

```
In [224]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import style
```

```
In [225]: df0 = pd.read_csv('states0.csv')
df1 = pd.read_csv('states1.csv')
df2 = pd.read_csv('states2.csv')
df3 = pd.read_csv('states3.csv')
df4 = pd.read_csv('states4.csv')
df5 = pd.read_csv('states5.csv')
df6 = pd.read_csv('states6.csv')
df7 = pd.read_csv('states7.csv')
df8 = pd.read_csv('states8.csv')
df9 = pd.read_csv('states9.csv')
```

```
In [226]: df0
```

Out[226]:

	Unnamed: 0	State	TotalPop		Hispanic		White
0	0	Alabama	4830620	3.7516156462584975%	61.878656462585%	31.25297619047	
1	1	Alaska	733375	5.909580838323351%	60.910179640718574%	2.8485029940119	
2	2	Arizona	6641928	29.565921052631502%	57.120000000000026%	3.8509868421052	
3	3	Arkansas	2958208	6.215474452554738%	71.13781021897813%	18.968759124087	
4	4	California	38421464	37.291874687968054%	40.21578881677474%	5.677396405391	
5	5	Colorado	5278906	20.78438003220608%	69.89557165861504%	3.546376811594	

```
In [227]: combined_df = pd.concat([df1,df2,df3,df4,df5,df6,df7,df8,df9] , ignore_index=True)
```

```
In [228]: combined_df.to_csv('combined_df' , index = False)
```

```
In [229]: df = pd.read_csv('combined_df')
df
```

Out[229]:

Unnamed: 0	State	TotalPop	Hispanic	White
0	Colorado	5278906	20.78438003220608%	69.89557165861504%
1	Connecticut	3593222	15.604830917874388%	67.6770531400966%
2	Delaware	926454	8.82476635514019%	64.63271028037383%
3	District of Columbia	647484	9.165921787709499%	33.103910614525134%
4	Florida	19645772	21.3385426653884%	59.08374880153398%
5	Georgia	10006693	8.418242207460397%	54.28630556974962%
6	Georgia	10006693	8.418242207460397%	54.28630556974962%
7	Hawaii	1406299	9.186708860759486%	25.032278481012657%
8	Idaho	1616547	11.505369127516781%	83.1362416107383%

```
In [230]: df = df.iloc[:,1:]
df
```

[illegible]

In [231]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54 entries, 0 to 53
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   State       54 non-null    object
1   TotalPop    54 non-null    int64
2   Hispanic    54 non-null    object
3   White       54 non-null    object
4   Black       54 non-null    object
5   Native      54 non-null    object
6   Asian       54 non-null    object
7   Pacific     50 non-null    object
8   Income      54 non-null    object
9   GenderPop   54 non-null    object
dtypes: int64(1), object(9)
memory usage: 4.3+ KB
```

In [232]: `df.isnull()`

Out[232]:

	State	TotalPop	Hispanic	White	Black	Native	Asian	Pacific	Income	GenderPop
0	False	False	False	False	False	False	False	True	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	True	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False	False	False
10	False	False	False	False	False	False	False	False	False	False

In [233]: `df = df.dropna()`

In [234]: `df = df.drop_duplicates()`

In [235]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 42 entries, 1 to 53
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   State       42 non-null    object
 1   TotalPop    42 non-null    int64
 2   Hispanic    42 non-null    object
 3   White       42 non-null    object
 4   Black       42 non-null    object
 5   Native      42 non-null    object
 6   Asian       42 non-null    object
 7   Pacific     42 non-null    object
 8   Income      42 non-null    object
 9   GenderPop   42 non-null    object
dtypes: int64(1), object(9)
memory usage: 3.6+ KB
```

In [236]: df = df.reset\_index(drop = True)

In [237]: df[['Male', 'Female']] = df['GenderPop'].str.split('\_', expand = True)

In [238]: df['Male'] = df['Male'].str.replace('M', '')

In [239]: df['Female'] = df['Female'].str.replace('F', '')

In [240]: df['Male'] = df['Male'].astype(int)

In [241]: df["Female"][12] = 2900655

C:\Users\mosai\AppData\Local\Temp\ipykernel\_7092\2094601370.py:1: 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/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

```
df["Female"][12] = 2900655
```

```
In [242]: df["Female"][14] = 2900655
```

