In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Load the dataset
data = pd.read_csv('MERGED2018-19.csv')
# Checking for missing values
print(data.isnull().sum())
# Define the columns to keep
columns_to_keep = [
    'ST_FIPS', 'CONTROL', 'PREDDEG', 'UGDS_WHITE', 'UGDS_BLACK', 'UGDS_HISP',
    'UGDS_ASIAN', 'UGDS_AIAN', 'UGDS_NHPI', 'UGDS_2MOR', 'UGDS_NRA', 'UGDS_UNKN','LATITUDE'

# Filter the dataset to only include these columns
filtered_data = data[columns_to_keep]
```

```
C:\Users\harin\AppData\Local\Temp\ipykernel_28440\131281525.py:2: DtypeWarning: Columns (1725,
1726,1727,1728,1815,1818,1823,1824,1830,1831,1879,1880,1881,1882,1883,1884,1885,1886,1887,188
8,1889,1890,1891,1892,1893,1910,1911,1912,1913,1957,1958,1959,1960,1961,1962,1963,1964,1965,19
66,1967,1968,1969,1970,1971,1972) have mixed types. Specify dtype option on import or set low_
memory=False.
  data = pd.read_csv('MERGED2018-19.csv')

UNITID                        0
OPEID                         0
OPEID6                        0
INSTNM                        0
CITY                          0
                           ...
SCUGFFN                     782
POOLYRS_FTFTAIDPCT          774
FTFTPCTPELL_POOLED_SUPP     977
FTFTPCTFLOAN_POOLED_SUPP    977
SCUGFFN_POOLED              774
Length: 1986, dtype: int64
```

In [3]:
```
filtered_data.describe()
```

Out[3]:

	ST_FIPS	CONTROL	PREDDEG	UGDS_WHITE	UGDS_BLACK	UGDS_HISP	UGDS_ASIAN	UGDS_AIAN
count	6806.000000	6806.000000	6806.000000	6041.000000	6041.000000	6041.000000	6041.000000	6041.000000
mean	29.032912	2.087570	1.833823	0.491790	0.179735	0.181379	0.037811	0.013504
std	16.769898	0.835281	1.070275	0.283737	0.217884	0.227826	0.079448	0.071663
min	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	13.000000	1.000000	1.000000	0.252400	0.035600	0.036300	0.002400	0.000000
50%	29.000000	2.000000	2.000000	0.532100	0.095900	0.091000	0.014400	0.002300
75%	42.000000	3.000000	3.000000	0.723100	0.239000	0.235300	0.037300	0.006700
max	78.000000	3.000000	4.000000	1.000000	1.000000	1.000000	1.000000	1.000000

In [4]:

```
# Checking for missing values
print(filtered_data.isnull().sum())

# Dropping rows with missing values in specific crucial columns
filtered_data = filtered_data.dropna(subset=['ST_FIPS', 'CONTROL', 'PREDDEG'])  # Example o

# Display the cleaned data
print(f"Original data size: {data.shape}, Cleaned data size: {filtered_data.shape}")

```

```
ST_FIPS         0
CONTROL         0
PREDDEG         0
UGDS_WHITE    765
UGDS_BLACK    765
UGDS_HISP     765
UGDS_ASIAN    765
UGDS_AIAN     765
UGDS_NHPI     765
UGDS_2MOR     765
UGDS_NRA      765
UGDS_UNKN     765
LATITUDE      475
LONGITUDE     475
dtype: int64
Original data size: (6806, 1986), Cleaned data size: (6806, 14)
```

In [5]:

```python
# Mapping for 'CONTROL' variable
control_mapping = {
    1: 'Public',
    2: 'Private Nonprofit',
    3: 'Private For-profit'
}

# Mapping for 'PREDDEG' (Predominant Undergraduate Degree Awarded)
preddeg_mapping = {
    0: 'Non-degree-granting',
    1: 'Certificate',
    2: 'Associate degree',
    3: 'Bachelor\'s degree',
    4: 'Graduate degree'
}
# Mapping for FIPS code
st_fips_mapping = {
    1: 'Alabama', 2: 'Alaska', 4: 'Arizona', 5: 'Arkansas', 6: 'California',
    8: 'Colorado', 9: 'Connecticut', 10: 'Delaware', 11: 'District of Columbia',
    12: 'Florida', 13: 'Georgia', 15: 'Hawaii', 16: 'Idaho', 17: 'Illinois',
    18: 'Indiana', 19: 'Iowa', 20: 'Kansas', 21: 'Kentucky', 22: 'Louisiana',
    23: 'Maine', 24: 'Maryland', 25: 'Massachusetts', 26: 'Michigan',
    27: 'Minnesota', 28: 'Mississippi', 29: 'Missouri', 30: 'Montana',
    31: 'Nebraska', 32: 'Nevada', 33: 'New Hampshire', 34: 'New Jersey',
    35: 'New Mexico', 36: 'New York', 37: 'North Carolina', 38: 'North Dakota',
    39: 'Ohio', 40: 'Oklahoma', 41: 'Oregon', 42: 'Pennsylvania',
    44: 'Rhode Island', 45: 'South Carolina', 46: 'South Dakota',
    47: 'Tennessee', 48: 'Texas', 49: 'Utah', 50: 'Vermont', 51: 'Virginia',
    53: 'Washington', 54: 'West Virginia', 55: 'Wisconsin', 56: 'Wyoming',
    60: 'American Samoa', 64: 'Federated States of Micronesia', 66: 'Guam',
    69: 'Northern Mariana Islands', 70: 'Palau', 72: 'Puerto Rico', 78: 'Virgin Islands'
}

# Apply the mappings
filtered_data['CONTROL'] = filtered_data['CONTROL'].map(control_mapping).astype('category')
filtered_data['PREDDEG'] = filtered_data['PREDDEG'].map(preddeg_mapping).astype('category')
filtered_data['ST_FIPS'] = filtered_data['ST_FIPS'].map(st_fips_mapping).astype('category')
```

In [6]:
```
1 filtered_data
```

Out[6]:

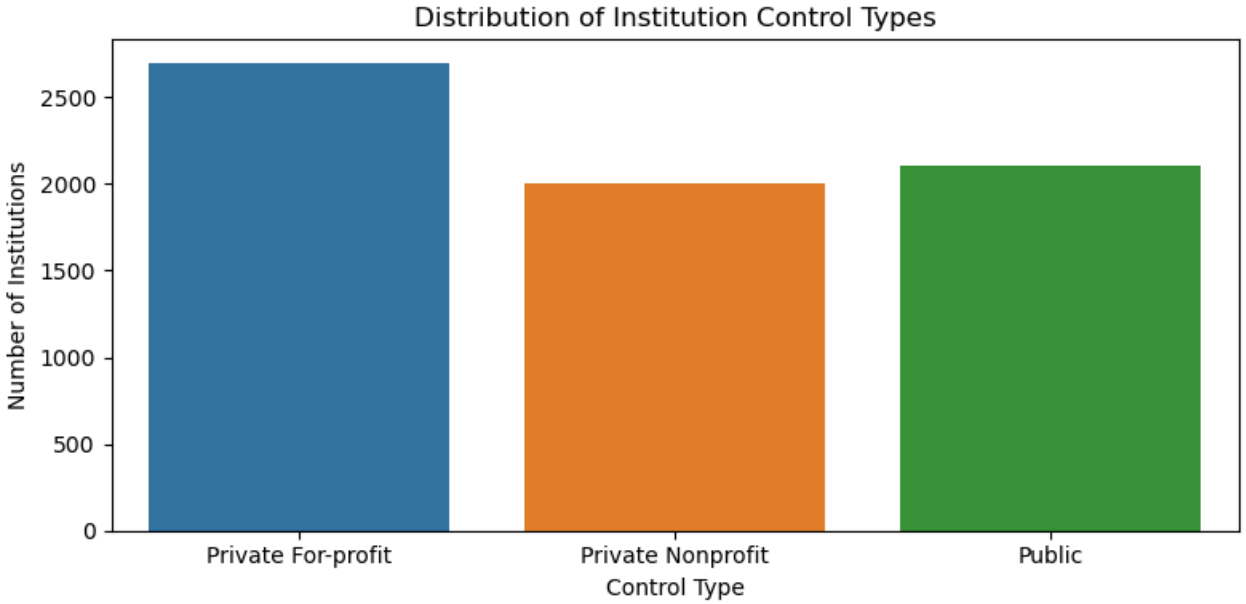
	ST_FIPS	CONTROL	PREDDEG	UGDS_WHITE	UGDS_BLACK	UGDS_HISP	UGDS_ASIAN	UGDS_AIAN	UGDS_N
0	Arizona	Private Nonprofit	Graduate degree	NaN	NaN	NaN	NaN	NaN	
1	California	Private Nonprofit	Graduate degree	NaN	NaN	NaN	NaN	NaN	
2	California	Private Nonprofit	Graduate degree	NaN	NaN	NaN	NaN	NaN	
3	California	Private Nonprofit	Graduate degree	NaN	NaN	NaN	NaN	NaN	
4	California	Public	Graduate degree	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	
6801	California	Private For-profit	Certificate	0.2162	0.0270	0.7027	0.0	0.0000	0
6802	Puerto Rico	Private For-profit	Certificate	0.0000	0.0000	1.0000	0.0	0.0000	0
6803	New Mexico	Private For-profit	Certificate	0.2432	0.0541	0.6216	0.0	0.0541	0
6804	Michigan	Private For-profit	Certificate	0.8462	0.0154	0.1231	0.0	0.0000	0
6805	Missouri	Private For-profit	Associate degree	0.8939	0.0455	0.0303	0.0	0.0303	0

6806 rows × 14 columns

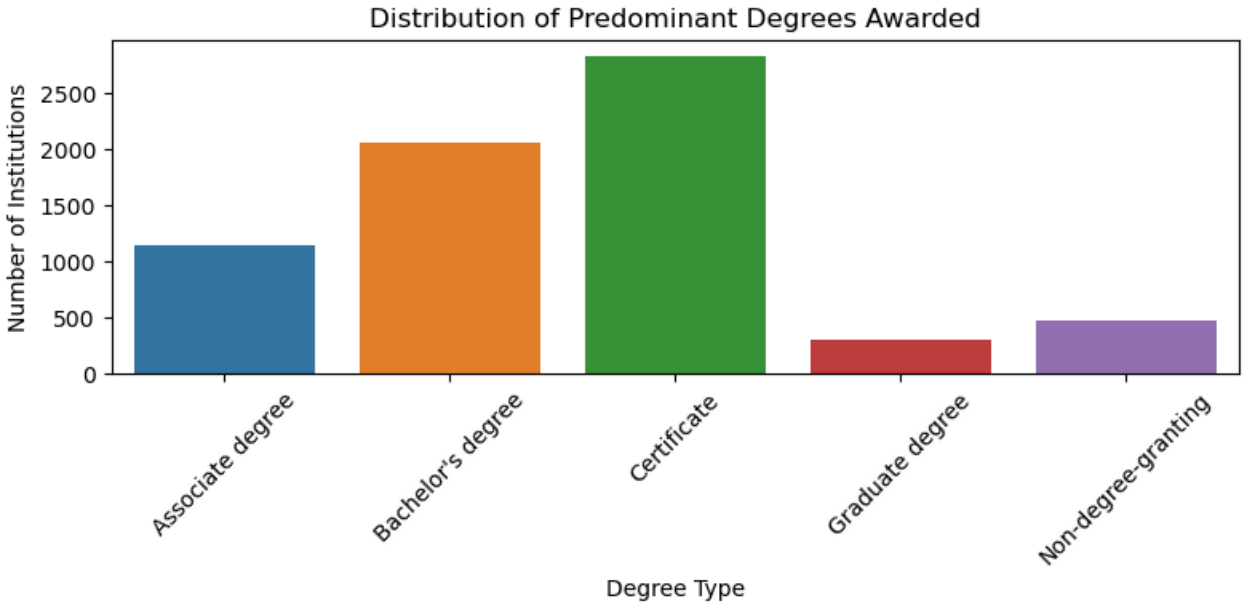
In [7]:
```
1 print(filtered_data.describe())
```

```
       UGDS_WHITE   UGDS_BLACK    UGDS_HISP   UGDS_ASIAN    UGDS_AIAN  \
count  6041.000000  6041.000000  6041.000000  6041.000000  6041.000000
mean      0.491790     0.179735     0.181379     0.037811     0.013504
std       0.283737     0.217884     0.227826     0.079448     0.071663
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.252400     0.035600     0.036300     0.002400     0.000000
50%       0.532100     0.095900     0.091000     0.014400     0.002300
75%       0.723100     0.239000     0.235300     0.037300     0.006700
max       1.000000     1.000000     1.000000     1.000000     1.000000

        UGDS_NHPI    UGDS_2MOR     UGDS_NRA    UGDS_UNKN     LATITUDE  \
count  6041.000000  6041.000000  6041.000000  6041.000000  6331.000000
mean      0.004573     0.030770     0.021385     0.035911    37.379213
std       0.031879     0.038606     0.063130     0.072023     5.849565
min       0.000000     0.000000     0.000000     0.000000   -14.322636
25%       0.000000     0.000000     0.000000     0.000000    33.973926
50%       0.000300     0.025000     0.000000     0.013300    38.833361
75%       0.002600     0.041800     0.017300     0.039200    41.332940
max       0.997300     0.631600     1.000000     1.000000    71.324702

         LONGITUDE
count  6331.000000
mean    -90.309695
std      17.980750
min    -170.742774
25%     -97.407742
50%     -86.266315
75%     -78.787266
max     171.378129
```

In [8]:
```
# Visualization: Distribution of institution control types
plt.figure(figsize=(8, 4))
sns.countplot(x='CONTROL', data=filtered_data)
plt.title('Distribution of Institution Control Types')
plt.xlabel('Control Type')
plt.ylabel('Number of Institutions')
plt.tight_layout()
plt.show()
```

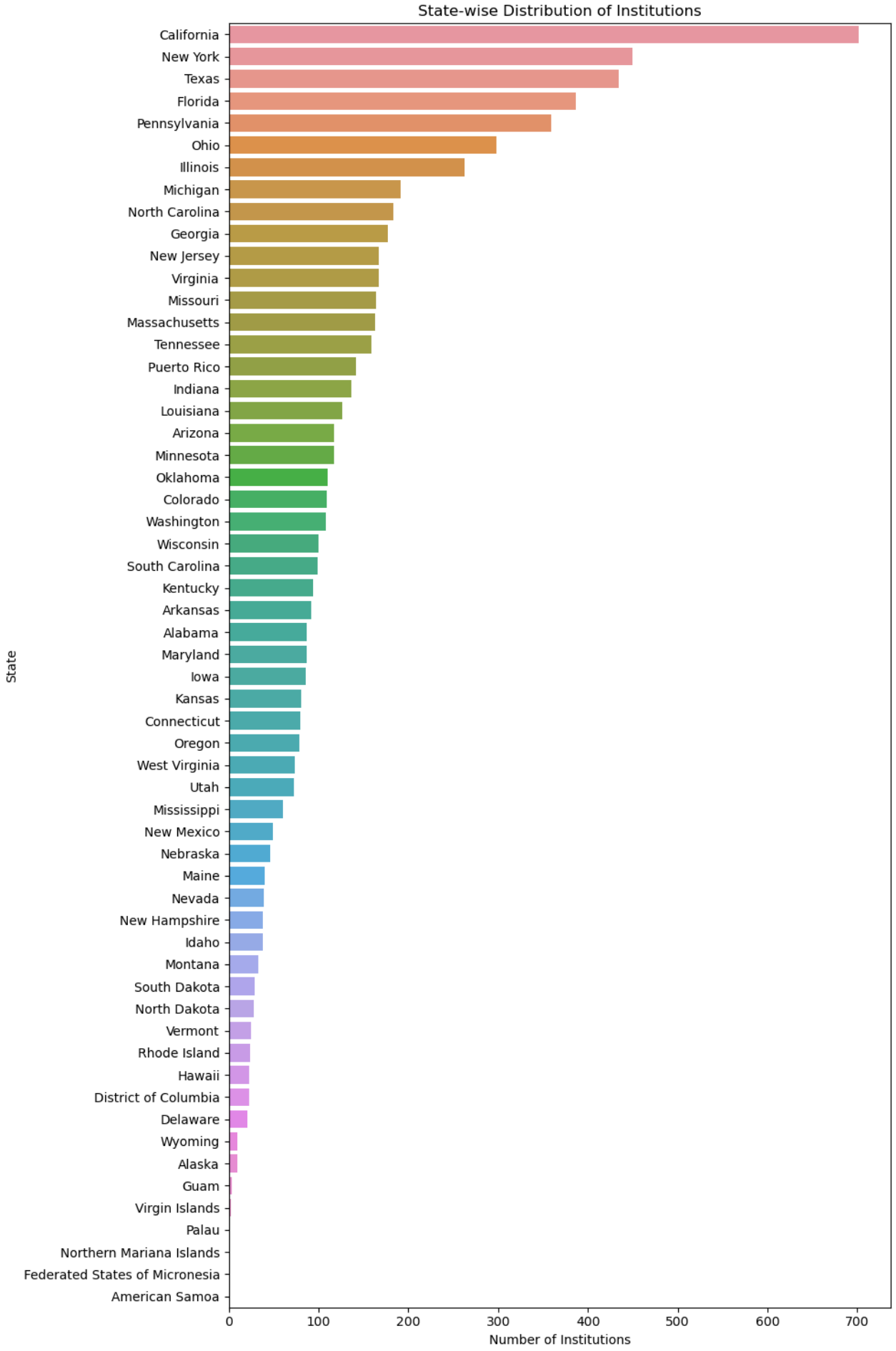


In [9]:
```
# Visualization: Distribution of predominant degree awarded
plt.figure(figsize=(8, 4))
sns.countplot(x='PREDDEG', data=filtered_data)
plt.title('Distribution of Predominant Degrees Awarded')
plt.xlabel('Degree Type')
plt.ylabel('Number of Institutions')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



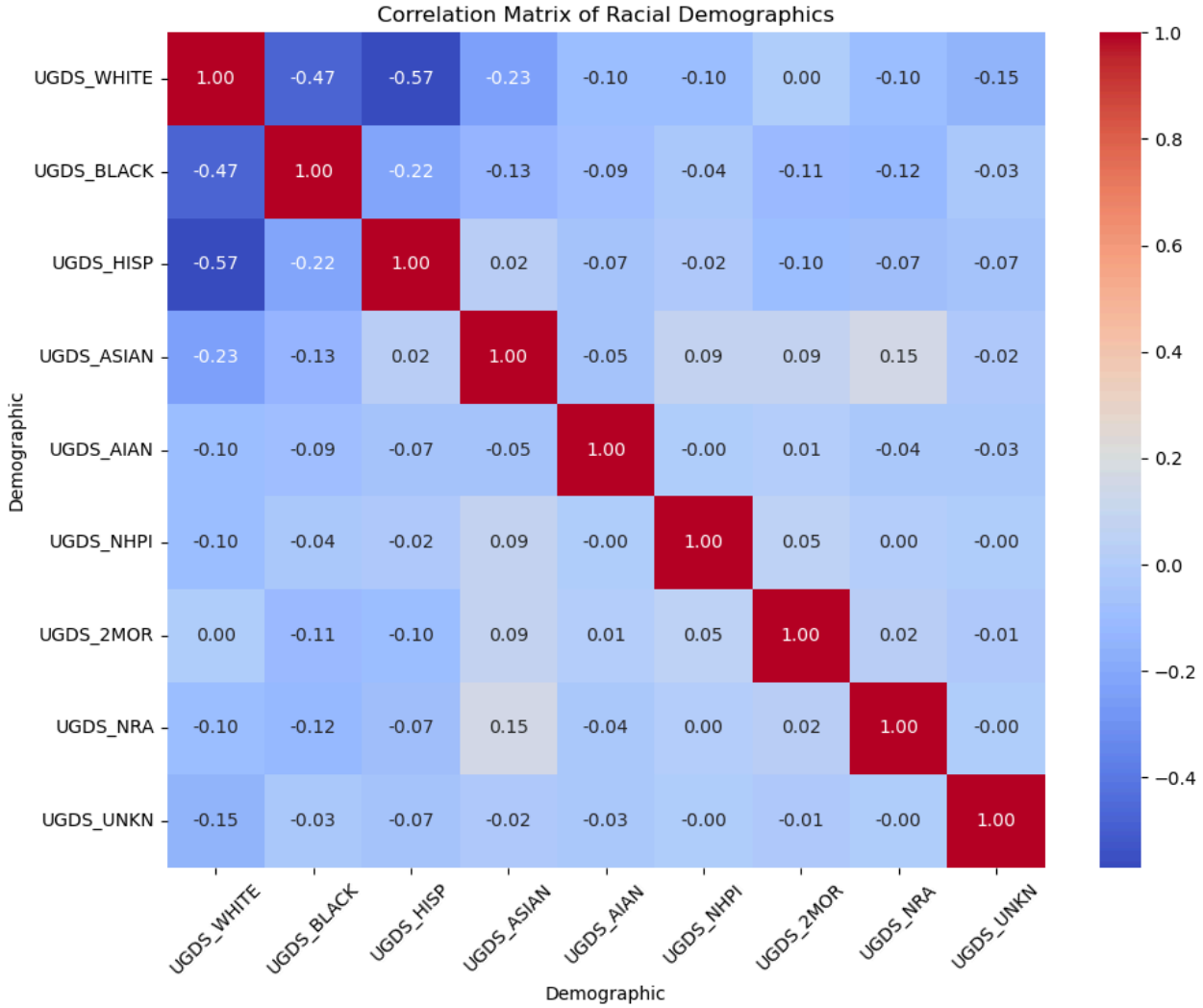
In [10]:

```
# Visualization: State-wise distribution of institutions
plt.figure(figsize=(10, 15))
sns.countplot(y='ST_FIPS', data=filtered_data, order = filtered_data['ST_FIPS'].value_counts
plt.title('State-wise Distribution of Institutions')
plt.xlabel('Number of Institutions')
plt.ylabel('State')
plt.tight_layout()
plt.show()
```


State-wise Distribution of Institutions
California
New York
Texas
Florida
Pennsylvania
Ohio
Illinois
Michigan
North Carolina
Georgia
New Jersey
Virginia
Missouri
Massachusetts
Tennessee
Puerto Rico
Indiana
Louisiana
Arizona
Minnesota
Oklahoma
Colorado
Washington
Wisconsin
South Carolina
Kentucky
Arkansas
Alabama
Maryland
Iowa
Kansas
Connecticut
Oregon
West Virginia
Utah
Mississippi
New Mexico
Nebraska
Maine
Nevada
New Hampshire
Idaho
Montana
South Dakota
North Dakota
Vermont
Rhode Island
Hawaii
District of Columbia
Delaware
Wyoming
Alaska
Guam
Virgin Islands
Palau
Northern Mariana Islands
Federated States of Micronesia
American Samoa
State
0
100
200
300
400
500
600
700
Number of Institutions

In [11]:

```
# Correlation analysis for numeric variables
# Assuming UGDS_WHITE to UGDS_UNKN are percentage values as characters, convert to float
race_columns = ['UGDS_WHITE', 'UGDS_BLACK', 'UGDS_HISP', 'UGDS_ASIAN', 'UGDS_AIAN', 'UGDS_NI
for col in race_columns:
    filtered_data[col] = pd.to_numeric(filtered_data[col], errors='coerce')

# Drop rows with missing values after conversion
filtered_data = filtered_data.dropna(subset=race_columns)

# Plotting a heatmap of the correlation matrix
plt.figure(figsize=(10, 8))
correlation_matrix = filtered_data[race_columns].corr()
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm')
plt.title('Correlation Matrix of Racial Demographics')
plt.xlabel('Demographic')
plt.ylabel('Demographic')
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()

```



In [13]:

```
filtered_data = filtered_data.dropna(subset=['ST_FIPS', 'CONTROL', 'PREDDEG'])
# Save the filtered dataset to a new file
filtered_data.to_csv('Filtered_Dataset.csv', index=False)
```