In [1]:

```
import pandas as pd
```

In [2]:

```
race_df = pd.read_csv("/content/Population by Race - US, States, Counties.csv")
race_df.drop(columns = ["Not Hispanic", "Hispanic"], inplace = True)
race_df
```

Out[2]:

	IBRC_Geo_ID	Statefips	Countyfips	Description	Year	Total Population	White Alone	Black Alone
0	0	0	0	U.S.	1990	249622814.0	209366661.0	30648345.0
1	0	0	0	U.S.	1991	252980941.0	211606011.0	31290743.0
2	0	0	0	U.S.	1992	256514224.0	213945622.0	31979982.0
3	0	0	0	U.S.	1993	259918588.0	216187073.0	32634735.0
4	0	0	0	U.S.	1994	263125821.0	218304774.0	33258981.0
...	...	...	...	...	...	...	...	...
95795	72000	72	0	Puerto Rico	2015	3473232.0	NaN	NaN
95796	72000	72	0	Puerto Rico	2016	3406672.0	NaN	NaN
95797	72000	72	0	Puerto Rico	2017	3325286.0	NaN	NaN
95798	72000	72	0	Puerto Rico	2018	3193354.0	NaN	NaN
95799	72000	72	0	Puerto Rico	2019	3193694.0	NaN	NaN

95800 rows × 12 columns

In [3]:

```
mylist = ["IBRC_Geo_ID", "Statefips", "Countyfips"]
race_df.drop(columns = mylist, inplace = True)
```

In [4]:

```
raceUSA2009 = race_df.loc[(race_df["Description"] == "U.S.") & (race_df["Year"] == 200
raceUSA2009
```

Out[4]:

	Description	Year	Total Population	White Alone	Black Alone	American Indian or Alaskan Native	Asian Alone	Hawaiian or Pacific Islander Alone	Two Mo Ra
19	U.S.	2009	306771529.0	241034399.0	39902270.0	3636766.0	14833733.0	657983.0	670637

In [5]:

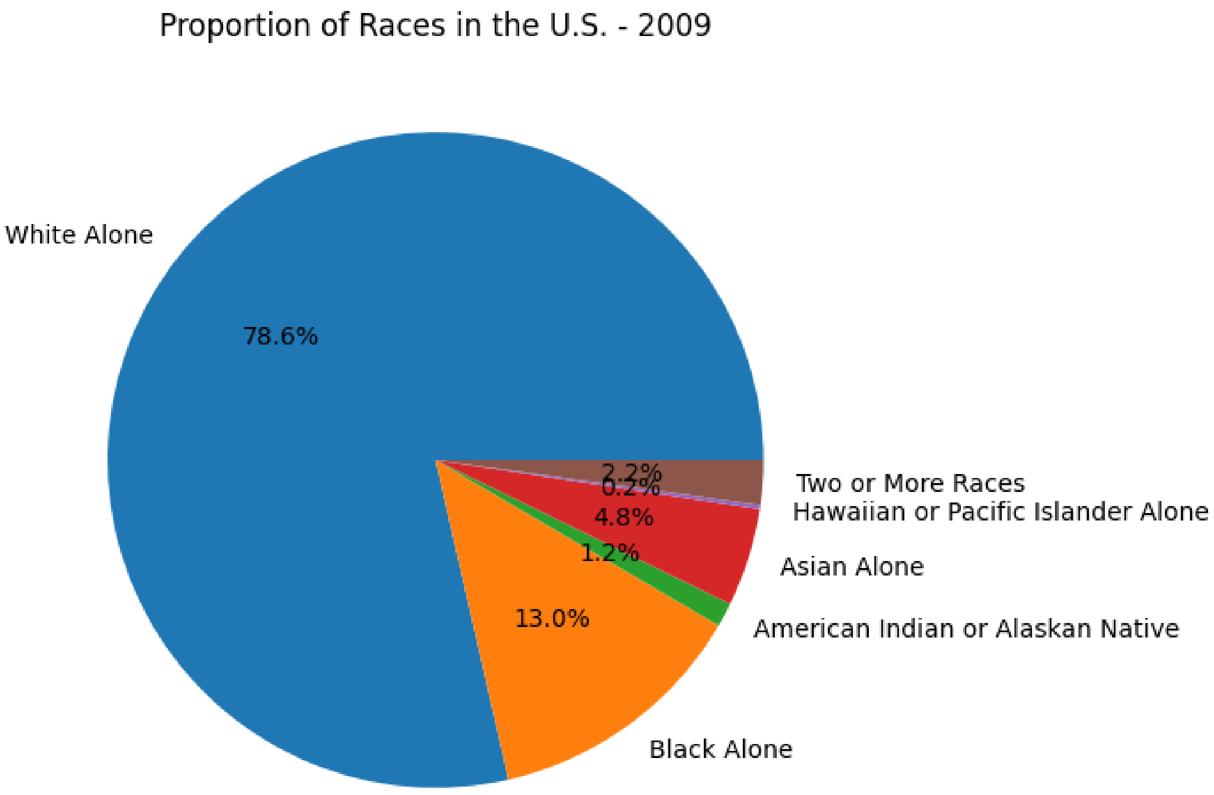
```
raceUSA2019 = race_df.loc[(race_df["Description"] == "U.S.") & (race_df["Year"] == 201
raceUSA2019
```

Out[5]:

	Description	Year	Total Population	White Alone	Black Alone	American Indian or Alaskan Native	Asian Alone	Hawaiian or Pacific Islander Alone	Two Mo Ra
29	U.S.	2019	328239523.0	250522190.0	44075086.0	4188092.0	19504862.0	806937.0	914235

In [6]:

