#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Fri Nov 10 07:56:48 2023

@author: Machindra reddy

"""

Import numpy as np

Import pandas as pd

Import matplotlib.pyplot as plt

df = pd.read\_csv("bigmart sales.csv")

df.head()

df.tail()

df.describe()

# Line chart

# Function to create an improved multiple line chart with a different grouping

defplot\_improved\_line\_chart\_by\_item\_type(data, x\_column, y\_column, groupby\_column, title, x\_label, y\_label):

"""

Creates an improved multiple line chart grouped by 'Item\_Type' instead of 'Outlet\_Size'.

:param data: The Pandas the DataFrame containing the data.

:paramx\_column: The x-axis column name.

:paramy\_column: The y-axis column name.

:paramgroupby\_column: The name of the column to group and construct several lines from.

:param title: The chart's title.

:paramx\_label: The x-axis label.

:paramy\_label: The y-axis label.

"""

# Group data by using 'groupby\_column' and 'x\_column' and calculate the mean of 'y\_column'

grouped\_data = data.groupby([groupby\_column, x\_column])[y\_column].mean().unstack(groupby\_column)

# Set the color palette

colors = plt.get\_cmap('tab10')

# Start plotting with a larger figure size for better readability

plt.figure(figsize=(14, 7))

for (column, color) in zip(grouped\_data.columns, colors.colors):

plt.plot(grouped\_data.index, grouped\_data[column], marker='o', color=color, label=column)

plt.title(title, fontsize=16)

plt.xlabel(x\_label, fontsize=12)

plt.ylabel(y\_label, fontsize=12)

plt.legend(title=groupby\_column, bbox\_to\_anchor=(1.04,1), loc="upper left")

plt.grid(True)

plt.xticks(grouped\_data.index, rotation=45) # Rotate x-axis labels for better readability

plt.tight\_layout() # Adjust layout to make room for the legend

plt.show()

# Let's use the improved function with the bigmart sales data, now grouping by 'Item\_Type'.

plot\_improved\_line\_chart\_by\_item\_type(df,

x\_column='Outlet\_Establishment\_Year',

y\_column='Item\_Outlet\_Sales',

groupby\_column='Item\_Type',

title='Average Item Sales over the Years by Item Type',

x\_label='Year of Establishment',

y\_label='Average Item Sales')

#Histogram

# Function to create a histogram

defplot\_histogram(data, column, title, x\_label, y\_label, bins=None, color=None):

"""

Creates a histogram for a given numerical column in the data.

:param data: Pandas DataFrame containing the data.

:param column: The column name for which the histogram will be created.

:param title: The chart's title.

:paramx\_label: x-axis label.

:paramy\_label: y-axis label .

:param bins: number in bins for histogram. Optional; the default is calculated automatically.

:param color: The color for the histogram bars. Optional; a default color palette is used if not provided.

"""

# Let's plot and find out the distribution of our column of interest.

plt.figure(figsize=(10, 6))

plt.hist(data[column], bins=bins, color=color)

# Setting up titles and labels. A histogram without labels is like a shop with no signboard!

plt.title(title, fontsize=16)

plt.xlabel(x\_label, fontsize=12)

plt.ylabel(y\_label, fontsize=12)

# And of course, the grid for better readability.

plt.grid(axis='y', alpha=0.75)

# Now, let the histogram speak for itself!

plt.show()

# Now we'll use the histogram function to visualize the distribution of the Item Maximum Retail Price (MRP).

plot\_histogram(df,

column='Item\_MRP',

title='Distribution of Item Maximum Retail Price (MRP)',

x\_label='Item MRP',

y\_label='Frequency',

bins=30, # Let's go for 30 bins.

color='skyblue') # A soothing sky blue color for our bars.

# Piechart

# Function to create a pie chart

defplot\_pie\_chart(data, column, title):

"""

Creates a pie chart for a given categorical column in the data.

:param data: Pandas DataFrame containing the data.

:param column: The column name for which the pie chart will be created.

:param title: The title of the pie chart.

"""

# First, let's sum up the total sales by the category of interest.

sales\_by\_category = data.groupby(column).size()

# Time to plot! Let's bring out the pie (chart), shall we?

fig, ax = plt.subplots(figsize= (10,8))

ax.pie(sales\_by\_category, labels=sales\_by\_category.index, autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.colors)

ax.axis('equal')

# And what's a pie without a title? Let's add one.

plt.title(title, fontsize=16)

# Serve the pie chart hot!

plt.show()

# Now, let's use the pie chart function to visualize the sales distribution by 'Outlet\_Type'.

plot\_pie\_chart(df,

column='Outlet\_Type',

title='Proportion of Total Sales by Outlet Type')