

1. Data Description:

We used a few numpy arrays and lists to store and manipulate the numerical values and values of other data types in the dataset.

```
# Create arrays of values
c_sales = np.array([741999.80, 719047.03, 836154.03])
cat = np.array(["Furniture", "Office Supplies", "Technology"])

# Create arrays of values
y2 = np.array([23.24, 28.49, 16.20, 32.04])
regions = ["Central", "East", "South", "West"]

# Arrays of data and values
xTicks = ['2014-10-21', '2015-8-8', '2016-5-26', '2017-3-13', '2017-12-30']
l_sales = np.array([325515.91, 350134.39, 451053.18, 517327.27, 653153.66])
```

2. Design of Sub-Plots with a Distinct Feature:

The first plot is a bar chart, It represents the sales by category. We added labels above to help us distinguish the exact sales values at first glance.

The second one is a pie chart. It represents the total percent of sales for each region. We designed the pie chart to be hollow in the middle to add some explanatory text.

The third plot is a line chart. It represents the sales trend over time from 2014 to 2017. The chart plots both the original sales data and a 2-year moving average for the given period to show the increments more directly.

3. Implementation:

```
# -*- coding: utf-8 -*-
"""
```

Created on Sun Mar 28 13:07:13 2025

Project 1 - Exploratory Data Analysis

Author: Yiyang Lei, Yisi Lu, Alex Fan

Course: ISTM 631

Purpose: This script analyzes the superstore dataset to identify sales trends and generate visualizations.

It performs exploratory data analysis (EDA) and identification of top-selling categories, sales by region and monthly sales trend.

Description: This script creates a 1x3 grid of business metric plots using NumPy arrays, which help financial analytics on the superstore dataset.

Each presents different trends and uses different features.

"""

```
# Import required libraries
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
import pandas as pd
```

```
# Create a frame and restrict its size to 15x4
```

```
plt.figure(figsize = (15,4))
```

```
# Plot 1: Bar chart - Sales by Category
```

```
# Feature: Amounts added the top of each bar
```

```
plt.subplot(1,3,1)
```

```
# Create arrays of values
```

```
c_sales = np.array([741999.80, 719047.03, 836154.03])
```

```
cat = np.array(["Furniture", "Office Supplies", "Technology"])
```

```
# Set up the bar chart
```

```
plt.bar(cat, c_sales, width = 0.6)
```

```
plt.title("Sales by Category")
```

```
plt.xlabel("Category")
```

```
plt.ylabel("Sales")
```

```
plt.grid(visible=True, color='gray', linestyle='--', linewidth=0.5, alpha=0.7)
```

```
# Add values of profit of each category to the chart
```

```
i = 0 # Initialize an index counter
```

```
for v in c_sales:
```

```
    plt.text(i, v + 10000, f"${v:,.0f}", ha='center', fontweight='bold')
```

```
    i += 1 # Manually increment the index
```

```
# Plot 2: Pie Chart - Sales by Region
```

```
# Feature: Hollow the pie chart to make the plot more readable
```

```

plt.subplot(1,3,2)

# Create arrays of values
y2 = np.array([23.24, 28.49, 16.20, 32.04])
regions = ["Central", "East", "South", "West"]

mycolors = ["orange", "hotpink", "b", "#4CAF50"]
plt.title("Sales by Region")

# Place explanatory text in the middle of the chart
plt.text(0, 0, "Total\nSales", ha='center', va='center', fontsize=12, fontweight='bold')

# Plot the pie chart with a blank area in the middle for better visualization
plt.pie(y2, labels=regions, autopct='%1.1f%%', wedgeprops={'width':0.5})

# Plot 3 - Line Graph: Sales over Time
# Feature: Calculate the simple moving average using pandas library
plt.subplot(1,3,3)

# Arrays of data and values
xTicks = ['2014-10-21', '2015-8-8', '2016-5-26', '2017-3-13', '2017-12-30']
l_sales = np.array([325515.91, 350134.39, 451053.18, 517327.27, 653153.66])

# Convert dates to numeric (ordinal format)
x_numeric = pd.to_datetime(xTicks).map(pd.Timestamp.toordinal)

# Compute moving average
window_size = 2
moving_avg = np.convolve(l_sales, np.ones(window_size)/window_size, mode='valid')

# Plot original data using x_numeric (not xTicks as strings)
plt.plot(x_numeric, l_sales, label="Total Sales", marker='o', color="black")

# Plot moving average using correctly aligned x-values
plt.plot(x_numeric[window_size-1:], moving_avg, label="2-Year Moving Average", linestyle="--",
color="red")

# Format x-axis: use original string dates as labels
plt.xticks(x_numeric, xTicks, rotation=30)

# Plot the line of sales trend
plt.xlabel("Time")
plt.xticks(rotation = 30)

```

```
plt.ylabel('Sales')
plt.title("Sales Trend Over 2014 - 2017")
plt.grid(color = "green", linestyle = '--', linewidth = 1)

plt.legend(loc = 'upper left')

# Add a super title and display the final plot
plt.suptitle('Exploratory Data Analysis', fontsize=16, ha='center', va='top')
plt.tight_layout()
plt.show()
```

4. Recommendation for Additional Coding Ideas: Suggest at least one additional feature you would consider adding to each plot.

We can add chunks of bars with a different color on the bar chart turning it into a stacked bar chart. we now can show the profit and sales revenue at the same time.

We can add interactive functionality to the pie chart so that when hovering over a section, it displays the zoomed-in category and exact sales value, and the sizes of each section will be automatically proportioned based on sales values.

We can change to smoothed lines or reduce noise for better readability in large datasets.