

```
In [1]: #Load Libraries
import pandas as pd
import matplotlib.pyplot as plt
import squarify
import numpy as np
import matplotlib.ticker as plticker # for plot ticks

In [2]: #import data as dataframe
data = pd.read_csv('expenditures.txt', names=['Year', 'Category', 'Expenditure', 'Sex'])
data2 = pd.read_csv('unemployment-rate-1948-2010.csv')

In [3]: #data.info()

In [4]: #data.head()

In [5]: #data2.head()

In [6]: #create date column for unemployment df
data2['Month'] = pd.to_numeric(data2['Period'].str[-2:])
data2['Day'] = 1
data2['Date'] = pd.to_datetime(data2[['Year', 'Month', 'Day']])

In [7]: data2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 746 entries, 0 to 745
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   Series id   746 non-null    object
 1   Year        746 non-null    int64
 2   Period      746 non-null    object
 3   Value       746 non-null    float64
 4   Month       746 non-null    int64
 5   Day         746 non-null    int64
 6   Date        746 non-null    datetime64[ns]
dtypes: datetime64[ns](1), float64(1), int64(3), object(2)
memory usage: 40.9+ KB
```

## Treemap

Resources: <https://www.analyticsvidhya.com/blog/2021/06/build-treemaps-in-python-using-squarify/> (<https://www.analyticsvidhya.com/blog/2021/06/build-treemaps-in-python-using-squarify/>)

<https://jingwen-z.github.io/data-viz-with-matplotlib-series5-treemap/> (<https://jingwen-z.github.io/data-viz-with-matplotlib-series5-treemap/>)

```
In [8]: # get filtered data
#selecting rows based year column
treemapdata = data[((data['Year'] >= 1999) & (data['Category'] == "Alcoholic Beverage"))]
```

```
In [9]: treemapdata.info()
```

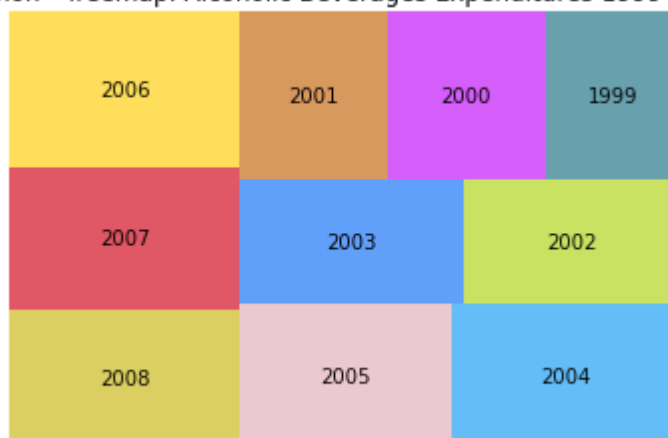
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10 entries, 1 to 127
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Year            10 non-null    int64
1   Category        10 non-null    object
2   Expenditure     10 non-null    int64
3   Sex             10 non-null    int64
dtypes: int64(3), object(1)
memory usage: 400.0+ bytes
```

```
In [10]: def get_cmap(n, name='hsv'):
'''Returns a function that maps each index in 0, 1, ..., n-1 to a distinct
RGB color; the keyword argument name must be a standard mpl colormap name.'''
return plt.cm.get_cmap(name, n)
```

```
In [11]: squarify.plot(sizes=treemapdata['Expenditure'],
                      label=treemapdata['Year'],
                      color=np.random.rand(len(treemapdata['Year']),3), #creates a random color
                      alpha=0.7 )
#plt.xlabel("not used") # X-axis Label
#plt.ylabel("not used") # Y-axis Label
plt.axis('off') # turns axis off
plt.title("Python - Treemap: Alcoholic Beverages Expenditures 1999-2008") # title
```

```
Out[11]: Text(0.5, 1.0, 'Python - Treemap: Alcoholic Beverages Expenditures 1999-2008')
```

Python - Treemap: Alcoholic Beverages Expenditures 1999-2008



```
In [42]: tmdata2 = data.loc[data["Year"] == 2008].sort_values(by=['Expenditure'], ascending=True)
```

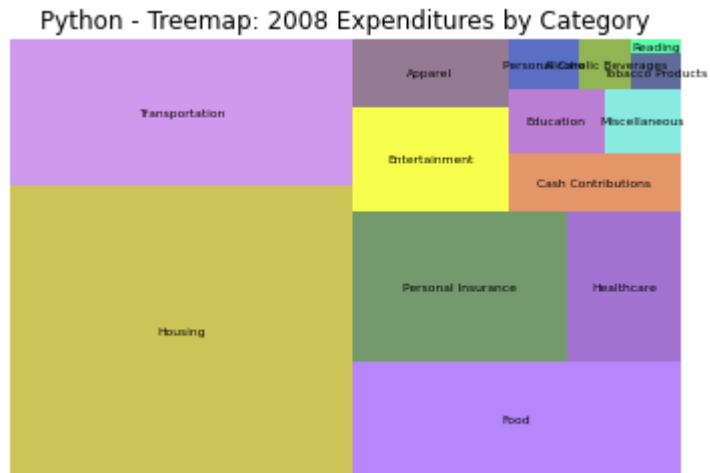
```
In [43]: tmdata2
```

Out[43]:

	Year	Category	Expenditure	Sex
2	2008	Housing	17109	1
4	2008	Transportation	8604	1
0	2008	Food	6443	1
13	2008	Personal Insurance	5605	1
5	2008	Healthcare	2976	1
6	2008	Entertainment	2835	1
3	2008	Apparel	1801	1
12	2008	Cash Contributions	1737	1
9	2008	Education	1046	1
11	2008	Miscellaneous	840	1
7	2008	Personal Care	616	1
1	2008	Alcoholic Beverages	444	1
10	2008	Tobacco Products	317	1
8	2008	Reading	116	1

```
In [54]: squarify.plot(sizes=tmdata2['Expenditure'],
                      label=tmdata2['Category'],
                      color=np.random.rand(len(tmdata2['Category']),3), #creates a random
                      alpha=0.7,
                      text_kwargs={'fontsize':6})
#plt.xlabel("not used") # X-axis Label
#plt.ylabel("not used") # Y-axis Label
plt.axis('off') # turns axis off
plt.title("Python - Treemap: 2008 Expenditures by Category") # title
```

Out[54]: Text(0.5, 1.0, 'Python - Treemap: 2008 Expenditures by Category')



## Area Chart

### References

<https://jingwen-z.github.io/data-viz-with-matplotlib-series7-area-chart/> (<https://jingwen-z.github.io/data-viz-with-matplotlib-series7-area-chart/>)

```
In [12]: # area chart data
areadata = data2[((data2['Year'] == 2001) | (data2['Year'] == 2008))]

#areadata=areadata.set_index('Date')
#Year2001 = data2[data2['Year'] == 2001]
#Year2001 = data2[(data2['Year'] == 2001)]
#Year2008 = data2[(data2['Year'] == 2008)]
```

```
In [13]: pivareadata=areadata.pivot(index='Month',columns='Year',values='Value')
pivareadata.columns = ['2001','2008']
```

In [14]: `pivareadata.info()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 12 entries, 1 to 12
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  -
 0   2001    12 non-null     float64
 1   2008    12 non-null     float64
dtypes: float64(2)
memory usage: 288.0 bytes
```

In [15]: `pivareadata`

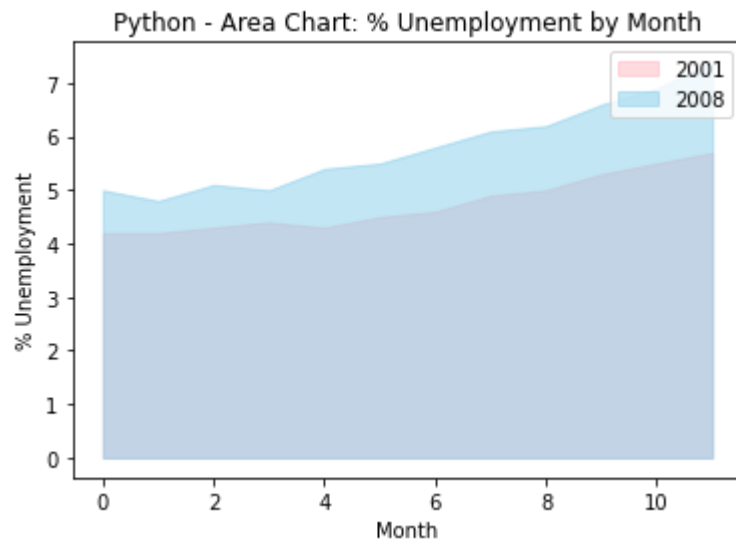
Out[15]:

	2001	2008
Month		
1	4.2	5.0
2	4.2	4.8
3	4.3	5.1
4	4.4	5.0
5	4.3	5.4
6	4.5	5.5
7	4.6	5.8
8	4.9	6.1
9	5.0	6.2
10	5.3	6.6
11	5.5	6.9
12	5.7	7.4

```
In [16]: plt.fill_between(np.arange(12), pivareadata['2001'], color="lightpink", alpha=0.5)
plt.fill_between(np.arange(12), pivareadata['2008'], color="skyblue", alpha=0.5,

plt.xlabel("Month") # X-axis Label
plt.ylabel("% Unemployment") # Y-axis Label
plt.title("Python - Area Chart: % Unemployment by Month") # title

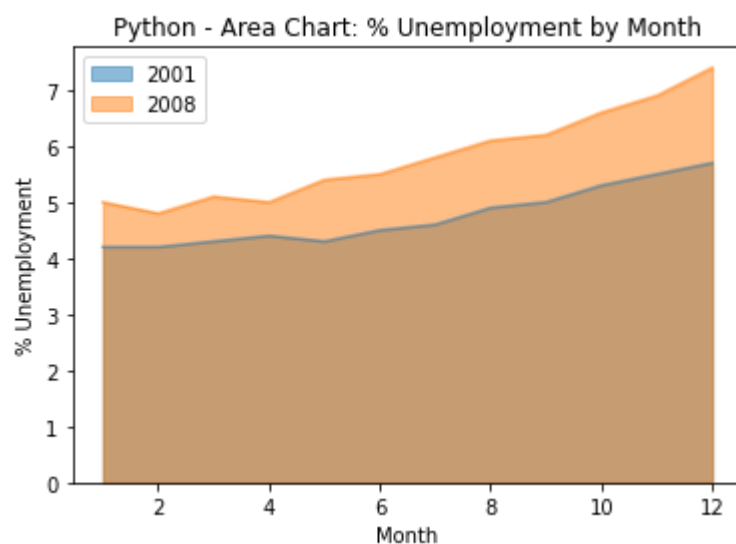
plt.legend()
plt.show()
```



```
In [17]: pivareadata.plot(kind='area', stacked=False)

plt.xlabel("Month") # X-axis Label
plt.ylabel("% Unemployment") # Y-axis Label
plt.title("Python - Area Chart: % Unemployment by Month") # title

plt.show(block=True)
```



# Stacked Area Chart

```
plt.stackplot(pivareadata.index, [pivareadata['2001'], pivareadata['2008']], labels=['2001', '2008'],
alpha=0.8)
```

```
plt.xlabel("Month") # X-axis label plt.ylabel("% Unemployment") # Y-axis label plt.title("Python -
Stacked Area Chart: % Unemployment by Month") # title
```

```
plt.legend(loc=2, fontsize='large') plt.show()
```

In [18]: data

Out[18]:

	Year	Category	Expenditure	Sex
0	2008	Food	6443	1
1	2008	Alcoholic Beverages	444	1
2	2008	Housing	17109	1
3	2008	Apparel	1801	1
4	2008	Transportation	8604	1
...	...	...	...	...
345	1984	Education	303	1
346	1984	Tobacco Products	228	1
347	1984	Miscellaneous	451	1
348	1984	Cash Contributions	706	1
349	1984	Personal Insurance	1897	1

350 rows × 4 columns

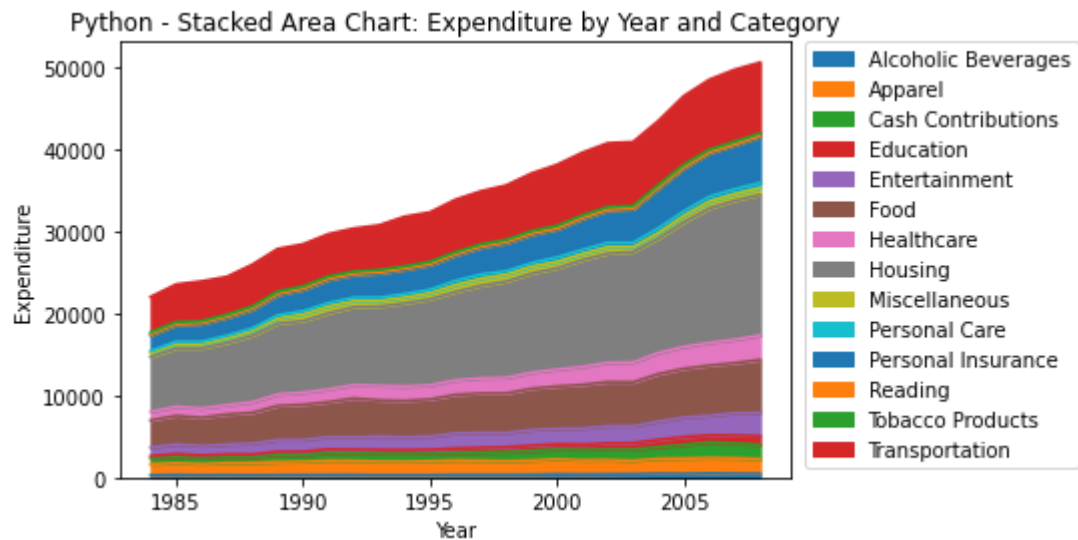
In [19]: newdf=data.pivot(index='Year',columns='Category',values='Expenditure')

In [20]: #newdf

```
In [21]: newdf.plot.area()

plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
plt.xlabel("Year") # X-axis Label
plt.ylabel("Expenditure") # Y-axis Label
plt.title("Python - Stacked Area Chart: Expenditure by Year and Category") # tit

plt.show()
```



In [ ]: