In [2]:
```
greetings = "Assalam-o-Alaikum!"
print(greetings)
```

```
Assalam-o-Alaikum!
```

Import Libraries

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Import Dataset

In [4]:
```
df = pd.read_excel("Sample - Superstore.xls")
df.head(5)
```

Out[4]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Category	Cate
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-BO-10001798	Furniture	Book
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-CH-10000454	Furniture	C
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-LA-10000240	Office Supplies	L
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	FUR-TA-10000577	Furniture	T
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	OFF-ST-10000760	Office Supplies	St

5 rows × 21 columns

Exploratory Data Analysis

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 non-null   datetime64[ns]
 3   Ship Date      9994 non-null   datetime64[ns]
 4   Ship Mode      9994 non-null   object
 5   Customer ID    9994 non-null   object
 6   Customer Name  9994 non-null   object
 7   Segment        9994 non-null   object
 8   Country        9994 non-null   object
 9   City           9994 non-null   object
 10  State          9994 non-null   object
 11  Postal Code    9994 non-null   int64
 12  Region         9994 non-null   object
 13  Product ID     9994 non-null   object
 14  Category       9994 non-null   object
 15  Sub-Category   9994 non-null   object
 16  Product Name   9994 non-null   object
 17  Sales          9994 non-null   float64
 18  Quantity       9994 non-null   int64
 19  Discount       9994 non-null   float64
 20  Profit         9994 non-null   float64
dtypes: datetime64[ns](2), float64(3), int64(3), object(13)
memory usage: 1.6+ MB
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
Row ID           0
Order ID         0
Order Date       0
Ship Date        0
Ship Mode        0
Customer ID      0
Customer Name    0
Segment          0
Country          0
City             0
State            0
Postal Code      0
Region           0
Product ID       0
Category         0
Sub-Category     0
Product Name     0
Sales            0
Quantity         0
Discount         0
Profit           0
dtype: int64
```

In [7]:
```
df.describe()
```

Out[7]:

	Row ID	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	4997.500000	55190.379428	229.858001	3.789574	0.156203	28.656896
std	2885.163629	32063.693350	623.245101	2.225110	0.206452	234.260108
min	1.000000	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	2499.250000	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	4997.500000	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	7495.750000	90008.000000	209.940000	5.000000	0.200000	29.364000
max	9994.000000	99301.000000	22638.480000	14.000000	0.800000	8399.976000

In [8]:
```
df = df.drop(columns = ["Row ID", "Order ID", "Customer ID", "Postal Code", "Product ID"])
df.head()
```

Out[8]:

	Order Date	Ship Date	Ship Mode	Customer Name	Segment	Country	City	State	Region	Category	Sub-Category	Product Name	Sales	Quantity
0	2016-11-08	2016-11-11	Second Class	Claire Gute	Consumer	United States	Henderson	Kentucky	South	Furniture	Bookcases	Bush Somerset Collection Bookcase	261.9600	2
1	2016-11-08	2016-11-11	Second Class	Claire Gute	Consumer	United States	Henderson	Kentucky	South	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.9400	3
2	2016-06-12	2016-06-16	Second Class	Darrin Van Huff	Corporate	United States	Los Angeles	California	West	Office Supplies	Labels	Self-Adhesive Address Labels for Typewriters b...	14.6200	2
3	2015-10-11	2015-10-18	Standard Class	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	South	Furniture	Tables	Bretford CR4500 Series Slim Rectangular Table	957.5775	5
4	2015-10-11	2015-10-18	Standard Class	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	South	Office Supplies	Storage	Eldon Fold 'N Roll Cart System	22.3680	2

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 16 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Order Date     9994 non-null   datetime64[ns]
 1   Ship Date      9994 non-null   datetime64[ns]
 2   Ship Mode      9994 non-null   object
 3   Customer Name  9994 non-null   object
 4   Segment        9994 non-null   object
 5   Country        9994 non-null   object
 6   City           9994 non-null   object
 7   State          9994 non-null   object
 8   Region         9994 non-null   object
 9   Category       9994 non-null   object
 10  Sub-Category   9994 non-null   object
 11  Product Name   9994 non-null   object
 12  Sales          9994 non-null   float64
 13  Quantity       9994 non-null   int64
 14  Discount       9994 non-null   float64
 15  Profit         9994 non-null   float64
dtypes: datetime64[ns](2), float64(3), int64(1), object(10)
memory usage: 1.2+ MB
```

Data Insights

Orders By Ship Mode

In [10]:
```
sm = df["Ship Mode"].value_counts()
sm
```

Out[10]:
```
Standard Class    5968
Second Class      1945
First Class       1538
Same Day           543
Name: Ship Mode, dtype: int64
```

In [11]:
```
sm = sm.to_frame().reset_index()
sm
```

Out[11]:

	index	Ship Mode
0	Standard Class	5968
1	Second Class	1945
2	First Class	1538
3	Same Day	543

In [12]:
```
sm.columns = ["Ship Mode", "Orders"]
sm
```

Out[12]:

	Ship Mode	Orders
0	Standard Class	5968
1	Second Class	1945
2	First Class	1538
3	Same Day	543

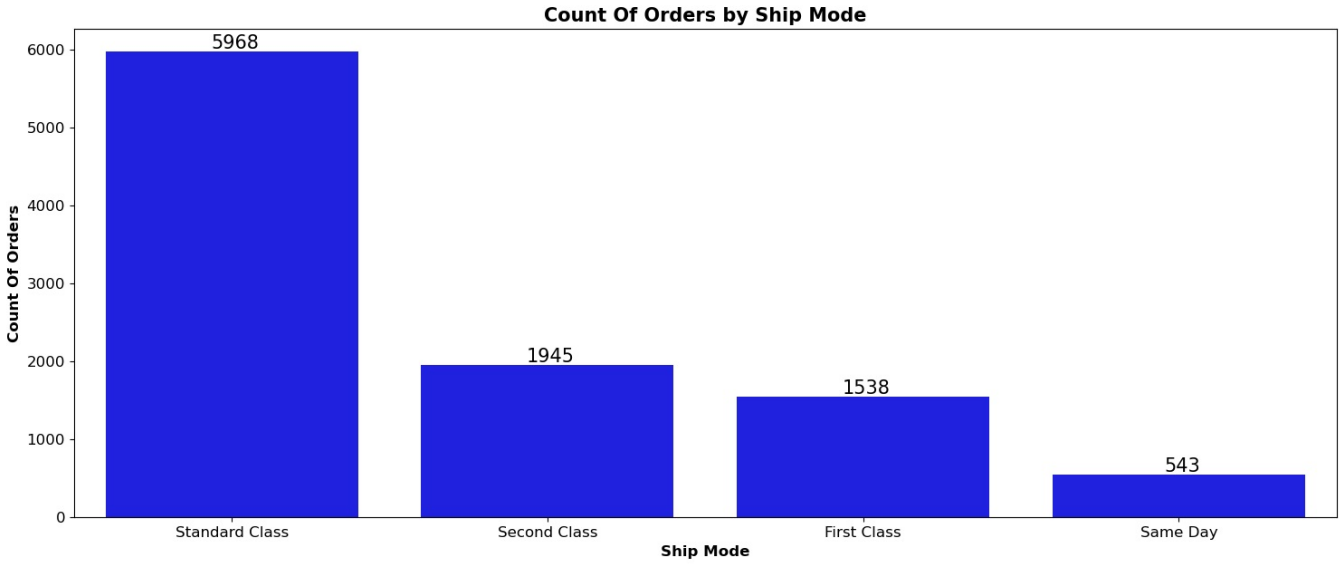
In [13]:
```
sm["Percentage"] = ((sm["Orders"] / sm["Orders"].sum()) * 100).round(1)
sm
```

Out[13]:

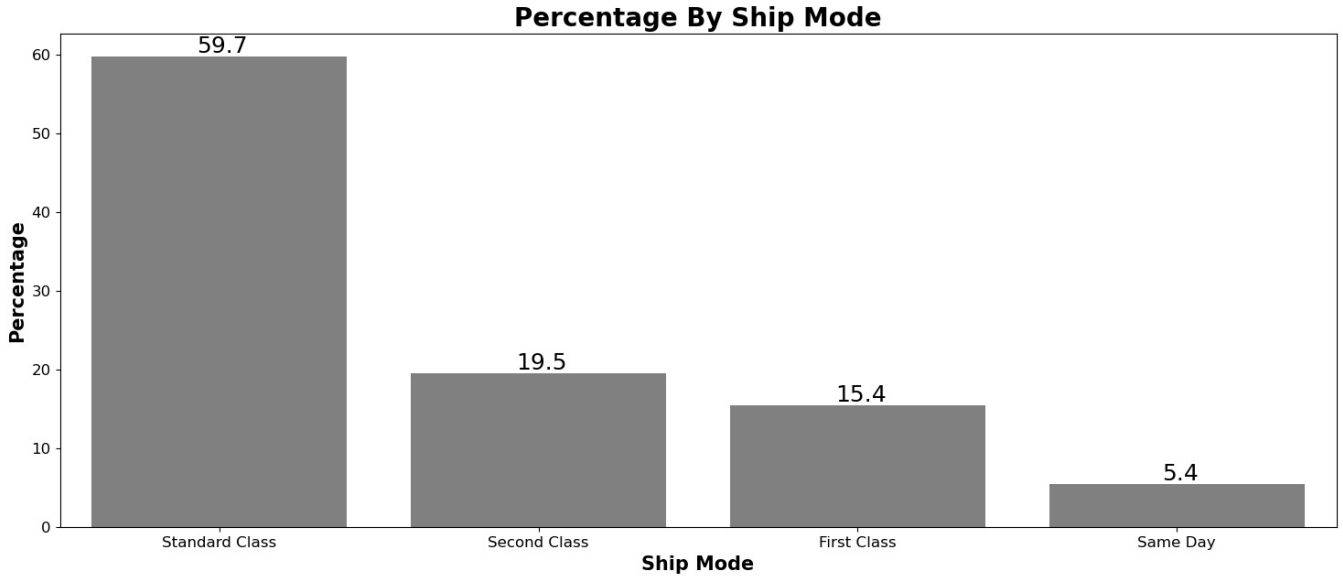
	Ship Mode	Orders	Percentage
0	Standard Class	5968	59.7
1	Second Class	1945	19.5
2	First Class	1538	15.4
3	Same Day	543	5.4

In [14]:
```
plt.figure(figsize = (18, 7))
graph = sns.barplot(x = "Ship Mode", y = "Orders", data = sm, color = "blue")
for p in graph.patches:
        graph.annotate('{:.0f}'.format(p.get_height()),
                       (p.get_x()+0.41, p.get_height()),
                        ha='center', va='bottom',color= 'black', size = 15)
plt.title("Count Of Orders by Ship Mode", size = 15, weight = "bold")
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Ship Mode", weight = "bold", size = 12)
plt.ylabel("Count Of Orders", weight = "bold", size = 12)
```

```
plt.show()
```



In [15]:
```
plt.figure(figsize = (18, 7))
graph = sns.barplot(x ="Ship Mode", y ="Percentage", data = sm, color ="grey")
for p in graph.patches:
        graph.annotate('{:.01f}'.format(p.get_height()),
                       (p.get_x()+0.41, p.get_height()),
                        ha='center', va='bottom',color= 'black', size = 18)
plt.title("Percentage By Ship Mode", size = 20, weight ="bold")
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Ship Mode", weight = "bold", size = 15)
plt.ylabel("Percentage", weight = "bold", size = 15)
plt.show()
```



In [16]:
```
df.head(1)
```

Out[16]:

	Order Date	Ship Date	Ship Mode	Customer Name	Segment	Country	City	State	Region	Category	Sub-Category	Product Name	Sales	Quantity	Disc
0	2016-11-08	2016-11-11	Second Class	Claire Gute	Consumer	United States	Henderson	Kentucky	South	Furniture	Bookcases	Bush Somerset Collection Bookcase	261.96	2	

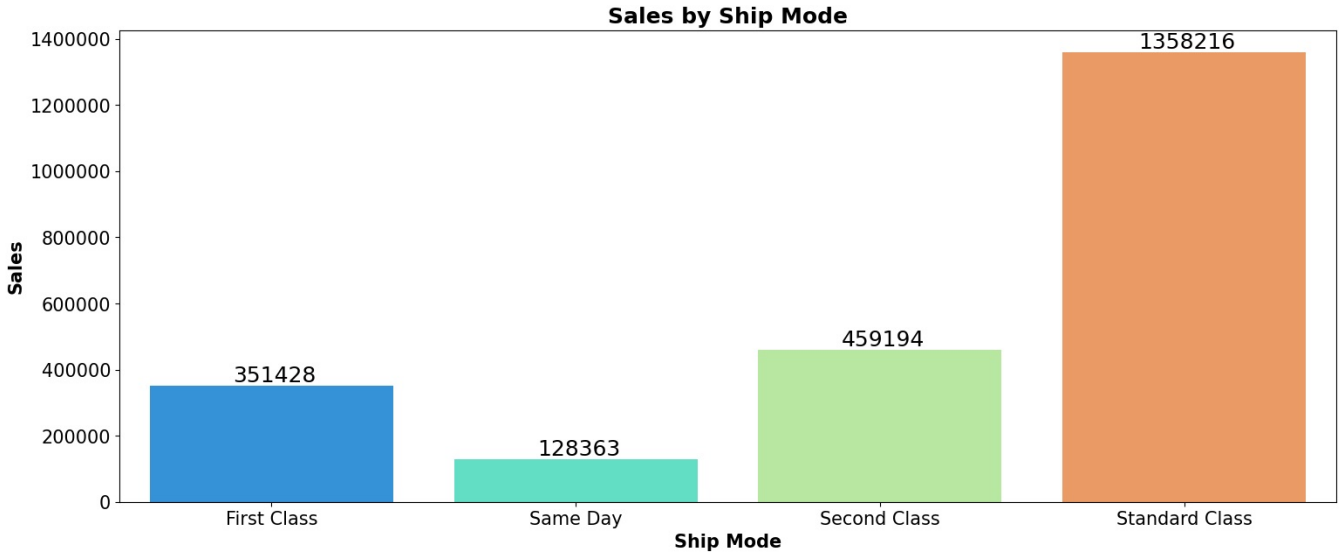
In [17]:
```
sqsm =df.groupby("Ship Mode")[["Sales", "Profit"]].agg("sum").round().reset_index()
sqsm
```

Out[17]:

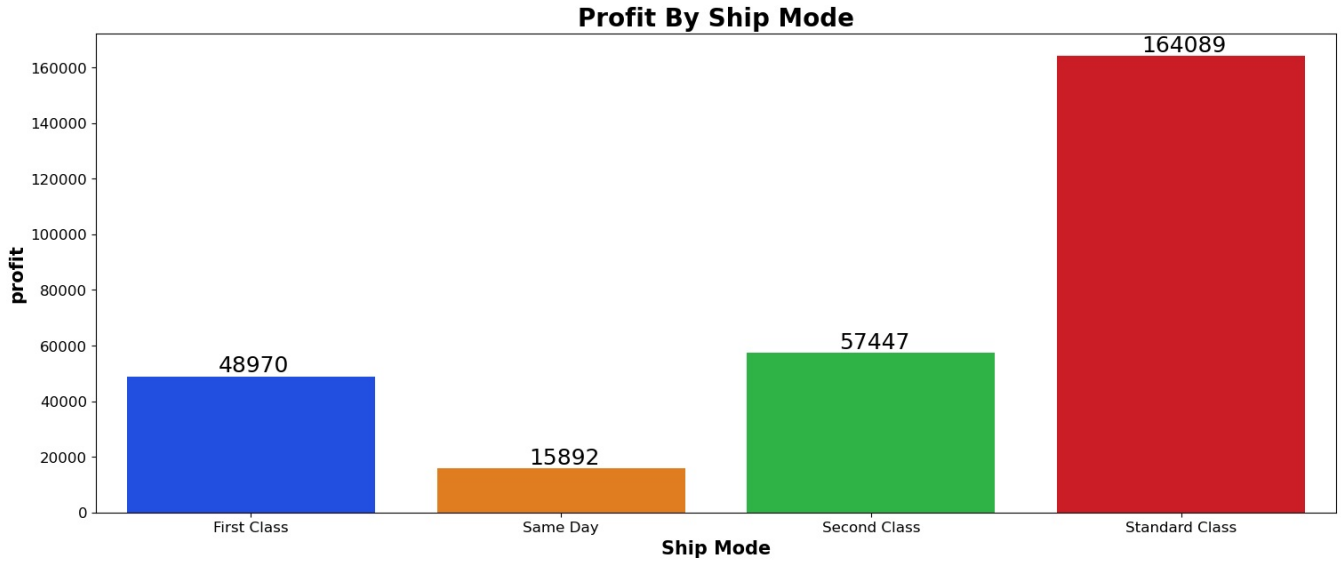
	Ship Mode	Sales	Profit
0	First Class	351428.0	48970.0
1	Same Day	128363.0	15892.0
2	Second Class	459194.0	57447.0
3	Standard Class	1358216.0	164089.0

In [18]:
```
plt.figure(figsize = (18,7))
```

```
graph = sns.barplot(x ="Ship Mode", y ="Sales", data = sqsm, palette = "rainbow")
plt.ticklabel_format(style= "plain", axis = "y")
for p in graph.patches:
        graph.annotate('{:.0f}'.format(p.get_height()),
                       (p.get_x()+0.41, p.get_height()),
                        ha='center', va='bottom',color= 'black', size = 18)
plt.title("Sales by Ship Mode", weight = "bold", size = 18)
plt.xlabel("Ship Mode", weight = "bold", size =15)
plt.ylabel("Sales", weight ="bold", size =15)
plt.xticks(size =15)
plt.yticks(size =15)
plt.show()
```



In [19]:
```
plt.figure(figsize = (18,7))
graph = sns.barplot(x = "Ship Mode", y ="Profit", data =sqsm, palette ="bright")
plt.ticklabel_format(style ="Plain", axis ="y")
for p in graph.patches:
        graph.annotate('{:.0f}'.format(p.get_height()),
                       (p.get_x()+0.41, p.get_height()),
                        ha='center', va='bottom',color= 'black', size = 18)
plt.title("Profit By Ship Mode", weight ="bold", size =20)
plt.xlabel("Ship Mode", weight ="bold", size =15)
plt.ylabel("profit", weight ="bold", size =15)
plt.xticks(size = 12)
plt.yticks(size =12)
plt.show()
```



In [20]:
```
os =df["Segment"].value_counts()
os =os.to_frame().reset_index()
os.columns = ["Segment", "Orders"]
os["Percentage"] =((os["Orders"] / os["Orders"].sum()) * 100).round(2)
os
```

Out[20]:

	Segment	Orders	Percentage
0	Consumer	5191	51.94
1	Corporate	3020	30.22
2	Home Office	1783	17.84

In [21]:
```
sns = df.groupby("Segment")[["Sales", "Profit"]].agg(["sum"]).reset_index()
```

In [21]:
```
sps = df.groupby('Segment')[['Sales', 'Profit']].agg(['sum']).reset_index()
sps
```

Out[21]:

	Segment	Sales	Profit
		sum	sum
0	Consumer	1.161401e+06	134119.2092
1	Corporate	7.061464e+05	91979.1340
2	Home Office	4.296531e+05	60298.6785

In [22]:
```
sps = sps.droplevel(1, axis = 1)
sps
```

Out[22]:

	Segment	Sales	Profit
0	Consumer	1.161401e+06	134119.2092
1	Corporate	7.061464e+05	91979.1340
2	Home Office	4.296531e+05	60298.6785

In [23]:
```
sps["Sales"] = sps["Sales"].astype(int)
sps
```

Out[23]:

	Segment	Sales	Profit
0	Consumer	1161401	134119.2092
1	Corporate	706146	91979.1340
2	Home Office	429653	60298.6785

In [24]:
```
sps["Profit"] =sps["Profit"].round(2)
sps
```

Out[24]:

	Segment	Sales	Profit
0	Consumer	1161401	134119.21
1	Corporate	706146	91979.13
2	Home Office	429653	60298.68

In [25]:
```
sps = sps.melt("Segment", var_name = "Category", value_name = "Dollar")
sps
```

Out[25]:

	Segment	Category	Dollar
0	Consumer	Sales	1161401.00
1	Corporate	Sales	706146.00
2	Home Office	Sales	429653.00
3	Consumer	Profit	134119.21
4	Corporate	Profit	91979.13
5	Home Office	Profit	60298.68

In [26]:
```
plt.figure(figsize = (18,7))
graph = sns.barplot(x ="Segment", y= "Dollar", hue = "Category", data = sps, palette = "muted")
for p in graph.patches:
        graph.annotate('{:.0f}'.format(p.get_height()),
                       (p.get_x()+0.2, p.get_height()),
                        ha='center', va='bottom',color= 'black', size = 15)
plt.title("Sales and Profit By Segments", weight ="bold", size =18)
plt.xticks(size =12)
plt.yticks(size =12)
plt.xlabel("Segment", weight ="bold", size =15)
plt.ylabel("Dollar", weight ="bold", size =15)
plt.ticklabel_format(style ="Plain", axis ="y")
plt.show()
```