{

"cells": [

{

"cell\_type": "code",

"execution\_count": 1,

"id": "0af4301b",

"metadata": {},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"C:\\Users\\rahul\\AppData\\Local\\Temp/ipykernel\_10396/1600307531.py:7: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float64` here.\n",

"Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations\n",

" from numpy import random, float, array\n"

]

}

],

"source": [

"%matplotlib inline\n",

"\n",

"import pandas as pd\n",

"from sklearn.cluster import KMeans\n",

"import matplotlib.pyplot as plt\n",

"from sklearn.preprocessing import scale\n",

"from numpy import random, float, array\n",

"import numpy as np\n",

"import seaborn as sns"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"id": "46514955",

"metadata": {},

"outputs": [],

"source": [

"crime = pd.read\_csv(\"crime\_data.csv\")"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"id": "1adf2bd1",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Unnamed: 0</th>\n",

" <th>Murder</th>\n",

" <th>Assault</th>\n",

" <th>UrbanPop</th>\n",

" <th>Rape</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>Alabama</td>\n",

" <td>13.2</td>\n",

" <td>236</td>\n",

" <td>58</td>\n",

" <td>21.2</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>Alaska</td>\n",

" <td>10.0</td>\n",

" <td>263</td>\n",

" <td>48</td>\n",

" <td>44.5</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>Arizona</td>\n",

" <td>8.1</td>\n",

" <td>294</td>\n",

" <td>80</td>\n",

" <td>31.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>Arkansas</td>\n",

" <td>8.8</td>\n",

" <td>190</td>\n",

" <td>50</td>\n",

" <td>19.5</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>California</td>\n",

" <td>9.0</td>\n",

" <td>276</td>\n",

" <td>91</td>\n",

" <td>40.6</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Unnamed: 0 Murder Assault UrbanPop Rape\n",

"0 Alabama 13.2 236 58 21.2\n",

"1 Alaska 10.0 263 48 44.5\n",

"2 Arizona 8.1 294 80 31.0\n",

"3 Arkansas 8.8 190 50 19.5\n",

"4 California 9.0 276 91 40.6"

]

},

"execution\_count": 3,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"crime.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"id": "61db0e3d",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(50, 5)"

]

},

"execution\_count": 4,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"crime.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"id": "ddfa0ab0",

"metadata": {},

"outputs": [],

"source": [

"# Normalization function \n",

"def norm\_func(i):\n",

" x = (i-i.min())/(i.max()-i.min())\n",

" return (x)"

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"id": "4f41c493",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Murder</th>\n",

" <th>Assault</th>\n",

" <th>UrbanPop</th>\n",

" <th>Rape</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>count</th>\n",

" <td>50.000000</td>\n",

" <td>50.000000</td>\n",

" <td>50.000000</td>\n",

" <td>50.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>mean</th>\n",

" <td>0.420964</td>\n",

" <td>0.430685</td>\n",

" <td>0.568475</td>\n",

" <td>0.360000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>std</th>\n",

" <td>0.262380</td>\n",

" <td>0.285403</td>\n",

" <td>0.245335</td>\n",

" <td>0.242025</td>\n",

" </tr>\n",

" <tr>\n",

" <th>min</th>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>25%</th>\n",

" <td>0.197289</td>\n",

" <td>0.219178</td>\n",

" <td>0.381356</td>\n",

" <td>0.200904</td>\n",

" </tr>\n",

" <tr>\n",

" <th>50%</th>\n",

" <td>0.388554</td>\n",

" <td>0.390411</td>\n",

" <td>0.576271</td>\n",

" <td>0.330749</td>\n",

" </tr>\n",

" <tr>\n",

" <th>75%</th>\n",

" <td>0.629518</td>\n",

" <td>0.698630</td>\n",

" <td>0.775424</td>\n",

" <td>0.487726</td>\n",

" </tr>\n",

" <tr>\n",

" <th>max</th>\n",

" <td>1.000000</td>\n",

" <td>1.000000</td>\n",

" <td>1.000000</td>\n",

" <td>1.000000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Murder Assault UrbanPop Rape\n",

"count 50.000000 50.000000 50.000000 50.000000\n",

"mean 0.420964 0.430685 0.568475 0.360000\n",

"std 0.262380 0.285403 0.245335 0.242025\n",

"min 0.000000 0.000000 0.000000 0.000000\n",

"25% 0.197289 0.219178 0.381356 0.200904\n",

"50% 0.388554 0.390411 0.576271 0.330749\n",

"75% 0.629518 0.698630 0.775424 0.487726\n",

"max 1.000000 1.000000 1.000000 1.000000"

]

},

"execution\_count": 6,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"# Normalized data frame (considering the numerical part of data)\n",

"df\_norm = norm\_func(crime.iloc[:,1:])\n",

"df\_norm.describe()"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"id": "9adf22af",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Murder</th>\n",

" <th>Assault</th>\n",

" <th>UrbanPop</th>\n",

" <th>Rape</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>count</th>\n",

" <td>50.000000</td>\n",

" <td>50.000000</td>\n",

" <td>50.000000</td>\n",

" <td>50.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>mean</th>\n",

" <td>0.420964</td>\n",

" <td>0.430685</td>\n",

" <td>0.568475</td>\n",

" <td>0.360000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>std</th>\n",

" <td>0.262380</td>\n",

" <td>0.285403</td>\n",

" <td>0.245335</td>\n",

" <td>0.242025</td>\n",

" </tr>\n",

" <tr>\n",

" <th>min</th>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>25%</th>\n",

" <td>0.197289</td>\n",

" <td>0.219178</td>\n",

" <td>0.381356</td>\n",

" <td>0.200904</td>\n",

" </tr>\n",

" <tr>\n",

" <th>50%</th>\n",

" <td>0.388554</td>\n",

" <td>0.390411</td>\n",

" <td>0.576271</td>\n",

" <td>0.330749</td>\n",

" </tr>\n",

" <tr>\n",

" <th>75%</th>\n",

" <td>0.629518</td>\n",

" <td>0.698630</td>\n",

" <td>0.775424</td>\n",

" <td>0.487726</td>\n",

" </tr>\n",

" <tr>\n",

" <th>max</th>\n",

" <td>1.000000</td>\n",

" <td>1.000000</td>\n",

" <td>1.000000</td>\n",

" <td>1.000000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Murder Assault UrbanPop Rape\n",

"count 50.000000 50.000000 50.000000 50.000000\n",

"mean 0.420964 0.430685 0.568475 0.360000\n",

"std 0.262380 0.285403 0.245335 0.242025\n",

"min 0.000000 0.000000 0.000000 0.000000\n",

"25% 0.197289 0.219178 0.381356 0.200904\n",

"50% 0.388554 0.390411 0.576271 0.330749\n",

"75% 0.629518 0.698630 0.775424 0.487726\n",

"max 1.000000 1.000000 1.000000 1.000000"

]

},

"execution\_count": 7,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"# Normalized data frame (considering the numerical part of data)\n",

"df\_norm = norm\_func(crime.iloc[:,1:])\n",

"df\_norm.describe()"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"id": "2e2a3cd6",

"metadata": {},

"outputs": [],

"source": [

"from scipy.cluster.hierarchy import linkage \n",

"import scipy.cluster.hierarchy as sch # for creating dendrogram "

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"id": "d6fa9822",

"metadata": {},

"outputs": [],

"source": [

"z = linkage(df\_norm, method=\"complete\",metric=\"euclidean\")"

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"id": "b82bf3c2",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1080x360 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"plt.figure(figsize=(15, 5))\n",

"plt.title('Hierarchical Clustering Dendrogram')\n",

"plt.xlabel('Features')\n",

"plt.ylabel('Crime')\n",

"sch.dendrogram(z,\n",

" leaf\_rotation=0., # rotates the x axis labels\n",

" leaf\_font\_size=8., # font size for the x axis labels\n",

" )\n",

"plt.show()"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"id": "4dc49a93",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Murder</th>\n",

" <th>Assault</th>\n",

" <th>UrbanPop</th>\n",

" <th>Rape</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>Murder</th>\n",

" <td>1.000000</td>\n",

" <td>0.801873</td>\n",

" <td>0.069573</td>\n",

" <td>0.563579</td>\n",

" </tr>\n",

" <tr>\n",

" <th>Assault</th>\n",

" <td>0.801873</td>\n",

" <td>1.000000</td>\n",

" <td>0.258872</td>\n",

" <td>0.665241</td>\n",

" </tr>\n",

" <tr>\n",

" <th>UrbanPop</th>\n",

" <td>0.069573</td>\n",

" <td>0.258872</td>\n",

" <td>1.000000</td>\n",

" <td>0.411341</td>\n",

" </tr>\n",

" <tr>\n",

" <th>Rape</th>\n",

" <td>0.563579</td>\n",

" <td>0.665241</td>\n",

" <td>0.411341</td>\n",

" <td>1.000000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Murder Assault UrbanPop Rape\n",

"Murder 1.000000 0.801873 0.069573 0.563579\n",

"Assault 0.801873 1.000000 0.258872 0.665241\n",

"UrbanPop 0.069573 0.258872 1.000000 0.411341\n",

"Rape 0.563579 0.665241 0.411341 1.000000"

]

},

"execution\_count": 12,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"crime.corr()"

]

},

{

"cell\_type": "code",

"execution\_count": 13,

"id": "13c9e670",

"metadata": {},

"outputs": [],

"source": [

"###### screw plot or elbow curve ############\n",

"k = list(range(2,15))\n",

"#k"

]

},

{

"cell\_type": "code",

"execution\_count": 14,

"id": "fcc057b9",

"metadata": {},

"outputs": [],

"source": [

"from sklearn.cluster import\tKMeans\n",

"from scipy.spatial.distance import cdist \n",

"import numpy as np"

]

},

{

"cell\_type": "code",

"execution\_count": 15,

"id": "c59e99e0",

"metadata": {},

"outputs": [],

"source": [

"TWSS = [] # variable for storing total within sum of squares for each kmeans \n",

"for i in k:\n",

" kmeans = KMeans(n\_clusters = i)\n",

" kmeans.fit(df\_norm)\n",

" WSS = [] # variable for storing within sum of squares for each cluster \n",

" for j in range(i):\n",

" WSS.append(sum(cdist(df\_norm.iloc[kmeans.labels\_==j,:],kmeans.cluster\_centers\_[j].reshape(1,df\_norm.shape[1]),\"euclidean\")))\n",

" TWSS.append(sum(WSS))"

]

},

{

"cell\_type": "code",

"execution\_count": 16,

"id": "48628ea4",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"([<matplotlib.axis.XTick at 0x2372e839550>,\n",

" <matplotlib.axis.XTick at 0x2372e839520>,\n",

" <matplotlib.axis.XTick at 0x2372e8320a0>,\n",

" <matplotlib.axis.XTick at 0x2372e3c03d0>,\n",

" <matplotlib.axis.XTick at 0x2372e3c0a60>,\n",

" <matplotlib.axis.XTick at 0x2372e3c71f0>,\n",

" <matplotlib.axis.XTick at 0x2372e3c0ac0>,\n",

" <matplotlib.axis.XTick at 0x2372e3c71c0>,\n",

" <matplotlib.axis.XTick at 0x2372e3cc070>,\n",

" <matplotlib.axis.XTick at 0x2372e3cc700>,\n",

" <matplotlib.axis.XTick at 0x2372e3cce50>,\n",

" <matplotlib.axis.XTick at 0x2372e3d45e0>,\n",

" <matplotlib.axis.XTick at 0x2372e3cce20>],\n",

" [Text(0, 0, ''),\n",

" Text(0, 0, ''),\n",

" Text(0, 0, ''),\n",

" Text(0, 0, ''),\n",

" Text(0, 0, ''),\n",

" Text(0, 0, ''),\n",

" Text(0, 0, ''),\n",

" Text(0, 0, ''),\n",

" Text(0, 0, ''),\n",

" Text(0, 0, ''),\n",

" Text(0, 0, ''),\n",

" Text(0, 0, ''),\n",

" Text(0, 0, '')])"

]

},

"execution\_count": 16,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1152x432 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"# Scree plot \n",

"\n",

"plt.figure(figsize=(16,6))\n",

"plt.plot(k,TWSS,'ro-');plt.xlabel(\"No\_of\_Clusters\");plt.ylabel(\"total\_within\_SS\");plt.xticks(k)"

]

},

{

"cell\_type": "code",

"execution\_count": 17,

"id": "1de89202",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"KMeans(n\_clusters=4)"

]

},

"execution\_count": 17,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"# Selecting 4 clusters from the above scree plot which is the optimum number of clusters \n",

"model=KMeans(n\_clusters=4) \n",

"model.fit(df\_norm)"

]

},

{

"cell\_type": "code",

"execution\_count": 18,

"id": "a0a14bc4",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([1, 2, 2, 1, 2, 2, 3, 3, 2, 1, 3, 0, 2, 3, 0, 3, 0, 1, 0, 2, 3, 2,\n",

" 0, 1, 3, 0, 0, 2, 0, 3, 2, 2, 1, 0, 3, 3, 3, 3, 3, 1, 0, 1, 2, 3,\n",

" 0, 3, 3, 0, 0, 3])"

]

},

"execution\_count": 18,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"model.labels\_ # getting the labels of clusters assigned to each row "

]

},

{

"cell\_type": "code",

"execution\_count": 19,

"id": "8362d6fe",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([[0.1686747 , 0.11485774, 0.34028683, 0.12601868],\n",

" [0.79141566, 0.6802226 , 0.36864407, 0.36466408],\n",

" [0.6124498 , 0.75 , 0.75423729, 0.67980189],\n",

" [0.30439405, 0.32937147, 0.70588235, 0.31098951]])"

]

},

"execution\_count": 19,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"model.cluster\_centers\_"

]

},

{

"cell\_type": "code",

"execution\_count": 20,

"id": "f5041488",

"metadata": {},

"outputs": [],

"source": [

"import seaborn as sns"

]

},

{

"cell\_type": "code",

"execution\_count": 21,

"id": "dd1be0e7",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Murder</th>\n",

" <th>Assault</th>\n",

" <th>Rape</th>\n",

" <th>UrbanPop</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>13.2</td>\n",

" <td>236</td>\n",

" <td>21.2</td>\n",

" <td>58</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>10.0</td>\n",

" <td>263</td>\n",

" <td>44.5</td>\n",

" <td>48</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>8.1</td>\n",

" <td>294</td>\n",

" <td>31.0</td>\n",

" <td>80</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>8.8</td>\n",

" <td>190</td>\n",

" <td>19.5</td>\n",

" <td>50</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>9.0</td>\n",

" <td>276</td>\n",

" <td>40.6</td>\n",

" <td>91</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Murder Assault Rape UrbanPop\n",

"0 13.2 236 21.2 58\n",

"1 10.0 263 44.5 48\n",

"2 8.1 294 31.0 80\n",

"3 8.8 190 19.5 50\n",

"4 9.0 276 40.6 91"

]

},

"execution\_count": 21,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"X = crime[['Murder', 'Assault', 'Rape', 'UrbanPop']]\n",

"clusters = KMeans(4) # 4 clusters!\n",

"clusters.fit( X )\n",

"clusters.cluster\_centers\_\n",

"clusters.labels\_\n",

"crime['Crime\_clusters'] = clusters.labels\_\n",

"crime.head()\n",

"crime.sort\_values(by=['Crime\_clusters'],ascending = True)\n",

"X.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 22,

"id": "4b31d89a",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Unnamed: 0</th>\n",

" <th>Murder</th>\n",

" <th>Assault</th>\n",

" <th>UrbanPop</th>\n",

" <th>Rape</th>\n",

" <th>Crime\_clusters</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>33</th>\n",

" <td>North Dakota</td>\n",

" <td>0.8</td>\n",

" <td>45</td>\n",

" <td>44</td>\n",

" <td>7.3</td>\n",

" <td>2</td>\n",

" </tr>\n",

" <tr>\n",

" <th>28</th>\n",

" <td>New Hampshire</td>\n",

" <td>2.1</td>\n",

" <td>57</td>\n",

" <td>56</td>\n",

" <td>9.5</td>\n",

" <td>2</td>\n",

" </tr>\n",

" <tr>\n",

" <th>18</th>\n",

" <td>Maine</td>\n",

" <td>2.1</td>\n",

" <td>83</td>\n",

" <td>51</td>\n",

" <td>7.8</td>\n",

" <td>2</td>\n",

" </tr>\n",

" <tr>\n",

" <th>14</th>\n",

" <td>Iowa</td>\n",

" <td>2.2</td>\n",

" <td>56</td>\n",

" <td>57</td>\n",

" <td>11.3</td>\n",

" <td>2</td>\n",

" </tr>\n",

" <tr>\n",

" <th>44</th>\n",

" <td>Vermont</td>\n",

" <td>2.2</td>\n",

" <td>48</td>\n",

" <td>32</td>\n",

" <td>11.2</td>\n",

" <td>2</td>\n",

" </tr>\n",

" <tr>\n",

" <th>48</th>\n",

" <td>Wisconsin</td>\n",

" <td>2.6</td>\n",

" <td>53</td>\n",

" <td>66</td>\n",

" <td>10.8</td>\n",

" <td>2</td>\n",

" </tr>\n",

" <tr>\n",

" <th>11</th>\n",

" <td>Idaho</td>\n",

" <td>2.6</td>\n",

" <td>120</td>\n",

" <td>54</td>\n",

" <td>14.2</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>22</th>\n",

" <td>Minnesota</td>\n",

" <td>2.7</td>\n",

" <td>72</td>\n",

" <td>66</td>\n",

" <td>14.9</td>\n",

" <td>2</td>\n",

" </tr>\n",

" <tr>\n",

" <th>43</th>\n",

" <td>Utah</td>\n",

" <td>3.2</td>\n",

" <td>120</td>\n",

" <td>80</td>\n",

" <td>22.9</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>6</th>\n",

" <td>Connecticut</td>\n",

" <td>3.3</td>\n",

" <td>110</td>\n",

" <td>77</td>\n",

" <td>11.1</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>38</th>\n",

" <td>Rhode Island</td>\n",

" <td>3.4</td>\n",

" <td>174</td>\n",

" <td>87</td>\n",

" <td>8.3</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>40</th>\n",

" <td>South Dakota</td>\n",

" <td>3.8</td>\n",

" <td>86</td>\n",

" <td>45</td>\n",

" <td>12.8</td>\n",

" <td>2</td>\n",

" </tr>\n",

" <tr>\n",

" <th>46</th>\n",

" <td>Washington</td>\n",

" <td>4.0</td>\n",

" <td>145</td>\n",

" <td>73</td>\n",

" <td>26.2</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>26</th>\n",

" <td>Nebraska</td>\n",

" <td>4.3</td>\n",

" <td>102</td>\n",

" <td>62</td>\n",

" <td>16.5</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>20</th>\n",

" <td>Massachusetts</td>\n",

" <td>4.4</td>\n",

" <td>149</td>\n",

" <td>85</td>\n",

" <td>16.3</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>36</th>\n",

" <td>Oregon</td>\n",

" <td>4.9</td>\n",

" <td>159</td>\n",

" <td>67</td>\n",

" <td>29.3</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>10</th>\n",

" <td>Hawaii</td>\n",

" <td>5.3</td>\n",

" <td>46</td>\n",

" <td>83</td>\n",

" <td>20.2</td>\n",

" <td>2</td>\n",

" </tr>\n",

" <tr>\n",

" <th>47</th>\n",

" <td>West Virginia</td>\n",

" <td>5.7</td>\n",

" <td>81</td>\n",

" <td>39</td>\n",

" <td>9.3</td>\n",

" <td>2</td>\n",

" </tr>\n",

" <tr>\n",

" <th>7</th>\n",

" <td>Delaware</td>\n",

" <td>5.9</td>\n",

" <td>238</td>\n",

" <td>72</td>\n",

" <td>15.8</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>25</th>\n",

" <td>Montana</td>\n",

" <td>6.0</td>\n",

" <td>109</td>\n",

" <td>53</td>\n",

" <td>16.4</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>15</th>\n",

" <td>Kansas</td>\n",

" <td>6.0</td>\n",

" <td>115</td>\n",

" <td>66</td>\n",

" <td>18.0</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>37</th>\n",

" <td>Pennsylvania</td>\n",

" <td>6.3</td>\n",

" <td>106</td>\n",

" <td>72</td>\n",

" <td>14.9</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>35</th>\n",

" <td>Oklahoma</td>\n",

" <td>6.6</td>\n",

" <td>151</td>\n",

" <td>68</td>\n",

" <td>20.0</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>49</th>\n",

" <td>Wyoming</td>\n",

" <td>6.8</td>\n",

" <td>161</td>\n",

" <td>60</td>\n",

" <td>15.6</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>13</th>\n",

" <td>Indiana</td>\n",

" <td>7.2</td>\n",

" <td>113</td>\n",

" <td>65</td>\n",

" <td>21.0</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>34</th>\n",

" <td>Ohio</td>\n",

" <td>7.3</td>\n",

" <td>120</td>\n",

" <td>75</td>\n",

" <td>21.4</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>29</th>\n",

" <td>New Jersey</td>\n",

" <td>7.4</td>\n",

" <td>159</td>\n",

" <td>89</td>\n",

" <td>18.8</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>5</th>\n",

" <td>Colorado</td>\n",

" <td>7.9</td>\n",

" <td>204</td>\n",

" <td>78</td>\n",

" <td>38.7</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>Arizona</td>\n",

" <td>8.1</td>\n",

" <td>294</td>\n",

" <td>80</td>\n",

" <td>31.0</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>45</th>\n",

" <td>Virginia</td>\n",

" <td>8.5</td>\n",

" <td>156</td>\n",

" <td>63</td>\n",

" <td>20.7</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>Arkansas</td>\n",

" <td>8.8</td>\n",

" <td>190</td>\n",

" <td>50</td>\n",

" <td>19.5</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>24</th>\n",

" <td>Missouri</td>\n",

" <td>9.0</td>\n",

" <td>178</td>\n",

" <td>70</td>\n",

" <td>28.2</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>California</td>\n",

" <td>9.0</td>\n",

" <td>276</td>\n",

" <td>91</td>\n",

" <td>40.6</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>16</th>\n",

" <td>Kentucky</td>\n",

" <td>9.7</td>\n",

" <td>109</td>\n",

" <td>52</td>\n",

" <td>16.3</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>Alaska</td>\n",

" <td>10.0</td>\n",

" <td>263</td>\n",

" <td>48</td>\n",

" <td>44.5</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>12</th>\n",

" <td>Illinois</td>\n",

" <td>10.4</td>\n",

" <td>249</td>\n",

" <td>83</td>\n",

" <td>24.0</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>31</th>\n",

" <td>New York</td>\n",

" <td>11.1</td>\n",

" <td>254</td>\n",

" <td>86</td>\n",

" <td>26.1</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>19</th>\n",

" <td>Maryland</td>\n",

" <td>11.3</td>\n",

" <td>300</td>\n",

" <td>67</td>\n",

" <td>27.8</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>30</th>\n",

" <td>New Mexico</td>\n",

" <td>11.4</td>\n",

" <td>285</td>\n",

" <td>70</td>\n",

" <td>32.1</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>21</th>\n",

" <td>Michigan</td>\n",

" <td>12.1</td>\n",

" <td>255</td>\n",

" <td>74</td>\n",

" <td>35.1</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>27</th>\n",

" <td>Nevada</td>\n",

" <td>12.2</td>\n",

" <td>252</td>\n",

" <td>81</td>\n",

" <td>46.0</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>42</th>\n",

" <td>Texas</td>\n",

" <td>12.7</td>\n",

" <td>201</td>\n",

" <td>80</td>\n",

" <td>25.5</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>32</th>\n",

" <td>North Carolina</td>\n",

" <td>13.0</td>\n",

" <td>337</td>\n",

" <td>45</td>\n",

" <td>16.1</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>41</th>\n",

" <td>Tennessee</td>\n",

" <td>13.2</td>\n",

" <td>188</td>\n",

" <td>59</td>\n",

" <td>26.9</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>Alabama</td>\n",

" <td>13.2</td>\n",

" <td>236</td>\n",

" <td>58</td>\n",

" <td>21.2</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>39</th>\n",

" <td>South Carolina</td>\n",

" <td>14.4</td>\n",

" <td>279</td>\n",

" <td>48</td>\n",

" <td>22.5</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>8</th>\n",

" <td>Florida</td>\n",

" <td>15.4</td>\n",

" <td>335</td>\n",

" <td>80</td>\n",

" <td>31.9</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>17</th>\n",

" <td>Louisiana</td>\n",

" <td>15.4</td>\n",

" <td>249</td>\n",

" <td>66</td>\n",

" <td>22.2</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>23</th>\n",

" <td>Mississippi</td>\n",

" <td>16.1</td>\n",

" <td>259</td>\n",

" <td>44</td>\n",

" <td>17.1</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>9</th>\n",

" <td>Georgia</td>\n",

" <td>17.4</td>\n",

" <td>211</td>\n",

" <td>60</td>\n",

" <td>25.8</td>\n",

" <td>3</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Unnamed: 0 Murder Assault UrbanPop Rape Crime\_clusters\n",

"33 North Dakota 0.8 45 44 7.3 2\n",

"28 New Hampshire 2.1 57 56 9.5 2\n",

"18 Maine 2.1 83 51 7.8 2\n",

"14 Iowa 2.2 56 57 11.3 2\n",

"44 Vermont 2.2 48 32 11.2 2\n",

"48 Wisconsin 2.6 53 66 10.8 2\n",

"11 Idaho 2.6 120 54 14.2 1\n",

"22 Minnesota 2.7 72 66 14.9 2\n",

"43 Utah 3.2 120 80 22.9 1\n",

"6 Connecticut 3.3 110 77 11.1 1\n",

"38 Rhode Island 3.4 174 87 8.3 3\n",

"40 South Dakota 3.8 86 45 12.8 2\n",

"46 Washington 4.0 145 73 26.2 3\n",

"26 Nebraska 4.3 102 62 16.5 1\n",

"20 Massachusetts 4.4 149 85 16.3 3\n",

"36 Oregon 4.9 159 67 29.3 3\n",

"10 Hawaii 5.3 46 83 20.2 2\n",

"47 West Virginia 5.7 81 39 9.3 2\n",

"7 Delaware 5.9 238 72 15.8 0\n",

"25 Montana 6.0 109 53 16.4 1\n",

"15 Kansas 6.0 115 66 18.0 1\n",

"37 Pennsylvania 6.3 106 72 14.9 1\n",

"35 Oklahoma 6.6 151 68 20.0 3\n",

"49 Wyoming 6.8 161 60 15.6 3\n",

"13 Indiana 7.2 113 65 21.0 1\n",

"34 Ohio 7.3 120 75 21.4 1\n",

"29 New Jersey 7.4 159 89 18.8 3\n",

"5 Colorado 7.9 204 78 38.7 3\n",

"2 Arizona 8.1 294 80 31.0 0\n",

"45 Virginia 8.5 156 63 20.7 3\n",

"3 Arkansas 8.8 190 50 19.5 3\n",

"24 Missouri 9.0 178 70 28.2 3\n",

"4 California 9.0 276 91 40.6 0\n",

"16 Kentucky 9.7 109 52 16.3 1\n",

"1 Alaska 10.0 263 48 44.5 0\n",

"12 Illinois 10.4 249 83 24.0 0\n",

"31 New York 11.1 254 86 26.1 0\n",

"19 Maryland 11.3 300 67 27.8 0\n",

"30 New Mexico 11.4 285 70 32.1 0\n",

"21 Michigan 12.1 255 74 35.1 0\n",

"27 Nevada 12.2 252 81 46.0 0\n",

"42 Texas 12.7 201 80 25.5 3\n",

"32 North Carolina 13.0 337 45 16.1 0\n",

"41 Tennessee 13.2 188 59 26.9 3\n",

"0 Alabama 13.2 236 58 21.2 0\n",

"39 South Carolina 14.4 279 48 22.5 0\n",

"8 Florida 15.4 335 80 31.9 0\n",

"17 Louisiana 15.4 249 66 22.2 0\n",

"23 Mississippi 16.1 259 44 17.1 0\n",

"9 Georgia 17.4 211 60 25.8 3"

]

},

"execution\_count": 22,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"stats =crime.sort\_values(\"Murder\", ascending=True)\n",

"stats"

]

},

{

"cell\_type": "code",

"execution\_count": 23,

"id": "3310fad0",

"metadata": {},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"C:\\Users\\rahul\\anaconda3\\lib\\site-packages\\seaborn\\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.\n",

" warnings.warn(\n",

"C:\\Users\\rahul\\anaconda3\\lib\\site-packages\\seaborn\\regression.py:581: UserWarning: The `size` parameter has been renamed to `height`; please update your code.\n",

" warnings.warn(msg, UserWarning)\n"

]

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 514x432 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"# Plot between pairs Murder~Assault\n",

"sns.lmplot( 'Murder','Assault', data=crime,\n",

" hue = 'Crime\_clusters',\n",

" fit\_reg=False, size = 6 );"

]

},

{

"cell\_type": "code",

"execution\_count": 25,

"id": "f4f66893",

"metadata": {},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"C:\\Users\\rahul\\anaconda3\\lib\\site-packages\\seaborn\\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.\n",

" warnings.warn(\n",

"C:\\Users\\rahul\\anaconda3\\lib\\site-packages\\seaborn\\regression.py:581: UserWarning: The `size` parameter has been renamed to `height`; please update your code.\n",

" warnings.warn(msg, UserWarning)\n"

]

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 514x432 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"# Plot between pairs Murder~Rape\n",

"sns.lmplot( 'Murder','Rape', data=crime,\n",

" hue = 'Crime\_clusters',\n",

" fit\_reg=False, size = 6 );"

]

},

{

"cell\_type": "code",

"execution\_count": 26,

"id": "cf9f8fee",

"metadata": {},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"C:\\Users\\rahul\\anaconda3\\lib\\site-packages\\seaborn\\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.\n",

" warnings.warn(\n",

"C:\\Users\\rahul\\anaconda3\\lib\\site-packages\\seaborn\\regression.py:581: UserWarning: The `size` parameter has been renamed to `height`; please update your code.\n",

" warnings.warn(msg, UserWarning)\n"

]

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 514x432 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"# Plot between pairs Assault~Rape\n",

"\n",

"sns.lmplot( 'Assault','Rape', data=crime,\n",

" hue = 'Crime\_clusters', fit\_reg=False, size = 6 );\n"

]

},

{

"cell\_type": "markdown",

"id": "f1faa717",

"metadata": {},

"source": [

"# All dots are states of US and different colors are one cluster showing clustering for the crime data."

]

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3 (ipykernel)",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.9.7"

}

},

"nbformat": 4,

"nbformat\_minor": 5

}