```
import numpy as np
import scipy as sp
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data = pd.read_csv('/content/Product sale history.csv')
print(data.shape)
data.head(10)
```

```
(15000, 8)
```

	week	sales_method	customer_id	nb_sold	revenue	years_as_customer	nb_site_visits	state
0	2	Email	2e72d641-95ac-497b-bbf8-4861764a7097	10	NaN	0	24	Arizona
1	6	Email + Call	3998a98d-70f5-44f7-942e-789bb8ad2fe7	15	225.47	1	28	Kansas
2	5	Call	d1de9884-8059-4065-b10f-86eef57e4a44	11	52.55	6	26	Wisconsin
3	4	Email	78aa75a4-ffeb-4817-b1d0-2f030783c5d7	11	NaN	3	25	Indiana
4	3	Email	10e6d446-10a5-42e5-8210-1b5438f70922	9	90.49	0	28	Illinois
5	6	Call	6489e678-40f2-4fed-a48e-d0dff9c09205	13	65.01	10	24	Mississippi
6	4	Email	eb6bd5f1-f115-4e4b-80a6-5e67fcfbfb94	11	113.38	9	28	Georgia
7	1	Email	047df079-071b-4380-9012-2bfe9bce45d5	10	99.94	1	22	Oklahoma
8	5	Email	771586bd-7b64-40be-87df-afe884d2af9e	11	108.34	10	31	Massachusetts
9	5	Call	56491dae-bbe7-49f0-a651-b823a01103d8	11	53.82	7	23	Missouri

Next steps: Generate code with data View recommended plots

```
data.dtypes
```

```
week                 int64
sales_method        object
customer_id         object
nb_sold              int64
revenue            float64
years_as_customer    int64
nb_site_visits       int64
state               object
dtype: object
```

```
data['sales_method'] = data['sales_method'].str.lower()
data['sales_method'].replace({'em + call':'email + call'},inplace=True)
data['sales_method'].value_counts()
```

```
sales_method
email           7466
call            4962
email + call    2572
Name: count, dtype: int64
```

```
data[['week', 'nb_sold', 'sales_method']] = data[['week', 'nb_sold', 'sales_method']].astype('category')
```

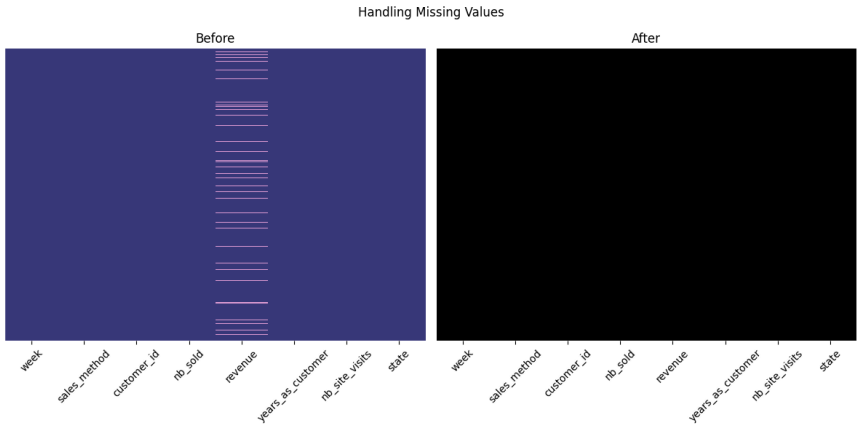
```
# Drop duplicates
print(data.shape)
data.drop_duplicates(inplace= True)
data.shape
```

```
(15000, 8)
(15000, 8)
```

```
data.isna().sum().sort_values()
```

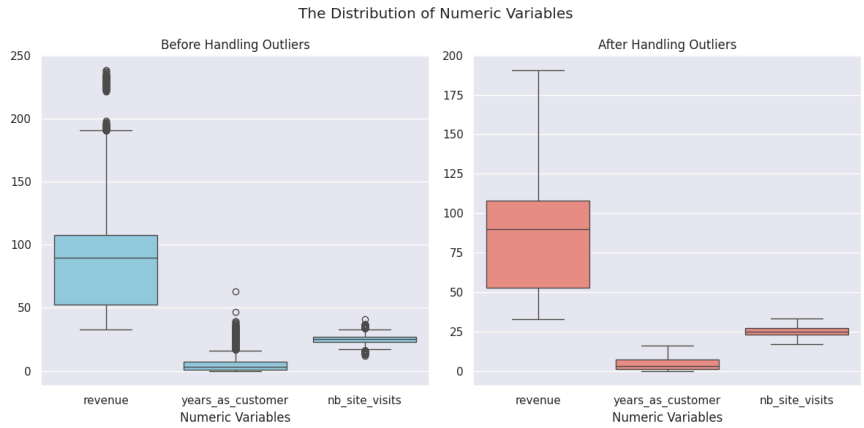
```
week                 0
sales_method         0
customer_id          0
nb_sold              0
years_as_customer    0
nb_site_visits       0
state                0
revenue           1074
dtype: int64
```

```
data_orig = data.copy()
data=data.sort_values(by='nb_sold', ascending=True)
data['revenue'].fillna(method='bfill', inplace=True)

# To check Missing Values
fig, axes = plt.subplots(1, 2, figsize=(12, 6))
sns.heatmap(data_orig.isnull(), cbar=False, yticklabels=False, cmap="tab20b",ax=axes[0])
sns.heatmap(data.isnull(), cbar=False, yticklabels=False, cmap="inferno", ax=axes[1])
axes[0].set_title("Before")
axes[1].set_title("After")
fig.suptitle("Handling Missing Values")
axes[0].set_xticklabels(axes[0].get_xticklabels(), rotation=45)
axes[1].set_xticklabels(axes[1].get_xticklabels(), rotation=45)
plt.tight_layout()
```

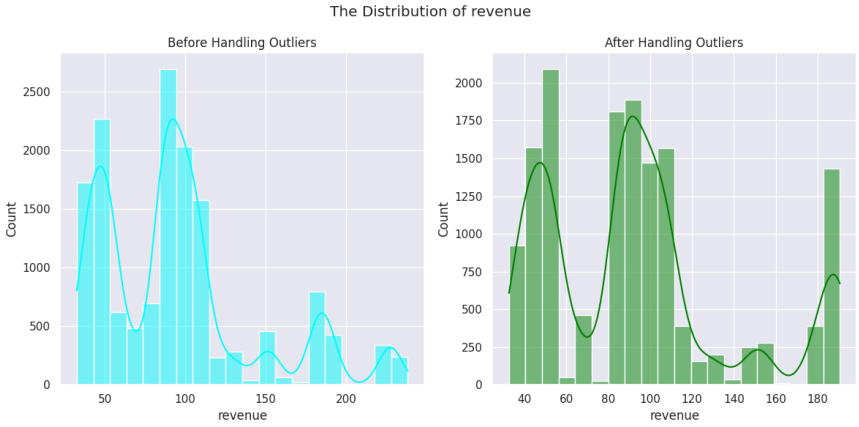


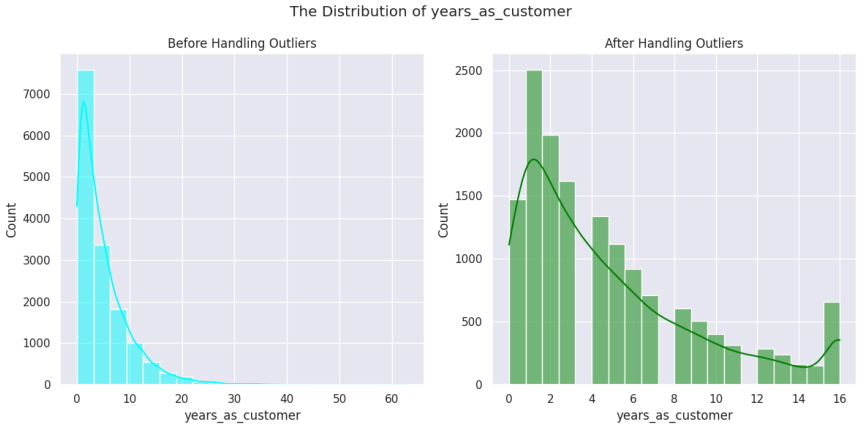

```
data_bef =data.copy()
# Handling Outliers
def handle_outlier(df,col):
    Q3 = df[col].quantile(0.75)
    Q1 = df[col].quantile(0.25)
    IQR = Q3 - Q1
    upper = Q3 + (1.5 * IQR)
    lower = Q1 - (1.5 * IQR)
    df[col] = np.where(df[col] < lower, lower, df[col])
    df[col] = np.where(df[col] > upper, upper, df[col])

data_num = data.select_dtypes(include = ["float64", "int64"])

for column in data_num.columns:
    handle_outlier(data, column)
```

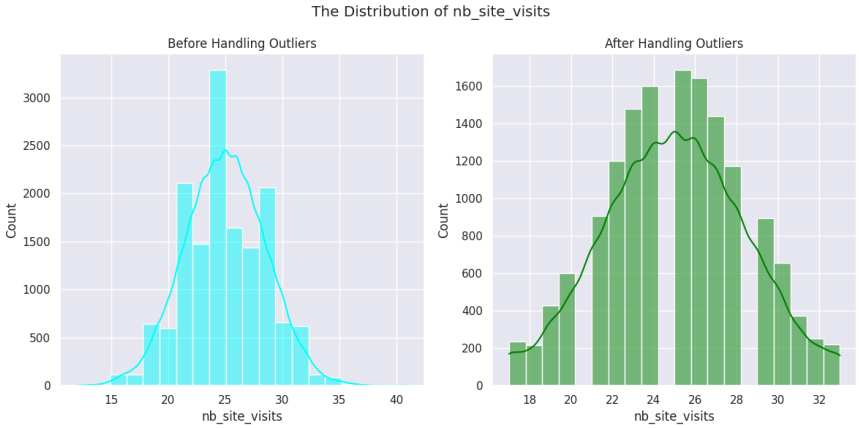
```
import seaborn as sns
import matplotlib.pyplot as plt



sns.set(style="darkgrid")
fig, axes = plt.subplots(1, 2, figsize=(12, 6))


sns.boxplot(data_bef, ax=axes[0], color='skyblue')

sns.boxplot(data, ax=axes[1], color='salmon')

axes[0].set_title("Before Handling Outliers")
axes[1].set_title("After Handling Outliers")
fig.suptitle("The Distribution of Numeric Variables")
axes[0].set_xlabel("Numeric Variables")
axes[1].set_xlabel("Numeric Variables")

plt.tight_layout()
plt.show()
```




```
data_num = data.select_dtypes(include = ["float64", "int64"])
for col in data_num.columns:
  fig, axes = plt.subplots(1, 2, figsize=(12, 6))
  sns.histplot(data_bef[col], kde=True, color='cyan',ax=axes[0], bins=20)
  sns.histplot(data[col], kde=True, color='green',ax=axes[1], bins=20)
  axes[0].set_title("Before Handling Outliers")
  axes[1].set_title("After Handling Outliers")
  fig.suptitle('The Distribution of '+col)
  plt.tight_layout()
  plt.show()
```


The Distribution of revenue
Before Handling Outliers
After Handling Outliers
Count
revenue


The Distribution of years_as_customer
Before Handling Outliers
After Handling Outliers
Count
years_as_customer

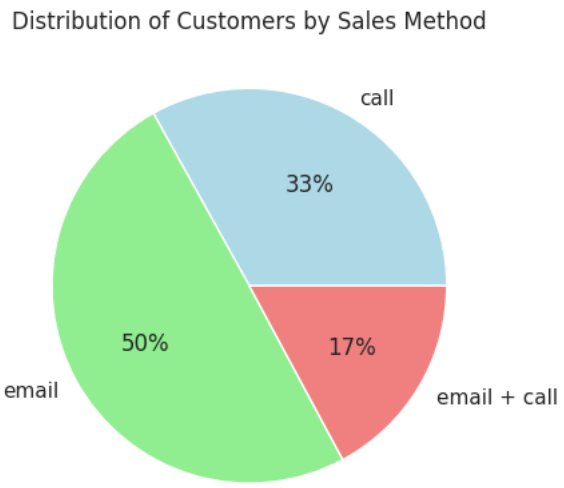

The Distribution of nb_site_visits
Before Handling Outliers
After Handling Outliers
Count
nb_site_visits

```
data[['years_as_customer', 'nb_site_visits']] = data[['years_as_customer', 'nb_site_visits']].astype('int64')
```

```
data.sample(n=10)
```

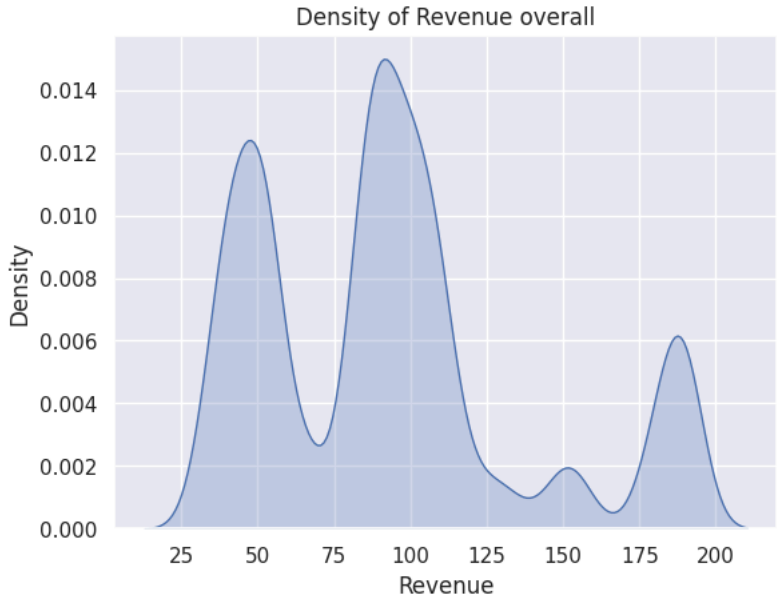
	week	sales_method	customer_id	nb_sold	revenue	years_as_customer	nb_site
14583	1	call	17baf754-a343-4d87-bb56-1ba896cb2a58	7	35.500	3	
5879	4	email	395849d7-e98d-4894-a6aa-beba256a5484	11	107.390	16	
5474	2	email	35146781-ec28-4e20-8e55-0acecc4a7b49	9	93.840	3	
3464	3	email	bd039f5d-ea59-49b7-8f3f-616125d56539	9	91.080	1	

```
import matplotlib.pyplot as plt
Customers = data.groupby('sales_method').size()
colors = ['lightblue', 'lightgreen', 'lightcoral', 'lightskyblue', 'lightpink']
Customers.plot(kind='pie', autopct='%1.0f%%', colors=colors)
plt.title('Distribution of Customers by Sales Method')
plt.show()
```

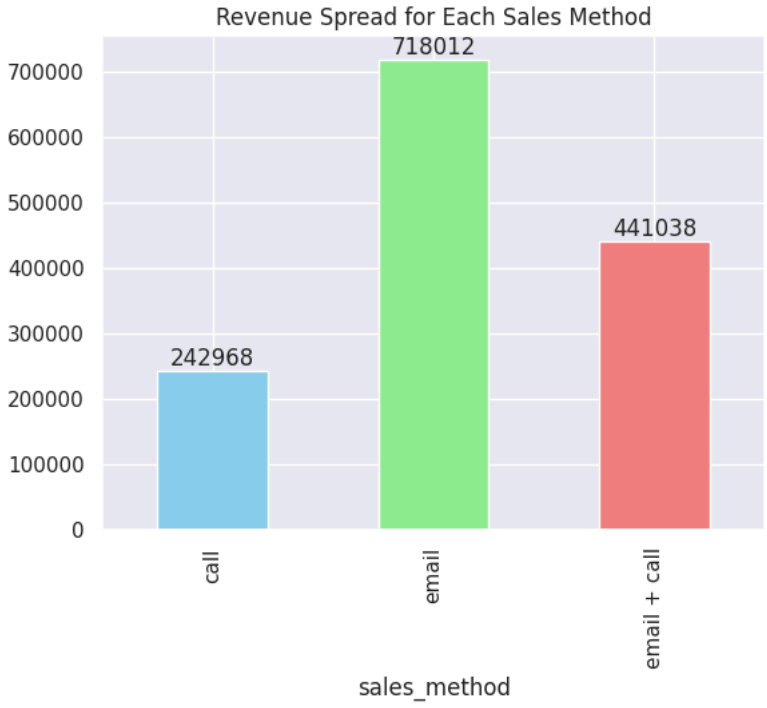



```
sns.kdeplot(data['revenue'], shade=True)
plt.xlabel('Revenue')
plt.ylabel('Density')
plt.title('Density of Revenue overall')
plt.show()
```

```
<ipython-input-42-55f0a268ac69>:1: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data['revenue'], shade=True)
```




```
import matplotlib.pyplot as plt

Mean_revenue = data.groupby('sales_method')['revenue'].sum()

colors = ['skyblue', 'lightgreen', 'lightcoral', 'lightskyblue', 'lightpink']

cx = Mean_revenue.plot.bar(color=colors)
cx.bar_label(cx.containers[0], label_type='edge')
plt.title('Revenue Spread for Each Sales Method')
plt.show()
```




```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_theme(style="whitegrid")
custom_palette = ["#FF5733", "#FFC300", "#DAF7A6", "#9AECDB", "#A3C4BC"]
sns.barplot(data=data, x="sales_method", y="revenue", hue="week", palette=custom_palette)
plt.title('The revenue of all sales methods increased each week')
plt.show()
```