Duke History in the Rubenstein Library Card Catalog

Duke University has a long and storied past, with many individuals shaping the school through leadership and monetary support. We plan to dig deeper into what of Duke history is present in the card catalog files and see what that can tell us about the institution.

Duke Presidential Last Names in the Card Catalog

Packages used: Like most of our code, we have used the Python pandas package as a way to work with the dataset in the format of a pandas dataframe. The plots are created using the Matplotlib package and scipy, which is used to perform the linear regression. Re, or regular expression is used to detect patterns in text, allowing us to search for instances of presidential names in the catalog.

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import re
from scipy import stats
import operator
pd.set_option('display.max_colwidth', None)
```

First we will be exploring the presidents of Duke University and their prevalence within the card catalog. The current President, Vincent Price, will be excluded since his last name is commonly found in the cards with another meaning and his time at Duke is well past the time of the catalog. Interestingly enough, the penultimate President's name appears in the cards (Richard Brodhead) despite his not coming to Duke until after the card catalog was digitized. Perhaps this eponymous figure was an ancestor of the one we know in relation to Duke today. We will be searching by last name, to see the frequency of the Presidents' last names in the catalog. A few of the presidents have surnames that have alternate meanings or uses (e.g., York, Wood) and we will attempt to remove the non-name occuraces of these. A disclamer to this analysis is that this method only captures names that were both properly spelled in the catloging process and translated into text accurately by our OCR software; it is likely that there are some misspelled names that did not make it thorugh the calculations.

In [2]:
```
# Read in dataset
df = pd.read_csv("main_file_dataset.csv")
```

First Approach

Name Counting

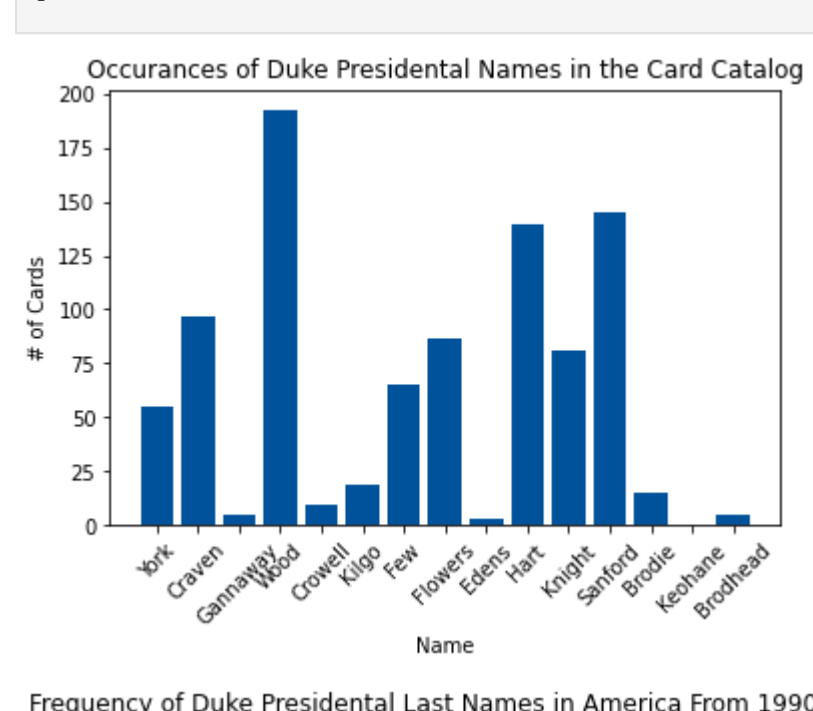
In [3]:
```
# Duke University presidents in chronological order
presidents = {"York": 0, "Craven": 0, "Gannaway": 0, "Craven": 0, "Wood": 0, "Crowell": 0, "Kilgo": 0, "Few": 0
    "Edens": 0, "Hart": 0, "Knight": 0,"Sanford": 0, "Brodie": 0, "Keohane": 0, "Brodhead": 0}

# Check for occurances of presidental last names in the cards
for index, row in df.iterrows():
    for pres in presidents.keys():
        if pres in str(row['Text']):
            # Check for homonyms
            x = str(row['Text'])
            if pres == "York" and ("New" in x or "Yorktown" in x or "Yorkshire" in x or "England" in x or "York
                continue
            elif pres == "Craven" and "Craven Co" in x.title():
                continue
            elif pres == "Wood" and (re.search(r"Wood[a-z-]", x) or "Wood County" in x):
                continue
            elif pres == "Kilgo" and re.search(r"Kilgo[a-z]", x):
                continue
            elif pres == "Hart" and re.search(r"Hart[a-z]", x):
                continue
            elif pres == "Knight" and (re.search(r"Knight[a-z]", x) or re.search(r"[a-z]Knight", x)):
                continue
            elif pres == "Sanford" and (re.search(r"Sanford,.*North Carolina", x) or "in Sanford" in x or "of S
                continue
            else:
                presidents[pres] = presidents.get(pres) + 1

print(presidents)
```

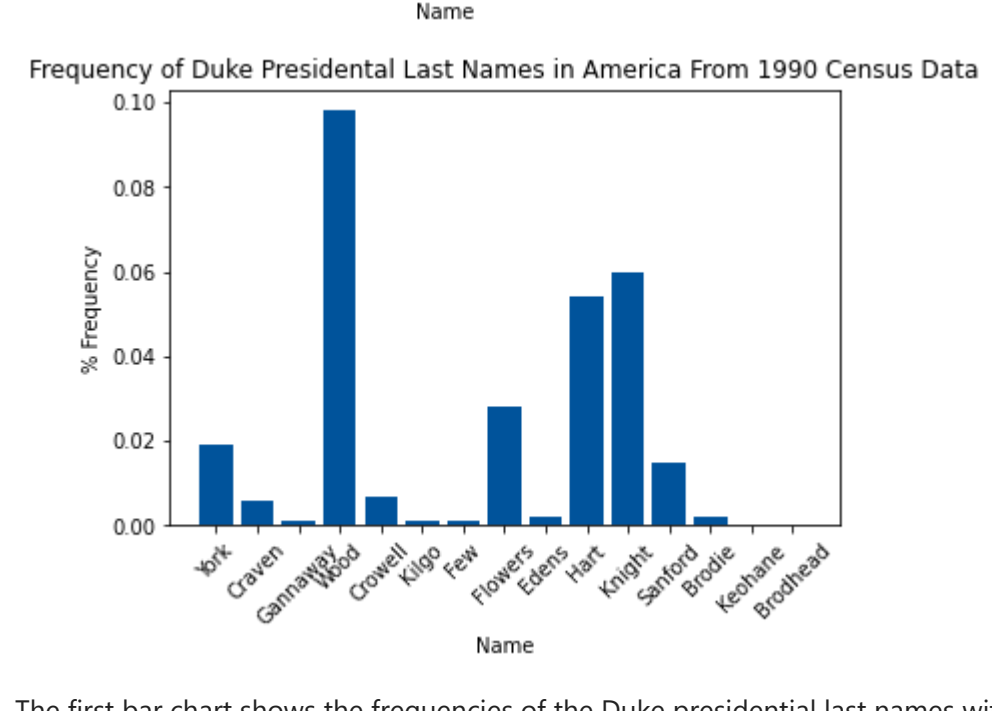
```
{'York': 55, 'Craven': 97, 'Gannaway': 5, 'Wood': 192, 'Crowell': 10, 'Kilgo': 19, 'Few': 65, 'Flowers': 87, 'E
dens': 3, 'Hart': 139, 'Knight': 81, 'Sanford': 145, 'Brodie': 15, 'Keohane': 0, 'Brodhead': 5}
```

In [5]:
```
# Counts of last name frequency after running above code
president_counts = {'York': 55, 'Craven': 97, 'Gannaway': 5, 'Wood': 192, 'Crowell': 10, 'Kilgo': 19, 'Few': 65
    'Flowers': 87, 'Edens': 3, 'Hart': 139, 'Knight': 81, 'Sanford': 145, 'Brodie': 15, 'Keohane': 0, 'Brodhead

