432e+09	0.0	-453624200.0	674876900.0	-24389
5	5	Phagu	-1.801418e+09	0.0	-453631400.0	674829500.0	-24386
6	6	Sarail	-1.801236e+09	0.0	-453631400.0	674904800.0	-24389
7	7	Sulla	-1.801442e+09	0.0	-453631400.0	674839100.0	-2438
8	8	Terchibari	-1.801093e+09	0.0	-453606200.0	674923600.0	-24389

9 rows × 21 columns

In [24]:
```
indices.drop(columns="system:index", inplace=True)
indices.columns
```

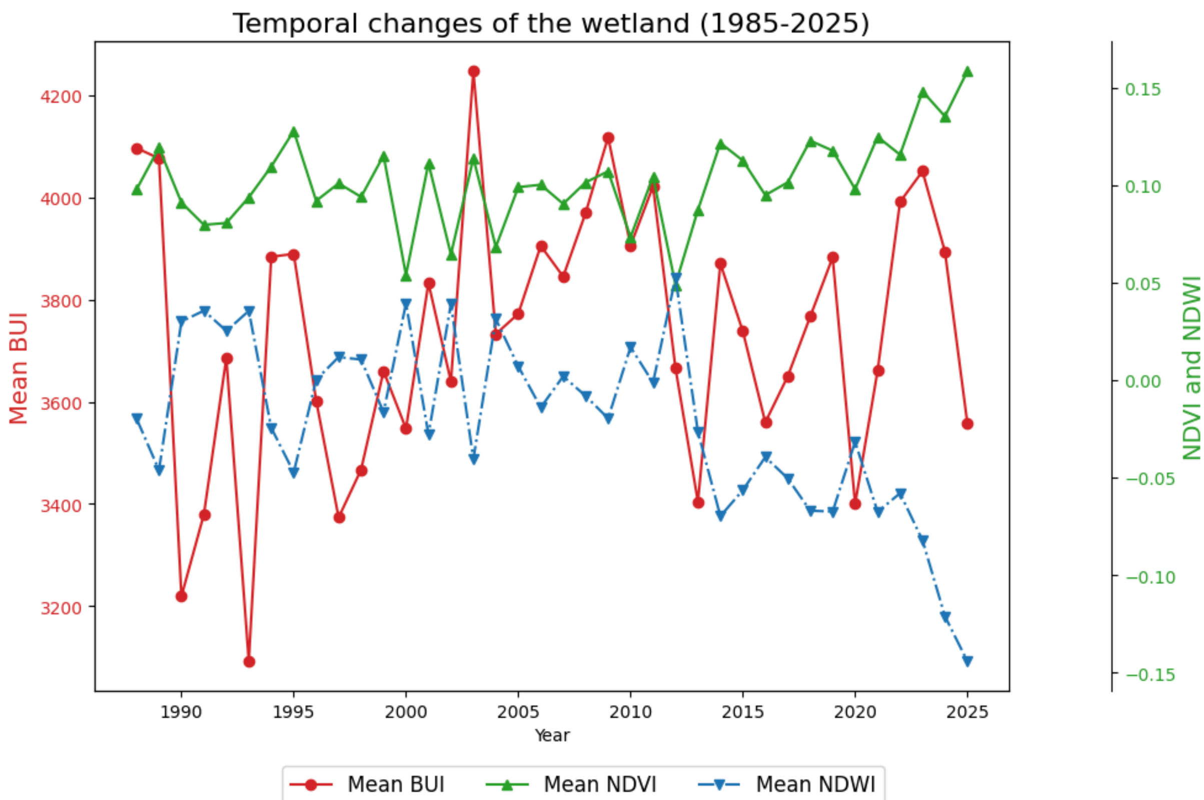
Out[24]:
```
Index(['mean_bui', 'mean_lst', 'mean_ndvi', 'mean_ndwi', 'year', '.geo'], d
type='object')
```

In [25]:
```
indices.replace(to_replace=-9999, value=0, inplace=True)
```

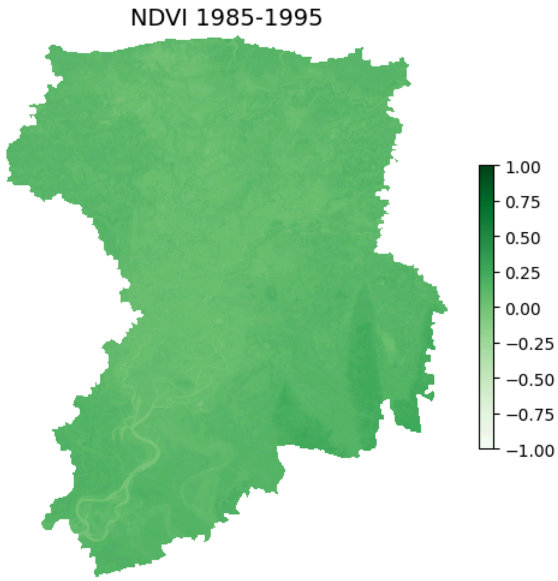
In [26]:
```
indices = indices[3:]
```

In [28]:
```
fig, ax1 = plt.subplots(figsize=(10, 6))

color_bui = 'tab:red'
color_lst = 'tab:red'

df = indices.copy()

ax1.set_xlabel('Year', fontsize=10)
ax1.set_ylabel('Mean BUI', color=color_bui, fontsize=14)
ax1.plot(df['year'], df['mean_bui'], color=color_bui, marker='o', label='Mea
ax1.tick_params(axis='y', labelcolor=color_bui)

ax2 = ax1.twinx()
ax2.spines['right'].set_position(('outward', 60))  # Offset third axis
ax2.set_ylabel('NDVI and NDWI', color='tab:green', fontsize=14)
ax2.plot(df['year'], df['mean_ndvi'], color='tab:green', marker='^', label='
ax2.plot(df['year'], df['mean_ndwi'], color='tab:blue', marker='v', label='M
ax2.tick_params(axis='y', labelcolor='tab:green')
```

```
fig.tight_layout()
fig.legend(loc='upper center', bbox_to_anchor=(0.48, 0.01), ncol=4, fontsize
plt.title('Temporal changes of the wetland (1985-2025)', fontsize=16)

plt.savefig("Figure/1985_2025_ChangesIndices.png", dpi=500)
plt.show()
```



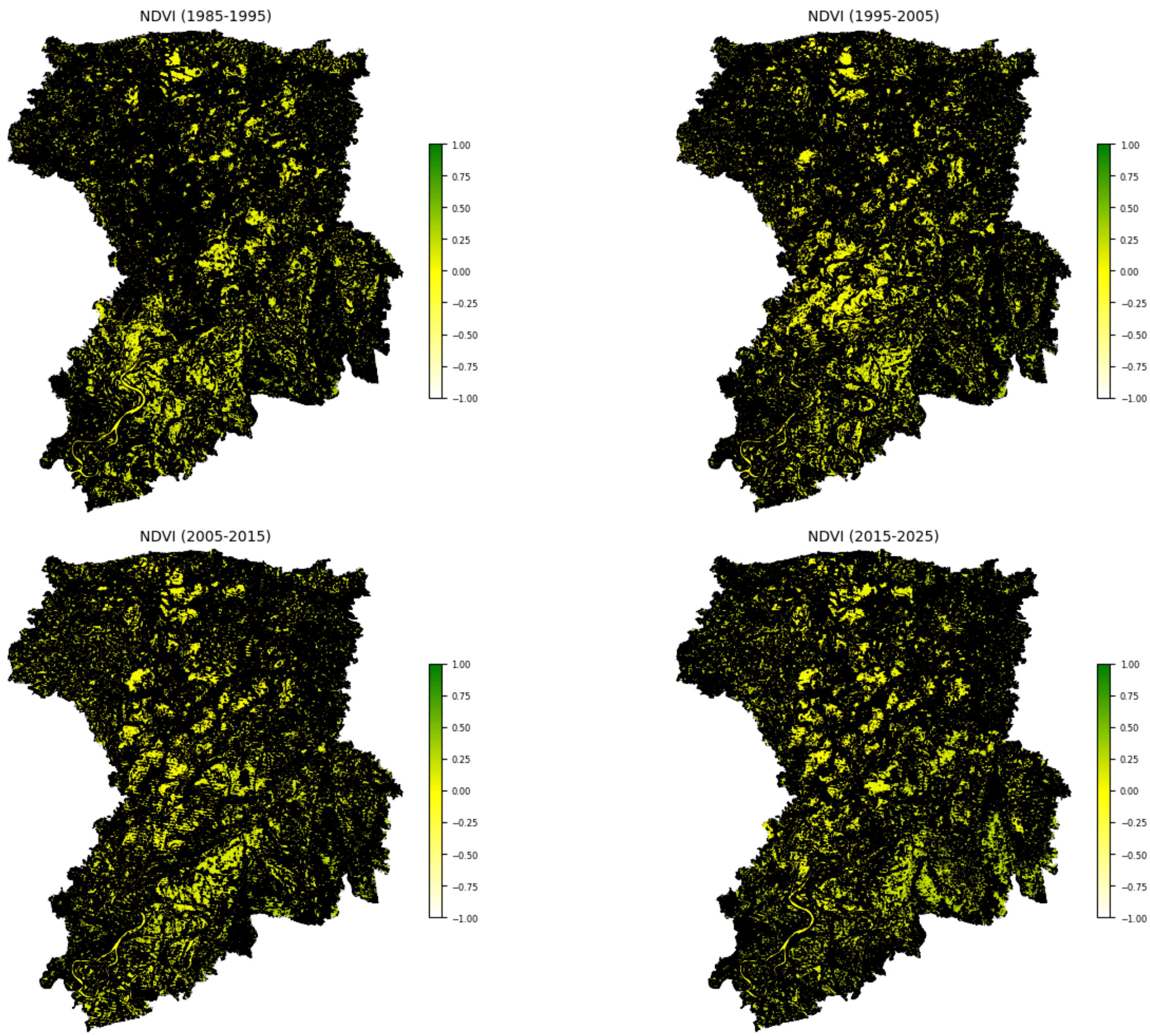
In [63]:
```
files = [
    'gis/NDVI_1985-1995.tif',
    'gis/NDVI_1996-2005.tif',
    'gis/NDVI_2006-2015.tif',
    'gis/NDVI_2016-2025.tif'
]

fig, axs = plt.subplots(2, 2, figsize=(12, 10))

titles = ['NDVI 1985-1995', 'NDVI 1996-2005', 'NDVI 2006-2015', 'NDVI 2016-2

for i, file in enumerate(files):
    with rasterio.open(file) as src:
        ndvi_data = src.read(1)
        ax = axs[i // 2, i % 2]
        cax = ax.imshow(ndvi_data, cmap='Greens', vmin=-1, vmax=1)
        ax.set_title(titles[i], fontsize=14)
        ax.axis('off')
        fig.colorbar(cax, ax=ax, fraction=0.02, pad=0.04)

plt.tight_layout()
plt.show()
```



In [6]:

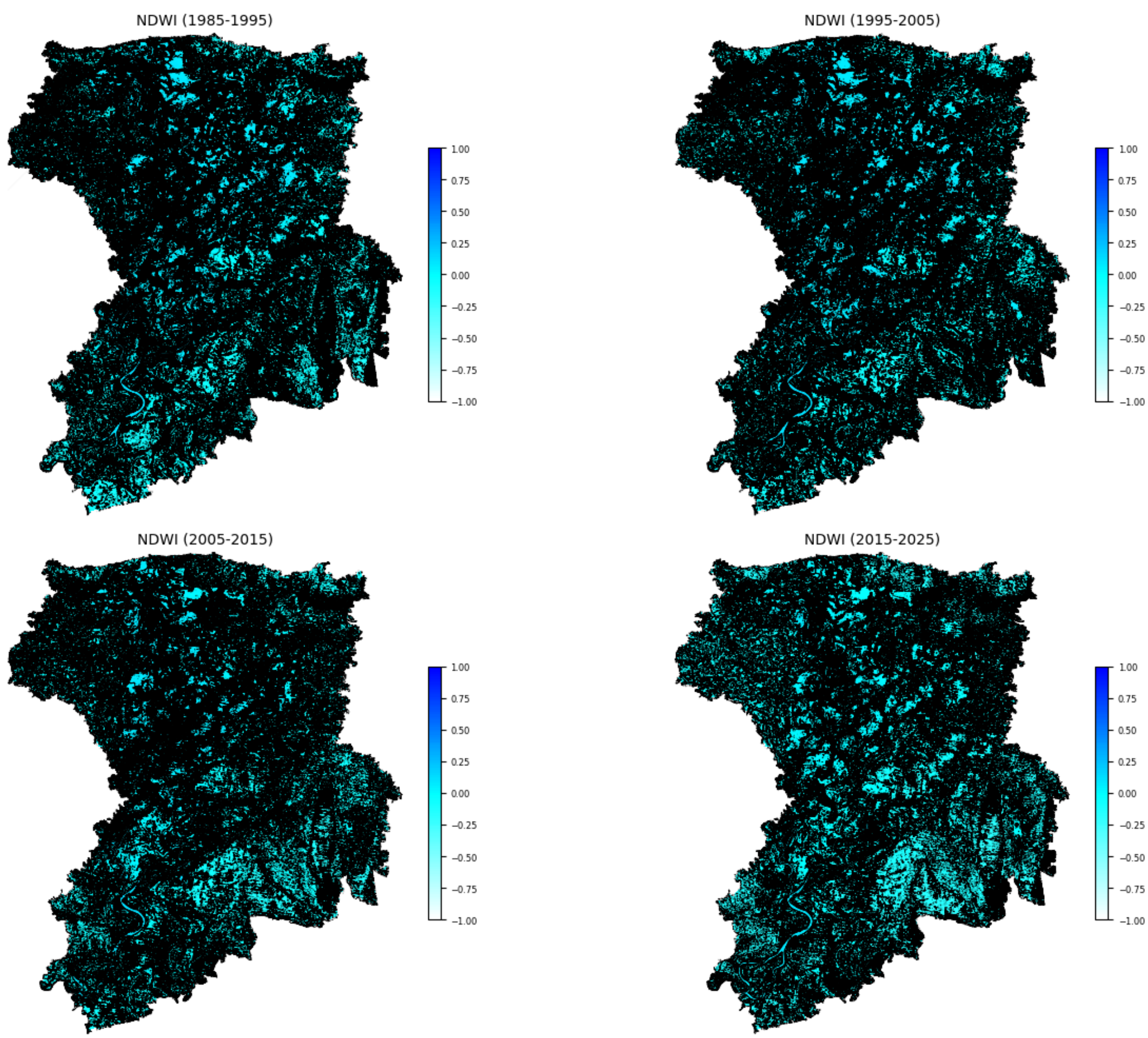
```
ndvi_files = [
    'gis/NDVI_1985-1995.tif',
    'gis/NDVI_1996-2005.tif',
    'gis/NDVI_2006-2015.tif',
    'gis/NDVI_2016-2025.tif'
]

colors = ["white", "yellow", "green"]

n_bins = 100
cmap = LinearSegmentedColormap.from_list("WhiteYellowGreen", colors, N=n_bir

years = ["1985-1995","1995-2005","2005-2015","2015-2025"]

plt.figure(figsize=(15, 10))

for i, ndvi_file in enumerate(ndvi_files):
    with rasterio.open(ndvi_file) as src:
        ndvi_data = src.read(1)
```

```
    ndvi_data = np.ma.masked_equal(ndvi_data, 0)
    ax = plt.subplot(2, 2, i+1)
    img = ax.imshow(ndvi_data, cmap=cmap, vmin=-1, vmax=1)
    ax.set_title(f'NDVI ({years[i]})', fontsize=10)
    ax.contour(ndvi_data, colors='black', linewidths=0.5)
    ax.set_axis_off()
    cbar = plt.colorbar(img, ax=ax, fraction=0.015, pad=0.03)
    cbar.ax.tick_params(labelsize=6)

plt.tight_layout()
plt.savefig("Figure/NDVIChanges.png", dpi=500)
plt.show()
```

