Module 5: Data Visualisation

Case Study Solution

Case Study Solution

Domain – Retail

focus – Visualize the sales data

Business challenge/requirement
BigMart is one of the biggest retailer of Europe and has operations across multiple countries. You are a data analyst in IT team of BigMart. Invoice and SKU wise Sales Data for Year 2011 is shared with you. You need to prepare meaningful charts to show case the various sales trends for 2011 to top management.

Key issues
Data should be displayed pictorially to capture the attention of top management

Considerations
NONE

Data volume
- Approx 500K records – file BigMartSalesData.csv

Additional information
- NA

Business benefits
This exercise is an annual exercise and BigMart makes important investment decision based on trends

Approach to Solve
You have to use fundamentals of Matplotlib covered in module 5 and plot following 4 chart/graph

1. Plot Total Sales Per Month for Year 2011. How the total sales has increased over months in Year 2011. Which month has lowest Sales?
2. Plot Total Sales Per Month for Year 2011 as Bar Chart. Is Bar Chart Better to visualize than Simple Plot
3. Plot Pie Chart for Year 2011 Country Wise. Which Country contributes highest towards sales ?

4. Plot Scatter Plot for the invoice amounts and see the concentration of amount. In which range most of the invoice amounts are concentrated

Enhancements for code

You can try these enhancements in code

1. Change the bar chart to show the value of bar
2. In Pie Chart Play with Parameters shadow=True, startangle=90 and see how different the chart looks
3. In scatter plot change the color of Scatter Points

Solution

```
import pandas as pd
import matplotlib.pyplot as plt


plt.show()

#  Read BigMartSales.csv as a dataframe called salesdata
sales_data = pd.read_csv('BigMartSalesData.csv')

# Check the basic data & Verify 541874 records are there
print(sales_data.head())
print(sales_data.info())

# Assignment 1: Plot Total Sales Per Month for Year 2011
# How the total sales has increased over months in Year 2011. Which month has lowest Sales?
# Some of the functions on pandas will be covered in next module -- so just execute the code in case you don't understand it

# Get Sales for the Year 2011
print (" Getting Sales Data for Year 2011")
sales_2011 = sales_data[sales_data['Year'] == 2011]
print (" getting Amount for Each Month")
sales_2011_month = sales_2011.groupby('Month').sum()['Amount']
print(sales_2011_month)

# Simply Plot the Sales Data for 2011 , X Axis Sales for each month, Y -Axis Month Number
plt.plot(sales_2011_month.index,sales_2011_month.values)
plt.xlabel("Sales in Euro")
```

```
plt.ylabel("Month Number")
plt.title("Sales Per Month in Year 2011")
plt.show()

# Save the Plot Locally
plt.savefig("Year2011MonthWiseSales")
# December should be the answer

# Assignment 2: Plot Total Sales Per Month for Year 2011 as Bar Chart
# Is Bar Chart Better to visualize than Simple Plot

plt.bar(sales_2011_month.index,sales_2011_month.values,color="red") # Change the color and play
plt.xlabel("Sales in Euro")
plt.ylabel("Month Number")
plt.title("Sales Per Month in Year 2011")
plt.show()

# Enhancement Can you show the value of the bar
# Refer to https://matplotlib.org/gallery/api/barchart.html#sphx-glr-gallery-api-barchart-py


# Assignment 3: Plot Pie Chart for Year 2011 Country Wise
# Which Country contributes highest towards sales  and what Percentage?

sales_country_wise = sales_2011.groupby('Country').sum()['Amount']
plt.title("Country Wise Contribution For 2011")
plt.pie(sales_country_wise.values,labels=sales_country_wise.index,autopct='%1.1f%%')
plt.show()
# Answer: UK -- 83.5 %
# Enhancement -- Play With Parameters  shadow=True, startangle=90 etc in plt.pie and see how different the chart looks


# Assignment 4: Plot Scatter Plot for the invoice amounts and see the concentration of amount
# In which range most of the invoice amounts are concentrated ?

sales_invoice_wise = sales_2011.groupby('InvoiceNo').sum()['Amount']
print(sales_invoice_wise)
plt.scatter(sales_invoice_wise.values,sales_invoice_wise.values)
plt.grid(True)
```

```
plt.show()
# Enhancement -- Change the color of points use color=['red','green','blue']
```