C:\Users\mosai\AppData\Local\Temp\ipykernel\_7092\3515725951.py:1: 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/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

```
df["Female"][14] = 2900655
```

```
In [243]: df["Female"][18] = 2900655
```

C:\Users\mosai\AppData\Local\Temp\ipykernel\_7092\3063036996.py:1: 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/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

```
df["Female"][18] = 2900655
```

```
In [244]: df['Female']
```

```
Out[244]: 0      1841615
1       340810
2     10045763
3       5123362
4        696428
5       806083
6     6556862
7     3333382
8     1558931
9     2233145
10    2364097
11      679019
12    2900655
13    3455936
14    2900655
15    2727005
16    1536358
17    3081445
18    2900655
19    2900655
```

```
In [245]: df['Female'] = df['Female'].astype(int)
```

```
In [246]: df['Female'].mean()
```

```
Out[246]: 3107844.8571428573
```

In [247]: `df.head()`

Out[247]:

	State	TotalPop	Hispanic	White	Black
0	Connecticut	3593222	15.604830917874388%	67.6770531400966%	10.34806763285027%
1	District of Columbia	647484	9.165921787709499%	33.103910614525134%	51.77653631284915%
2	Florida	19645772	21.3385426653884%	59.08374880153398%	15.165675934803444%
3	Georgia	10006693	8.418242207460397%	54.28630556974962%	32.08829841594277%
4	Hawaii	1406299	9.186708860759486%	25.032278481012657%	2.052848101265823%

In [248]: `df['Hispanic'] = df['Hispanic'].str.replace("%", "").astype(float)`

In [249]: `df['White'] = df['White'].str.replace("%", "").astype(float)`

In [250]: `df['Black'] = df['Black'].str.replace("%", "").astype(float)`

In [251]: `df['Native'] = df['Native'].str.replace("%", "").astype(float)`

In [252]: `df['Asian'] = df['Asian'].str.replace("%", "").astype(float)`

In [253]: `df['Pacific'] = df['Pacific'].str.replace("%", "").astype(float)`

In [254]: df.info()

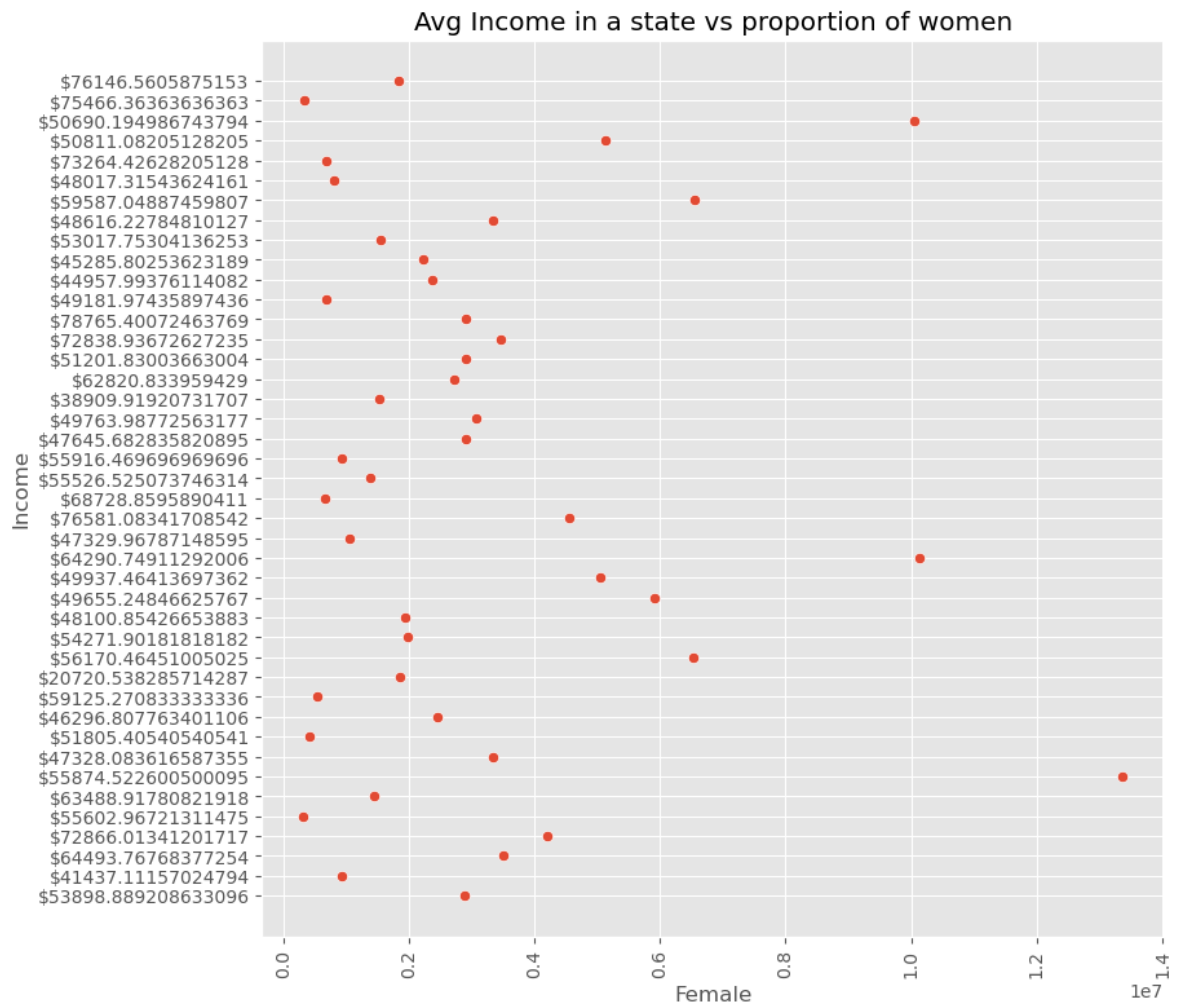
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42 entries, 0 to 41
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype  
---  -
0   State       42 non-null    object  
1   TotalPop    42 non-null    int64   
2   Hispanic    42 non-null    float64  
3   White       42 non-null    float64  
4   Black       42 non-null    float64  
5   Native      42 non-null    float64  
6   Asian       42 non-null    float64  
7   Pacific     42 non-null    float64  
8   Income      42 non-null    object  
9   GenderPop   42 non-null    object  
10  Male        42 non-null    int32   
11  Female      42 non-null    int32   
dtypes: float64(6), int32(2), int64(1), object(3)
memory usage: 3.7+ KB
```

In [ ]:

## Graph-1

The first visualization your boss wants you to make is a scatterplot that shows average income in a state vs proportion of women in that state.

```
In [255]: style.use('ggplot')
plt.figure(figsize = (9,9))
sns.scatterplot(x=df['Female'],y=df['Income'])
plt.xticks(rotation = 90)
plt.title("Avg Income in a state vs proportion of women")
plt.show()
```



```
In [256]: df.head()
```

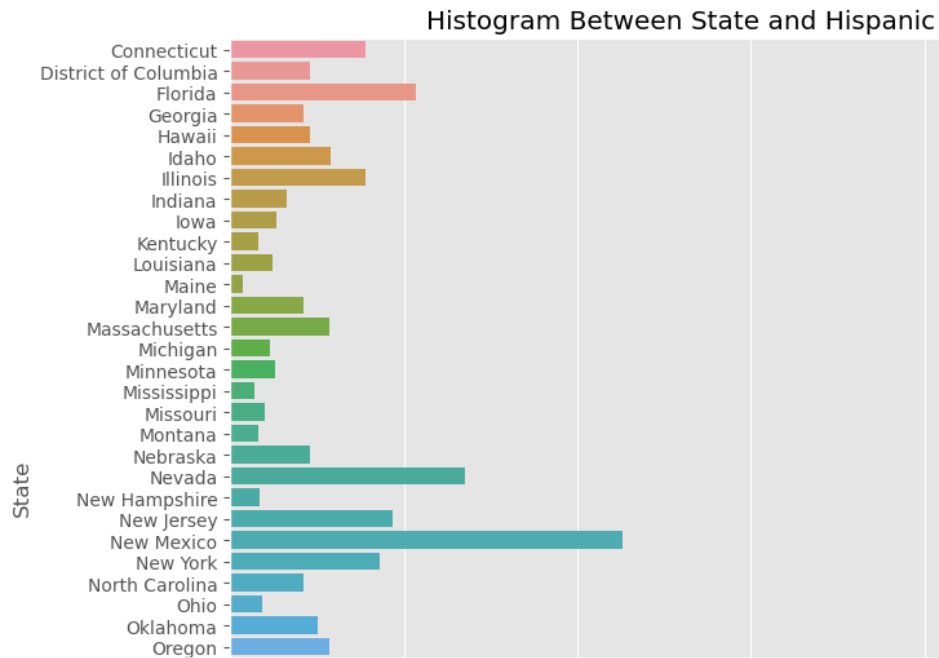
Out[256]:

	State	TotalPop	Hispanic	White	Black	Native	Asian	Pacific	
0	Connecticut	3593222	15.604831	67.677053	10.348068	0.126208	4.021981	0.018599	\$76
1	District of Columbia	647484	9.165922	33.103911	51.776536	0.200559	3.383240	0.029609	\$754
2	Florida	19645772	21.338543	59.083749	15.165676	0.210451	2.283174	0.051510	\$5069
3	Georgia	10006693	8.418242	54.286306	32.088298	0.187583	3.097649	0.046602	\$508
4	Hawaii	1406299	9.186709	25.032278	2.052848	0.144937	36.592089	8.758861	\$732



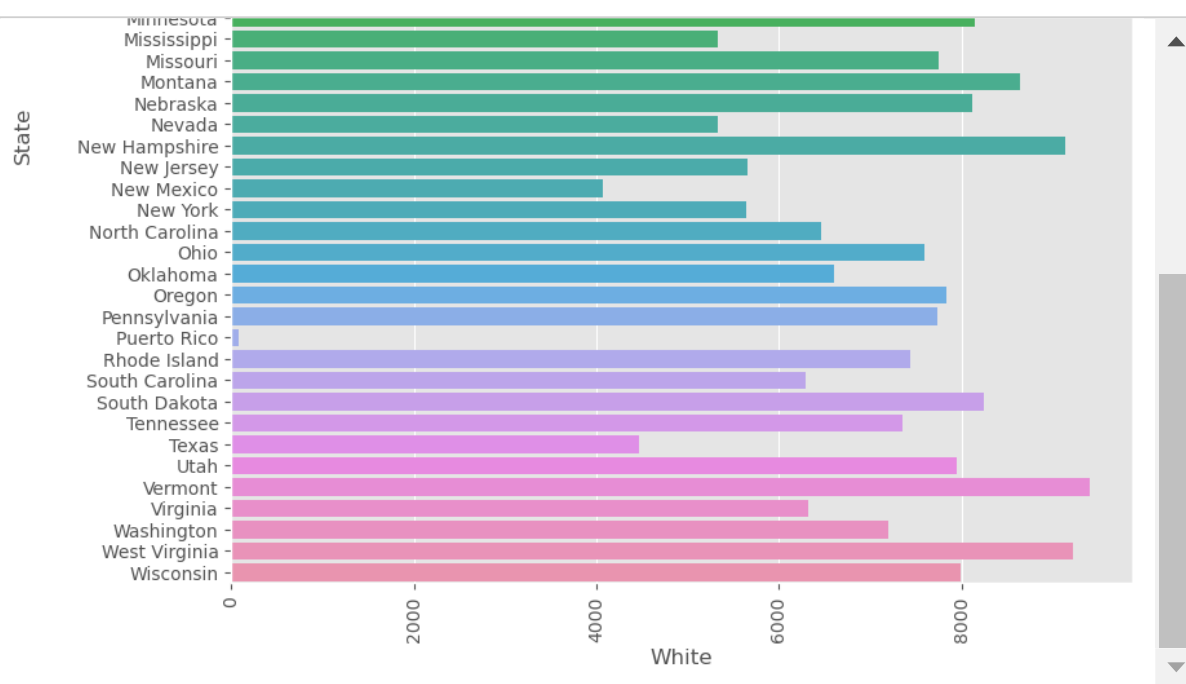
```
In [266]: # Converting From % to normal numbers
df["Hispanic"] = df["Hispanic"] * 100
```

```
In [267]: style.use('ggplot')
plt.figure(figsize = (9,9))
sns.barplot(x = df['Hispanic'] , y= df['State'])
plt.xticks(rotation = 90)
plt.title("Histogram Between State and Hispanic")
plt.show()
```



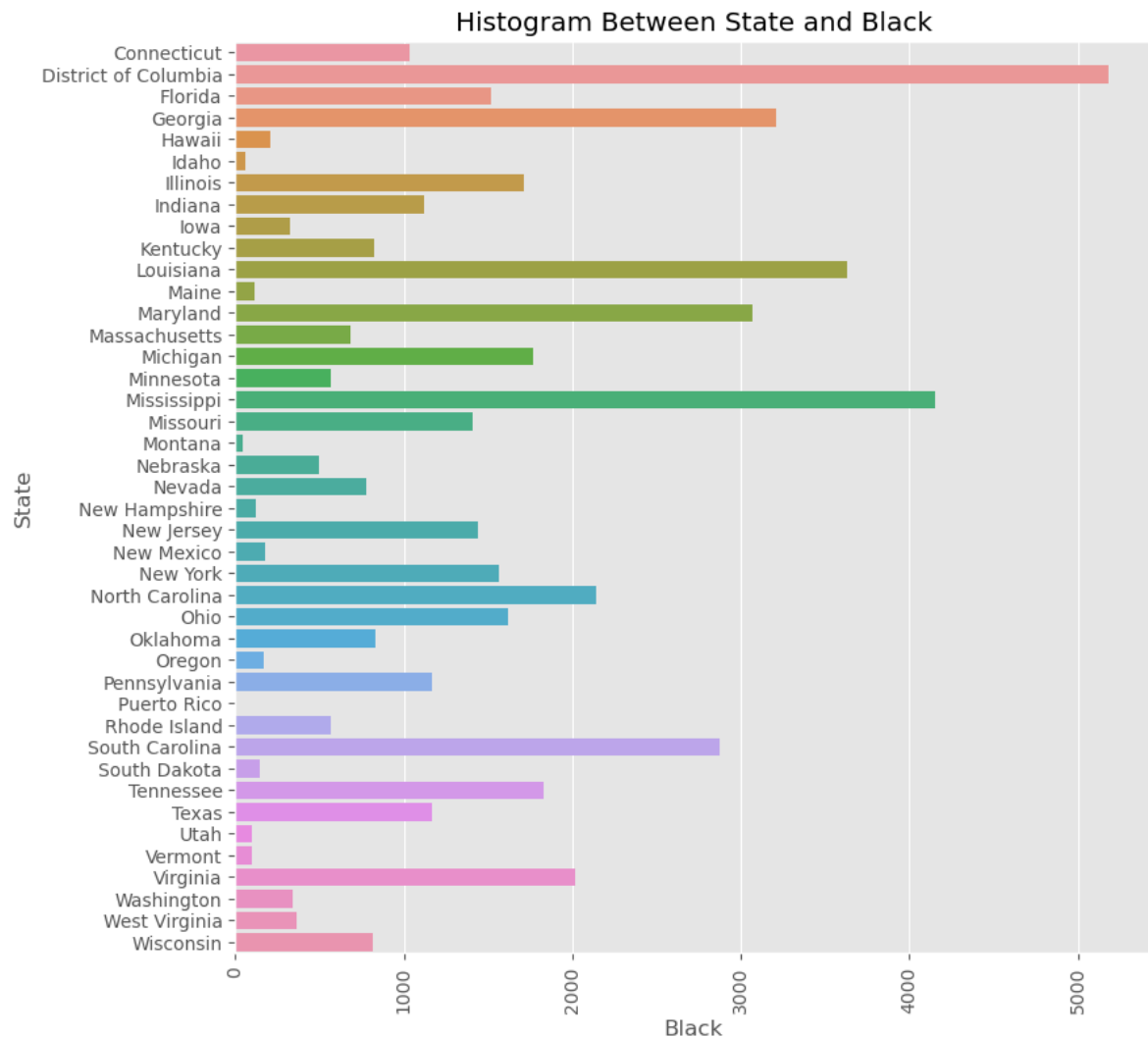
```
In [268]: # Converting From % to normal numbers
df["White"] = df["White"] * 100
```

```
In [269]: style.use('ggplot')
plt.figure(figsize = (9,9))
sns.barplot(x = df['White'] , y= df['State'])
plt.xticks(rotation = 90)
plt.title("Histogram Between State and White")
plt.show()
```



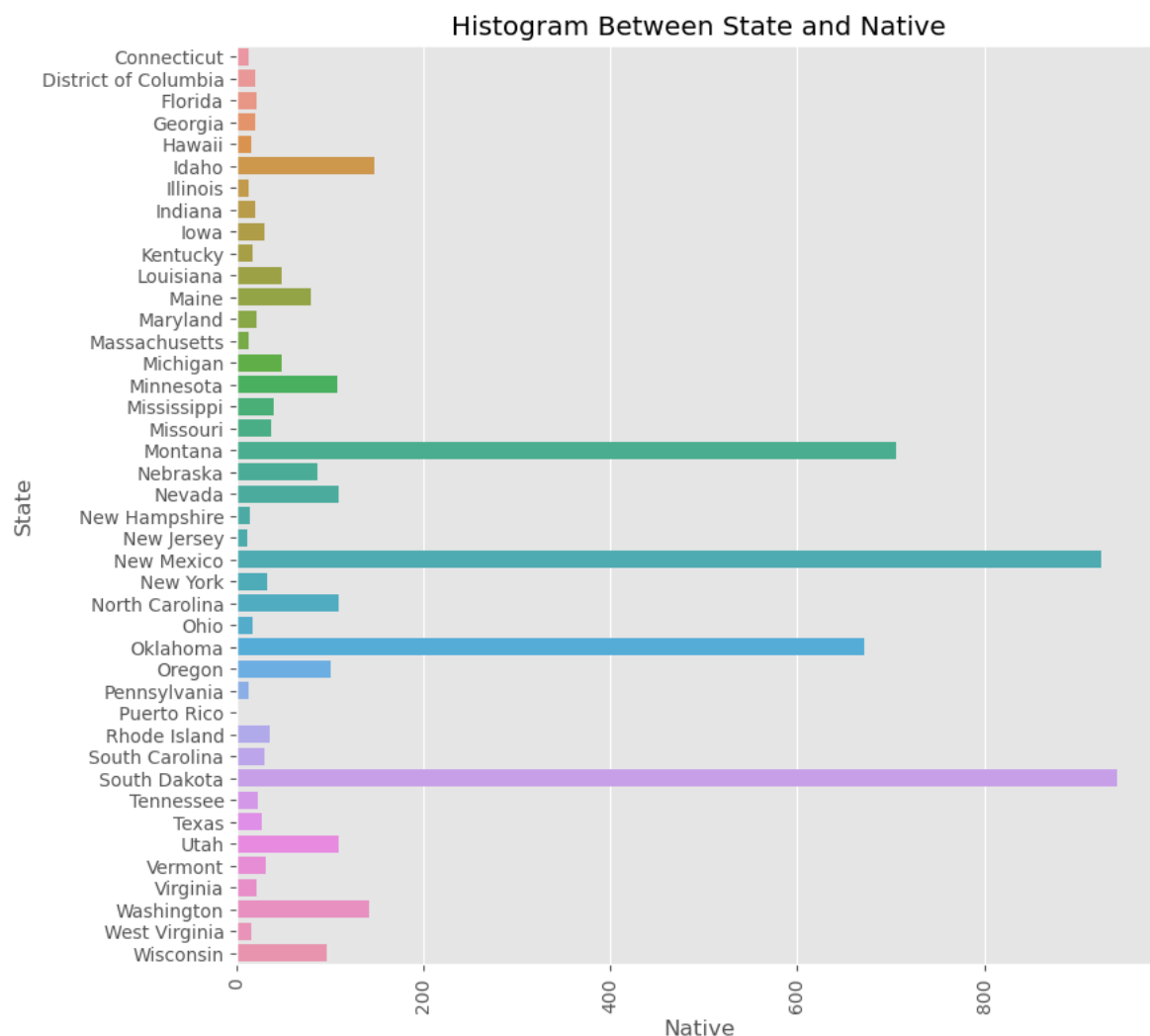
```
In [270]: # Converting From % to normal numbers
df["Black"] = df["Black"] * 100
```

```
In [271]: style.use('ggplot')
plt.figure(figsize = (9,9))
sns.barplot(x = df['Black'] , y= df['State'])
plt.xticks(rotation = 90)
plt.title("Histogram Between State and Black")
plt.show()
```



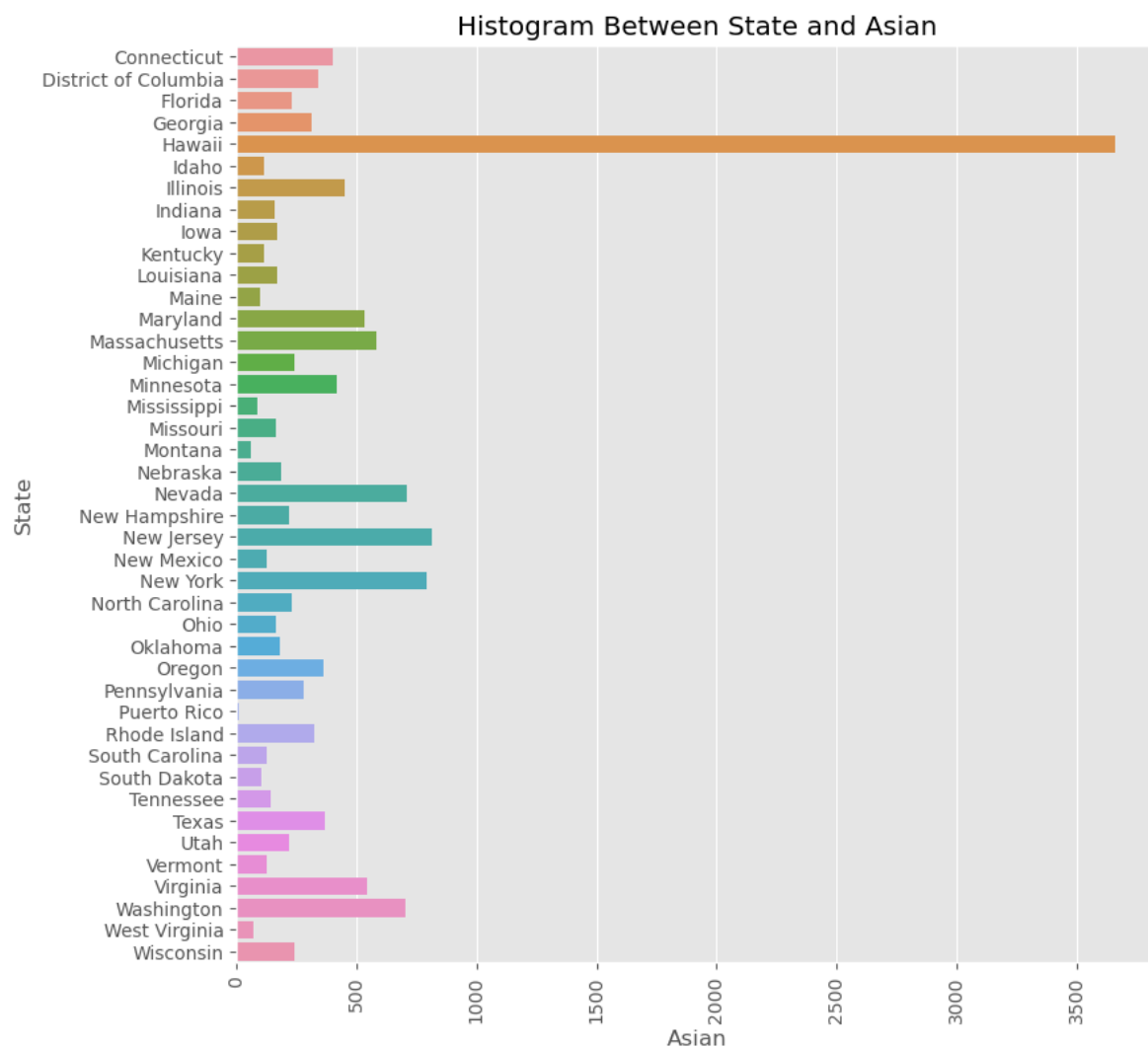
```
In [272]: # Converting From % to normal numbers
df["Native"] = df["Native"] * 100
```

```
In [273]: style.use('ggplot')
plt.figure(figsize = (9,9))
sns.barplot(x = df['Native'] , y= df['State'])
plt.xticks(rotation = 90)
plt.title("Histogram Between State and Native")
plt.show()
```



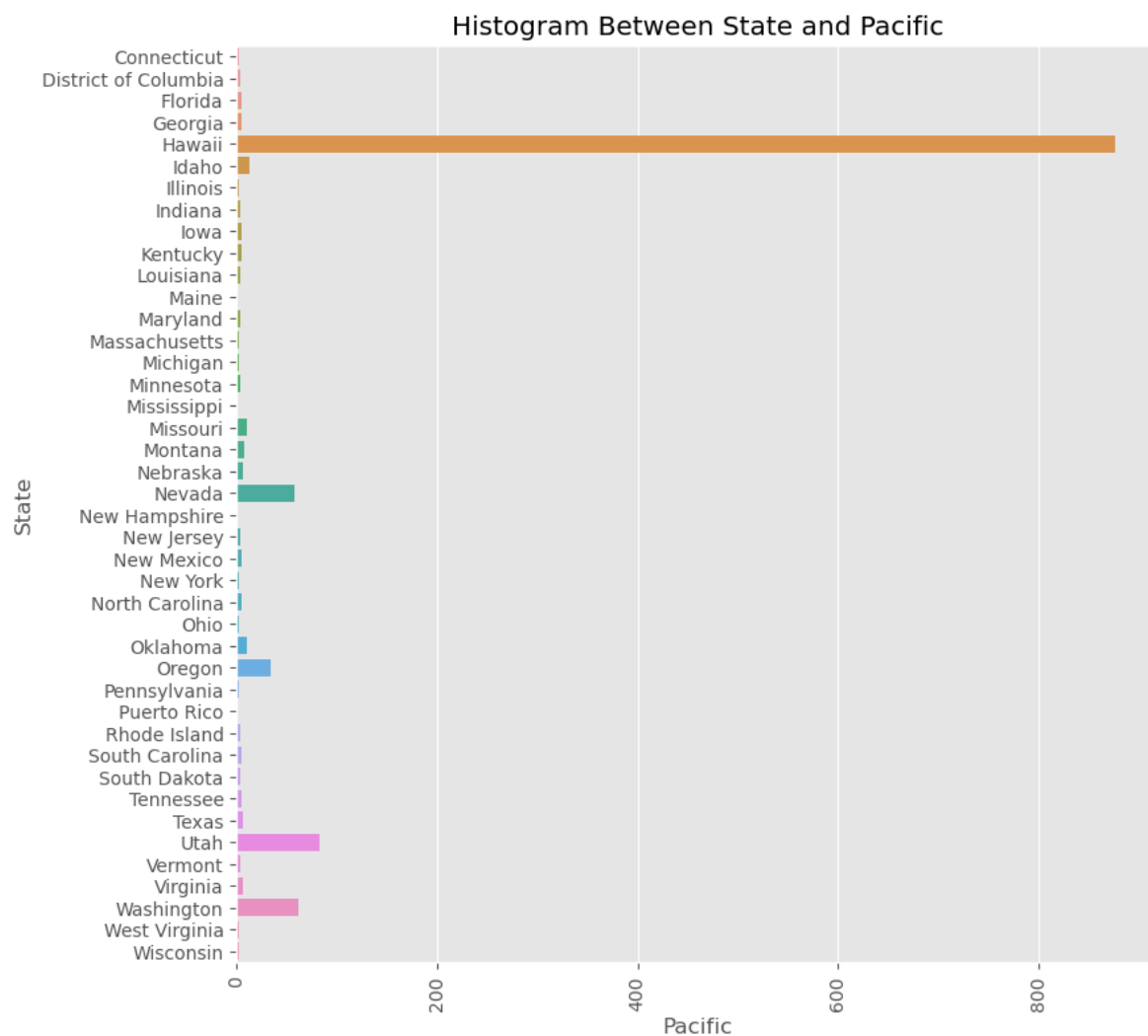
```
In [274]: # Converting From % to normal numbers
df["Asian"] = df["Asian"] * 100
```

```
In [275]: style.use('ggplot')
plt.figure(figsize = (9,9))
sns.barplot(x = df['Asian'] , y= df['State'])
plt.xticks(rotation = 90)
plt.title("Histogram Between State and Asian")
plt.show()
```



```
In [276]: # Converting From % to normal numbers
df["Pacific"] = df["Pacific"] * 100
```

```
In [277]: style.use('ggplot')
plt.figure(figsize = (9,9))
sns.barplot(x = df['Pacific'] , y= df['State'])
plt.xticks(rotation = 90)
plt.title("Histogram Between State and Pacific")
plt.show()
```



In [ ]: