Name - ARYAN BAJAJ

Task 3 - Exploratory Data Analysis - Retail (Level - Beginner)

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns

data=pd.read_csv('C:/Users/HP/Desktop/SampleSuperstore.csv',encoding='latin1')
data
```

Out[1]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.5820
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.8714
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.0310
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.5164
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.2480	3	0.20	4.1028
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.9600	2	0.00	15.6332
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	258.5760	2	0.20	19.3932
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	29.6000	4	0.00	13.3200
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	243.1600	2	0.00	72.9480

9994 rows × 13 columns

Understanding the Data

In [2]:
```
data.head(5)
```

Out[2]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.5820
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.8714
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.0310
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.5164

In [3]:
```
data.tail(5)
```

Out[3]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.248	3	0.2	4.1028
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.960	2	0.0	15.6332
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	258.576	2	0.2	19.3932
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	29.600	4	0.0	13.3200
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	243.160	2	0.0	72.9480

In [4]:
```
data.describe()
```

Out[4]:

	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	55190.379428	229.858001	3.789574	0.156203	28.656896
std	32063.693350	623.245101	2.225110	0.206452	234.260108
min	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	90008.000000	209.940000	5.000000	0.200000	29.364000
max	99301.000000	22638.480000	14.000000	0.800000	8399.976000

In [5]:
```
data.shape
```

Out[5]:
```
(9994, 13)
```

In [6]:
```
data.dtypes
```

Out[6]:
```
Ship Mode        object
Segment          object
Country          object
City             object
State            object
Postal Code       int64
Region           object
Category         object
Sub-Category     object
Sales           float64
Quantity          int64
Discount        float64
Profit          float64
dtype: object
```

In [7]:
```
data.nunique()
```

Out[7]:
```
Ship Mode          4
Segment            3
Country            1
City             531
State             49
Postal Code      631
Region             4
Category           3
Sub-Category      17
Sales           5825
Quantity          14
Discount          12
Profit          7287
dtype: int64
```

In [8]:
```
data['Ship Mode'].unique()
```

Out[8]:
```
array(['Second Class', 'Standard Class', 'First Class', 'Same Day'],
      dtype=object)
```

```
In [9]:  data['Segment'].unique()
```

```
Out[9]:  array(['Consumer', 'Corporate', 'Home Office'], dtype=object)
```

```
In [10]:  data['Country'].unique()
```

```
Out[10]:  array(['United States'], dtype=object)
```

```
In [11]: data['City'].unique()
```

```
Out[11]: array(['Henderson', 'Los Angeles', 'Fort Lauderdale', 'Concord',
                'Seattle', 'Fort Worth', 'Madison', 'West Jordan', 'San Francisco',
                'Fremont', 'Philadelphia', 'Orem', 'Houston', 'Richardson',
                'Naperville', 'Melbourne', 'Eagan', 'Westland', 'Dover',
                'New Albany', 'New York City', 'Troy', 'Chicago', 'Gilbert',
                'Springfield', 'Jackson', 'Memphis', 'Decatur', 'Durham',
                'Columbia', 'Rochester', 'Minneapolis', 'Portland', 'Saint Paul',
                'Aurora', 'Charlotte', 'Orland Park', 'Urbandale', 'Columbus',
                'Bristol', 'Wilmington', 'Bloomington', 'Phoenix', 'Roseville',
                'Independence', 'Pasadena', 'Newark', 'Franklin', 'Scottsdale',
                'San Jose', 'Edmond', 'Carlsbad', 'San Antonio', 'Monroe',
                'Fairfield', 'Grand Prairie', 'Redlands', 'Hamilton', 'Westfield',
                'Akron', 'Denver', 'Dallas', 'Whittier', 'Saginaw', 'Medina',
                'Dublin', 'Detroit', 'Tampa', 'Santa Clara', 'Lakeville',
                'San Diego', 'Brentwood', 'Chapel Hill', 'Morristown',
                'Cincinnati', 'Inglewood', 'Tamarac', 'Colorado Springs',
                'Belleville', 'Taylor', 'Lakewood', 'Arlington', 'Arvada',
                'Hackensack', 'Saint Petersburg', 'Long Beach', 'Hesperia',
                'Murfreesboro', 'Layton', 'Austin', 'Lowell', 'Manchester',
                'Harlingen', 'Tucson', 'Quincy', 'Pembroke Pines', 'Des Moines',
                'Peoria', 'Las Vegas', 'Warwick', 'Miami', 'Huntington Beach',
                'Richmond', 'Louisville', 'Lawrence', 'Canton', 'New Rochelle',
                'Gastonia', 'Jacksonville', 'Auburn', 'Norman', 'Park Ridge',
                'Amarillo', 'Lindenhurst', 'Huntsville', 'Fayetteville',
                'Costa Mesa', 'Parker', 'Atlanta', 'Gladstone', 'Great Falls',
                'Lakeland', 'Montgomery', 'Mesa', 'Green Bay', 'Anaheim',
                'Marysville', 'Salem', 'Laredo', 'Grove City', 'Dearborn',
                'Warner Robins', 'Vallejo', 'Mission Viejo', 'Rochester Hills',
                'Plainfield', 'Sierra Vista', 'Vancouver', 'Cleveland', 'Tyler',
                'Burlington', 'Waynesboro', 'Chester', 'Cary', 'Palm Coast',
                'Mount Vernon', 'Hialeah', 'Oceanside', 'Evanston', 'Trenton',
                'Cottage Grove', 'Bossier City', 'Lancaster', 'Asheville',
                'Lake Elsinore', 'Omaha', 'Edmonds', 'Santa Ana', 'Milwaukee',
                'Florence', 'Lorain', 'Linden', 'Salinas', 'New Brunswick',
                'Garland', 'Norwich', 'Alexandria', 'Toledo', 'Farmington',
                'Riverside', 'Torrance', 'Round Rock', 'Boca Raton',
                'Virginia Beach', 'Murrieta', 'Olympia', 'Washington',
                'Jefferson City', 'Saint Peters', 'Rockford', 'Brownsville',
                'Yonkers', 'Oakland', 'Clinton', 'Encinitas', 'Roswell',
                'Jonesboro', 'Antioch', 'Homestead', 'La Porte', 'Lansing',
                'Cuyahoga Falls', 'Reno', 'Harrisonburg', 'Escondido', 'Royal Oak',
                'Rockville', 'Coral Springs', 'Buffalo', 'Boynton Beach',
                'Gulfport', 'Fresno', 'Greenville', 'Macon', 'Cedar Rapids',
                'Providence', 'Pueblo', 'Deltona', 'Murray', 'Middletown',
                'Freeport', 'Pico Rivera', 'Provo', 'Pleasant Grove', 'Smyrna',
                'Parma', 'Mobile', 'New Bedford', 'Irving', 'Vineland', 'Glendale',
                'Niagara Falls', 'Thomasville', 'Westminster', 'Coppell', 'Pomona',
                'North Las Vegas', 'Allentown', 'Tempe', 'Laguna Niguel',
                'Bridgeton', 'Everett', 'Watertown', 'Appleton', 'Bellevue',
                'Allen', 'El Paso', 'Grapevine', 'Carrollton', 'Kent', 'Lafayette',
                'Tigard', 'Skokie', 'Plano', 'Suffolk', 'Indianapolis', 'Bayonne',
                'Greensboro', 'Baltimore', 'Kenosha', 'Olathe', 'Tulsa', 'Redmond',
                'Raleigh', 'Muskogee', 'Meriden', 'Bowling Green', 'South Bend',
                'Spokane', 'Keller', 'Port Orange', 'Medford', 'Charlottesville',
                'Missoula', 'Apopka', 'Reading', 'Broomfield', 'Paterson',
                'Oklahoma City', 'Chesapeake', 'Lubbock', 'Johnson City',
                'San Bernardino', 'Leominster', 'Bozeman', 'Perth Amboy',
                'Ontario', 'Rancho Cucamonga', 'Moorhead', 'Mesquite', 'Stockton',
                'Ormond Beach', 'Sunnyvale', 'York', 'College Station',
                'Saint Louis', 'Manteca', 'San Angelo', 'Salt Lake City',
                'Knoxville', 'Little Rock', 'Lincoln Park', 'Marion', 'Littleton',
                'Bangor', 'Southaven', 'New Castle', 'Midland', 'Sioux Falls',
                'Fort Collins', 'Clarksville', 'Sacramento', 'Thousand Oaks',
                'Malden', 'Holyoke', 'Albuquerque', 'Sparks', 'Coachella',
                'Elmhurst', 'Passaic', 'North Charleston', 'Newport News',
                'Jamestown', 'Mishawaka', 'La Quinta', 'Tallahassee', 'Nashville',
                'Bellingham', 'Woodstock', 'Haltom City', 'Wheeling',
                'Summerville', 'Hot Springs', 'Englewood', 'Las Cruces', 'Hoover',
                'Frisco', 'Vacaville', 'Waukesha', 'Bakersfield', 'Pompano Beach',
                'Corpus Christi', 'Redondo Beach', 'Orlando', 'Orange',
                'Lake Charles', 'Highland Park', 'Hempstead', 'Noblesville',
                'Apple Valley', 'Mount Pleasant', 'Sterling Heights', 'Eau Claire',
                'Pharr', 'Billings', 'Gresham', 'Chattanooga', 'Meridian',
                'Bolingbrook', 'Maple Grove', 'Woodland', 'Missouri City',
                'Pearland', 'San Mateo', 'Grand Rapids', 'Visalia',
                'Overland Park', 'Temecula', 'Yucaipa', 'Revere', 'Conroe',
                'Tinley Park', 'Dubuque', 'Dearborn Heights', 'Santa Fe',
                'Hickory', 'Carol Stream', 'Saint Cloud', 'North Miami',
                'Plantation', 'Port Saint Lucie', 'Rock Hill', 'Odessa',
                'West Allis', 'Chula Vista', 'Manhattan', 'Altoona', 'Thornton',
                'Champaign', 'Texarkana', 'Edinburg', 'Baytown', 'Greenwood',
                'Woonsocket', 'Superior', 'Bedford', 'Covington', 'Broken Arrow',
                'Miramar', 'Hollywood', 'Deer Park', 'Wichita', 'Mcallen',
                'Iowa City', 'Boise', 'Cranston', 'Port Arthur', 'Citrus Heights',
                'The Colony', 'Daytona Beach', 'Bullhead City', 'Portage', 'Fargo',
                'Elkhart', 'San Gabriel', 'Margate', 'Sandy Springs', 'Mentor',
                'Lawton', 'Hampton', 'Rome', 'La Crosse', 'Lewiston',
                'Hattiesburg', 'Danville', 'Logan', 'Waterbury', 'Athens',
                'Avondale', 'Marietta', 'Yuma', 'Wausau', 'Pasco', 'Oak Park',
                'Pensacola', 'League City', 'Gaithersburg', 'Lehi', 'Tuscaloosa',
                'Moreno Valley', 'Georgetown', 'Loveland', 'Chandler', 'Helena',
                'Kirkwood', 'Waco', 'Frankfort', 'Bethlehem', 'Grand Island',
                'Woodbury', 'Rogers', 'Clovis', 'Jupiter', 'Santa Barbara',
                'Cedar Hill', 'Norfolk', 'Draper', 'Ann Arbor', 'La Mesa',
                'Pocatello', 'Holland', 'Milford', 'Buffalo Grove', 'Lake Forest',
                'Redding', 'Chico', 'Utica', 'Conway', 'Cheyenne', 'Owensboro',
                'Caldwell', 'Kenner', 'Nashua', 'Bartlett', 'Redwood City',
                'Lebanon', 'Santa Maria', 'Des Plaines', 'Longview',
                'Hendersonville', 'Waterloo', 'Cambridge', 'Palatine', 'Beverly',
                'Eugene', 'Oxnard', 'Renton', 'Glenview', 'Delray Beach',
                'Commerce City', 'Texas City', 'Wilson', 'Rio Rancho', 'Goldsboro',
                'Montebello', 'El Cajon', 'Beaumont', 'West Palm Beach', 'Abilene',
                'Normal', 'Saint Charles', 'Camarillo', 'Hillsboro', 'Burbank',
                'Modesto', 'Garden City', 'Atlantic City', 'Longmont', 'Davis',
                'Morgan Hill', 'Clifton', 'Sheboygan', 'East Point', 'Rapid City',
                'Andover', 'Kissimmee', 'Shelton', 'Danbury', 'Sanford',
                'San Marcos', 'Greeley', 'Mansfield', 'Elyria', 'Twin Falls',
                'Coral Gables', 'Romeoville', 'Marlborough', 'Laurel', 'Bryan',
                'Pine Bluff', 'Aberdeen', 'Hagerstown', 'East Orange',
                'Arlington Heights', 'Oswego', 'Coon Rapids', 'San Clemente',
                'San Luis Obispo', 'Springdale', 'Lodi', 'Mason'], dtype=object)
```

```
In [12]: data['State'].unique()
```

```
Out[12]: array(['Kentucky', 'California', 'Florida', 'North Carolina',
                'Washington', 'Texas', 'Wisconsin', 'Utah', 'Nebraska',
                'Pennsylvania', 'Illinois', 'Minnesota', 'Michigan', 'Delaware',
                'Indiana', 'New York', 'Arizona', 'Virginia', 'Tennessee',
                'Alabama', 'South Carolina', 'Oregon', 'Colorado', 'Iowa', 'Ohio',
                'Missouri', 'Oklahoma', 'New Mexico', 'Louisiana', 'Connecticut',
                'New Jersey', 'Massachusetts', 'Georgia', 'Nevada', 'Rhode Island',
                'Mississippi', 'Arkansas', 'Montana', 'New Hampshire', 'Maryland',
                'District of Columbia', 'Kansas', 'Vermont', 'Maine',
                'South Dakota', 'Idaho', 'North Dakota', 'Wyoming',
                'West Virginia'], dtype=object)
```

```
In [13]: data['Postal Code'].unique()
```

```
Out[13]: array([42420, 90036, 33311, 90032, 28027, 98103, 76106, 53711, 84084,
                94109, 68025, 19140, 84057, 90049, 77095, 75080, 77041, 60540,
                32935, 55122, 48185, 19901, 47150, 10024, 12180, 90004, 60610,
                85234, 22153, 10009, 49201, 38109, 77070, 35601, 94122, 27707,
                60623, 29203, 55901, 55407, 97206, 55106, 80013, 28205, 60462,
                10035, 50322, 43229, 37620, 19805, 61701, 85023, 95661, 64055,
                91104, 43055, 53132, 85254, 95123, 98105, 98115, 73034, 90045,
                19134, 88220, 78207, 77036, 62521, 71203,  6824, 75051, 92374,
                45011,  7090, 19120, 44312, 80219, 75220, 37064, 90604, 48601,
                44256, 43017, 48227, 38401, 33614, 95051, 55044, 92037, 77506,
                94513, 27514,  7960, 45231, 94110, 90301, 33319, 80906,  7109,
                48180,  8701, 22204, 80004,  7601, 33710, 19143, 90805, 92345,
                37130, 84041, 78745,  1852, 31907,  6040, 78550, 85705, 62301,
                 2038, 33024, 98198, 61604, 89115,  2886, 33180, 28403, 92646,
                40475, 80027,  1841, 39212, 48187, 10801, 28052, 32216, 47201,
                13021, 73071, 94521, 60068, 79109, 11757, 90008, 92024, 77340,
                14609, 72701, 92627, 80134, 30318, 64118, 59405, 48234, 33801,
                36116, 85204, 60653, 54302, 45503, 92804, 98270, 97301, 78041,
                75217, 43123, 10011, 48126, 31088, 94591, 92691, 48307,  7060,
                85635, 98661, 60505, 76017, 40214, 75081, 44105, 75701, 27217,
                22980, 19013, 27511, 32137, 10550, 48205, 33012, 11572, 92105,
                60201, 48183, 55016, 71111, 50315, 93534, 23223, 28806, 92530,
                68104, 98026, 92704, 53209, 41042, 44052,  7036, 93905,  8901,
                17602,  3301, 21044, 75043,  6360, 22304, 43615, 87401, 92503,
                90503, 78664, 92054, 33433, 23464, 92563, 28540, 52601, 98502,
                20016, 65109, 63376, 61107, 33142, 78521, 10701, 94601, 28110,
                20735, 30076, 72401, 47374, 94509, 33030, 46350, 48911, 44221,
                89502, 22801, 92025, 48073, 20852, 33065, 14215, 33437, 39503,
                93727, 27834, 11561, 35630, 31204, 52402,  2908, 81001, 94533,
                32725, 42071,  6457, 11520, 90660, 84604, 84062, 30080, 24153,
                44134, 36608,  2740, 75061,  8360, 85301, 14304, 27360, 92683,
                38301, 75019, 91767, 89031, 18103, 19711, 85281, 92677,  8302,
                 2149, 13601, 54915, 98006, 75002, 79907, 76051, 75007, 37167,
                98031, 70506, 97224, 60076, 75023, 23434, 46203,  7002, 28314,
                27405, 21215, 53142, 66062, 98002, 74133, 97756, 27604, 74403,
                 6450, 42104, 46614,  6010, 89015, 99207, 76248, 45014, 32127,
                97504, 22901, 59801, 33178, 29501, 97477, 32712, 19601, 80020,
                65807,  7501, 73120, 23320, 79424, 65203, 37604, 36830, 92404,
                 1453, 59715, 85345, 44107,  8861, 91761, 91730, 56560, 75150,
                95207, 32174, 94086,  3820, 17403, 77840, 63116,  2169, 95336,
                44240, 76903, 84106, 35810, 37918, 72209, 48146, 43302, 80122,
                 5408,  4401, 38671, 47362, 48640, 57103, 80525, 47905, 37042,
                95823, 91360,  2148,  1040, 87105, 89431, 92236, 60126,  7055,
                29406, 23602, 14701, 46544, 43402, 92253, 32303, 37211, 98226,
                60098, 76117, 60090, 29483, 71901, 80112, 43130, 88001, 35244,
                75034, 95687, 84107, 53186, 93309, 33068, 45373, 78415, 90278,
                32839,  7050, 70601, 60035, 11550, 46060, 55124, 29464, 48310,
                54703, 78577, 59102, 97030, 37421, 83642, 92307, 60440, 55369,
                95695, 77489, 77581, 94403, 49505, 93277, 66212, 92592, 92399,
                 2151, 77301, 60477, 52001, 48127, 87505, 28601, 60188, 56301,
                33161, 46226, 33317, 34952, 29730, 79762, 53214, 91911, 66502,
                16602, 80229, 61821, 47401, 71854, 78539, 77520, 46142, 90712,
                 2895, 54880, 76021, 98042, 74012, 33023, 33021, 77536, 67212,
                78501, 52240, 83704,  2920, 61032, 77642, 95610, 75056, 98052,
                32114, 86442, 46368, 58103, 46514, 91776, 33063, 30328, 44060,
                73505, 23666, 13440, 54601, 83501, 39401, 94526, 48858, 84321,
                 6708, 30605,  4240, 61832, 85323, 30062, 85364, 54401, 99301,
                60302, 32503, 77573, 20877, 84043, 35401, 92553, 40324, 80538,
                85224, 59601, 63122, 76706, 48066, 60423, 18018, 55113, 68801,
                55125, 48237, 72756, 88101, 33458, 93101, 75104, 68701, 84020,
                48104, 91941, 83201, 49423,  6460, 60089, 92630, 96003, 95928,
                13501, 72032, 82001, 42301, 83605, 70065,  3060, 38134, 94061,
                37087, 93454, 60016, 98632, 37075, 50701,  2138, 60067,  1915,
                97405, 93030, 98059, 60025, 33445, 80022, 77590, 27893, 87124,
                27534, 98208, 90640, 92020, 77705, 33407, 79605, 61761, 63301,
                60174, 93010, 97123, 91505, 95351, 67846,  8401, 80501, 95616,
                26003, 95037,  7011, 53081, 30344, 57701,  1810, 34741,  6484,
                 6810, 52302, 32771, 78666, 80634, 76063, 44035, 83301, 33134,
                60441,  1752, 20707, 77803, 71603, 57401, 21740,  7017, 60004,
                60543, 55433, 92672, 94568, 93405, 72762, 95240, 77571, 45040,
                30188], dtype=int64)
```

In [14]:
```
data['Region'].unique()
```

Out[14]:
```
array(['South', 'West', 'Central', 'East'], dtype=object)
```

In [15]:
```
data['Category'].unique()
```

Out[15]:
```
array(['Furniture', 'Office Supplies', 'Technology'], dtype=object)
```

In [16]:
```
data['Sub-Category'].unique()
```

Out[16]:
```
array(['Bookcases', 'Chairs', 'Labels', 'Tables', 'Storage',
       'Furnishings', 'Art', 'Phones', 'Binders', 'Appliances', 'Paper',
       'Accessories', 'Envelopes', 'Fasteners', 'Supplies', 'Machines',
       'Copiers'], dtype=object)
```

In [17]:
```
data['Sales'].unique()
```

Out[17]:
```
array([261.96 , 731.94 ,  14.62 , ..., 437.472,  97.98 , 243.16 ])
```

In [18]:
```
data['Quantity'].unique()
```

Out[18]:
```
array([ 2,  3,  5,  7,  4,  6,  9,  1,  8, 14, 11, 13, 10, 12],
      dtype=int64)
```

In [19]:
```
data['Discount'].unique()
```

Out[19]:
```
array([0.  , 0.45, 0.2 , 0.8 , 0.3 , 0.5 , 0.7 , 0.6 , 0.32, 0.1 , 0.4 ,
       0.15])
```

In [20]:
```
data.columns
```

Out[20]:
```
Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Postal Code',
       'Region', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount',
       'Profit'],
      dtype='object')
```

Cleaning the Data

In [21]:

```
# Finding all the NULL Values

data.isnull().sum()
```

Out[21]:

```
Ship Mode       0
Segment         0
Country         0
City            0
State           0
Postal Code     0
Region          0
Category        0
Sub-Category    0
Sales           0
Quantity        0
Discount        0
Profit          0
dtype: int64
```

Relationship Analysis

In [22]:

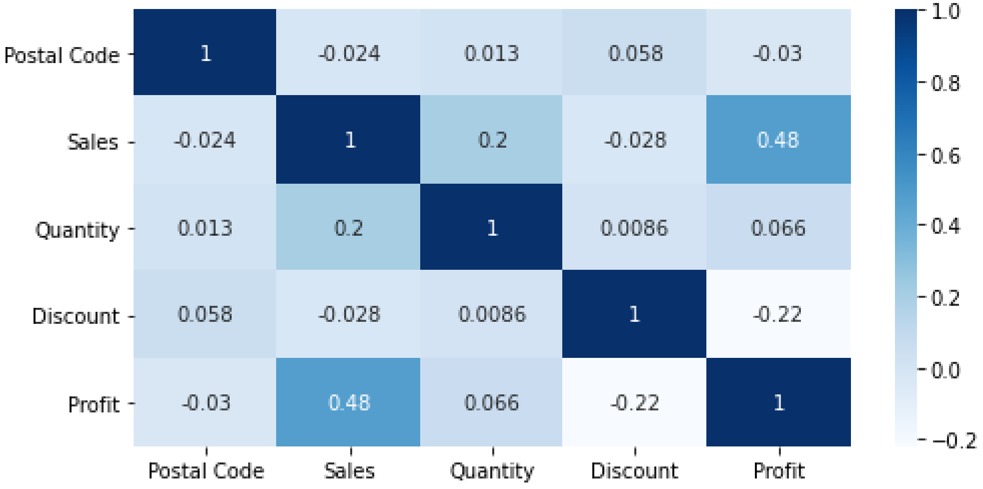
```
corelation = data.corr()
corelation
```

Out[22]:

	Postal Code	Sales	Quantity	Discount	Profit
Postal Code	1.000000	-0.023854	0.012761	0.058443	-0.029961
Sales	-0.023854	1.000000	0.200795	-0.028190	0.479064
Quantity	0.012761	0.200795	1.000000	0.008623	0.066253
Discount	0.058443	-0.028190	0.008623	1.000000	-0.219487
Profit	-0.029961	0.479064	0.066253	-0.219487	1.000000

In [23]:
```
import matplotlib.pyplot as plt
plt.figure(figsize=(8,4))
sns.heatmap(corelation,cmap='Blues',annot=True)
```

Out[23]:
```
<AxesSubplot:>
```


Postal Code
Sales
Quantity
Discount
Profit
Postal Code
Sales
Quantity
Discount
Profit