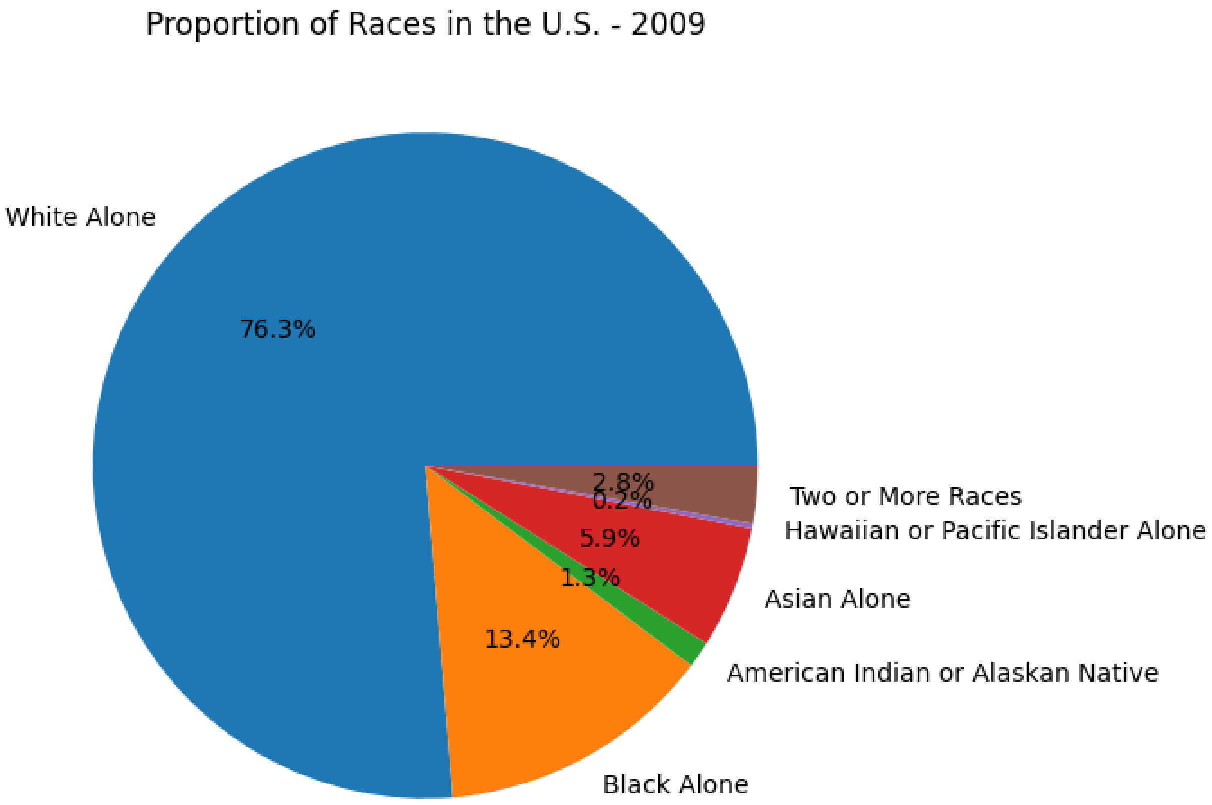
```
data_pie = raceUSA2009[['White Alone', 'Black Alone', 'American Indian or Alaskan Nati
                         'Asian Alone', 'Hawaiian or Pacific Islander Alone', 'Two or More
data_pie = data_pie.reset_index()
data_pie.columns = ['Race', 'Population']


import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))
plt.pie(data_pie['Population'], labels=data_pie['Race'], autopct='%1.1f%%')


plt.title('Proportion of Races in the U.S. - 2009')


plt.show()
```



In [7]:

```
data_pie = raceUSA2019[['White Alone', 'Black Alone', 'American Indian or Alaskan Nati
                         'Asian Alone', 'Hawaiian or Pacific Islander Alone', 'Two or More
data_pie = data_pie.reset_index()
```

```
data_pie.columns = ['Race', 'Population']


plt.figure(figsize=(8, 6))
plt.pie(data_pie['Population'], labels=data_pie['Race'], autopct='%1.1f%%')

plt.title('Proportion of Races in the U.S. - 2009')

plt.show()
```



In [8]:
```
race_df = race_df[~race_df["Description"].str.contains("County")]
race_df = race_df[~race_df["Description"].str.contains(",")]
race_df = race_df[~race_df["Description"].str.contains("U.S.")]
race_df = race_df[~race_df["Description"].str.contains("District of Columbia")]
race_df = race_df[~race_df["Description"].str.contains("Puerto Rico")]
```

In [9]:
```
race_df = race_df.loc[(race_df["Year"] == 2009) | (race_df["Year"] == 2019)]
race_df
```

Out[9]:

	Description	Year	Total Population	White Alone	Black Alone	American Indian or Alaskan Native	Asian Alone	Hawaiian or Pacific Islander Alone	Two or More Races
49	Alabama	2009	4757938.0	3353244.0	1252307.0	31825.0	53747.0	4796.0	62019.0
59	Alabama	2019	4903185.0	3389396.0	1313291.0	34799.0	73715.0	5110.0	86874.0
2089	Alaska	2009	698895.0	477369.0	23979.0	105769.0	37726.0	7191.0	46861.0
2099	Alaska	2019	731545.0	477488.0	27108.0	113953.0	47799.0	10498.0	54699.0
2949	Arizona	2009	6343154.0	5391934.0	274594.0	330731.0	183714.0	15728.0	146453.0
...	...	...	...	...	...	...	...	...	...
91229	West Virginia	2019	1792147.0	1675122.0	64610.0	4585.0	14703.0	530.0	32597.0
92899	Wisconsin	2009	5669264.0	5030324.0	363022.0	58819.0	129070.0	2562.0	85467.0
92909	Wisconsin	2019	5822434.0	5067781.0	390543.0	68628.0	175279.0	3419.0	116784.0
95089	Wyoming	2009	559851.0	525475.0	5402.0	14304.0	4596.0	503.0	9571.0
95099	Wyoming	2019	578759.0	535371.0	7467.0	15778.0	6571.0	596.0	12976.0

100 rows × 9 columns

In [10]:

```
race_df["Description"].unique()
```

Out[10]:

```
array(['Alabama', 'Alaska', 'Arizona', 'Arkansas', 'California',
       'Colorado', 'Connecticut', 'Delaware', 'Florida', 'Georgia',
       'Hawaii', 'Idaho', 'Illinois', 'Indiana', 'Iowa', 'Kansas',
       'Kentucky', 'Louisiana', 'Maine', 'Maryland', 'Massachusetts',
       'Michigan', 'Minnesota', 'Mississippi', 'Missouri', 'Montana',
       'Nebraska', 'Nevada', 'New Hampshire', 'New Jersey', 'New Mexico',
       'New York', 'North Carolina', 'North Dakota', 'Ohio', 'Oklahoma',
       'Oregon', 'Pennsylvania', 'Rhode Island', 'South Carolina',
       'South Dakota', 'Tennessee', 'Texas', 'Utah', 'Vermont',
       'Virginia', 'Washington', 'West Virginia', 'Wisconsin', 'Wyoming'],
      dtype=object)
```

In [11]:

```
race_df2009 = race_df.loc[race_df["Year"] == 2009]

race_df2009.reset_index(inplace = True)

race_df2009["White Prop"] = (race_df2009["White Alone"] / race_df2009["Total Populatic
race_df2009["Black Prop"] = (race_df2009["Black Alone"] / race_df2009["Total Populatic
race_df2009["Native Prop"] = (race_df2009["American Indian or Alaskan Native"] / race_
race_df2009["Asian Prop"] = (race_df2009["Asian Alone"] / race_df2009["Total Populatic
race_df2009["Pacific Prop"] = (race_df2009["Hawaiian or Pacific Islander Alone"] / rac
race_df2009["Mixed Prop"] = (race_df2009["Two or More Races"] / race_df2009["Total Pop

race_df2009.drop(columns = ["Total Population", "White Alone", "Black Alone", "Americā

race_df2009
```

```
<ipython-input-11-bc6164c59bf9>:5: 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/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  race_df2009["White Prop"] = (race_df2009["White Alone"] / race_df2009["Total Popula
tion"]) * 100
<ipython-input-11-bc6164c59bf9>:6: 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/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  race_df2009["Black Prop"] = (race_df2009["Black Alone"] / race_df2009["Total Popula
tion"]) * 100
<ipython-input-11-bc6164c59bf9>:7: 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/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  race_df2009["Native Prop"] = (race_df2009["American Indian or Alaskan Native"] / ra
ce_df2009["Total Population"]) * 100
<ipython-input-11-bc6164c59bf9>:8: 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/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  race_df2009["Asian Prop"] = (race_df2009["Asian Alone"] / race_df2009["Total Popula
tion"]) * 100
<ipython-input-11-bc6164c59bf9>:9: 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/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  race_df2009["Pacific Prop"] = (race_df2009["Hawaiian or Pacific Islander Alone"] /
race_df2009["Total Population"]) * 100
<ipython-input-11-bc6164c59bf9>:10: 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/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  race_df2009["Mixed Prop"] = (race_df2009["Two or More Races"] / race_df2009["Total
Population"]) * 100
<ipython-input-11-bc6164c59bf9>:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  race_df2009.drop(columns = ["Total Population", "White Alone", "Black Alone", "Amer
ican Indian or Alaskan Native", "Asian Alone", "Hawaiian or Pacific Islander Alone",
"Two or More Races", "index", "Year"],  inplace = True)
```

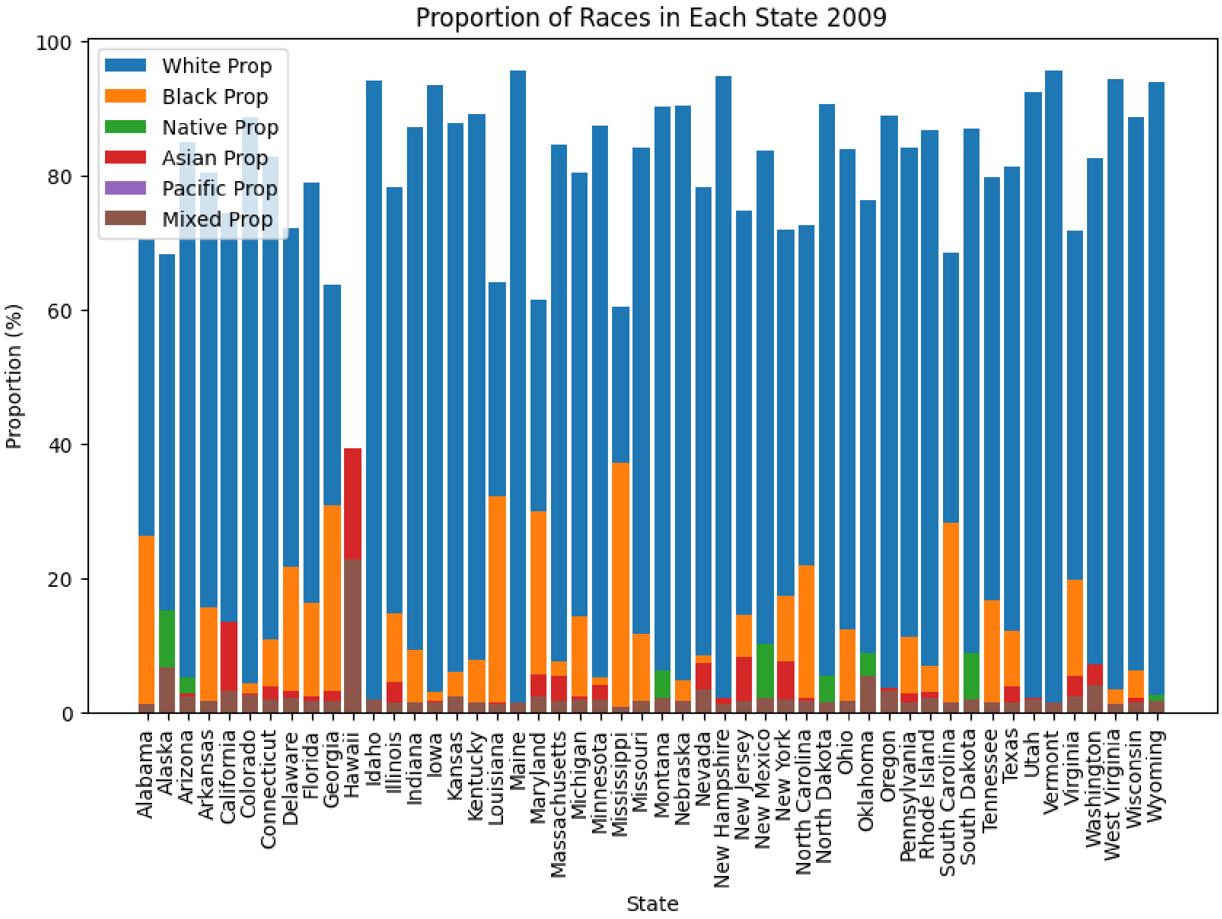
Out[11]:

	Description	White Prop	Black Prop	Native Prop	Asian Prop	Pacific Prop	Mixed Prop
0	Alabama	70.476833	26.320372	0.668882	1.129628	0.100800	1.303485
1	Alaska	68.303393	3.430987	15.133747	5.397950	1.028910	6.705013
2	Arizona	85.003990	4.328982	5.213983	2.896256	0.247952	2.308836
3	Arkansas	80.476574	15.556004	0.868739	1.251707	0.211817	1.635159
4	California	74.480970	6.681507	1.613769	13.396662	0.479925	3.347167
5	Colorado	88.732642	4.236419	1.499177	2.832652	0.166184	2.532926
6	Connecticut	82.847555	10.860864	0.451737	3.841084	0.095289	1.903472
7	Delaware	72.191695	21.721822	0.633936	3.227771	0.075359	2.149417
8	Florida	78.919605	16.317011	0.469435	2.484275	0.099568	1.710106
9	Georgia	63.700822	30.790619	0.482411	3.267665	0.103577	1.654906
10	Hawaii	25.774977	1.707486	0.381520	39.270834	10.138210	22.726972
11	Idaho	94.259730	0.714341	1.614473	1.265151	0.173696	1.972609
12	Illinois	78.313971	14.857373	0.549529	4.638191	0.058648	1.582289
13	Indiana	87.142914	9.248908	0.367221	1.586079	0.055161	1.599718
14	Iowa	93.408257	2.943581	0.438067	1.719988	0.072835	1.417271
15	Kansas	87.852843	6.045390	1.140783	2.397850	0.094115	2.469019
16	Kentucky	89.187329	7.851012	0.270693	1.130117	0.074425	1.486423
17	Louisiana	64.162107	32.231110	0.716018	1.551658	0.054791	1.284317
18	Maine	95.669643	1.214660	0.659301	1.014749	0.028956	1.412691
19	Maryland	61.619161	29.935512	0.519529	5.541422	0.090448	2.293928
20	Massachusetts	84.649886	7.618709	0.442202	5.383136	0.089787	1.816278
21	Michigan	80.558771	14.308226	0.677103	2.434376	0.035863	1.985661
22	Minnesota	87.414932	5.219095	1.256059	4.025806	0.057354	2.026754
23	Mississippi	60.461123	37.076336	0.553168	0.880872	0.057084	0.971416
24	Missouri	84.265691	11.665052	0.504623	1.635222	0.111322	1.818091
25	Montana	90.197382	0.442488	6.373186	0.653366	0.079981	2.253598
26	Nebraska	90.470645	4.671749	1.261224	1.778965	0.111713	1.705704
27	Nevada	78.371603	8.494915	1.577031	7.373136	0.699342	3.483973
28	New Hampshire	94.905638	1.223917	0.274979	2.170652	0.043158	1.381656
29	New Jersey	74.734187	14.551392	0.537325	8.337474	0.086722	1.752901
30	New Mexico	83.702540	2.354279	10.141388	1.487773	0.152494	2.161526
31	New York	72.006099	17.396885	0.912417	7.520335	0.119780	2.044485
32	North Carolina	72.558454	21.852146	1.519266	2.207064	0.101603	1.761467
33	North Dakota	90.709027	1.228029	5.470188	0.981250	0.049025	1.562481

	Description	White Prop	Black Prop	Native Prop	Asian Prop	Pacific Prop	Mixed Prop
34	Ohio	83.925972	12.325820	0.252097	1.668269	0.044991	1.782851
35	Oklahoma	76.287507	7.570452	8.884885	1.759267	0.133232	5.364657
36	Oregon	89.045712	1.924539	1.709500	3.737646	0.374757	3.207845
37	Pennsylvania	84.149455	11.218717	0.303540	2.748322	0.053936	1.526030
38	Rhode Island	86.697050	7.057873	0.843357	2.989809	0.152043	2.259867
39	South Carolina	68.449752	28.223641	0.521213	1.299753	0.083161	1.422480
40	South Dakota	86.958951	1.251941	8.895296	0.943540	0.066042	1.884230
41	Tennessee	79.791910	16.797333	0.409117	1.450725	0.084507	1.466409
42	Texas	81.292949	12.178587	0.960484	3.899933	0.122552	1.545495
43	Utah	92.320137	1.203193	1.452695	2.067437	0.925454	2.031085
44	Vermont	95.695700	1.045746	0.380751	1.277654	0.027848	1.572300
45	Virginia	71.793076	19.751330	0.503259	5.480362	0.100770	2.371202
46	Washington	82.575960	3.751328	1.807654	7.212603	0.633243	4.019212
47	West Virginia	94.315163	3.432074	0.219291	0.681414	0.024408	1.327651
48	Wisconsin	88.729754	6.403336	1.037507	2.276662	0.045191	1.507550
49	Wyoming	93.859795	0.964900	2.554966	0.820933	0.089845	1.709562