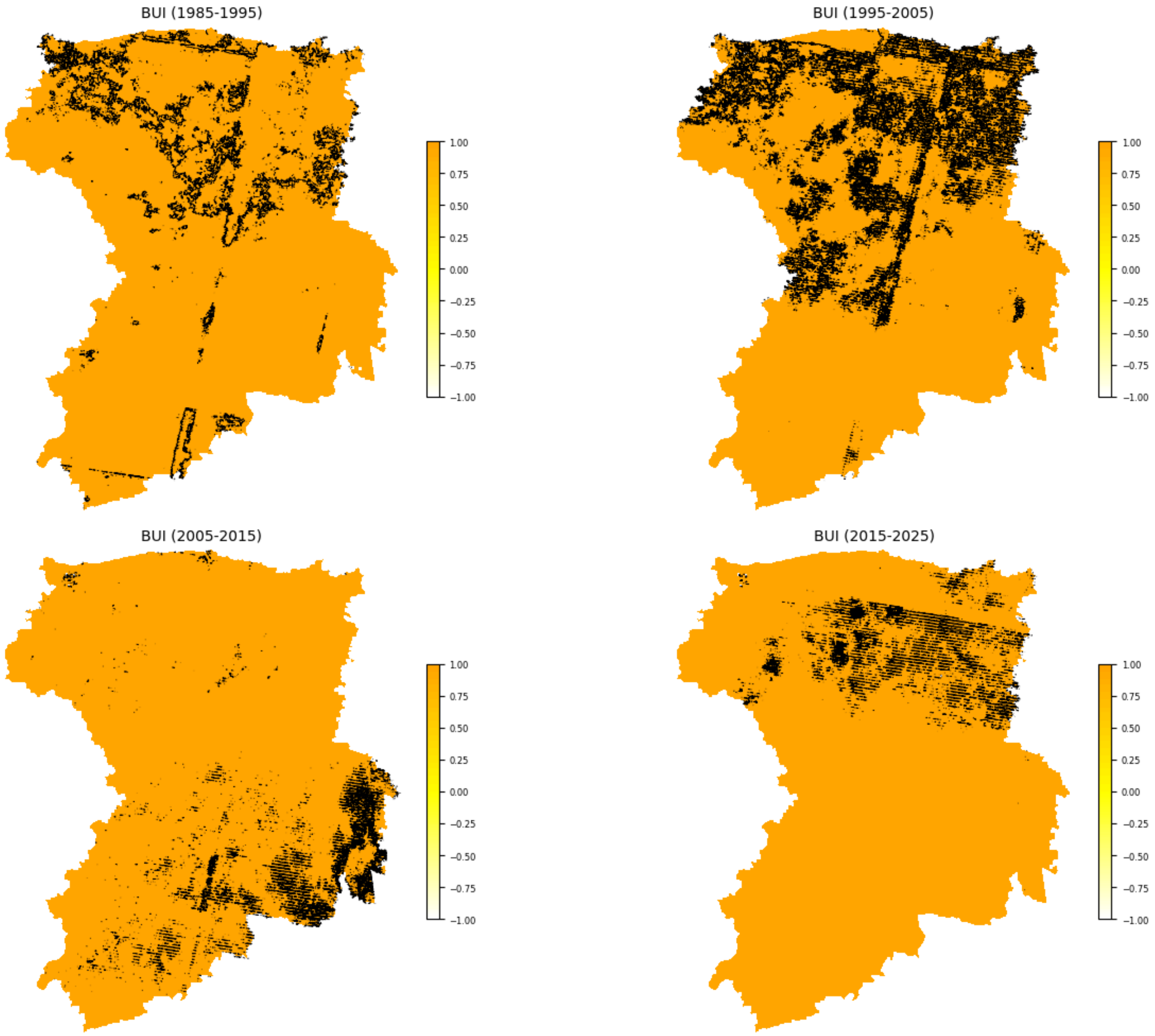


In [8]:
```
ndvi_files = [
    'gis/NDWI_1985-1995.tif',
    'gis/NDWI_1996-2005.tif',
    'gis/NDWI_2006-2015.tif',
    'gis/NDWI_2016-2025.tif'
]

colors = ["white", "cyan", "blue"]
n_bins = 100
cmap = LinearSegmentedColormap.from_list("WhiteCyanBlue", colors, N=n_bins)
```

```python
years = ["1985-1995","1995-2005","2005-2015","2015-2025"]
plt.figure(figsize=(15, 10))
for i, ndvi_file in enumerate(ndvi_files):
    with rasterio.open(ndvi_file) as src:
        ndvi_data = src.read(1)

    ndvi_data = np.ma.masked_equal(ndvi_data, 0)

    ax = plt.subplot(2, 2, i+1)
    img = ax.imshow(ndvi_data, cmap=cmap, vmin=-1, vmax=1)
    ax.set_title(f'NDWI ({years[i]})', fontsize=10)

    ax.contour(ndvi_data, colors='black', linewidths=0.5)
    ax.set_axis_off()

    cbar = plt.colorbar(img, ax=ax, fraction=0.015, pad=0.03)
    cbar.ax.tick_params(labelsize=6)

plt.tight_layout()
plt.savefig("Figure/NDWIChanges.png", dpi=500)
plt.show()
```

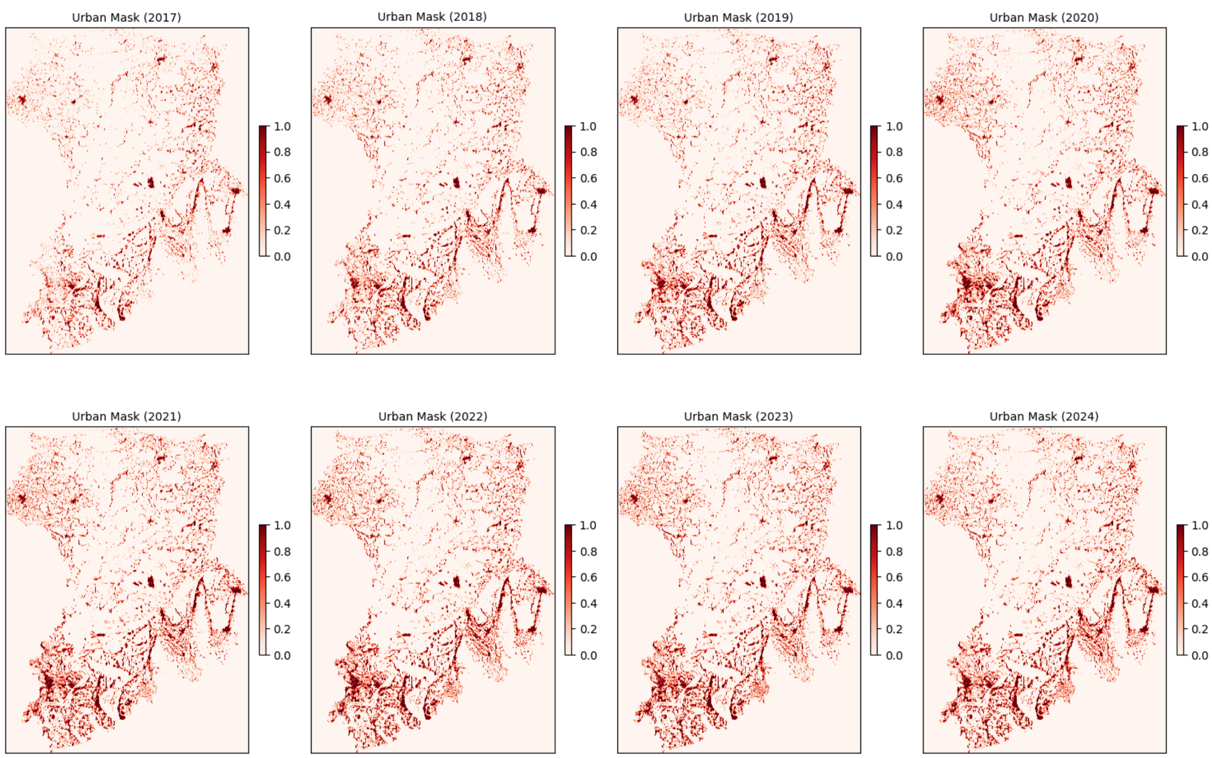


In [10]:
```python
ndvi_files = [
    'gis/BUI_1985-1995.tif',
    'gis/BUI_1996-2005.tif',
    'gis/BUI_2006-2015.tif',
    'gis/BUI_2016-2025.tif'
]

colors = ["white", "yellow", "orange"]

n_bins = 100
cmap = LinearSegmentedColormap.from_list("WhiteYellowOrange", colors, N=n_bi

years = ["1985-1995","1995-2005","2005-2015","2015-2025"]

plt.figure(figsize=(15, 10))

for i, ndvi_file in enumerate(ndvi_files):
    with rasterio.open(ndvi_file) as src:
        ndvi_data = src.read(1)

    ndvi_data = np.ma.masked_equal(ndvi_data, 0)

    ax = plt.subplot(2, 2, i+1)
    img = ax.imshow(ndvi_data, cmap=cmap, vmin=-1, vmax=1)
    ax.set_title(f'BUI ({years[i]})', fontsize=10)

    ax.contour(ndvi_data, colors='black', linewidths=0.5)
    ax.set_axis_off()

    cbar = plt.colorbar(img, ax=ax, fraction=0.015, pad=0.03)
    cbar.ax.tick_params(labelsize=6)

plt.tight_layout()
plt.savefig("Figure/BUIChanges.png", dpi=500)
plt.show()
```



In [13]:
```
lulc_files = [
    'gis/LULC2017c.tif',
    'gis/LULC2018c.tif',
    'gis/LULC2019c.tif',
    'gis/LULC2020c.tif',
    'gis/LULC2021c.tif',
    'gis/LULC2022c.tif',
    'gis/LULC2023c.tif',
    'gis/LULC2024c.tif',
]

WATER_CLASS = 1
VEGETATION_CLASS = 2
FLOODED_CLASS = 9
FLOOD_CLASS = 4
URBAN_CLASS = 7

plt.figure(figsize=(15, 10))

for i, lulc_file in enumerate(lulc_files):
    with rasterio.open(lulc_file) as src:
        lulc_data = src.read(1)
```

```
    urban_mask = (lulc_data == URBAN_CLASS)

    ax = plt.subplot(2, 4, i+1)  # 2 rows, 4 columns, each subplot
    img = ax.imshow(urban_mask, cmap='Reds')
    ax.set_title(f'Urban Mask ({2017 + i})', fontsize=10)

    ax.set_xticks([])
    ax.set_yticks([])

    plt.colorbar(img, ax=ax, fraction=0.025, pad=0.04)

plt.tight_layout()
plt.savefig("Figure/Hydro_Urban.png", dpi=500)
plt.show()
```



In [44]:
```
stock = pt.Stock + ps.Stock
stock.index = pt.Location
stock
```

Out[44]:
```
Location
Ajmiriganj    702.24
Balaganj      297.42
Goainghat     299.30
Hakaluki      218.28
Kanairghat    302.92
Phagu         196.68
Sarail        192.50
Sulla         450.06
Terchibari    257.66
Name: Stock, dtype: float64
```

In [38]:
```
hydro_veg.drop(columns=["Flooded Area Change (m²)", "Unnamed: 0", "Location"
hydro_veg
```

Out[38]:

	Water Area Change (m²)	Flood Area Change (m²)	Urban Area Change (m²)	Vegetation Area Change (m²)	SOC%_1985	SOC%_20
0	-1.801432e+09	-453631400.0	674840200.0	-243894900.0	2.630	2.6
1	-1.801236e+09	-453561100.0	674877600.0	-243896300.0	0.960	0.9
2	-1.801432e+09	-453631400.0	674916100.0	-243895900.0	1.100	1.1
3	-1.801432e+09	-453631400.0	674816400.0	-243893000.0	1.325	1.3
4	-1.801432e+09	-453624200.0	674876900.0	-243895900.0	1.220	1.2
5	-1.801418e+09	-453631400.0	674829500.0	-243863300.0	0.635	0.6
6	-1.801236e+09	-453631400.0	674904800.0	-243895900.0	1.035	1.0
7	-1.801442e+09	-453631400.0	674839100.0	-243895100.0	1.425	1.4
8	-1.801093e+09	-453606200.0	674923600.0	-243895900.0	1.140	1.1

In [50]:

```
df85 = hydro_veg[['Water Area Change (m²)', 'Flood Area Change (m²)', 'Urbar
                  'Vegetation Area Change (m²)', 'SOC%_1985', 'Stock_1985', 'p
                  'SBD_1985', 'CEC_1985', 'Clay_1985']]

df25 = hydro_veg[['Water Area Change (m²)', 'Flood Area Change (m²)', 'Urbar
                  'Vegetation Area Change (m²)', 'SOC%_2025', 'pH_2025', 'TN_2
                  'SBD_2025', 'CEC_2025', 'Clay_2025', 'Stock']]
```

In [51]:

```
df25["Stock"] = stock.values
df25.corr()
```

```
/var/folders/nk/5ry1y2d128x8fgnl550m4c_h0000gp/T/ipykernel_10775/1289109096.
py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df25["Stock"] = stock.values
```

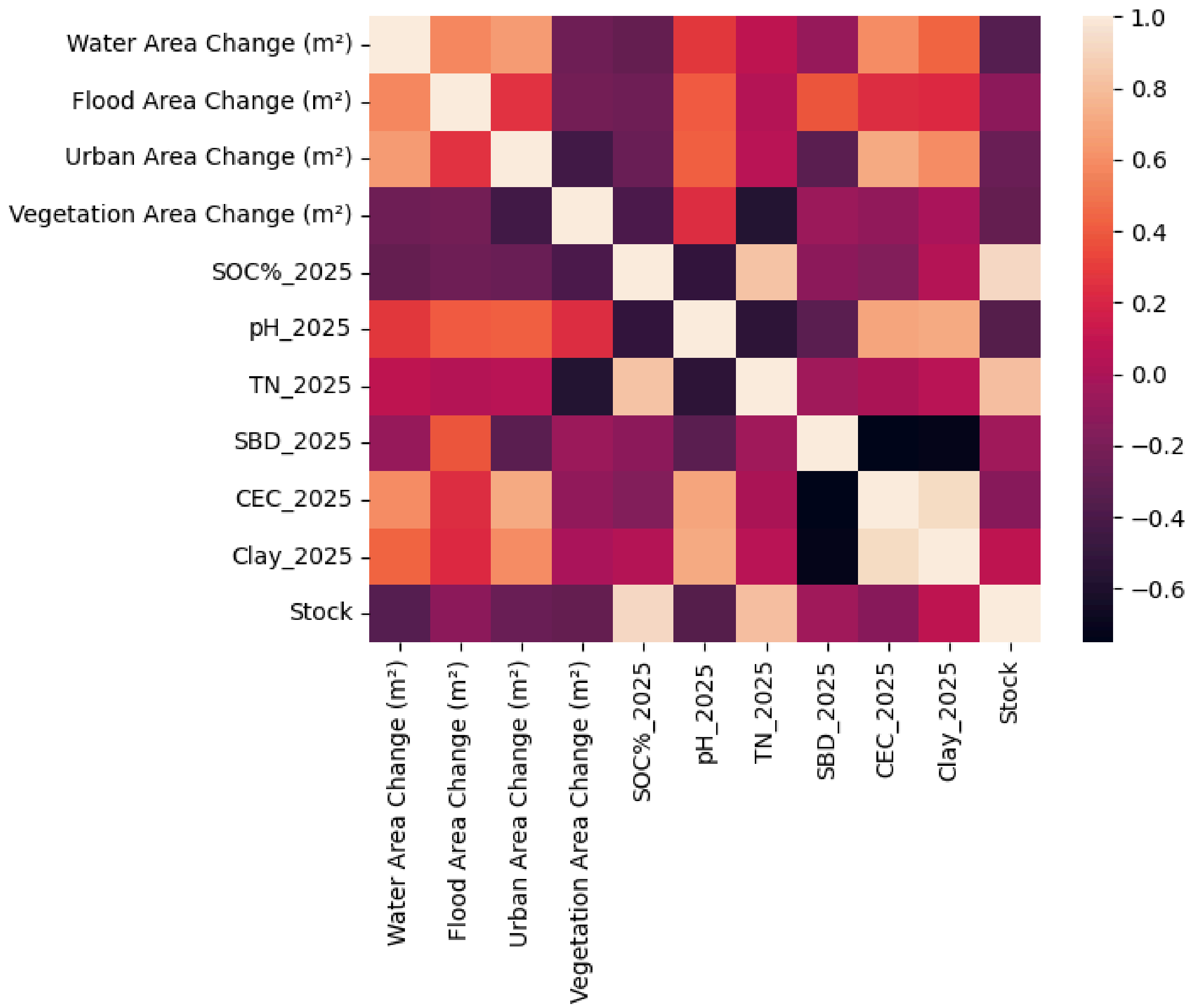
Out[51]:

	Water Area Change (m²)	Flood Area Change (m²)	Urban Area Change (m²)	Vegetation Area Change (m²)	SOC%_2025	pH_2025
Water Area Change (m²)	1.000000	0.573615	0.652505	-0.237553	-0.288625	0.280374
Flood Area Change (m²)	0.573615	1.000000	0.266292	-0.221624	-0.246244	0.406621
Urban Area Change (m²)	0.652505	0.266292	1.000000	-0.441788	-0.272020	0.423422
Vegetation Area Change (m²)	-0.237553	-0.221624	-0.441788	1.000000	-0.399503	0.238732
SOC%_2025	-0.288625	-0.246244	-0.272020	-0.399503	1.000000	-0.515605
pH_2025	0.280374	0.406621	0.423422	0.238732	-0.515605	1.000000
TN_2025	0.089248	0.034240	0.063594	-0.569955	0.823218	-0.531426
SBD_2025	-0.086337	0.382489	-0.327582	-0.072338	-0.129697	-0.328970
CEC_2025	0.597194	0.229355	0.717392	-0.094457	-0.159026	0.687553
Clay_2025	0.431829	0.226821	0.587549	-0.015874	0.036066	0.713822
Stock	-0.341031	-0.121136	-0.270824	-0.285629	0.917883	-0.348410

In [52]:

```
sns.heatmap(df25.corr())
```

Out[52]: <Axes: >



In [53]:
```
df25.columns
```

Out[53]:
```
Index(['Water Area Change (m²)', 'Flood Area Change (m²)',
       'Urban Area Change (m²)', 'Vegetation Area Change (m²)', 'SOC%_202
5',
       'pH_2025', 'TN_2025', 'SBD_2025', 'CEC_2025', 'Clay_2025', 'Stock'],
      dtype='object')
```

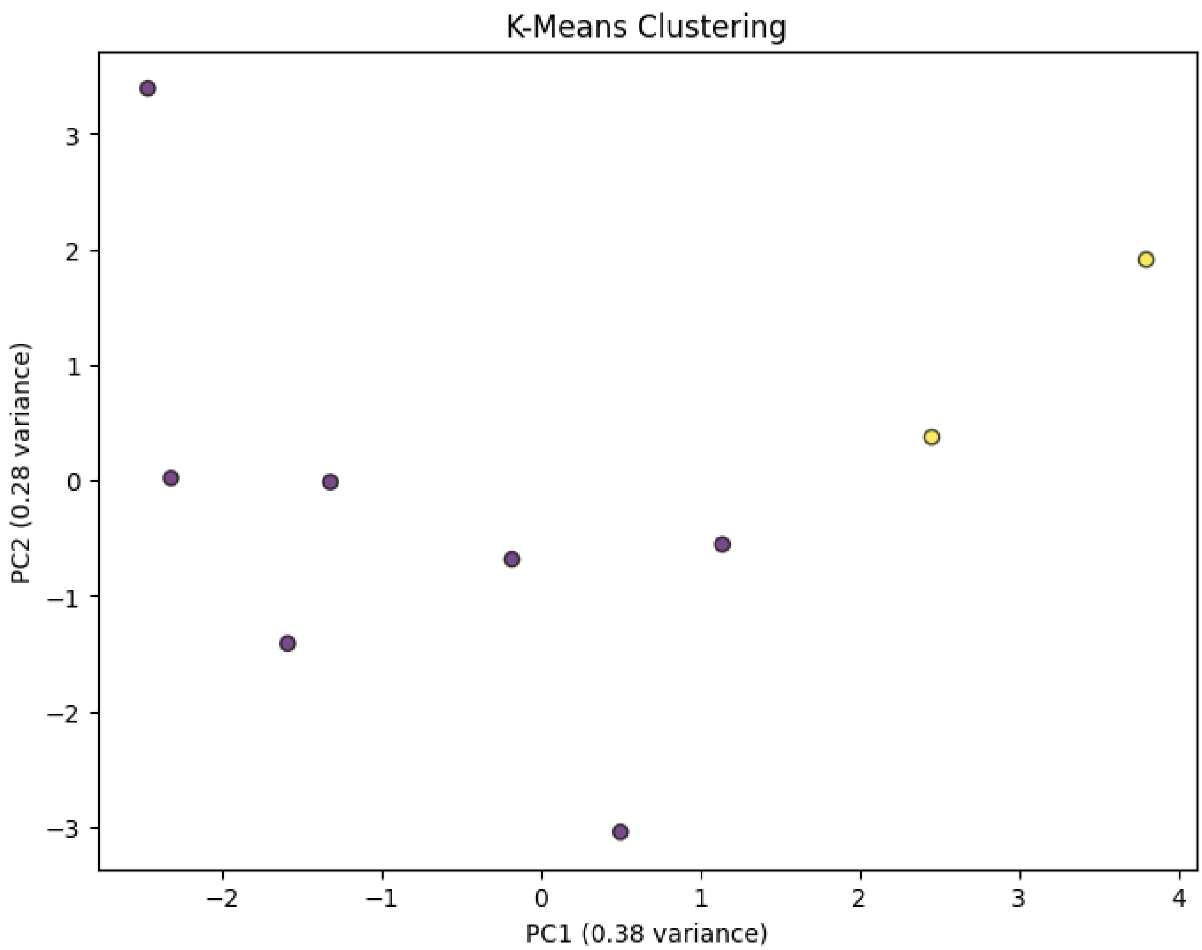
In [62]:
```
df = df25.copy()

features = df.drop(columns='Stock')
scaler = StandardScaler()
scaled_data = scaler.fit_transform(features)

pca = PCA(n_components=2)
pca_result = pca.fit_transform(scaled_data)

loadings = pca.components_.T * np.sqrt(pca.explained_variance_)

loadings_df = pd.DataFrame(loadings, index=features.columns, columns=[f'PC{i
print("PCA Loadings:\n", loadings_df)
```

```
PCA Loadings:
                                     PC1       PC2
Water Area Change (m²)          0.723329 -0.173338
Flood Area Change (m²)          0.426687  0.013525
Urban Area Change (m²)          0.836305 -0.260917
Vegetation Area Change (m²) -0.087001  0.798647
SOC%_2025                      -0.368086 -0.860072
pH_2025                         0.837895  0.470879
TN_2025                        -0.160376 -0.992185
SBD_2025                       -0.600548  0.200734
CEC_2025                        1.016864 -0.157205
Clay_2025                       0.937974 -0.194617
```

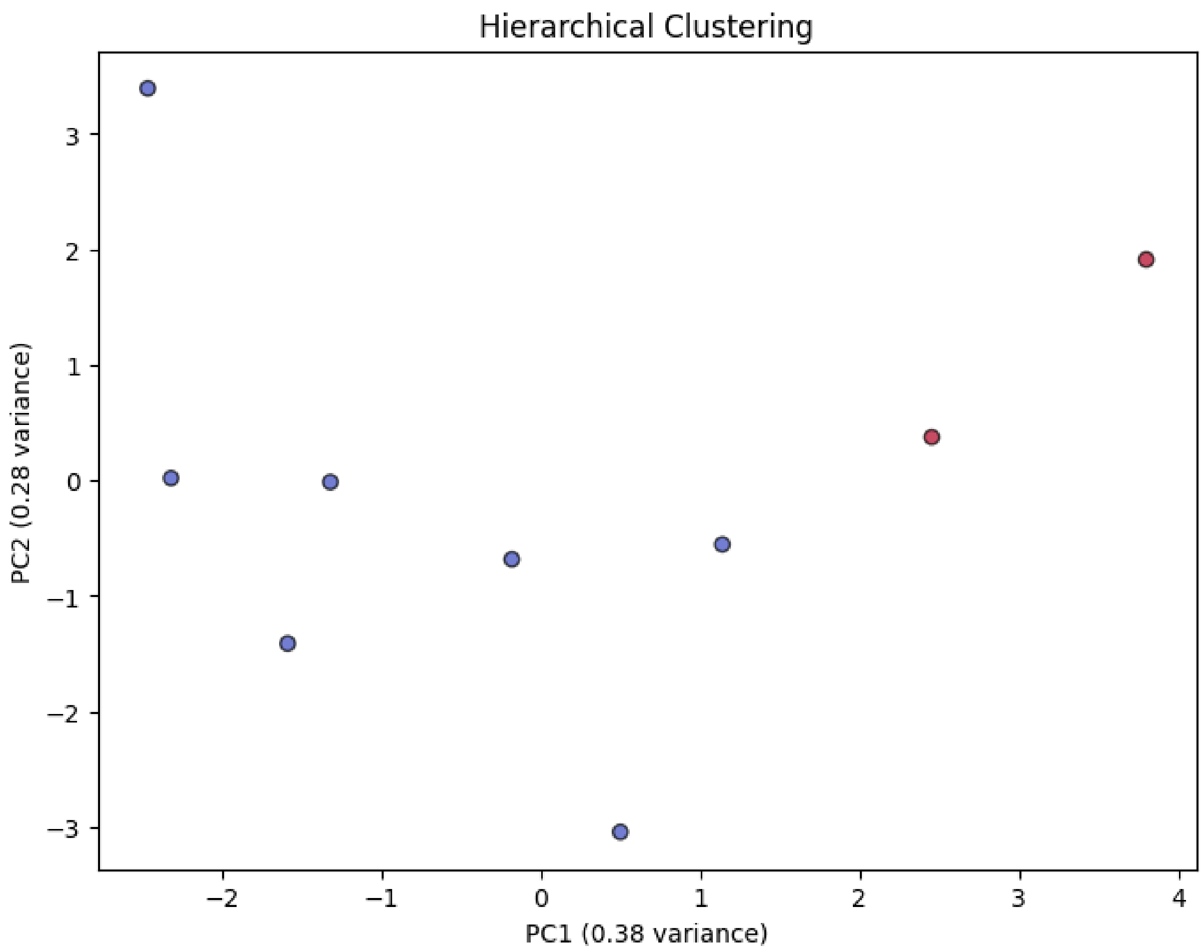
In [63]:
```
pca.explained_variance_ratio_
```

Out[63]:
```
array([0.40444296, 0.24754743])
```

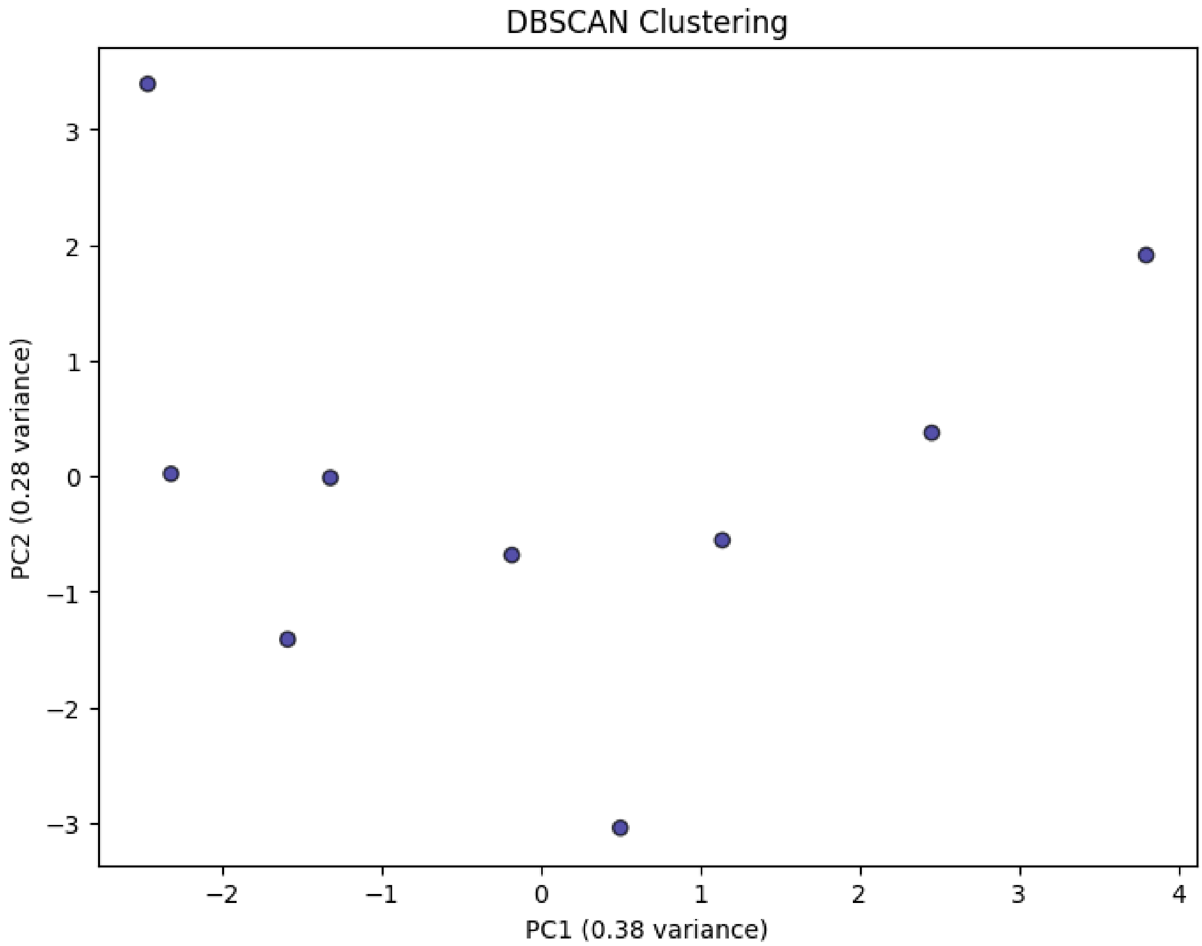
In [73]:
```
kmeans = KMeans(n_clusters=2, random_state=42)
kmeans_labels = kmeans.fit_predict(pca_result)

plt.figure(figsize=(8, 6))
plt.scatter(pca_result[:, 0], pca_result[:, 1], c=kmeans_labels, cmap='viric
plt.title('K-Means Clustering')
plt.xlabel(f'PC1 ({pca.explained_variance_ratio_[0]:.2f} variance)')
plt.ylabel(f'PC2 ({pca.explained_variance_ratio_[1]:.2f} variance)')
plt.grid(False)
plt.show()
```



In [72]:
```
hierarchical = AgglomerativeClustering(n_clusters=2)
hierarchical_labels = hierarchical.fit_predict(pca_result)

plt.figure(figsize=(8, 6))
plt.scatter(pca_result[:, 0], pca_result[:, 1], c=hierarchical_labels, cmap=
plt.title('Hierarchical Clustering')
plt.xlabel(f'PC1 ({pca.explained_variance_ratio_[0]:.2f} variance)')
plt.ylabel(f'PC2 ({pca.explained_variance_ratio_[1]:.2f} variance)')
plt.grid(False)
plt.show()
```



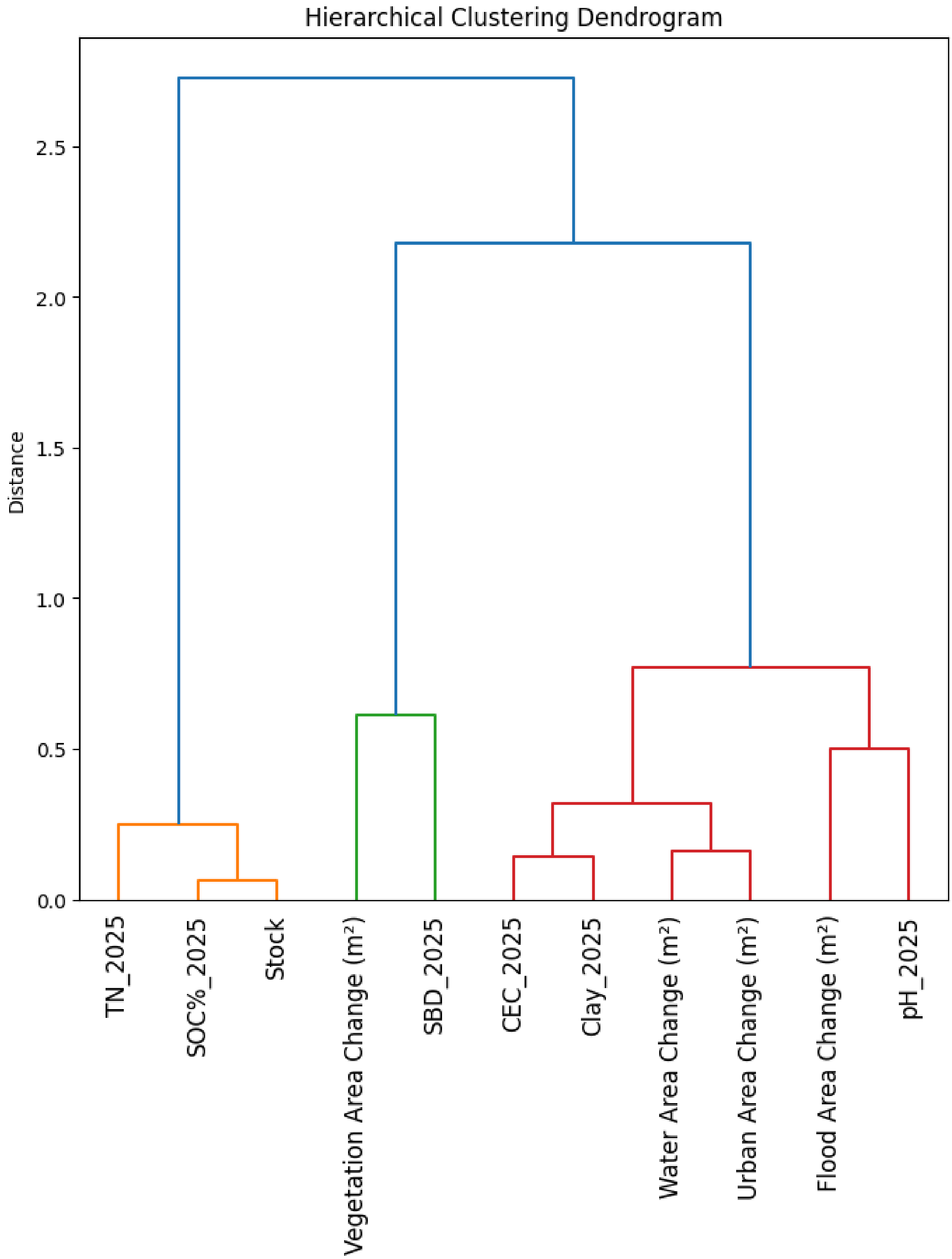
In [71]:
```
dbscan = DBSCAN(eps=0.5, min_samples=2)
dbscan_labels = dbscan.fit_predict(pca_result)

plt.figure(figsize=(8, 6))
plt.scatter(pca_result[:, 0], pca_result[:, 1], c=dbscan_labels, cmap='plasm
plt.title('DBSCAN Clustering')
plt.xlabel(f'PC1 ({pca.explained_variance_ratio_[0]:.2f} variance)')
plt.ylabel(f'PC2 ({pca.explained_variance_ratio_[1]:.2f} variance)')
plt.grid(False)
plt.show()

print("Silhouette Score for K-Means:", silhouette_score(pca_result, kmeans_l
print("Silhouette Score for Hierarchical:", silhouette_score(pca_result, hie
```




```
Silhouette Score for K-Means: 0.3785980121054481
Silhouette Score for Hierarchical: 0.3158913948359037
```

In [81]:

```python
from scipy.cluster.hierarchy import dendrogram, linkage

Z = linkage(loadings, method='ward')

plt.figure(figsize=(8, 8))
dendrogram(Z, labels=features.columns, orientation='top', leaf_rotation=90,
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('')
plt.ylabel('Distance')
plt.show()
```



Hierarchical Clustering Dendrogram
Distance
2.5
2.0
1.5
1.0
0.5
0.0
TN_2025
SOC%_2025
Stock
Vegetation Area Change (m²)
SBD_2025
CEC_2025
Clay_2025
Water Area Change (m²)
Urban Area Change (m²)
Flood Area Change (m²)
pH_2025

In [ ]: