Deborah Barndt

ITMD 513 Open Source Programming

Professor Dr. Sam

Hw6

2-28-19

Question #1: Expense Pie Chart

'''

Deborah Barndt

2-28-19

ExpensePieChart.py

hw6: Question 1 Expense Pie Chart

This program will read data from a text file and use the matplotlib to plot

out and display a pie chart showing how you spend your money.

Written by Deborah Barndt.

'''

import matplotlib.pyplot as plt

# Function to read a text file, and then build and display a pie chart.

def main():

# Open the text file in read mode.

budget = open('expenses.txt', 'r')

# Create the labels for the pie chart.

categories = ['Rent', 'Gas', 'Food', 'Clothing', 'Car Payment', 'Misc']

# Store the expense amounts in the order of the category labels.

expenses = []

# Read the content of the expenses file and then read the file line by line.

content = budget.read()

for data in content.split('\n'):

# Add the current expense in the file to the expenses.

expenses.append(int(data))

# Create the color scheme for the pie chart.

colors = ['lightskyblue', 'yellowgreen', 'lightcoral', 'gold','lightgreen', 'orange']

# Explode the first slice of the expense pie chart.

explode = (0.1, 0, 0, 0, 0, 0)

# Plot out the expenses pie chart from the values.

plt.pie(expenses, labels = categories, colors = colors, explode = explode, autopct = '%1.1f%%', shadow = True, startangle = 90)

# Set the axis of the expense pie chart as equal.

plt.axis('equal')

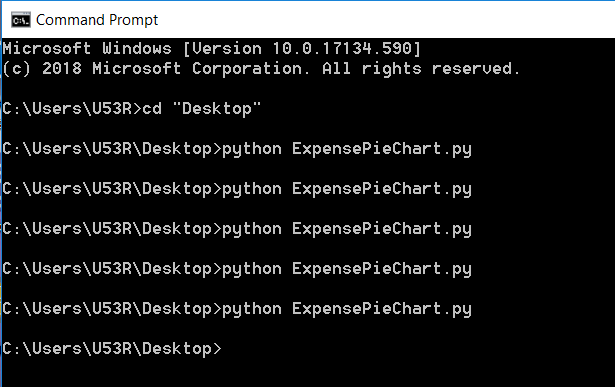
# Display the expense pie chart.

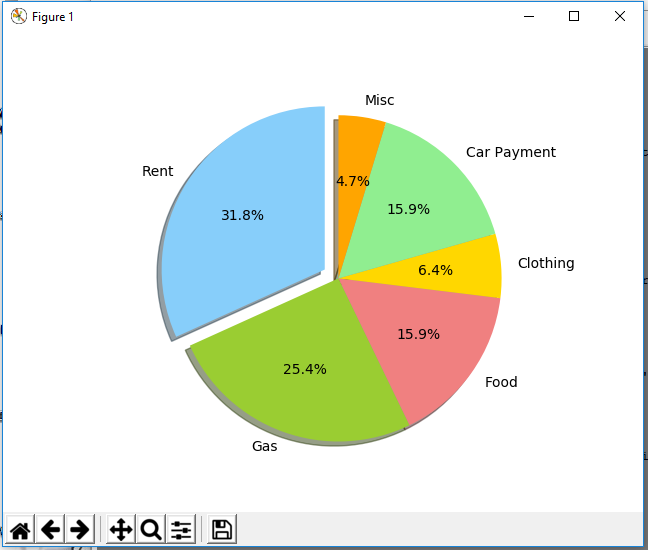
plt.show()

# Call the main function to run the program.

main()

Output Result:





Question #2: 1994 Weekly Gas Averages

'''

Deborah Barndt

2-28-19

WeeklyGasAverages.py

hw6: Question 2 1994 Weekly Gas Averages

This program will read data from 1994\_Weekly\_Gas\_averages.txt and then plot the

data as either a line graph or a bar chart.

Written by Deborah Barndt.

'''

import matplotlib.pyplot as plt

# Function to read a text file, and then build and display a line graph or bar chart.

def main():

# Ask the user if they would like the values to be displayed as a line graph or bar chart.

display = input('Would you like to display a line graph or bar chart? (line/bar) ')

# Open the text file in read mode.

averages = open('1994\_weekly\_gas\_averages.txt', 'r')

# Read the content of the gas averages.

content = averages.read()

# Split the content of the gas averages.

gas = content.split()

# Close the text file.

averages.close()

# For loop to strip the gas price into a float.

for i in range(0, len(gas)):

gas[i] = float(gas[i].strip())

# Create the range for the x-coordinates.

x\_coords = list(range(1,53))

# Build the line graph or bar chart dependent on user input.

if (display == 'line'):

# Build the line graph.

plt.plot(x\_coords, gas, color = 'lightskyblue')

# Set the limit of x-axis.

plt.xlim([1, 52])

# Create the label for the x-axis.

plt.xlabel('Week')

# Create the label for the y-axis.

plt.ylabel('Gas Price')

# Create the title of the line graph.

plt.title('Weekly Average Gas Price in 1994')

# Display the line graph.

plt.show()

elif (display == 'bar'):

# Build the bar chart.

plt.bar(x\_coords, gas, color = 'lightskyblue')

# Set the limit of x-axis.

plt.xlim([1, 52])

# Create the label for the x-axis.

plt.xlabel('Week')

# Create the label for the y-axis.

plt.ylabel('Gas Price')

# Create the title of the line graph.

plt.title('Weekly Average Gas Price in 1994')

# Display the line graph.

plt.show()

else:

print('Invalid input: Type either line or bar.')

# Call the main function the run the program.

main()

Output Result:

