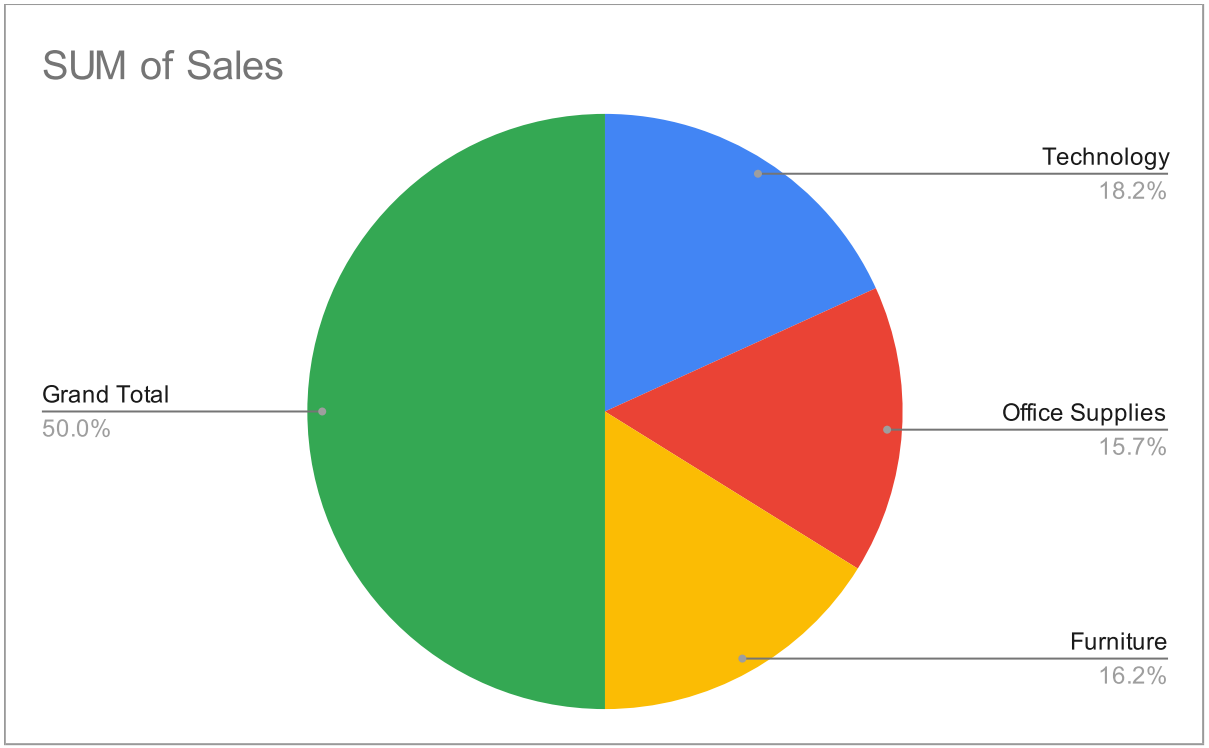
Category	SUM of Sales
Technology	836154.033
Office Supplies	719047.032
Furniture	741999.7953
Grand Total	**2297200.86**

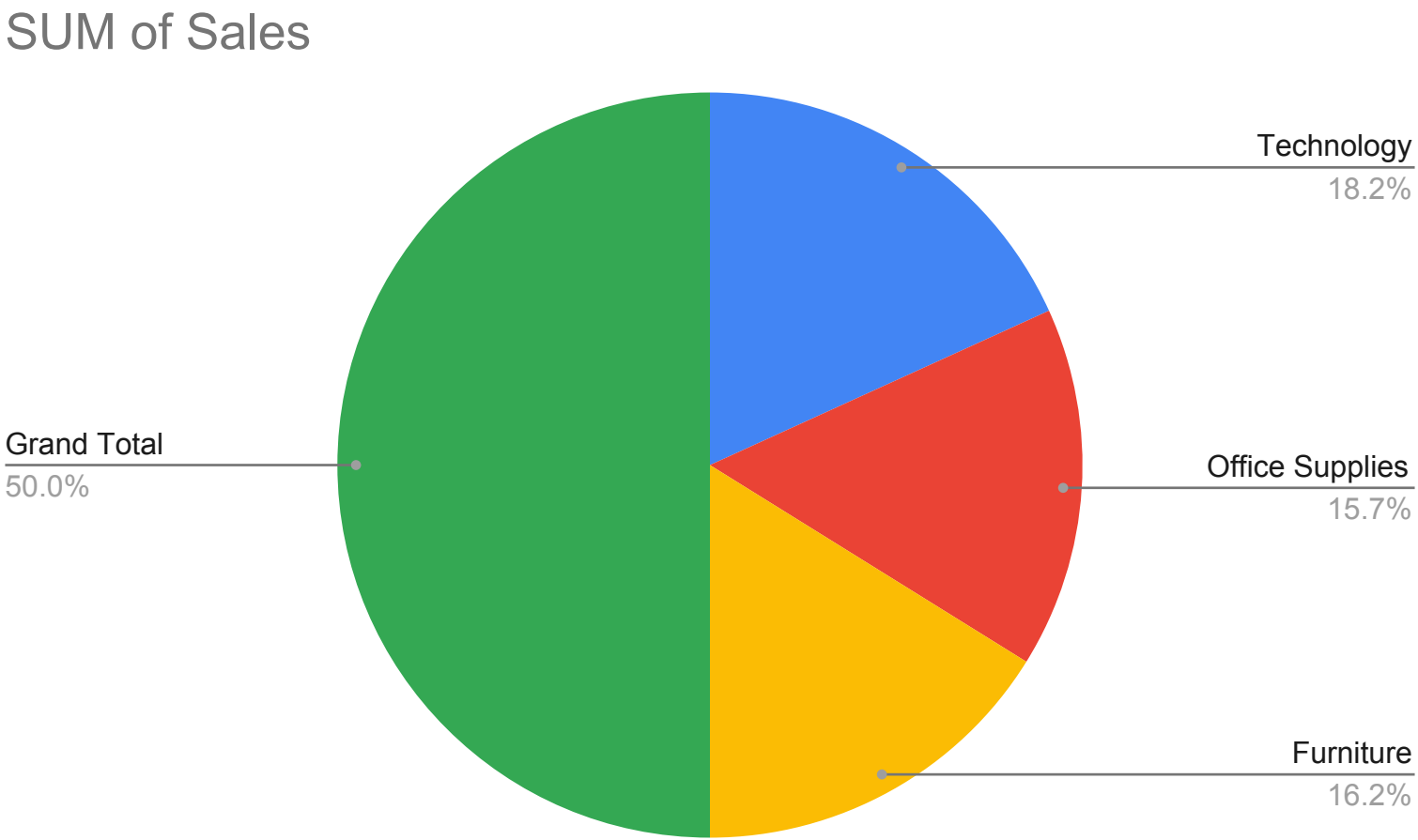
Region	SUM of Sales	SUM of Profit
West	725457.8245	108418.4489
South	391721.905	46749.4303
East	678781.24	91522.78
Central	501239.8908	39706.3625
Grand Total	**2297200.86**	**286397.0217**

Segment	SUM of Sales
Home Office	429653.1485
Corporate	706146.3668
Consumer	1161401.345
Grand Total	**2297200.86**

Category	SUM of Profit Ma
Technology	288.3869841
Office Supplies	831.7705556
Furniture	82.25987395
Grand Total	**1202.417414**

Region	SUM of Profit Ma
West	703.0156373
South	264.9008333
East	476.262381
Central	-241.7614379
Grand Total	**1202.417414**


SUM of Sales
Technology
18.2%
Office Supplies
15.7%
Furniture
16.2%
Grand Total
50.0%


SUM of Sales
Technology
18.2%
Office Supplies
15.7%
Furniture
16.2%
Grand Total
50.0%

```
import pandas as pd
import numpy as np
```

```
df = pd.read_csv("/content/Superstore.csv",encoding="latin-1")
df.head(2)
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Regi
0	1	CA-2013-152156	09-11-2013	12-11-2013	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	Sou
1	2	CA-2013-152156	09-11-2013	12-11-2013	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	Sou

2 rows × 21 columns

```
from datetime import datetime
df["Order Date"] = pd.to_datetime(df["Order Date"],format="mixed")
```

```
df["Order Date"].dtype
```

```
dtype('<M8[ns]')
```

```
df["Ship Date"] = pd.to_datetime(df["Ship Date"],format="mixed")
```

Date time foramts are converted to standard datetime format used when analyzing data.

Let's now check for missing values and handle them

```
missing_values = df.isnull().sum()
missing_values
```

	0
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

Since There are no missing values we can move onto next step

Pivot tables are auto adjusted when using pandas module in python

Create a Pivot Table that summarizes Total Sales by Category, then sort categories from highest to lowest to find business-driving segments.

```
cat_table = df.groupby("Category")["Sales"].sum().sort_values(ascending=False)
cat_table
```

Therefore, We can see that Technology followed by Furniture are the top 2 business driving categories with Office supplies close to Furniture.

Category	Sales
Technology	836154.0330
Furniture	741999.7953
Office Supplies	719047.0320

Add another Pivot Table showing Sales by Region and Segment, then highlight top-performing regions using conditional formatting.

```
reg_table = df.groupby("Region")["Sales"].sum().sort_values(ascending=False)
reg_table
```

Region	Sales
West	725457.8245
East	678781.2400
Central	501239.8908
South	391721.9050

dtype: float64

```
seg_table = df.groupby("Segment")["Sales"].sum().sort_values(ascending=False)
seg_table
```

Segment	Sales
Consumer	1.161401e+06
Corporate	7.061464e+05
Home Office	4.296531e+05

dtype: float64

```
Start coding or generate with AI.
```