# Frequency of presidential last names as of the 1990 US census
census_freqs = {'York': 0.019, 'Craven': 0.006, 'Gannaway': 0.001, 'Wood': 0.098, 'Crowell': 0.007, 'Kilgo': 0.
    'Few': 0.001, 'Flowers': 0.028, 'Edens': 0.002, 'Hart': 0.054, 'Knight': 0.060, 'Sanford': 0.015, 'Brodie':
    'Keohane': 0.000, 'Brodhead': 0.000}
```

Frequency Plotting

In [6]:
```
# Disply bar chart of last name occurances
plt.bar(*zip(*president_counts.items()), color='#00539B')
plt.xticks(rotation = 45)
plt.title("Occurances of Duke Presidental Names in the Card Catalog")
plt.xlabel("Name")
plt.ylabel("# of Cards")
plt.show()

# Display bar chart of last name frequencies based on the 1990 census data
plt.bar(*zip(*census_freqs.items()), color='#00539B')
plt.xticks(rotation = 45)
plt.title("Frequency of Duke Presidental Last Names in America From 1990 Census Data")
plt.xlabel("Name")
plt.ylabel("% Frequency")
plt.show()
```



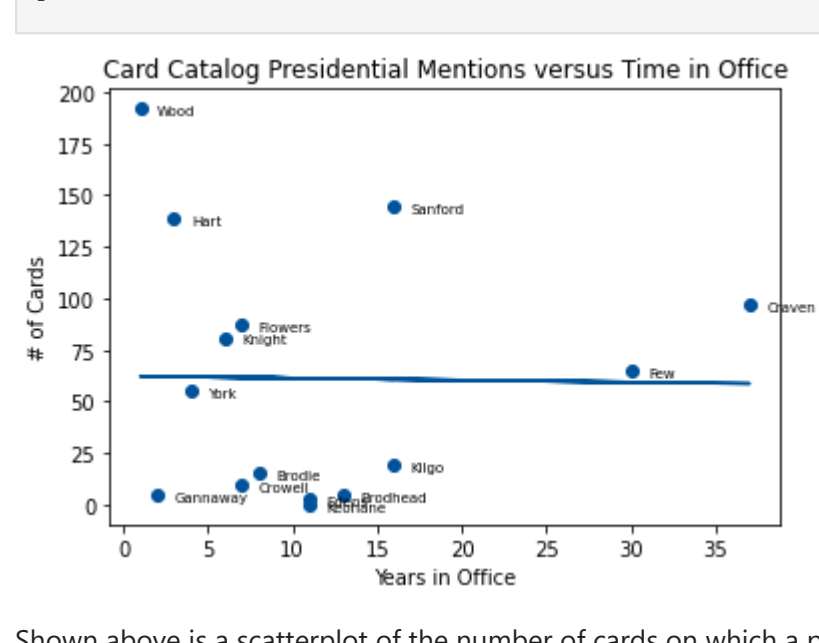


The first bar chart shows the frequencies of the Duke presidential last names within the card catalog, in order of presidental appointment. There does not appear to be any trend over time amongst the names. When comparing to the 1990 census data of last name frequency in the United States, some names are similarly more frequent than others (e.g., Wood and Hart) or similarly infrequent (e.g. Gannaway, Edens, and Keohane). Some presidents, however, have statistically uncommon last names but a large amount of occurances in the card catalog (Craven, Few, Sanford). Are there other factors at play? Let's see how this compares to the amount of time each person spent in office.

Occurances versus Time in Office

Each of these Presidents served varied time spans in office, let's see if there is a relationship between the length of time they spent in office and the amount of times they were mentioned in cards in the catalog. Preseident Craven served two nonconsecutive terms, so his time in office will be the addition of the two terms.

In [34]:
```
terms = [4, 37, 2, 1, 7, 16, 30, 7, 11, 3, 6, 16, 8, 11, 13]
counts = [55, 97, 5, 192, 10, 19, 65, 87, 3, 139, 81, 145, 15, 0, 5]
last = ["York", "Craven", "Gannaway", "Wood", "Crowell", "Kilgo", "Few", "Flowers", "Edens", "Hart", "Knight",

# Regression code adapted from https://www.w3schools.com/python/python_ml_linear_regression.asp
slope, intercept, r, p, std_err = stats.linregress(terms, counts)
def myfunc(x):
  return slope * x + intercept
mymodel = list(map(myfunc, terms))

# Plot points with name labels, regression line
plt.scatter(terms, counts, color='#00539B')
for i in range(len(terms)):
    plt.annotate(last[i], (terms[i] + 1, counts[i] - 3), fontsize=7)
plt.plot(terms, mymodel, color='#00539B')
plt.title("Card Catalog Presidential Mentions versus Time in Office")
plt.xlabel("Years in Office")
plt.ylabel("# of Cards")
plt.show()
```



Shown above is a scatterplot of the number of cards on which a presidential last name is mentioned versus the amount of time they spent in office. We ran a linear regression on the data and found **no correlation** between these two variables. Time in office appears not to impact the prevalence of presidential last names in the catalog. What else is going on here? Let's check the catalog for both first *and* last names.

Second Approach

Presidental First and Last Names

A common occurrance in the card catalog is the presence of multiple generations of family members. By looking at the *last* names of presidents, we were able to look for more general trends in regards to the presidents' names, but are unable to determine just how many of the last names gleaned are actually related to the president or their family members. Next we will look into instances where the specific presidents' *first and last* names occur in the files, with hopes of learning more about the history and prevalence of these figures within the card catalog.

In [5]:
```
# Read in dataframe and pull out wanted, non-null columns
df = df[~df.Name.isnull()]
df = df[~df.Text.isnull()]
df = df.iloc[:,[2,3,8,9,10]]

# Using these regexes, we can find if both words occur in a row, e.g. Craven, Braxton
base = r'^{}'
expr = '(?=.*{})'

# Names to look for occuring together
full_names = [["York","Brantley"], ["Craven", "Braxton"], ["Gannaway", "William"], ["Wood", "Marquis"], ["Crowe
    ["Kilgo", "John"], ["Few", "William"], ["Flowers", "Robert"], ["Edens", "Arthur"], ["Hart", "Julian"],
    ["Knight", "Douglas"], ["Sanford", "Terry"], ["Brodie", "Keith"], ["Keohane", "Nannerl"], ["Brodhead", "Ric
```

York was found 55 times according to the previous method, but most talk about cities & counties or unrelated people with the same last name. There are probably very few talking about the president. This better extracts the president names by checking for the occurence of both last and first names.

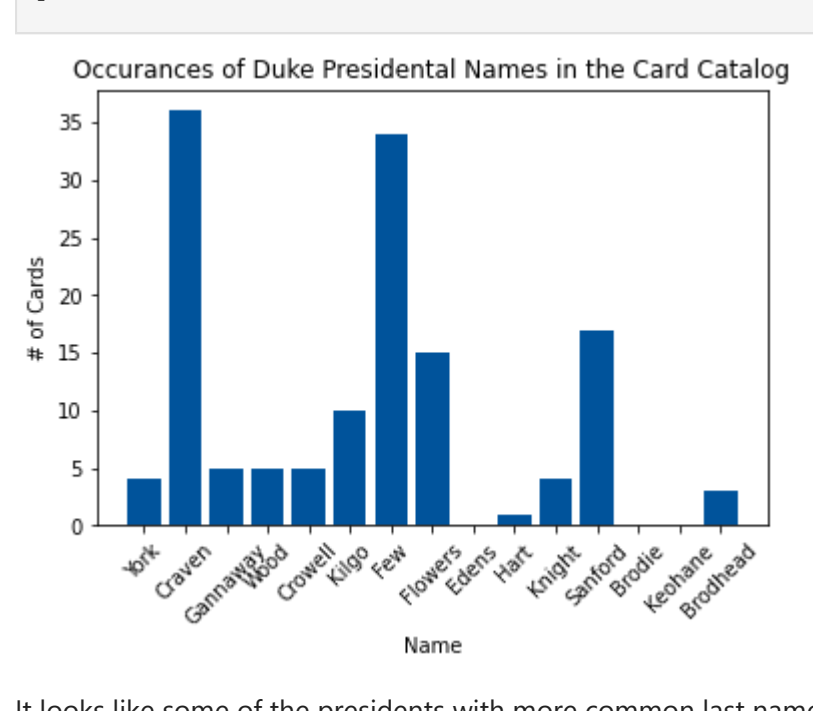
In [24]:
```
name_counts = []

for name in full_names:
    name_counts.append(len(df[df.Text.str.contains(base.format(''.join(expr.format(w) for w in name)),case=True

print(name_counts)
```

```
[4, 36, 5, 5, 5, 10, 34, 15, 0, 1, 4, 17, 0, 0, 3]
```

In [41]:
```
# Dictionary storing indices of president mentioning cards
president_indices = {"York": [7072, 14590, 50164, 50170], "Craven": [2004, 2005, 9541, 10235, 11146, 11155, 111
              "Gannaway": [6266, 6267, 6269, 16522, 41864], "Wood": [13320, 30929, 30984, 49501, 49528],
              "Crowell": [127, 11387, 19234, 32886, 46573], "Kilgo": [158, 6815, 21326, 25715, 31143, 31930, 42
              "Few": [4293, 4295, 5490, 5535, 6660, 7338, 7356, 13399, 13678, 13680, 14742, 15288, 15290, 15293
              "Flowers": [10743, 15710, 16296, 19004, 20519, 22959, 26399, 27309, 28173, 30657, 34052, 42059, 4
             "Edens": [], "Hart": [14123], "Knight": [5490, 25741, 30171, 47829],
              "Sanford": [5490, 9533, 9541, 9554, 19873, 19875, 22266, 39311, 39312, 39313, 39314, 39363, 39364
              "Brodie": [], "Keohane": [], "Brodhead": [2619, 34442, 35593]}
```

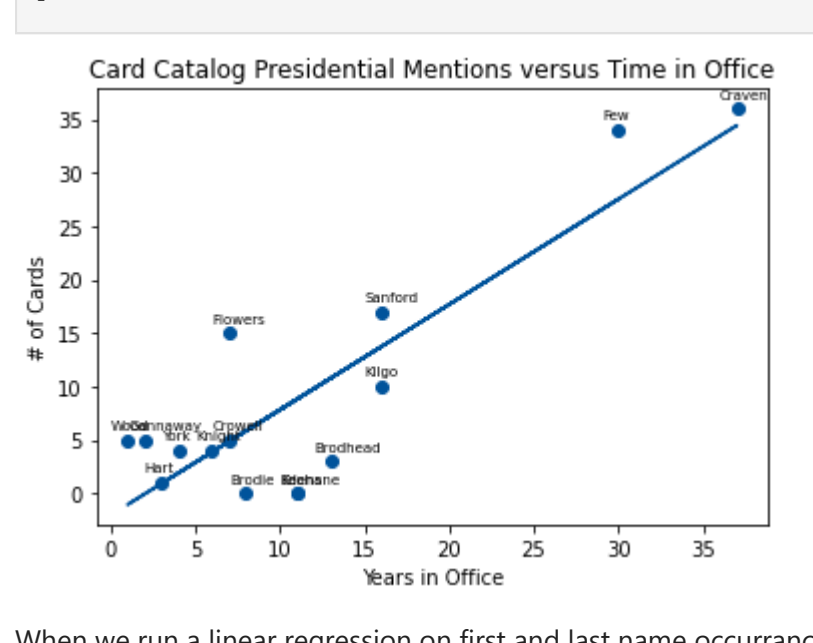
When we check for both the first and last names of the Duke presidents, we get markedly fewer results, but more accurate ones. Let's take a look at the frequency of each president's mentions.

In [25]:
```
last = ["York", "Craven", "Gannaway", "Wood", "Crowell", "Kilgo", "Few", "Flowers", "Edens", "Hart", "Knight",
name_counts = [4, 36, 5, 5, 5, 10, 34, 15, 0, 1, 4, 17, 0, 0, 3]

# Disply bar chart of first and last name occurances
plt.bar(last, name_counts, color='#00539B')
plt.xticks(rotation = 45)
plt.title("Occurances of Duke Presidental Names in the Card Catalog")
plt.xlabel("Name")
plt.ylabel("# of Cards")
plt.show()
```



It looks like some of the presidents with more common last names, like Wood, Knight, and Hart, have gone down in frequency when we also check for first names. Let's compare with the length of time in office.

In [33]:
```
terms = [4, 37, 2, 1, 7, 16, 30, 7, 11, 3, 6, 16, 8, 11, 13]

# Linear regression code adapted from https://www.w3schools.com/python/python_ml_linear_regression.asp
slope, intercept, r, p, std_err = stats.linregress(terms, name_counts)
def myfunc(x):
  return slope * x + intercept
mymodel = list(map(myfunc, terms))

# Plot labeled points, regression line
plt.scatter(terms, name_counts, color='#00539B')
for i in range(len(terms)):
    plt.annotate(last[i], (terms[i] - 1, name_counts[i] + 1), fontsize=7)
plt.plot(terms, mymodel, color='#00539B')
plt.title("Card Catalog Presidential Mentions versus Time in Office")
plt.xlabel("Years in Office")
plt.ylabel("# of Cards")
plt.show()
```



When we run a linear regression on first and last name occurrances versus years a president was in office, we find a **positive** correlation between the variables. So, as the number of years in office a president had increases, the number of cards on which they are mentioned increases.

Qualitative Analysis of Duke Presidents in the Catalog

Let's see what the collections containing the Duke Presidents' full names are talking about and whether they are indeed mentioning the presidents. To do this, we searched through the instances where a president's first and last name appeared in the catalog, briefly summarized the mention, and provided relevant links.

Keith Brodie, Nannerel Keohane, and Arthur Edens

Presidents Brodie, Keohane, and Edens were not mentioned in the card catalog.

Julian Hart

The card containing "Julian" and "Hart" is not referring to the Duke president.

Richard Brodhead

On two cards, a Richard Brodhead is mentioned, but not the one who was president of Duke. Upon further inspection, this man was a U.S. Democratic Senator from Pennsylvania. The cards upon which he is mentioned can be viewed here and here.

Douglas Knight

President Douglas Knight is mentioned as having correspondence with Herbert Clarence Bradshaw, being the recipient of a letter from a Mr. Matton and letters from William Murray Werber. The first two mention religion and all involve letters.

Brantley York

One card simply prompts a look to the Duke University Archives. York is mentioned here as correspondent to Tod Robinson Caldwell.

This collection mentions early foundations of Duke Univesity and President Craven.

Here is York's son's collection.

William Gannaway

Four cards mention William Gannaway Brownlow, former Governor of Tennesee and one is a prompt to see the archives.

Marquis Wood

Like many of the other presidents, Wood has a card prompting a check of the archives.

Marquis Wood also has two collections of papers associated with the Methodist Episcopal Church.

In William Clark Doub's collection, Wood's manuscript on the introduction of Methodism into the Yadkin Valley is mentioned.

John Crowell

We, of course, have the entry under Crowell's name to See Duke University Archives.

This card mentions a quarrel between Crowell and his faculty.

This and this mention letters to and from John Crowell.

Here Crowell is part of a list of unpublished sketches of well-known North Carolinians.

John Kilgo

Here we have the boilerplate John Kilgo card.

Kilgo is listed as a correspondent in the Hemphill Family Collection.

Kilgo appears to have been involved in the Methodist Episcopal Church here and here.

Correspondence with President Kilgo is mentioned here and here in relation to the Southgates.

Kilgo seems to be something of a controversial and outspoken character, as here and here he is spoken of positively and here and here he is said to be involved in a court case.

Robert Flowers

In addition to the requisite Flowers card we have correspondence between Flowers and others here, here, here, here, here, and here.

President Flowers is praised here along with some other notable Dukies and had gifted some items related to the Methodist Church here.

He is also in a photograph that is cataloged and is said to have written a biography of Edwin W. Fuller.

Terry Sanford

As Terry Sanford was a US Senator and NC Governor as well as a President of Duke, there appear to be many cards mentioning him. Cards mentioning correspondence with Sanford can be found here, here, here, here, here, and here.

This talks about the Southern Rural Poverty Project, directed by members of the Sanford Institute of Public Policy.

Here and here are collections that catalog items that talk about Sanford.

[Here] and on subsequent pages we have a restricted collection relating to manuscripts created by Terry Sanford, related to his time as Governor and Duke President. And here there are documents related to his time as a US Senator.

William Few

Few is mentioned as a professor here.

He is included in collections with other important Duke figures here, here, here, here, here, and here.

Correspondece with William Few is mentioned here, here, and here.

Few's son, Lynne Few, appears here.

He also has a collection of papers related to war and money.

Ella Howerton Parks remembers Few as the "prince of all hat doffers.".

This collection talks about a treatise Few signed related to the relations between the northern and southern colonies.

Braxton Craven

President Craven's great grandson has a rather extensive collection and his grandson is mentioned here.

Braxton Craven is cataloged in the 1850 census of Randolph County here along with many transactions under his name. Another money-related card mentions a statement for what Trinity College owed a company.

Correspondences involving Craven are cataloged here, here, and here.

Craven is mentioned, but not in much detail here, here, here, here, here, and here.

Duke Building Names in the Catalog

While going through the card catalog, it was striking to see how many of the last names present in the cards were names of buildings at Duke's Durham campus today. Let's explore the prevalence of these names in the catalog.

Taking the names from buildings on Duke's East and West campuses--including dorms, academic buildings, and public spaces--we will check for the building names in the collection header author name columns of our dataset.

In [74]:
```
buildings = {
'Alspaugh': 0, 'Baldwin': 0, 'Bassett': 0, 'Bivins': 0, 'Blackwell': 0, 'Branson': 0, 'Brodie': 0, 'Brown': 0,
'Epworth': 0, 'Friedl': 0, 'Gilbert-Addoms': 0, 'Giles': 0, 'Lilly': 0, 'Biddle': 0, 'Pegram': 0, 'Southgate':
'White': 0, 'Wilson': 0, 'Allen': 0, 'Bostock': 0, 'Brodhead': 0, 'Fitzpatrick': 0, 'Flowers': 0, 'Gray': 0,
'Gross': 0, 'Hart': 0, 'Karsh': 0, 'Levine': 0, 'Perkins': 0, 'Reuben-Cooke': 0,  'Rubenstein': 0, 'Sanford': 0
'Teer': 0, 'Wilkinson': 0, 'Craven': 0, 'Crowell': 0, 'Edens': 0, 'Few': 0, 'Keohane': 0,  'Kilgo': 0, 'Wannama
}

# Check each collection author's name for matches
for index, row in df.iterrows():
    for b in buildings.keys():
        if b in str(row['Text']):
            buildings[b] = buildings.get(b) + 1

print(buildings)
```

```
{'Alspaugh': 4, 'Baldwin': 148, 'Bassett': 37, 'Bivins': 1, 'Blackwell': 86, 'Branson': 58, 'Brodie': 15, 'Brow
n': 826, 'Crowell': 10, 'Epworth': 7, 'Friedl': 2, 'Gilbert-Addoms': 0, 'Giles': 80, 'Lilly': 12, 'Biddle': 13
3, 'Pegram': 62, 'Southgate': 55, 'White': 613, 'Wilson': 580, 'Allen': 402, 'Bostock': 3, 'Brodhead': 5, 'Fitz
patrick': 10, 'Flowers': 87, 'Gray': 239, 'Gross': 16, 'Hart': 357, 'Karsh': 2, 'Levine': 0, 'Perkins': 177, 'R
euben-Cooke': 0, 'Rubenstein': 0, 'Sanford': 174, 'Teer': 1, 'Wilkinson': 84, 'Craven': 161, 'Edens': 3, 'Few':
65, 'Keohane': 0, 'Kilgo': 25, 'Wannamaker': 2}
```

When we look in the dataset at large, nearly all of the names are present. However, this is certainly not entirely accurate when considering what we are looking for. As we saw with the presidents deep dive, many names have double meanings or can also be first names. Let's check in just the name column to see if we can yield more accurate results.

In [3]:
```
# Dictionary storing important Duke building names (excluding those not named after people)
buildings = {
'Alspaugh': 0, 'Baldwin': 0, 'Bassett': 0, 'Bivins': 0, 'Blackwell': 0, 'Branson': 0, 'Brodie': 0, 'Brown': 0,
'Epworth': 0, 'Friedl': 0, 'Gilbert-Addoms': 0, 'Giles': 0, 'Lilly': 0, 'Biddle': 0, 'Pegram': 0, 'Southgate':
'White': 0, 'Wilson': 0, 'Allen': 0, 'Bostock': 0, 'Brodhead': 0, 'Fitzpatrick': 0, 'Flowers': 0, 'Gray': 0,
'Gross': 0, 'Hart': 0, 'Karsh': 0, 'Levine': 0, 'Perkins': 0, 'Reuben-Cooke': 0,  'Rubenstein': 0, 'Sanford': 0
'Teer': 0, 'Wilkinson': 0, 'Craven': 0, 'Crowell': 0, 'Edens': 0, 'Few': 0, 'Keohane': 0,  'Kilgo': 0, 'Wannama
}

# Store indices of building names in dictionary
building_indices = {'Alspaugh': [], 'Baldwin': [], 'Bassett': [], 'Blackwell': [], 'Brodie': [], 'Brown': [], '
                    'Giles': [], 'Lilly': [], 'Biddle': [], 'Pegram': [], 'Southgate': [], 'White': [], 'Wilson
                    'Flowers': [], 'Gray': [], 'Hart': [], 'Perkins': [], 'Rubenstein': [], 'Sanford': [], 'Wil
                    'Craven': [], 'Edens': [], 'Few': [], 'Kilgo': []}

# Check each collection author's name for matches
for index, row in df.iterrows():
    if row['Coll_head'] == 1:
        for b in buildings.keys():
            if (b + ',') in str(row['Name']) or (b + ' ') in str(row['Name']):
                buildings[b] = buildings.get(b) + 1
                building_indices[b].append(index)

print(buildings)
print(building_indices)
```

```
{'Alspaugh': 1, 'Baldwin': 9, 'Bassett': 2, 'Bivins': 0, 'Blackwell': 36, 'Branson': 0, 'Brodie': 1, 'Brown': 7
4, 'Crowell': 1, 'Epworth': 0, 'Friedl': 0, 'Gilbert-Addoms': 0, 'Giles': 7, 'Lilly': 2, 'Biddle': 19, 'Pegra
m': 5, 'Southgate': 4, 'White': 42, 'Wilson': 49, 'Allen': 46, 'Bostock': 0, 'Brodhead': 0, 'Fitzpatrick': 0,
'Flowers': 4, 'Gray': 13, 'Gross': 0, 'Hart': 10, 'Karsh': 0, 'Levine': 0, 'Perkins': 6, 'Reuben-Cooke': 0, 'Ru
benstein': 0, 'Sanford': 10, 'Teer': 0, 'Wilkinson': 5, 'Craven': 10, 'Edens': 1, 'Few': 4, 'Keohane': 0, 'Kilg
o': 1, 'Wannamaker': 0}
{'Alspaugh': [644], 'Baldwin': [2093, 2094, 20279, 21958, 21987, 21988, 33105, 33124, 33125], 'Bassett': [2902,
2903], 'Blackwell': [4524, 4525, 4529, 4530, 4532, 4534, 4537, 4549, 4550, 4551, 4552, 4553, 4554, 4555, 4556,
4557, 4558, 4559, 4560, 4561, 4562, 4563, 4564, 4565, 4566, 4567, 4568, 4569, 4570, 4571, 4572, 4573, 4574, 457
5, 4577, 4578], 'Brodie': [5869], 'Brown': [672, 5635, 6019, 6023, 6045, 6046, 6047, 6049, 6059, 6060, 6062, 60
65, 6068, 6069, 6073, 6076, 6077, 6079, 6081, 6082, 6083, 6084, 6123, 6125, 6127, 6134, 6136, 6137, 6138, 6140,
6142, 6145, 6146, 6148, 6150, 6153, 6156, 6157, 6158, 6159, 6160, 6161, 6163, 6164, 6165, 6166, 6167, 6168, 616
9, 6170, 6171, 6172, 6173, 6174, 6175, 6176, 6184, 6188, 6192, 6195, 6196, 6198, 6201, 6202, 6206, 6215, 6223,
6224, 6225, 6226, 6240, 6241, 9600, 9605], 'Crowell': [11387], 'Giles': [17234, 17236, 17242, 17243, 43415, 434
19, 47180], 'Lilly': [10994, 27597], 'Biddle': [4287, 4291, 4292, 4293, 4296, 4299, 4300, 4314, 4315, 4316, 431
7, 4318, 4319, 4320, 4321, 4323, 4324, 38118, 40075], 'Pegram': [35032, 35033, 35034, 35035, 35036], 'Southgat
e': [42029, 42030, 42067, 42068], 'White': [27817, 48046, 48048, 48050, 48051, 48053, 48054, 48056, 48058, 4805
9, 48060, 48061, 48062, 48063, 48064, 48065, 48067, 48077, 48086, 48090, 48091, 48092, 48094, 48100, 48101, 481
10, 48112, 48114, 48118, 48119, 48120, 48122, 48125, 48126, 48129, 48131, 48133, 48134, 48141, 48143, 48144, 48
146], 'Wilson': [7645, 15252, 27146, 33430, 35342, 48955, 48958, 48960, 48961, 48964, 48967, 48968, 48969, 4897
0, 48971, 48980, 48986, 48987, 48990, 48996, 48998, 49000, 49004, 49007, 49008, 49009, 49019, 49021, 49024, 490
28, 49030, 49031, 49034, 49035, 49036, 49038, 49040, 49042, 49043, 49045, 49047, 49049, 49052, 49053, 49056, 49
059, 49062, 49064, 49066], 'Allen': [486, 487, 499, 503, 505, 507, 508, 509, 515, 518, 519, 520, 521, 522, 523,
524, 525, 529, 530, 533, 535, 538, 540, 544, 547, 548, 549, 550, 552, 553, 554, 563, 566, 14979, 15511, 17472,
22453, 34964, 35032, 35033, 35034, 35035, 35292, 35293, 37048, 49468], 'Flowers': [15700, 15708, 15710, 15711],
'Gray': [18368, 18369, 18370, 18372, 18375, 18381, 18383, 18386, 18388, 18390, 18391, 18392, 23022], 'Hart': [2
0569, 20574, 20575, 20577, 20579, 20580, 20582, 20584, 20588, 20593], 'Perkins': [35223, 35226, 35228, 35241, 3
5243, 35246], 'Rubenstein': [], 'Sanford': [37109, 39311, 39349, 39350, 39355, 39357, 39359, 39360, 39363, 3936
7], 'Wilkinson': [48566, 48572, 48574, 48575, 48577], 'Craven': [11142, 11146, 11147, 11153, 11154, 11167, 1118
4, 11192, 11193, 14234], 'Edens': [14394], 'Few': [15285, 15288, 15290, 15293], 'Kilgo': [25715]}
```

In [47]:
```
non_zero = {}

for x, y in buildings.items():
    if y != 0:
        non_zero[x] = y

non_zero = sorted(non_zero.items(), key=operator.itemgetter(1))
```

Alspaugh: 1, Brodie: 1, Crowell: 1, Edens: 1, Kilgo: 1, Bassett: 2, Lilly: 2, Southgate: 4, Flowers: 4, Few: 4, Pegram: 5, Wilkinson: 5, Perkins: 6, Giles: 7, Baldwin: 9, Hart: 10, Sanford: 10, Craven: 10, Gray: 13, Biddle: 19, Blackwell: 36, White: 42, Allen: 46, Wilson: 49, Brown: 74

Above, we can see the building names found in the author names of the collections of the card catalog, sorted in ascending order. Not all of the names were present in the files, but many had at least a couple instances in the files. As we've already looked into some of these in the presidents section, let's see what we can learn from some of the other notable Duke names.

Qualitative Analysis of Duke Building Names

Let's see if any of mentions of Duke building names in the dataset are actually related to the people that the building was named after.

Lilly, Pegram, Wilkinson, Gray, White, Allen, Wilson, Brown

Some of these names, unfortunately, do not have any hits that are directly relevant to the history of the building, but it is still interesting to see the prevalence of Duke-related names, regardless if it is the same specific individuals. Additionally, the library's history of East Campus buildings is found here.

Alspaugh, Bassett, Baldwin

Similar to the presidents, some of these names' only relevant cards are the "See Duke University Archives" cards assoicated with the person the building is named after. See Alspaugh, Bassett, and Baldwin.

There are, however, a couple mentions of the Bassett Affair here and here. The Bassett Affair was when John Spencer Bassett added a sentence in his journal praising Booker T. Washington as one of the best southerners in the past 100 years, enraging many Southern Democrats. President Kilgo and other faculty and students supported Bassett and the Trinity Board of Trustees voted not to accept his resignation, leading to favorable publicity for the college and setting a precident for academic freedom.

Southgate

James Southgate had numerous items recorded in the catalog. Starting with this card, there is a lot of information about James and family. Southgate's son, James Haywood was a Trinity College trustee and is discussed in his father's collections. The senior's letters are described being more of interest than Haywood's, who wrote anout the insurance business and family stress. Kilgo, apparently was a friend of J.H.

Giles

The Giles sisters were the first women to recieve degrees from Trinity college, both undergraduate and graduate. Mary Giles' collection includes papers concerning her and her sisters' education, international travels, and their lives after college.

Blackwell

William Thomas Blackwell was the founder of the Bull Durham Tobacco Company and has many associated collections in the catalog, starting here. The cards discuss his tobacco business and his financial woes. There are many money-related logs, ledgers, and journals.

Biddle

The card corresponding to Mary Duke Biddle's collection is found here. It contains a variety of documents relating to various aspects of her life.

Perkins

William Robertson Perkins was a judge who was counsel to James B. Duke and a trustee of the Duke Endowment. Starting here, his collection discusses his connection to the university and employment.

Duke's Nomenclature

The institution we now know as Duke University has gone through many naming iterations over the years. Starting with Brown School in the nineteenth century, it has also been called Union Institute, Normal College, and, finally, Trinity College before gaining the moniker which we have today. Are these names present in the catalog? And, if so, *when* are these names mentioned? Let's find out.

Finding Name Matches

In [4]:
```
# Lists to hold the dates of Duke name mentions
brown, union, normal, trinity, duke = [], [], [], [], []
counts = [0, 0, 0, 0, 0]

name_indices = {'brown': [], 'union': [], 'normal': [], 'trinity': [], 'duke': []}

# Check each card for a college name
curr_col = 0
for index, row in df.iterrows():
    if row['Coll_head'] == 1:
        curr_col = index
    if not pd.isnull(df.iloc[curr_col]['Year']):
        if "Brown School" in row['Text'].title():
            brown.append(df.iloc[curr_col]['Year'])
            name_indices['brown'].append(index)
            counts[0] += 1
        if "Union Institute" in row['Text'].title():
            union.append(df.iloc[curr_col]['Year'])
            name_indices['union'].append(index)
            counts[1] += 1
        if "Normal College" in row['Text'].title():
            normal.append(df.iloc[curr_col]['Year'])
            name_indices['normal'].append(index)
            counts[2] += 1
        if "Trinity College" in row['Text'].title():
            trinity.append(df.iloc[curr_col]['Year'])
            name_indices['trinity'].append(index)
            counts[3] += 1
        if "Duke University" in row['Text'].title():
            duke.append(df.iloc[curr_col]['Year'])
            name_indices['duke'].append(index)
            counts[4] += 1
```

In [49]:
```
# Dictionary storing indices of cards that mention Duke names
print(name_indices)
```

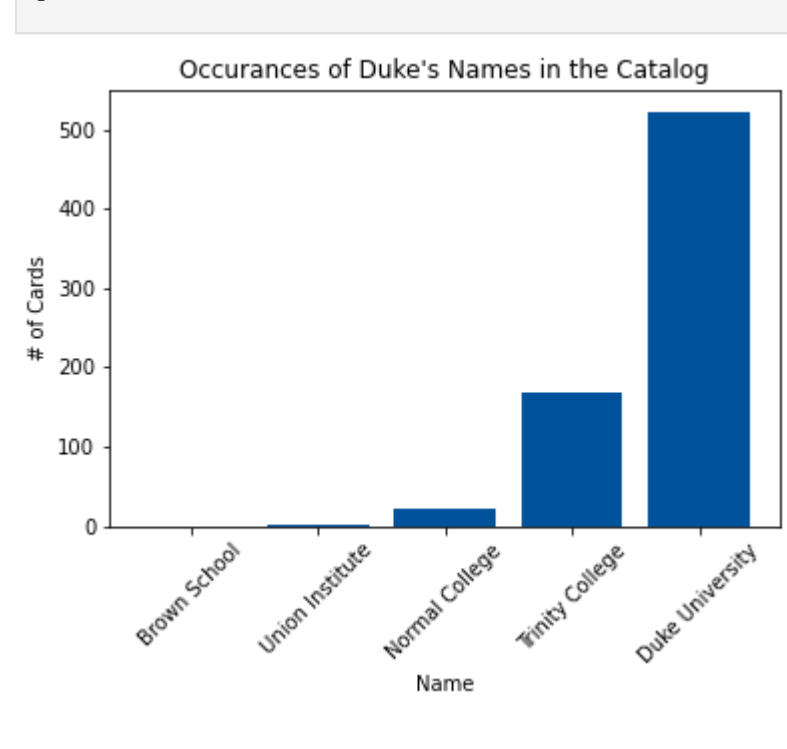
```
{'brown': [], 'union': [31358, 50170], 'normal': [410, 2004, 4821, 5148, 5149, 6904, 7031, 9828, 15130, 15131,
15133, 20074, 20075, 25093, 25094, 25195, 44683, 47523, 47901, 48481, 48488], 'trinity': [127, 158, 932, 949, 1
950, 2004, 2726, 4207, 4294, 4855, 4859, 5861, 5864, 5955, 6085, 6659, 6815, 6917, 6918, 6921, 7684, 7685, 768
7, 7688, 7698, 7700, 8356, 9180, 9645, 10235, 10741, 10906, 10920, 11155, 11156, 11158, 11161, 11164, 11197, 11
199, 11200, 11731, 11822, 12078, 13261, 13302, 13303, 13317, 13671, 13674, 14005, 15133, 16211, 17237, 17238, 1
7239, 17306, 17967, 19231, 19295, 19298, 19384, 20729, 20731, 21237, 21765, 22038, 22129, 22147, 22150, 22158,
23045, 24107, 24513, 24523, 24837, 25004, 25621, 25748, 27216, 27397, 27816, 27836, 28173, 28657, 28659, 28828,
30321, 30375, 30376, 30480, 30706, 30943, 30945, 30947, 30951, 31012, 31013, 31015, 31016, 31017, 31018, 31019,
31247, 31977, 32804, 32886, 34437, 34637, 34654, 35029, 35460, 37321, 37405, 37500, 37748, 37801, 38323, 38324,
39272, 39480, 39481, 39702, 40312, 40454, 40468, 40616, 40644, 40703, 41457, 41590, 41594, 41642, 41914, 42030,
42061, 42065, 42066, 42069, 42408, 42531, 42532, 42919, 42927, 43242, 43348, 43352, 43354, 43616, 43619, 44460,
44474, 44507, 44696, 45276, 45662, 45663, 45693, 45696, 45697, 45808, 46887, 47503, 47523, 49746, 49750, 49751,
50170, 50219], 'duke': [59, 60, 258, 415, 509, 515, 568, 570, 571, 572, 575, 577, 596, 749, 751, 752, 800, 878,
881, 888, 1084, 1085, 1408, 1432, 1594, 1619, 1941, 1950, 2095, 2168, 2178, 2388, 2820, 2829, 3592, 3885, 3912,
4139, 4143, 4273, 4294, 4463, 4465, 4466, 4467, 4471, 4472, 4504, 4664, 4665, 4860, 5148, 5149, 5231, 5488, 548
9, 5610, 5730, 6085, 6086, 6090, 6106, 6116, 6122, 6185, 6186, 6236, 6265, 6434, 6551, 6565, 6630, 6773, 6777,
6844, 6986, 6990, 7021, 7337, 7484, 7683, 7688, 7701, 7974, 8023, 8085, 8090, 8093, 8185, 8191, 8251, 8356, 903
8, 9179, 9180, 9260, 9273, 9404, 9405, 9407, 9410, 9412, 9414, 9417, 9419, 9420, 9645, 9663, 10329, 10354, 1038
6, 10504, 10662, 10743, 10752, 10753, 11037, 11045, 11168, 11175, 11197, 11337, 11735, 11737, 11823, 12145, 121
52, 12533, 12561, 12679, 13053, 13141, 13302, 13318, 13655, 13669, 13692, 13693, 13694, 13695, 13696, 13741, 13
743, 13745, 13746, 13747, 13750, 13751, 13752, 13753, 13754, 13755, 13756, 13757, 13758, 13759, 13760, 13761, 1
3762, 13763, 13764, 13765, 13766, 13767, 13768, 13769, 13770, 13771, 13772, 13773, 13774, 13775, 13776, 13777,
13778, 13779, 13780, 13781, 13782, 13783, 13793, 13795, 13796, 13797, 13798, 13799, 13800, 13801, 13806, 13807,
13808, 13809, 14399, 14484, 14512, 14516, 14517, 14585, 14623, 14861, 14979, 14983, 15294, 15511, 15697, 15751,
15752, 15927, 15939, 16184, 16278, 16529, 16542, 16556, 16557, 16797, 16842, 16922, 17179, 17472, 17493, 17494,
17497, 17498, 17754, 17756, 18432, 18666, 18712, 18713, 18952, 19185, 19309, 19311, 19334, 19384, 19444, 19616,
19647, 19648, 19650, 19651, 19652, 19653, 19658, 19660, 19662, 19666, 19771, 19865, 19909, 19912, 19920, 20156,
20210, 20212, 20214, 20318, 20523, 20542, 20549, 20823, 21005, 21396, 21508, 21566, 21616, 21723, 21815, 21954,
22050, 22265, 22266, 22267, 22272, 22285, 22594, 22595, 22597, 22618, 22772, 22868, 22869, 22871, 22893, 22895,
22905, 22921, 22922, 22923, 22926, 22943, 23288, 23343, 23775, 23909, 24166, 24237, 24525, 24619, 24838, 25049,
25074, 25251, 25262, 25263, 25264, 25665, 26306, 26307, 26393, 26398, 26399, 26603, 26801, 26854, 27338, 27340,
27457, 27486, 27719, 27720, 27790, 27802, 27889, 28247, 28561, 28601, 28658, 28718, 28828, 29203, 29456, 29573,
29623, 30375, 30376, 30650, 30666, 30677, 31230, 31322, 31509, 31813, 31892, 31899, 32183, 32297, 32343, 32851,
32877, 32971, 33042, 33092, 33642, 33672, 33917, 33922, 34078, 34080, 34136, 34437, 34537, 34539, 34654, 34850,
34882, 35029, 35030, 35031, 35234, 35237, 35251, 35367, 35373, 35418, 35620, 35990, 36033, 36215, 36299, 36493,
36495, 36496, 36542, 36652, 36711, 36804, 36868, 36869, 37052, 37205, 37214, 37221, 37260, 37261, 37263, 37405,
37612, 37613, 37656, 37657, 37801, 37847, 37849, 38056, 38063, 38064, 38071, 38235, 38401, 38412, 38679, 38866,
39017, 39135, 39255, 39258, 39271, 39294, 39295, 39297, 39311, 39313, 39314, 39673, 39675, 39679, 39680, 39681,
39744, 39745, 39747, 39748, 40073, 40076, 40077, 40078, 40080, 40082, 40083, 40235, 40319, 40337, 40338, 40616,
40703, 40878, 40952, 40978, 41173, 41174, 41568, 41644, 41703, 41736, 41775, 42007, 42051, 42069, 42377, 42533,
42780, 42991, 42994, 43032, 43238, 43242, 43243, 43244, 43247, 43338, 43348, 43352, 43369, 43370, 43417, 43598,
43599, 43616, 43760, 43762, 43763, 43996, 44012, 44200, 44245, 44447, 44459, 44513, 44535, 44812, 44857, 44859,
45317, 45318, 45704, 45710, 45866, 46093, 46469, 46470, 46473, 46685, 47102, 47152, 47173, 47272, 47601, 47703,
47864, 47880, 47979, 48169, 48439, 48644, 48709, 49487, 49585, 49586, 49592, 49716, 50170]}
```

It looks like we have 0 mentions of Brown School, 2 mentions of Union Institute, 21 mentions of Normal College, 169 mentions of Trinity College, and 523 mentions of Duke University. All mentions are associated with valid dates.

In [2]:
```
# Values stored after running above code
counts = [0, 2, 21, 169, 523]
names = ["Brown School", "Union Institute", "Normal College", "Trinity College", "Duke University"]