In [12]:

```python
import plotly.express as px

df_race_melted = race_df2009.melt(id_vars='Description', var_name='Race', value_name='

plt.figure(figsize=(10, 6))

# Creating a bar plot for each race
for race in df_race_melted['Race'].unique():
    race_data = df_race_melted[df_race_melted['Race'] == race]
    plt.bar(race_data['Description'], race_data['Proportion'], label=race)

# Adding labels and title
plt.xlabel('State')
plt.ylabel('Proportion (%)')
plt.title('Proportion of Races in Each State 2009')
plt.legend()


plt.xticks(rotation=90)
# Display the plot
plt.show()






# fig = px.bar(
#     df_race_melted,
#     x='Description',
#     y='Proportion',
#     color='Race',
#     title='Proportion of Races in Each State 2009',
#     labels={'Description': 'State', 'Proportion': 'Proportion (%)'}
# )


# fig.show()
```



In [13]:

```
race_df2019 = race_df.loc[race_df["Year"] == 2019]

race_df2019.reset_index(inplace = True)
race_df2019["White Prop"] = (race_df2019["White Alone"] / race_df2019["Total Populatic
race_df2019["Black Prop"] = (race_df2019["Black Alone"] / race_df2019["Total Populatic
race_df2019["Native Prop"] = (race_df2019["American Indian or Alaskan Native"] / race_
race_df2019["Asian Prop"] = (race_df2019["Asian Alone"] / race_df2019["Total Populatic
race_df2019["Pacific Prop"] = (race_df2019["Hawaiian or Pacific Islander Alone"] / rac
race_df2019["Mixed Prop"] = (race_df2019["Two or More Races"] / race_df2019["Total Pop

race_df2019.drop(columns = ["Total Population", "White Alone", "Black Alone", "America

race_df2019
```

```
<ipython-input-13-80f6c32d6c03>:4: 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/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  race_df2019["White Prop"] = (race_df2019["White Alone"] / race_df2019["Total Popula
tion"]) * 100
<ipython-input-13-80f6c32d6c03>:5: 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/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  race_df2019["Black Prop"] = (race_df2019["Black Alone"] / race_df2019["Total Popula
tion"]) * 100
<ipython-input-13-80f6c32d6c03>:6: 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/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  race_df2019["Native Prop"] = (race_df2019["American Indian or Alaskan Native"] / ra
ce_df2019["Total Population"]) * 100
<ipython-input-13-80f6c32d6c03>:7: 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/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  race_df2019["Asian Prop"] = (race_df2019["Asian Alone"] / race_df2019["Total Popula
tion"]) * 100
<ipython-input-13-80f6c32d6c03>:8: 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/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  race_df2019["Pacific Prop"] = (race_df2019["Hawaiian or Pacific Islander Alone"] /
race_df2019["Total Population"]) * 100
<ipython-input-13-80f6c32d6c03>:9: 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/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  race_df2019["Mixed Prop"] = (race_df2019["Two or More Races"] / race_df2019["Total
Population"]) * 100
<ipython-input-13-80f6c32d6c03>:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  race_df2019.drop(columns = ["Total Population", "White Alone", "Black Alone", "Amer
ican Indian or Alaskan Native", "Asian Alone", "Hawaiian or Pacific Islander Alone",
"Two or More Races", "index", "Year"],  inplace = True)
```

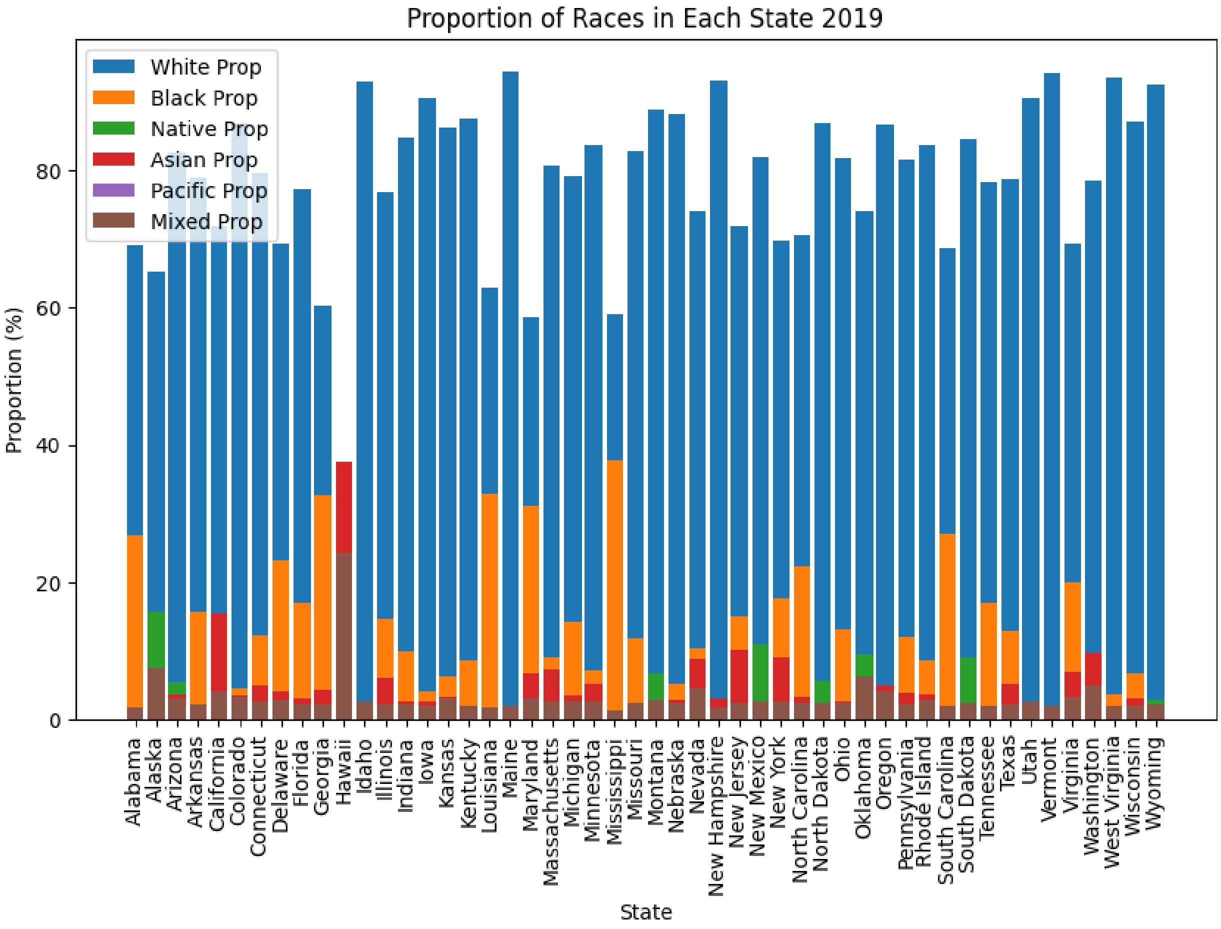
Out[13]:

	Description	White Prop	Black Prop	Native Prop	Asian Prop	Pacific Prop	Mixed Prop
0	Alabama	69.126415	26.784447	0.709722	1.503411	0.104218	1.771787
1	Alaska	65.271173	3.705582	15.577032	6.533979	1.435045	7.477189
2	Arizona	82.616785	5.179443	5.301800	3.691901	0.277096	2.932976
3	Arkansas	79.039527	15.675239	1.017859	1.670188	0.390549	2.206638
4	California	71.939104	6.460677	1.644711	15.465961	0.505679	3.983868
5	Colorado	86.939426	4.592657	1.614625	3.519245	0.196727	3.137320
6	Connecticut	79.700204	12.189341	0.576868	4.952729	0.108799	2.472059
7	Delaware	69.223652	23.162080	0.674085	4.093702	0.108753	2.737727
8	Florida	77.277192	16.917588	0.507269	2.955744	0.114728	2.227479
9	Georgia	60.198270	32.570493	0.527510	4.369855	0.116893	2.216979
10	Hawaii	25.519044	2.186497	0.393397	37.587437	10.138558	24.175067
11	Idaho	93.022078	0.914684	1.741403	1.550643	0.220697	2.550495
12	Illinois	76.760199	14.619177	0.595810	5.900415	0.066170	2.058228
13	Indiana	84.809956	9.946141	0.420961	2.595727	0.067422	2.159793
14	Iowa	90.618592	4.060290	0.540717	2.666153	0.152263	1.961985
15	Kansas	86.288193	6.134766	1.203543	3.187504	0.126557	3.059437
16	Kentucky	87.515872	8.471502	0.300022	1.600475	0.093830	2.018299
17	Louisiana	62.787101	32.798463	0.787215	1.811050	0.061392	1.754778
18	Maine	94.429004	1.688052	0.726597	1.295927	0.033849	1.826572
19	Maryland	58.545011	31.074172	0.609361	6.720551	0.110906	2.940000
20	Massachusetts	80.582700	9.022876	0.497787	7.218234	0.107508	2.570895
21	Michigan	79.219789	14.097428	0.739622	3.368267	0.041915	2.532979
22	Minnesota	83.763107	7.013862	1.373831	5.186633	0.074721	2.587846
23	Mississippi	59.072345	37.785709	0.628497	1.109891	0.060682	1.342876
24	Missouri	82.871652	11.822200	0.583942	2.172115	0.160605	2.389486
25	Montana	88.906302	0.597786	6.652083	0.920023	0.086080	2.837727
26	Nebraska	88.095634	5.208984	1.513900	2.749523	0.121433	2.310526
27	Nevada	73.942748	10.269058	1.686473	8.706247	0.798174	4.597300
28	New Hampshire	93.115155	1.792366	0.302564	2.963865	0.049422	1.776628
29	New Jersey	71.928252	15.057739	0.624047	9.984137	0.115287	2.290539
30	New Mexico	81.869146	2.612135	10.959120	1.790799	0.159336	2.609464
31	New York	69.636485	17.586102	0.975390	9.012227	0.138967	2.650831
32	North Carolina	70.551370	22.221056	1.578906	3.187417	0.125190	2.336061
33	North Dakota	86.942401	3.409696	5.572381	1.687527	0.080702	2.307293

	Description	White Prop	Black Prop	Native Prop	Asian Prop	Pacific Prop	Mixed Prop
34	Ohio	81.723452	13.051219	0.290955	2.493588	0.060355	2.380431
35	Oklahoma	73.979668	7.779157	9.378487	2.381948	0.216858	6.263882
36	Oregon	86.661070	2.222519	1.828350	4.854238	0.456287	3.977536
37	Pennsylvania	81.604991	12.030099	0.395931	3.764079	0.078066	2.126834
38	Rhode Island	83.589824	8.505882	1.081123	3.732061	0.200593	2.890516
39	South Carolina	68.600586	26.958518	0.548409	1.827854	0.098374	1.966258
40	South Dakota	84.560266	2.298287	9.043485	1.548506	0.088961	2.460496
41	Tennessee	78.404870	17.051272	0.478257	1.963283	0.096322	2.005997
42	Texas	78.652999	12.895697	1.017048	5.209257	0.149028	2.075971
43	Utah	90.585466	1.482771	1.550862	2.673990	1.059870	2.647040
44	Vermont	94.248777	1.406115	0.391193	1.917341	0.039584	1.996990
45	Virginia	69.388259	19.880584	0.545977	6.908894	0.118235	3.158050
46	Washington	78.500066	4.358879	1.927788	9.560029	0.792513	4.860725
47	West Virginia	93.470123	3.605173	0.255838	0.820413	0.029573	1.818880
48	Wisconsin	87.038874	6.707556	1.178682	3.010408	0.058721	2.005759
49	Wyoming	92.503270	1.290174	2.726178	1.135360	0.102979	2.242039

In [14]:

```python
df_race_melted19 = race_df2019.melt(id_vars='Description', var_name='Race', value_name


plt.figure(figsize=(10, 6))

# Creating a bar plot for each race
for race in df_race_melted19['Race'].unique():
    race_data = df_race_melted19[df_race_melted19['Race'] == race]
    plt.bar(race_data['Description'], race_data['Proportion'], label=race)

# Adding labels and title
plt.xlabel('State')
plt.ylabel('Proportion (%)')
plt.title('Proportion of Races in Each State 2019')
plt.legend()


plt.xticks(rotation=90)
# Display the plot
plt.show()
```



In [15]:

```
race_df2009.set_index('Description', inplace=True)
race_df2019.set_index('Description', inplace=True)

combined_race_df = race_df2009.merge(race_df2019, left_index=True, right_index=True, s
combined_race_df
```

Out[15]:

	White Prop 2009	Black Prop 2009	Native Prop 2009	Asian Prop 2009	Pacific Prop 2009	Mixed Prop 2009	White Prop 2019	Black Prop 2019
Description								
Alabama	70.476833	26.320372	0.668882	1.129628	0.100800	1.303485	69.126415	26.784447
Alaska	68.303393	3.430987	15.133747	5.397950	1.028910	6.705013	65.271173	3.705582
Arizona	85.003990	4.328982	5.213983	2.896256	0.247952	2.308836	82.616785	5.179443
Arkansas	80.476574	15.556004	0.868739	1.251707	0.211817	1.635159	79.039527	15.675239
California	74.480970	6.681507	1.613769	13.396662	0.479925	3.347167	71.939104	6.460677
Colorado	88.732642	4.236419	1.499177	2.832652	0.166184	2.532926	86.939426	4.592657
Connecticut	82.847555	10.860864	0.451737	3.841084	0.095289	1.903472	79.700204	12.189341
Delaware	72.191695	21.721822	0.633936	3.227771	0.075359	2.149417	69.223652	23.162080
Florida	78.919605	16.317011	0.469435	2.484275	0.099568	1.710106	77.277192	16.917588
Georgia	63.700822	30.790619	0.482411	3.267665	0.103577	1.654906	60.198270	32.570493
Hawaii	25.774977	1.707486	0.381520	39.270834	10.138210	22.726972	25.519044	2.186497
Idaho	94.259730	0.714341	1.614473	1.265151	0.173696	1.972609	93.022078	0.914684
Illinois	78.313971	14.857373	0.549529	4.638191	0.058648	1.582289	76.760199	14.619177
Indiana	87.142914	9.248908	0.367221	1.586079	0.055161	1.599718	84.809956	9.946141
Iowa	93.408257	2.943581	0.438067	1.719988	0.072835	1.417271	90.618592	4.060290
Kansas	87.852843	6.045390	1.140783	2.397850	0.094115	2.469019	86.288193	6.134766
Kentucky	89.187329	7.851012	0.270693	1.130117	0.074425	1.486423	87.515872	8.471502
Louisiana	64.162107	32.231110	0.716018	1.551658	0.054791	1.284317	62.787101	32.798463
Maine	95.669643	1.214660	0.659301	1.014749	0.028956	1.412691	94.429004	1.688052
Maryland	61.619161	29.935512	0.519529	5.541422	0.090448	2.293928	58.545011	31.074172
Massachusetts	84.649886	7.618709	0.442202	5.383136	0.089787	1.816278	80.582700	9.022876
Michigan	80.558771	14.308226	0.677103	2.434376	0.035863	1.985661	79.219789	14.097428
Minnesota	87.414932	5.219095	1.256059	4.025806	0.057354	2.026754	83.763107	7.013862
Mississippi	60.461123	37.076336	0.553168	0.880872	0.057084	0.971416	59.072345	37.785709
Missouri	84.265691	11.665052	0.504623	1.635222	0.111322	1.818091	82.871652	11.822200
Montana	90.197382	0.442488	6.373186	0.653366	0.079981	2.253598	88.906302	0.597786
Nebraska	90.470645	4.671749	1.261224	1.778965	0.111713	1.705704	88.095634	5.208984
Nevada	78.371603	8.494915	1.577031	7.373136	0.699342	3.483973	73.942748	10.269058
New Hampshire	94.905638	1.223917	0.274979	2.170652	0.043158	1.381656	93.115155	1.792366
New Jersey	74.734187	14.551392	0.537325	8.337474	0.086722	1.752901	71.928252	15.057739
New Mexico	83.702540	2.354279	10.141388	1.487773	0.152494	2.161526	81.869146	2.612135

	White Prop 2009	Black Prop 2009	Native Prop 2009	Asian Prop 2009	Pacific Prop 2009	Mixed Prop 2009	White Prop 2019	Black Prop 2019
Description								
New York	72.006099	17.396885	0.912417	7.520335	0.119780	2.044485	69.636485	17.586102
North Carolina	72.558454	21.852146	1.519266	2.207064	0.101603	1.761467	70.551370	22.221056
North Dakota	90.709027	1.228029	5.470188	0.981250	0.049025	1.562481	86.942401	3.409696
Ohio	83.925972	12.325820	0.252097	1.668269	0.044991	1.782851	81.723452	13.051219
Oklahoma	76.287507	7.570452	8.884885	1.759267	0.133232	5.364657	73.979668	7.779157
Oregon	89.045712	1.924539	1.709500	3.737646	0.374757	3.207845	86.661070	2.222519
Pennsylvania	84.149455	11.218717	0.303540	2.748322	0.053936	1.526030	81.604991	12.030099
Rhode Island	86.697050	7.057873	0.843357	2.989809	0.152043	2.259867	83.589824	8.505882
South Carolina	68.449752	28.223641	0.521213	1.299753	0.083161	1.422480	68.600586	26.958518
South Dakota	86.958951	1.251941	8.895296	0.943540	0.066042	1.884230	84.560266	2.298287
Tennessee	79.791910	16.797333	0.409117	1.450725	0.084507	1.466409	78.404870	17.051272
Texas	81.292949	12.178587	0.960484	3.899933	0.122552	1.545495	78.652999	12.895697
Utah	92.320137	1.203193	1.452695	2.067437	0.925454	2.031085	90.585466	1.482771
Vermont	95.695700	1.045746	0.380751	1.277654	0.027848	1.572300	94.248777	1.406115
Virginia	71.793076	19.751330	0.503259	5.480362	0.100770	2.371202	69.388259	19.880584
Washington	82.575960	3.751328	1.807654	7.212603	0.633243	4.019212	78.500066	4.358879
West Virginia	94.315163	3.432074	0.219291	0.681414	0.024408	1.327651	93.470123	3.605173
Wisconsin	88.729754	6.403336	1.037507	2.276662	0.045191	1.507550	87.038874	6.707556
Wyoming	93.859795	0.964900	2.554966	0.820933	0.089845	1.709562	92.503270	1.290174

In [16]:

```
#CLUSTERING TIME


#WHITE

from sklearn.cluster import KMeans
X_train = combined_race_df[["White Prop 2009", "White Prop 2019"]]

for n_clusters in range(3, 9):
    model = KMeans(n_clusters=n_clusters)
    model.fit(X_train)

    centroids = model.cluster_centers_
    clusters = model.labels_

    cluster_names = {i: f"cluster {i+1}" for i in range(n_clusters)}
    clusters_mapped = pd.Series(clusters).map(cluster_names)
    combined_race_df['Cluster'] = pd.Categorical(clusters_mapped)
    combined_race_df['Description'] = combined_race_df.index

    fig = px.scatter(
        combined_race_df,
        x="White Prop 2009",
        y="White Prop 2019",
        color='Cluster',
        hover_name='Description',
        title=f"KMeans Clustering with {n_clusters} Clusters (White Population Proport
    )

    fig.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value
of `n_init` explicitly to suppress the warning
  warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

In [17]:

```
#BLACK

from sklearn.cluster import KMeans
X_train = combined_race_df[["Black Prop 2009", "Black Prop 2019"]]

for n_clusters in range(3, 10):
    model = KMeans(n_clusters=n_clusters)
    model.fit(X_train)

    centroids = model.cluster_centers_
    clusters = model.labels_

    cluster_names = {i: f"cluster {i+1}" for i in range(n_clusters)}
    clusters_mapped = pd.Series(clusters).map(cluster_names)

    combined_race_df['Cluster'] = pd.Categorical(clusters_mapped)
    combined_race_df['Description'] = combined_race_df.index

    fig = px.scatter(
        combined_race_df,
        x="Black Prop 2009",
        y="Black Prop 2019",
        color='Cluster',
        hover_name='Description',
        title=f"KMeans Clustering with {n_clusters} Clusters (Black Population Proport
    )
    fig.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

In [18]:

```
# NATIVE PROPORTIONS

X_train = combined_race_df[["Native Prop 2009", "Native Prop 2019"]]

for n_clusters in range(3, 10):
    model = KMeans(n_clusters=n_clusters)
    model.fit(X_train)

    centroids = model.cluster_centers_
    clusters = model.labels_

    cluster_names = {i: f"cluster {i+1}" for i in range(n_clusters)}
    clusters_mapped = pd.Series(clusters).map(cluster_names)

    combined_race_df['Cluster'] = pd.Categorical(clusters_mapped)
    combined_race_df['Description'] = combined_race_df.index

    fig = px.scatter(
        combined_race_df,
        x="Native Prop 2009",
        y="Native Prop 2019",
        color='Cluster',
        hover_name='Description',
        title=f"KMeans Clustering with {n_clusters} Clusters (Native Population Propor
    )
    fig.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

In [19]:

```
# ASIAN PROPORTIONS

X_train = combined_race_df[["Asian Prop 2009", "Asian Prop 2019"]]

for n_clusters in range(3, 10):
    model = KMeans(n_clusters=n_clusters)
    model.fit(X_train)

    centroids = model.cluster_centers_
    clusters = model.labels_

    cluster_names = {i: f"cluster {i+1}" for i in range(n_clusters)}
    clusters_mapped = pd.Series(clusters).map(cluster_names)

    combined_race_df['Cluster'] = pd.Categorical(clusters_mapped)
    combined_race_df['Description'] = combined_race_df.index

    fig = px.scatter(
        combined_race_df,
        x="Asian Prop 2009",
        y="Asian Prop 2019",
        color='Cluster',
        hover_name='Description',
        title=f"KMeans Clustering with {n_clusters} Clusters (Asian Population Proport
    )
    fig.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

In [20]:

```
# Pacific Proportions

X_train = combined_race_df[["Pacific Prop 2009", "Pacific Prop 2019"]]

for n_clusters in range(2, 10):
    model = KMeans(n_clusters=n_clusters)
    model.fit(X_train)

    centroids = model.cluster_centers_
    clusters = model.labels_

    cluster_names = {i: f"cluster {i+1}" for i in range(n_clusters)}
    clusters_mapped = pd.Series(clusters).map(cluster_names)

    combined_race_df['Cluster'] = pd.Categorical(clusters_mapped)
    combined_race_df['Description'] = combined_race_df.index

    fig = px.scatter(
        combined_race_df,
        x="Pacific Prop 2009",
        y="Pacific Prop 2019",
        color='Cluster',
        hover_name='Description',
        title=f"KMeans Clustering with {n_clusters} Clusters (Pacific Population Propc
    )
    fig.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

In [21]:

```
# Mixed Proportions

X_train = combined_race_df[["Mixed Prop 2009", "Mixed Prop 2019"]]

for n_clusters in range(2, 10):
    model = KMeans(n_clusters=n_clusters)
    model.fit(X_train)

    centroids = model.cluster_centers_
    clusters = model.labels_

    cluster_names = {i: f"cluster {i+1}" for i in range(n_clusters)}
    clusters_mapped = pd.Series(clusters).map(cluster_names)

    combined_race_df['Cluster'] = pd.Categorical(clusters_mapped)
    combined_race_df['Description'] = combined_race_df.index

    fig = px.scatter(
        combined_race_df,
        x="Mixed Prop 2009",
        y="Mixed Prop 2019",
        color='Cluster',
        hover_name='Description',
        title=f"KMeans Clustering with {n_clusters} Clusters (Mixed Population Proport
    )
    fig.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
g:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
```

ethnicity population - https://www.statsamerica.org/downloads/default.aspx