

'''

This program reads the data from the price of gasoline csv file
and uses only the numeric data.
Each line of the file has the year, followed by 12 months of gasoline prices.
We put the numeric fields in a NumPy array and use the data for numeric operations.

The outputs of the program are the averages of each month across the years
and the averages for each year.

'''

```
import csv
import numpy as np

infile = 'Price_of_Gasoline.XL.csv'

# create new empty lists: years and prices come from data
yearsList = []
pricesList = []
# names of months for labeling results
monthList = ['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']

# read the data
with open(infile, 'rU') as csvfile:
    # the csv file reader returns a list of the csv items on each line - note delimiter is comma
    priceReader = csv.reader(csvfile, dialect='excel', delimiter=',')

    # from each line, a list of row items, make separate lists for years and for the prices
    for line in priceReader:
        # skip lines without data
        if line[0] == " or line[0].startswith('Price') or line[0].startswith('Year') or line[0].startswith('2001') or
line[0].startswith('2002') or line[0].startswith('2003'):
            continue
        else:
            try:
                # add the year to list
                yearsList.append(line[0])
                # append the prices (as strings) to the list
                pricesList.append(line[1:])
            except IndexError:
                print ('Error: ', line)
csvfile.close()

print ("Read", len(yearsList), "years of prices")

# make a numpy array for the strings
data = np.array(pricesList)
print('Shape of Prices data', data.shape)

# convert the empty strings to strings of zeros, using a Boolean mask to find empty strings
data[data == ""] = '0'

# now we can convert the whole thing to float without getting conversion errors for the empty strings
prices = data.astype(np.float)
```

```

#print(prices)

# compute the average price for each month (or use mean)
# sum along the columns
monthTotalPrices = np.sum(prices, axis = 0)
# divide by number of years to get average
monthAveragePrices = monthTotalPrices / len(yearsList)

# print(monthAveragePrices)
print ("\nAverage gas price for each month\n")

# print the average price for each month
for i, mon in enumerate(monthList):
    print (mon, ': ', monthAveragePrices[i])

# compute the average price for each year up to the last one with missing data
# sum along the rows
yearTotalPrices = np.sum(prices, axis = 1)
# divide by number of months to get average
yearAveragePrices = yearTotalPrices / 12

#print(monthAveragePrices)
print ("\nAverage gas price for each year\n")

# print the average price for the years
for i, year in enumerate(yearsList[ :-1]):
    print (year, ': ', yearAveragePrices[i])

# or display the monthly averages as a simple plot

import matplotlib.pyplot as pp

x = np.arange(12)
pp.xticks(x,monthList)
pp.plot(x, monthAveragePrices)
pp.show()

# or we can also display the years with a simple plot
x = np.arange(len(yearsList)-1)
pp.xticks(x,yearsList)
pp.plot(x, yearAveragePrices[ :-1])
pp.show()

# Done!

```

PROGRAM RESULTS

Read 26 years of prices
Shape of Prices data (26, 12)

Average gas price for each month

Jan : 1.07388461538
Feb : 1.073
Mar : 1.07719230769
Apr : 1.102
May : 1.13126923077
Jun : 1.14846153846
Jul : 1.14080769231
Aug : 1.14084615385
Sep : 1.14584615385
Oct : 1.06888461538
Nov : 1.06492307692
Dec : 1.05411538462

Average gas price for each year

1976 : 0.614333333333
1977 : 0.656333333333
1978 : 0.67025
1979 : 0.90325
1980 : 1.24516666667
1981 : 1.37825
1982 : 1.2955
1983 : 1.24116666667
1984 : 1.21225
1985 : 1.20175
1986 : 0.927416666667
1987 : 0.948416666667
1988 : 0.946166666667
1989 : 1.02216666667
1990 : 1.16433333333
1991 : 1.14008333333
1992 : 1.1265
1993 : 1.10791666667
1994 : 1.11183333333
1995 : 1.14716666667
1996 : 1.23091666667
1997 : 1.23366666667
1998 : 1.05933333333
1999 : 1.16508333333
2000 : 1.51