brown = []
union = ['1863-1890', '1879-1889']
normal = ['1833-1985', '1852-1853', '1881-1919', '1846-1933', '1846-1933', '1820-1907', '1783-1940', '1840-1925
trinity = ['1891-1913', '1901-1922', '1847-1890', '1865', '1833-1967', '1852-1853', '1924-1971', '1859', '1887-
duke = ['1973-1989', '1973-1989', '1967-1995', '1962', '1990-1995', '1923-1960', '1843-1971', '1843-1971', '184
```

Name Frequencies

In [12]:
```
# Create bar chart of Duke name frequencies
plt.bar(names, counts, color='#00539B')
plt.xticks(rotation = 45)
plt.title("Occurances of Duke's Names in the Catalog")
plt.xlabel("Name")
plt.ylabel("# of Cards")
plt.show()
```



Average Collection Dates by Name

Let's find the average date of a collection that mentions each of these college names. For this, we will find the average date for each instance of a found name--either a single date or the simple average of the collection's start and end dates--and then the mean of all the instances of the college name.

In [31]:
```
def find_mean(list):
    total, num = 0, 0
    for date in list:
        dates = date.split("-")
        if len(dates) == 1:
            total += int(dates[0])
        else:
            total += ((int(dates[0]) + int(dates[1])) / 2)
        num += 1
    return round(total / num)

averages = []
averages.append(find_mean(union))
averages.append(find_mean(normal))
averages.append(find_mean(trinity))
averages.append(find_mean(duke))
averages
```

Out[31]:
```
[1880, 1871, 1884, 1917]
```

Actual Dates of Institutions:

Brown School: 1838-1841

Union Institution: 1841-1851

Normal College: 1851-1859

Trinity College: 1859-1924

Duke University: 1924-present

Average Associated Date in the Catalog:

Brown School: N/A

Union Institution: 1880

Normal College: 1871

Trinity College: 1884

Duke University: 1917

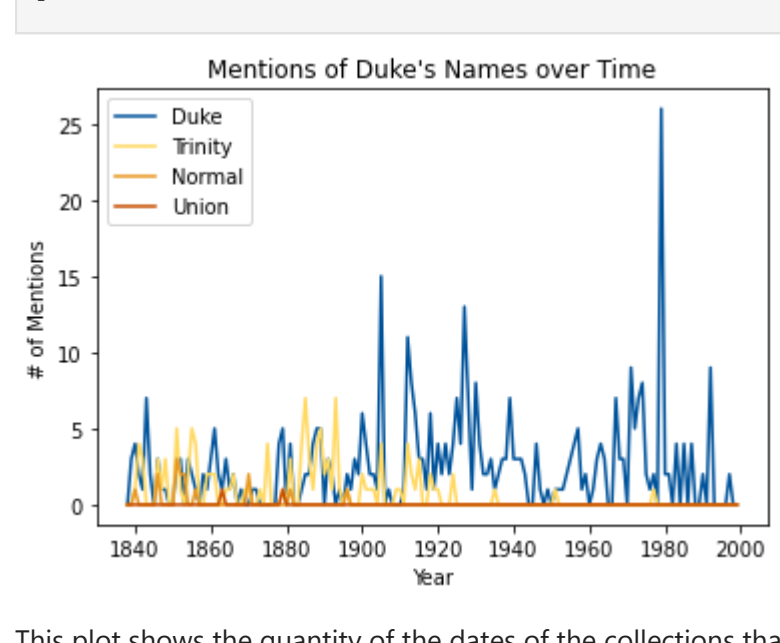
Names Time Series

Brown School will not be included in the following analysis, as there were no instances of it in the catalog. But, let's see how the other four names' instances are distributed over time in the catalog. For simplicity's sake, for year ranges of catalog collections, each year in the range was counted once (e.g., for 1940-1942, 1940, 1941, and 1941 each got a quantity increase of 1).

In [8]:
```
def start(college, dic):
    for i in range(1838, 2000):
        dic[i] = 0

    for date in college:
        dates = str(date).split("-")
        if int(dates[0]) >= 1838:
            dic[int(dates[0])] = dic.get(int(dates[0])) + 1

    return dic
```

In [9]:
```
dic1, dic2, dic3, dic4 = {}, {}, {}, {}
dic1 = start(union, dic1)
dic2 = start(normal, dic2)
dic3 = start(trinity, dic3)
dic4 = start(duke, dic4)
```

In [10]:
```
plt.plot(dic4.keys(), dic4.values(), label='Duke', color='#00539B')
plt.plot(dic3.keys(), dic3.values(), label='Trinity', color='#FFD960')
plt.plot(dic2.keys(), dic2.values(), label='Normal', color='#E89923')
plt.plot(dic1.keys(), dic1.values(), label='Union', color='#C84E00')
plt.legend(loc='upper left')
plt.title("Mentions of Duke's Names over Time")
plt.xlabel("Year")
plt.ylabel("# of Mentions")
plt.show()
```



This plot shows the quantity of the dates of the collections that mention one of Duke's names, for each of the names.

A major caveat to this is that we were only able to pull out the dates of when a collection was writen, not when it was acquired by the library, and this plot shows the dates when the collections that mention one of these full names were written.