In [3]:
```
from bs4 import BeautifulSoup

with open("female-detainee-cases.html", "r", encoding="utf-8") as f:
    soup = BeautifulSoup(f, "html.parser")
```

In [4]:
```
# grab every <a> tag
all_links = soup.find_all("a")

# keep only those whose text starts with “Case ” and whose href ends with “.
case_links = [
    a for a in all_links
    if a.text.strip().startswith("Case ")
    and a.get("href", "").endswith(".html")
]
```

In [5]:
```
records = []
for a in case_links:
    href = a["href"]
    text = a.get_text(strip=True)
    # e.g. "Case 2657 Moy Chin See his wife"
    records.append({"href": href, "raw_text": text})
```

In [6]:
```
import re

parsed = []
pattern = re.compile(r"Case\s+(\d+)\s+(.+)")
for rec in records:
    m = pattern.match(rec["raw_text"])
    if not m:
        # flag for manual review
        parsed.append({
            **rec,
            "case_number": None,
            "name": None,
            "descriptor": None,
            "note": "FAILED TO PARSE"
        })
        continue

    num = m.group(1)
    remainder = m.group(2)  # e.g. "Moy Chin See his wife"

    # Heuristic: split off a trailing descriptor like "his wife", "alias …",
    # You may need to refine this for cases like "Kwok Ah Ying and Kowk Sue
    parts = re.split(r"\s+(alias|nee|wife|daughter|and)\b", remainder, maxsp
    if len(parts) == 1:
        name, descriptor = parts[0], ""
    else:
        name = parts[0].strip()
        descriptor = remainder[len(name):].strip()

    parsed.append({
        **rec,
        "case_number": num,
        "name": name,
        "descriptor": descriptor
    })
```

In [7]:
```
import pandas as pd

df = pd.DataFrame(parsed)

# Optional cleaning:
df["case_number"] = df["case_number"].astype("Int64")  # integer column
df["name"] = df["name"].str.replace(r"^Mrs\.\s*", "", regex=True)
df["descriptor"] = df["descriptor"].str.replace(r"[()]", "", regex=True)


df[:10]
```

Out[7]:

	href	raw_text	case_number	name	descriptor
0	2657.html	Case 2657 Moy Chin See his wife	2657	Moy Chin See his	wife
1	2917.html	Case 2917 Lee Kin Sai alias Lee Wah Chung	2917	Lee Kin Sai	alias Lee Wah Chung
2	2950.html	Case 2950 Tie Yimm a woman	2950	Tie Yimm a woman	
3	3068.html	Case 3068 Lin Kum daughter, Wye See mother	3068	Lin Kum	daughter, Wye See mother
4	3100.html	Case 3100 Tarm How Yen wife	3100	Tarm How Yen	wife
5	3308.html	Case 3308 Yung Ah Chung woman	3308	Yung Ah Chung woman	
6	3549.html	Case 3549 Mrs. Fong Ah Chung	3549	Fong Ah Chung	
7	3644.html	Case 3644 Mrs. Ching Din	3644	Ching Din	
8	3745.html	Case 3745 Mrs. Lee nee Chun Ah On	3745	Lee	nee Chun Ah On
9	3763.html	Case 3763 Mrs. Leong nee Lee Ah Fung	3763	Leong	nee Lee Ah Fung

In [8]:
```
df.to_csv("female_detainee_cases.csv", index=False, encoding="utf-8")
```

updated code below

In [ ]:

In [10]:
```
from bs4 import BeautifulSoup
import pandas as pd
import re
from collections import defaultdict

# 1. Load the HTML file
with open("female-detainee-cases.html", "r", encoding="utf-8") as f:
    soup = BeautifulSoup(f, "html.parser")

# 2. Helper to determine if link is valid
def is_case_link(tag):
    href = tag.get("href", "")
    text = tag.get_text(strip=True)
    return (
        text.lower().startswith("case ")
        and ".html" in href
        and not href.endswith(".pdf")
        and "google.com" not in href
    )

# 3. Extract all case-related links
case_links = [a for a in soup.find_all("a") if is_case_link(a)]

# 4. Group by case number using dictionary
cases = defaultdict(lambda: {"hrefs": [], "raw_texts": []})

case_pattern = re.compile(r"Case\s+(\d+)\s+(.*)", re.IGNORECASE)

for tag in case_links:
    href = tag["href"]
    text = tag.get_text(strip=True)

    match = case_pattern.match(text)
    if match:
        case_num = int(match.group(1))
        remainder = match.group(2).strip()
        cases[case_num]["hrefs"].append(href)
        cases[case_num]["raw_texts"].append(remainder)

# 5. Normalize and clean names/descriptors
def clean_name_and_descriptor(raw_name):
    name = raw_name
    descriptor = ""

    # Remove "Mrs." and similar prefixes
    name = re.sub(r"^Mrs\.?\s*", "", name, flags=re.IGNORECASE)

    # Extract trailing known descriptors
    known_descriptors = [
        "a woman", "woman", "his wife", "wife", "daughter", "mother", "recor
        "testimony.*", "appeal", r"\(.*\)", "sisters"
    ]
    for desc in known_descriptors:
        pattern = rf"\b{desc}\b"
        match = re.search(pattern, name, re.IGNORECASE)
        if match:
            descriptor = match.group(0)
            name = re.sub(pattern, "", name, flags=re.IGNORECASE).strip()
            break

    # Handle "alias" and "nee"
    if ' alias ' in name:
        name, extra = name.split(' alias ', 1)
        descriptor = f"alias {extra.strip()}"
    elif ' nee ' in name:
        name, extra = name.split(' nee ', 1)
        descriptor = f"nee {extra.strip()}"
    elif ' and ' in name:
        # keep multi-person name together, e.g., sisters
        descriptor = descriptor or "multiple individuals"

    return name.strip(), descriptor.strip()

# 6. Build final structured data
records = []
for case_number, info in sorted(cases.items()):
    combined_text = " / ".join(info["raw_texts"])
    combined_links = "; ".join(sorted(set(info["hrefs"])))

    name, descriptor = clean_name_and_descriptor(combined_text)

    records.append({
        "case_number": case_number,
        "name": name,
        "descriptor": descriptor,
        "hrefs": combined_links,
        "raw_text": combined_text
    })

# 7. Output as DataFrame and CSV
fem_df = pd.DataFrame(records)
fem_df = df.sort_values(by="case_number")

fem_df.to_csv("female_detainee_cases_cleaned_grouped.csv", index=False, enco
print(f"✅ Cleaned {len(df)} grouped case records.")
```

```
✅ Cleaned 138 grouped case records.
```

In [11]:
```
fem_df.sample(10)
```

Out[11]:

	href	raw_text	case_number	name	descriptor
56	5062b4.html	Case 5062 Quock Ah Sip Testimony pages 72 to 99	5062	Quock Ah Sip Testimony pages 72 to 99	
133	10116.html	Case 10116 Chin Chon Loy	10116	Chin Chon Loy	
127	9722.html	Case 9722 Mar Chew Kook	9722	Mar Chew Kook	
120	8978.html	Case 8978 Dong Que Far	8978	Dong Que Far	
0	2657.html	Case 2657 Moy Chin See his wife	2657	Moy Chin See his	wife
7	3644.html	Case 3644 Mrs. Ching Din	3644	Ching Din	
47	5057.html	Case 5057 Che Tue Far	5057	Che Tue Far	
21	4969.html	Case 4969 Wong Chow Ling	4969	Wong Chow Ling	
85	5106.html	Case 5106 Wong You Choy	5106	Wong You Choy	
95	5316.html	Case 5316 Lee Ngau Yook	5316	Lee Ngau Yook	

In [12]:
```
hc_df= pd.read_csv('habeas-corpus-cases-1889-1892.csv')



#hc_df.to_csv('habeas_csv_sample.csv',index=False)
```

In [13]:
```
!pip install geopy
```

```
Requirement already satisfied: geopy in /opt/anaconda3/lib/python3.12/site-p
ackages (2.4.1)
Requirement already satisfied: geographiclib<3,>=1.52 in /opt/anaconda3/lib/
python3.12/site-packages (from geopy) (2.0)
```

In [14]:
```
import pandas as pd

df = pd.read_csv('habeas_csv_sample.csv')
print(df.info())
print(df.head())
df = df.dropna(axis=1, how='all')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 11 columns):
 #   Column                     Non-Null Count  Dtype 
---  ------                     --------------  ----- 
 0   CASE NUMBER                20 non-null     int64 
 1   YEAR                       20 non-null     int64 
 2   FOR RELIEF OF              20 non-null     object
 3   STEAM SHIP NUMBER          20 non-null     object
 4   CHARACTER OF CASE          20 non-null     object
 5   BY WHOM OR WHERE DETAINED  20 non-null     object
 6   ATTORNEY FOR PETITION      20 non-null     object
 7   REMARKS                    20 non-null     object
 8   NAME OF FATHER             20 non-null     object
 9   ADDRESS                    20 non-null     object
 10  Age or year of birth       20 non-null     object
dtypes: int64(2), object(9)
memory usage: 1.8+ KB
None
   CASE NUMBER  YEAR     FOR RELIEF OF STEAM SHIP NUMBER CHARACTER OF CASE 
\
0        10169  1890      Soho One Dun                         Native Born 
1        10197  1891      Gin Heng Lee                         Native Born 
2         9901  1890      Woo Moon Kee                         Native Born 
3        10004  1890      Jee Hung Hee                         Native Born 
4         9135  1890  Jong Foong Fooey                         Native Born 

  BY WHOM OR WHERE DETAINED ATTORNEY FOR PETITION     REMARKS  \
0                     China  Schaertzer, Henry C.  Discharged  
1            City of Peking          Mowry, Lyman    Remanded  
2                   Oceanic     Riordan, Thomas D    Remanded  
3                    Gaelic     Riordan, Thomas D  Discharged  
4    City of Rio de Janeiro       Stranahan, F.E.  Discharged  

     NAME OF FATHER         ADDRESS Age or year of birth  
0    So Ho Yee Gawk  723 Sacramento           about 1869  
1       Gin Wah Kew  727 Sacramento                 1861  
2  Woo Shoo Cheong      728 Dupont                 1868  
3     Jee Yooey Too      821 Dupont                 1874  
4          Jong Foo  808 Sacramento                 1868  
```

In [15]:
```python
import pandas as pd
df = pd.read_csv("habeas_csv_sample.csv")

df.head()
```

Out[15]:

	CASE NUMBER	YEAR	FOR RELIEF OF	STEAM SHIP NUMBER	CHARACTER OF CASE	BY WHOM OR WHERE DETAINED	ATTORNEY FOR PETITION	REMARKS
0	10169	1890	Soho One Dun		Native Born	China	Schaertzer, Henry C.	Dischargec
1	10197	1891	Gin Heng Lee		Native Born	City of Peking	Mowry, Lyman	Remandec
2	9901	1890	Woo Moon Kee		Native Born	Oceanic	Riordan, Thomas D	Remandec
3	10004	1890	Jee Hung Hee		Native Born	Gaelic	Riordan, Thomas D	Dischargec
4	9135	1890	Jong Foong Fooey		Native Born	City of Rio de Janeiro	Stranahan, F.E.	Dischargec

In [16]:

```
# Identify columns where >90% of entries are NaN/empty
empty_frac = df.isna().mean()
to_drop = empty_frac[ empty_frac > 0.9 ].index.tolist()
df.drop(columns=to_drop, inplace=True)

## what it do


hc_df.columns
```

Out[16]:

```
Index(['CASE NUMBER', 'YEAR', 'FOR RELIEF OF', 'STEAM SHIP NUMBER',
       'CHARACTER OF CASE', 'BY WHOM OR WHERE DETAINED',
       'ATTORNEY FOR PETITION', 'REMARKS', 'NAME OF FATHER', 'ADDRESS',
       'Age or year of birth'],
      dtype='object')
```

Rename Columns to Snake_Case

In [18]:

```
def to_snake(name):
    return (
      name.strip()
          .lower()
          .replace("%", "pct")
          .replace(" or ", "_")
          .replace("  ", " ")
          .replace(" ", "_")
    ) ##jfc

hc_df.columns=[to_snake(d) for d in hc_df.columns]
```

```
hc_df["for_relief_of"] = hc_df["for_relief_of"].str.strip().replace(r"\s+",
hc_df["name_of_father"]    = hc_df["name_of_father"].str.strip()
hc_df.columns
```

Out[18]:
```
Index(['case_number', 'year', 'for_relief_of', 'steam_ship_number',
       'character_of_case', 'by_whom_where_detained', 'attorney_for_petitio
n',
       'remarks', 'name_of_father', 'address', 'age_year_of_birth'],
      dtype='object')
```

In [19]:
```
hc_df["year"]        = hc_df["year"].astype(int)
hc_df["case_number"] = hc_df["case_number"].astype(int)
hc_df["address"] = hc_df["address"].str.strip()


missing_city = ~hc_df["address"].str.contains(r",")
hc_df.loc[missing_city, "address"] += ", San Francisco, CA"
```

Addressing the mixed ages columns

In [21]:
```
import pandas as pd
import re

def parse_birth_year(raw, case_year):
    """
    raw: the original cell (e.g. "25 years", "1869", "18")
    case_year: the year the case was filed
    returns: an int birth_year or None
    """
    if pd.isna(raw):
        return None
    s = str(raw).strip()
    # extract the first group of digits
    m = re.search(r"(\d{1,4})", s)
    if not m:
        return None
    val = int(m.group(1))
    # decide if this is an age or an actual year
    if val < 120:
        return case_year - val
    elif val >= 1800:
        return val
    else:
        # e.g. a weird 3-digit number like "189"—ambiguous
        return None

# apply it:
hc_df["birth_year"] = hc_df.apply(
    lambda row: parse_birth_year(row["age_year_of_birth"], row["year"]), axi
)

# (Optionally) drop the old mixed column
#df.drop(columns=["age_or_year_of_birth"], inplace=True)
```

```
# I didn't run that even though ChatGPT chugged it because it still has impo
hc_df.sample(1)
```

Out[21]:

	case_number	year	for_relief_of	steam_ship_number	character_of_case	by_w
1125	10061	1890	Jew Sin Yook		Native Born	

In [22]:

```
hc_df[hc_df["birth_year"].isna()][["age_year_of_birth", "year"]]
```

Out[22]:

	age_year_of_birth	year
30		1889
62		1889
102		1889
120		1889
150		1889
165	1688	1889
232		1890
255		1890
256		1890
259		1890
260		1890
289		1890
297		1890
301		1890
333	1668	1890
347		1890
349		1890
374		1890
409		1890
417		1890
441		1890
442		1890
476		1890
586		1890
654		1890
656		1890
792		1890
1149		1890
1205		1890
1230		1890
1259		1891
1260		1891

	age_year_of_birth	year
1263		1891
1266		1891
1267		1891
1268		1892
1271		1892
1272		1892
1273		1892
1274		1892
1275		1892
1276		1892
1277		1892
1278		1892
1279		1892
1280		1892
1281		1892

In [23]:
```python
print(hc_df[165:166]) # wrongly written baby entry?

hc_df[1280:1281] #missing data
```

```
    case_number  year for_relief_of steam_ship_number character_of_case  \
165        9072  1889   Leong Yun Po                          Native Born   

   by_whom_where_detained attorney_for_petition     remarks  name_of_father  
\
165                 Belgic     Riordan, Thomas D  Discharged  Leong Jung One   

                          address age_year_of_birth  birth_year  
165  940 Dupont, San Francisco, CA              1688         NaN  
```

Out[23]:

	case_number	year	for_relief_of	steam_ship_number	character_of_case	by_w
1280	10318	1892	Doo Dai Hoy (female)		Wife of resident merchant	

In [24]:
```python
def validate_age_year(raw, case_year):
    """
    raw: the original 'age_or_year_of_birth' entry (could be "25 years", "18
    case_year: the year the case was filed (int)
    returns: a flag string ("" if OK, otherwise a tag)
    """
    # 1) Missing entirely?
    if pd.isna(raw) or str(raw).strip() == "":
        return "Missing"

    s = str(raw).strip().lower()

    # 2) Must match 1–4 digits, optional 'year' or 'years' suffix, and nothi
    m = re.fullmatch(r"(\d{1,4})(?:\s*years?)?", s)
    if not m:
        return "Invalid Format"

    val = int(m.group(1))

    # 3) Now decide if it's an age or a birth year
    if val < 120:
        # treated as age → compute implied birth year
        birth = case_year - val
        # flag if that birth year is outside a reasonable window
        if birth < 1800 or birth > case_year:
            return "Suspicious Age"
    else:
        # treated as birth year
        if val < 1800 or val > case_year:
            return "Suspicious Year"

    # 4) If we got here, it passed all checks
    return ""

# Apply across your DataFrame:
hc_df["age_year_flag"] = hc_df.apply(
    lambda row: validate_age_year(row["age_year_of_birth"], row["year"]),
    axis=1
)

# Then inspect only the flagged rows:
flags = hc_df[hc_df["age_year_flag"] != ""]

flags.sample(5)
```

Out[24]:

	case_number	year	for_relief_of	steam_ship_number	character_of_case	by_w
165	9072	1889	Leong Yun Po		Native Born	
301	9218	1890	Chin Leong Shee (woman)		Merchant's wife	
476	9396	1890	Low Sun Kwy (female)		Native Born	
1274	10307	1892	Ho Hon		Resident Merchant	
441	9361	1890	Lum Toong		Resident Merchant	

In [25]:

```
def validate_age_year(raw, case_year):
    """
    raw: the original 'age_or_year_of_birth' entry
         (could be "25 years", "1869", "about 1869", "child", etc.)
    case_year: the year the case was filed (int)
    returns: a flag string ("" if OK, otherwise a tag)
    """
    # 1) Missing entirely?
    if pd.isna(raw) or str(raw).strip() == "":
        return "Missing"

    s = str(raw).strip().lower()

    # 2) Match 1–4 digits, optionally preceded by 'about' or 'circa',
    #    and optionally followed by 'year' or 'years'
    pattern = r"(?:(?:about|circa)\s*)?(\d{1,4})(?:\s*years?)?"
    m = re.fullmatch(pattern, s)
    if not m:
        return "Invalid Format"

    val = int(m.group(1))

    # 3) Decide if it's an age or a birth year
    if val < 120:
        # treated as age → compute implied birth year
        birth = case_year - val
        # flag if that birth year is outside a reasonable window
        if birth < 1800 or birth > case_year:
            return "Suspicious Age"
    else:
        # treated as birth year
        if val < 1800 or val > case_year:
            return "Suspicious Year (Baby?)"

    # 4) Passed all checks
    return ""

# Apply to DataFrame:
hc_df["age_year_flag"] = hc_df.apply(
    lambda row: validate_age_year(row["age_year_of_birth"], row["year"]),
    axis=1
)

# Inspect flagged rows:
#flags = hc_df[hc_df["age_year_flag"] != ""]
```

In [26]:
```
df= hc_df.sample(20)
```

Attempting to Geocode Latitude and Longitude

In [28]:
```
from geopy.geocoders import Nominatim
from time import sleep

geolocator = Nominatim(user_agent="habeas_geo")
latitudes, longitudes = [], []

for addr in df["address"]:
    try:
        loc = geolocator.geocode(addr, timeout=10)
        latitudes.append(loc.latitude if loc else None)
        longitudes.append(loc.longitude if loc else None)
    except Exception:
        latitudes.append(None)
        longitudes.append(None)
    sleep(1)  # be polite!
df["latitude"]  = latitudes
df["longitude"] = longitudes
```

In [29]:
```
##ok it took forever but I tried geocoding a sample of 20 with the ChatGPT c
df[df["latitude"].notna()]
```

Out[29]:

	case_number	year	for_relief_of	steam_ship_number	character_of_case	by_w
128	9032	1889	Lee Ah Sik		Native Born	
507	9429	1890	Wong Wah Yun		Native Born	
539	9462	1890	Wong Ah Loon		Native Born	
1009	9937	1890	Chin Ah Nong		Native Born	
136	9040	1889	Jin Ah Yen		Native Born	
200	9110	1890	Loui Wing Sing		Native Born	
640	9565	1890	Soo Yow		Native Born	
222	9135	1890	Jong Foong Fooey		Native Born	
305	9222	1890	Gee Bing Jow		Native Born	
565	9488	1890	Jung Ah Chung		Native Born	
805	9730	1890	Lee Suey Chung		Native Born	
696	9621	1890	Tom Ping Leaum		Native Born	
639	9564	1890	Lim Ah Chee		Native Born	

case_number year for_relief_of steam_ship_number character_of_case by_w

In [30]:
```
import numpy as np

# Calculate age at time of case:
#   age_at_case = case_year - birth_year
# We'll get NaN for any rows where birth_year is missing.
hc_df["age_at_case"] = hc_df["year"] - hc_df["birth_year"]

# Optionally, force to integer where non-null (e.g. 25.0 → 25)
hc_df["age_at_case"] = hc_df["age_at_case"].where(hc_df["age_at_case"].notna

# Quick sanity-check:
print(hc_df[["year", "birth_year", "age_at_case"]].head(10))
print("\nAny negative or implausible ages?")
print(hc_df.loc[hc_df["age_at_case"] < 0, ["year", "birth_year", "age_at_cas
```

```
   year  birth_year  age_at_case
0  1889      1870.0         19.0
1  1889      1871.0         18.0
2  1889      1864.0         25.0
3  1889      1872.0         17.0
4  1889      1869.0         20.0
5  1889      1874.0         15.0
6  1889      1869.0         20.0
7  1889      1868.0         21.0
8  1889      1864.0         25.0
9  1889      1874.0         15.0

Any negative or implausible ages?
Empty DataFrame
Columns: [year, birth_year, age_at_case]
Index: []
```

In [31]:
```
# If you have any missing birth_years, use the pandas nullable Int64 dtype:
hc_df["birth_year"] = hc_df["birth_year"].astype("Int64")

# Confirm the dtype change:
print(hc_df["birth_year"].dtype)
# → Int64
```

```
Int64
```

In [32]:
```
hc_df.to_csv("cleaned_habeas_corpus_cases.csv", index=False)
```

In [33]:
```
# Load the cleaned dataset
df = pd.read_csv('cleaned_habeas_corpus_cases.csv')

# Select 25 random rows
sample_df = df.sample(n=25)

# Save the sample to a new CSV file
sample_df.to_csv('sample_habeas_corpus_cases.csv', index=False)
```

Claude section

In [35]:
```
# Display basic information about the dataset
print(f"Total number of cases: {len(hc_df)}")

# Analyze case outcomes
outcome_counts = hc_df['remarks'].value_counts()
print("\nCase Outcomes:")
for outcome, count in outcome_counts.items():
    print(f"- {outcome}: {count}")

# Calculate percentages
outcome_percentages = outcome_counts / len(df) * 100
print("\nOutcome Percentages:")
for outcome, percentage in outcome_percentages.items():
    print(f"- {outcome}: {percentage:.1f}%")
```

```
Total number of cases: 1284

Case Outcomes:
- Discharged: 723
- Remanded: 496
- Remanded appealed to Circuit Court: 14
-  : 11
- Petition and writ dismissed: 9
- Petition dead: 9
- Petition Dead: 7
- Writ returned non est: 5
- Petition to dismiss: 2
- Writ returned: 2
- Writ not served: 2
- Boond exonerated: 1
- Bail exonerated: 1
- Writ and Petition dismissed: 1
- Landed by Customs House: 1

Outcome Percentages:
- Discharged: 56.3%
- Remanded: 38.6%
- Remanded appealed to Circuit Court: 1.1%
-  : 0.9%
- Petition and writ dismissed: 0.7%
- Petition dead: 0.7%
- Petition Dead: 0.5%
- Writ returned non est: 0.4%
- Petition to dismiss: 0.2%
- Writ returned: 0.2%
- Writ not served: 0.2%
- Boond exonerated: 0.1%
- Bail exonerated: 0.1%
- Writ and Petition dismissed: 0.1%
- Landed by Customs House: 0.1%
```

In [36]:
```
# Additional analysis: Examine if there's any relationship between age and c
print("\nAge Statistics by Outcome:")
```

```
age_by_outcome = df.groupby('remarks')['age_at_case'].agg(['mean', 'median',
print(age_by_outcome)

# Analyze outcomes by attorney
print("\nCase Outcomes by Attorney:")
attorney_outcomes = pd.crosstab(df['attorney_for_petition'], df['remarks'])
print(attorney_outcomes)

# Calculate success rates for attorneys with at least 3 cases
print("\nAttorney Success Rates (for attorneys with at least 3 cases):")
attorney_counts = df['attorney_for_petition'].value_counts()
frequent_attorneys = attorney_counts[attorney_counts >= 3].index

for attorney in frequent_attorneys:
    attorney_df = df[df['attorney_for_petition'] == attorney]
    total_cases = len(attorney_df)
    discharged = len(attorney_df[attorney_df['remarks'] == 'Discharged'])
    success_rate = discharged / total_cases * 100
    print(f"- {attorney}: {discharged}/{total_cases} ({success_rate:.1f}%)")
```

```
Age Statistics by Outcome:
                                        mean  median   min   max  count
remarks
                                   21.000000    20.5  14.0  29.0     10
Bail exonerated                    21.000000    21.0  21.0  21.0      1
Boond exonerated                   21.000000    21.0  21.0  21.0      1
Discharged                         20.608696    21.0   4.0  32.0    690
Landed by Customs House            20.000000    20.0  20.0  20.0      1
Petition Dead                      22.833333    22.5  21.0  26.0      6
Petition and writ dismissed        22.000000    22.0  18.0  24.0      9
Petition dead                      20.888889    22.0  14.0  25.0      9
Petition to dismiss                      NaN     NaN   NaN   NaN      0
Remanded                           20.429448    21.0   7.0  30.0    489
Remanded appealed to Circuit Court 21.538462    22.0  14.0  27.0     13
Writ and Petition dismissed        23.000000    23.0  23.0  23.0      1
Writ not served                     7.500000     7.5   5.0  10.0      2
Writ returned                      15.000000    15.0  15.0  15.0      2
Writ returned non est              23.000000    23.0  20.0  26.0      3

Case Outcomes by Attorney:
remarks                               Bail exonerated  Boond exonerated  \
attorney_for_petition
Bergen, Benjamin                 0                  0                 0
Blaney, Edward W.                0                  1                 0
Carroll Cook                     0                  0                 0
Cook, Carroll                    0                  0                 0
Cross & Denson                   0                  0                 0
Hilborn & Hall                   0                  0                 0
Lande, Edward                    0                  0                 0
McAllister                       0                  0                 0
McAllister & McAllister          0                  0                 0
McAllister, Jr. , Ward           0                  0                 0
Miller, H B M                    0                  0                 0
Mowry, Lyman                     3                  0                 1
Naphtaly, Joseph                 0                  0                 0
Perry, G. H. & Ricketts, Alfred  0                  0                 0
Perry, George H                  0                  0                 0
Ricketts, Alfred                 0                  0                 0
Riordan, Thomas D                5                  0                 0
Schaertzer, Henry C.             0                  0                 0
Schlesinger, Bert                0                  0                 0
Smith                            0                  0                 0
Stonehill & Whaley               0                  0                 0
Stranahan & Smith                0                  0                 0
Stranahan, F. E.                 0                  0                 0
Stranahan, F.E.                  3                  0                 0
Talcott, H.D.                    0                  0                 0
van Duzer, A P                   0                  0                 0

remarks                          Discharged  Landed by Customs House  \
attorney_for_petition
Bergen, Benjamin                          2                        0
Blaney, Edward W.                        32                        0
Carroll Cook                              1                        0
Cook, Carroll                             1                        0
Cross & Denson                            3                        0
```

```
Hilborn & Hall                        0                         0
Lande, Edward                         2                         0
McAllister                            0                         0
McAllister & McAllister               0                         0
McAllister, Jr. , Ward                1                         0
Miller, H B M                         1                         0
Mowry, Lyman                        189                         0
Naphtaly, Joseph                      1                         0
Perry, G. H. & Ricketts, Alfred       2                         0
Perry, George H                       0                         0
Ricketts, Alfred                    103                         0
Riordan, Thomas D                   170                         1
Schaertzer, Henry C.                 17                         0
Schlesinger, Bert                     8                         0
Smith                                 3                         0
Stonehill & Whaley                    2                         0
Stranahan & Smith                     1                         0
Stranahan, F. E.                     18                         0
Stranahan, F.E.                     160                         0
Talcott, H.D.                         3                         0
van Duzer, A P                        3                         0

remarks                           Petition Dead  Petition and writ dismissed
\
attorney_for_petition
Bergen, Benjamin                              0                            0
Blaney, Edward W.                             0                            0
Carroll Cook                                  0                            0
Cook, Carroll                                 0                            0
Cross & Denson                                0                            0
Hilborn & Hall                                0                            1
Lande, Edward                                 1                            0
McAllister                                    0                            0
McAllister & McAllister                       0                            0
McAllister, Jr. , Ward                        0                            0
Miller, H B M                                 0                            0
Mowry, Lyman                                  4                            4
Naphtaly, Joseph                              0                            0
Perry, G. H. & Ricketts, Alfred               0                            0
Perry, George H                               0                            0
Ricketts, Alfred                              0                            0
Riordan, Thomas D                             0                            1
Schaertzer, Henry C.                          0                            0
Schlesinger, Bert                             0                            0
Smith                                         0                            0
Stonehill & Whaley                            0                            0
Stranahan & Smith                             0                            0
Stranahan, F. E.                              0                            0
Stranahan, F.E.                               2                            3
Talcott, H.D.                                 0                            0
van Duzer, A P                                0                            0

remarks                           Petition dead  Petition to dismiss  Remande
d  \
attorney_for_petition
Bergen, Benjamin                              0                    0
```

```
5
Blaney, Edward W.                            0                    0        3
2
Carroll Cook                                 0                    0
0
Cook, Carroll                                0                    0
1
Cross & Denson                               0                    0
1
Hilborn & Hall                               0                    0
0
Lande, Edward                                0                    0
5
McAllister                                   0                    0
1
McAllister & McAllister                      0                    0
1
McAllister, Jr. , Ward                       0                    0
0
Miller, H B M                                0                    0
3
Mowry, Lyman                                 3                    2        6
8
Naphtaly, Joseph                             0                    0
0
Perry, G. H. & Ricketts, Alfred              0                    0
0
Perry, George H                              0                    0
4
Ricketts, Alfred                             0                    0       12
9
Riordan, Thomas D                            1                    0        6
4
Schaertzer, Henry C.                         0                    0
5
Schlesinger, Bert                            0                    0
1
Smith                                        0                    0
3
Stonehill & Whaley                           1                    0
0
Stranahan & Smith                            0                    0
0
Stranahan, F. E.                             0                    0
1
Stranahan, F.E.                              4                    0       16
7
Talcott, H.D.                                0                    0
1
van Duzer, A P                               0                    0
4

remarks                         Remanded appealed to Circuit Court  \
attorney_for_petition
Bergen, Benjamin                                                  0
Blaney, Edward W.                                                 1
```

```
Carroll Cook                                                    0
Cook, Carroll                                                   1
Cross & Denson                                                  0
Hilborn & Hall                                                  0
Lande, Edward                                                   0
McAllister                                                      0
McAllister & McAllister                                         0
McAllister, Jr. , Ward                                          0
Miller, H B M                                                   0
Mowry, Lyman                                                    6
Naphtaly, Joseph                                                0
Perry, G. H. & Ricketts, Alfred                                 0
Perry, George H                                                 0
Ricketts, Alfred                                                0
Riordan, Thomas D                                               3
Schaertzer, Henry C.                                            0
Schlesinger, Bert                                               0
Smith                                                           0
Stonehill & Whaley                                              0
Stranahan & Smith                                               0
Stranahan, F. E.                                                0
Stranahan, F.E.                                                 3
Talcott, H.D.                                                   0
van Duzer, A P                                                  0

remarks                          Writ and Petition dismissed  Writ not serve
d  \
attorney_for_petition
Bergen, Benjamin                                         0
0
Blaney, Edward W.                                        0
0
Carroll Cook                                             0
0
Cook, Carroll                                            0
0
Cross & Denson                                           0
0
Hilborn & Hall                                           0
0
Lande, Edward                                            0
0
McAllister                                               0
0
McAllister & McAllister                                  0
0
McAllister, Jr. , Ward                                   0
0
Miller, H B M                                            0
0
Mowry, Lyman                                             1
0
Naphtaly, Joseph                                         0
0
Perry, G. H. & Ricketts, Alfred                          0
0
```

```
Perry, George H                                       0
0
Ricketts, Alfred                                      0
0
Riordan, Thomas D                                     0
2
Schaertzer, Henry C.                                  0
0
Schlesinger, Bert                                     0
0
Smith                                                 0
0
Stonehill & Whaley                                    0
0
Stranahan & Smith                                     0
0
Stranahan, F. E.                                      0
0
Stranahan, F.E.                                       0
0
Talcott, H.D.                                         0
0
van Duzer, A P                                        0
0

remarks                          Writ returned  Writ returned non est
attorney_for_petition
Bergen, Benjamin                             0                      0
Blaney, Edward W.                            0                      0
Carroll Cook                                 0                      0
Cook, Carroll                                0                      0
Cross & Denson                               0                      0
Hilborn & Hall                               0                      0
Lande, Edward                                0                      1
McAllister                                   0                      0
McAllister & McAllister                      0                      0
McAllister, Jr. , Ward                       0                      0
Miller, H B M                                0                      0
Mowry, Lyman                                 0                      0
Naphtaly, Joseph                             0                      0
Perry, G. H. & Ricketts, Alfred              0                      0
Perry, George H                              0                      0
Ricketts, Alfred                             0                      1
Riordan, Thomas D                            0                      0
Schaertzer, Henry C.                         0                      1
Schlesinger, Bert                            0                      0
Smith                                        0                      0
Stonehill & Whaley                           0                      0
Stranahan & Smith                            0                      0
Stranahan, F. E.                             0                      0
Stranahan, F.E.                              2                      2
Talcott, H.D.                                0                      0
van Duzer, A P                               0                      0

Attorney Success Rates (for attorneys with at least 3 cases):
- Stranahan, F.E.: 160/346 (46.2%)
- Mowry, Lyman: 189/281 (67.3%)
- Riordan, Thomas D: 170/247 (68.8%)
- Ricketts, Alfred: 103/233 (44.2%)
- Blaney, Edward W.: 32/66 (48.5%)
- Schaertzer, Henry C.: 17/23 (73.9%)
- Stranahan, F. E.: 18/19 (94.7%)
- Lande, Edward: 2/9 (22.2%)
- Schlesinger, Bert: 8/9 (88.9%)
- van Duzer, A P: 3/7 (42.9%)
- Bergen, Benjamin: 2/7 (28.6%)
- Smith: 3/6 (50.0%)
- Cross & Denson: 3/4 (75.0%)
- Perry, George H: 0/4 (0.0%)
- Talcott, H.D.: 3/4 (75.0%)
- Miller, H B M: 1/4 (25.0%)
- Cook, Carroll: 1/3 (33.3%)
- Stonehill & Whaley: 2/3 (66.7%)
```

Reflection

What worked well was asking for options about approaches to cleaning the dataset. I asked it to outline two approaches for cleaning up the mixed data "Age or year or birth" column. While having the conversation about coding options it also chugged out a bonus code for creating flag markers for suspicious data.

I'm frustrated on how ChatGPT used the regular expression library from python, because I still don't understand how it works. On top of that I also had to think hard enough about ChatGPT's code to notice errors while not thinking about what the hell a regex is. Despite these frustrations the code ChatGPT and Claude gave all worked on the first try! I did have to prompt it to include more string words that were attached next to someone's age so they could be added to the "birth_year" column, but the AI still knows wayyy more about regex than I do.

There were also a lot of details about data cleaning that I forgot about (like snake case) so I had to keep repeatedly asking the AI for new things, which thankfully did not lock me out of the high-end model. It was funny seeing the AI churn out "sanity check" codes and snarky comments after a while because I asked for help with geocoding. I geocoded only a sample because of ChatGPT's "be polite" comments but it did in fact map locations of the cases!

Essentially my takeaways are that these AI models can be extremely helpful for coding and researching, but you really have to be critical enough to catch things you may have missed to ask or if there's some error in the code. There's also certain weird limitations that are mostly unknown to be careful of. My last lesson from this experience is Claude and ChatGPT's high reasoning models can potentially give you so much insight and knowledge on coding with python that you might not know and even adapt.

```
In [ ]:
```