Aaron J. Brown

DSC 640: Data Presentation & Visualization - Winter 2023

WEEKS 5-6 Exercises: Tree Maps, Area Charts, and Stacked Area Charts (PYTHON)

Importing Libraries.

In [1]:
```
import os
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import nltk

import warnings
warnings.filterwarnings('ignore')

# pip install squarify
```

Importing Data.

In [2]:
```
unemployment_df = pd.read_csv('/Users/aaronbrown/Documents/Classwork/DSC 640 - Data Presentation and Visualization/Data/unemployement-rate-1948-2010.csv')
employ_data = unemployment_df

employ_data
```

Out[2]:

	Series id	Year	Period	Value
0	LNS14000000	1948	M01	3.4
1	LNS14000000	1948	M02	3.8
2	LNS14000000	1948	M03	4.0
3	LNS14000000	1948	M04	3.9
4	LNS14000000	1948	M05	3.5
...	...	...	...	...
741	LNS14000000	2009	M10	10.1
742	LNS14000000	2009	M11	10.0
743	LNS14000000	2009	M12	10.0
744	LNS14000000	2010	M01	9.7
745	LNS14000000	2010	M02	9.7

746 rows × 4 columns

In [3]:
```
expenditures_df = pd.read_csv('/Users/aaronbrown/Documents/Classwork/DSC 640 - Data Presentation and Visualization/Data/expenditures.txt', sep="\t")
spending_data = expenditures_df
spending_08 = spending_data[spending_data["year"] == 2008]

spending_data
```

Out[3]:

	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 [4]:
```
spending_data2 = pd.pivot_table(spending_data, index = "year", columns = "category", values=  "expenditure", aggfunc = "mean")
spending_data2
```

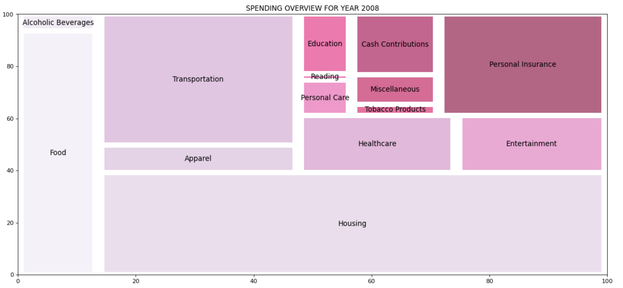
Out[4]:

category	Alcoholic Beverages	Apparel	Cash Contributions	Education	Entertainment	Food	Healthcare	Housing	Miscellaneous	Personal Care	Personal Insurance	Reading	Tobacco Products	Transportation
year														
1984	275	1319	706	303	1055	3290	1049	6674	451	289	1897	132	228	4304
1985	306	1420	805	321	1170	3477	1108	7087	529	303	2016	141	219	4587
1986	271	1346	746	314	1149	3448	1135	7292	522	303	2127	140	230	4842
1987	289	1446	741	337	1193	3664	1135	7569	562	330	2175	142	232	4600
1988	269	1489	693	342	1329	3748	1298	8079	578	334	2249	150	242	5093
1989	284	1582	900	367	1424	4152	1407	8609	643	366	2472	157	261	5187
1990	293	1618	816	406	1422	4296	1480	8703	842	364	2592	153	274	5120
1991	297	1735	950	447	1472	4271	1554	9252	860	399	2787	163	276	5151
1992	301	1710	958	426	1500	4723	1634	9477	765	387	2750	162	275	5228
1993	268	1676	961	455	1626	4399	1776	9636	715	385	2908	166	268	5453
1994	278	1644	960	460	1567	4411	1755	10106	749	397	2938	165	259	6044
1995	277	1704	925	471	1612	4505	1732	10458	766	403	2964	162	269	6014
1996	309	1752	940	524	1834	4698	1770	10747	855	513	3060	159	255	6382
1997	309	1729	1001	571	1813	4801	1841	11272	847	528	3223	164	264	6457
1998	309	1674	1109	580	1746	4810	1903	11713	860	401	3381	161	273	6616
1999	318	1743	1181	635	1891	5031	1959	12057	867	408	3436	159	300	7011
2000	372	1856	1192	632	1863	5158	2066	12319	776	564	3365	146	319	7417
2001	349	1743	1258	648	1953	5321	2182	13011	750	485	3737	141	308	7633
2002	376	1749	1277	752	2079	5375	2350	13283	792	526	3899	139	320	7759
2003	391	1640	1370	783	2060	5340	2416	13432	606	527	4055	127	290	7781
2004	459	1816	1408	905	2218	5781	2574	13918	690	581	4823	130	288	7801
2005	426	1886	1663	940	2388	5931	2664	15167	808	541	5204	126	319	8344
2006	497	1874	1869	888	2376	6111	2766	16366	846	585	5270	117	327	8508
2007	457	1881	1821	945	2698	6113	2853	16920	808	588	5336	118	323	8758
2008	444	1801	1737	1046	2835	6443	2976	17109	840	616	5605	116	317	8604

Generating Tree Map.

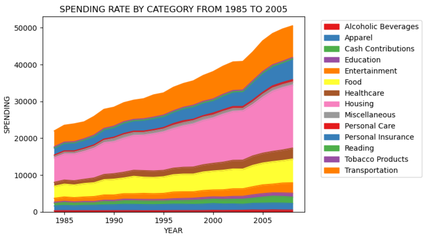
In [5]:
```
import squarify
```

In [6]:
```
plt.figure(figsize=(18, 8), dpi = 80)
squarify.plot(spending_08.expenditure, label = spending_08.category, alpha = 0.6,
              color = sns.color_palette("PuRd", len(spending_08['category'].value_counts())),
              pad = 1, text_kwargs = {'fontsize': 12})
plt.title('SPENDING OVERVIEW FOR YEAR 2008')
plt.show()
```



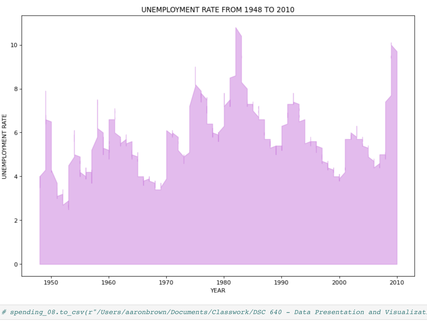
Generating Stacked Area Chart.

In [7]:
```
spending_data2.plot.area(color = sns.color_palette("Set1", len(spending_08['category'].value_counts())))
plt.legend(bbox_to_anchor = (1.05, 1.0), loc = "upper left")
plt.xlabel("YEAR"), plt.ylabel("SPENDING")
plt.title('SPENDING RATE BY CATEGORY FROM 1985 TO 2005")
plt.show()
```



Generating Area Chart.

In [8]:
```
plt.figure(figsize=(12, 8), dpi = 80)
plt.fill_between(employ_data.Year, employ_data.Value, color = "mediumorchid", alpha = 0.4)
plt.xlabel("YEAR"), plt.ylabel("UNEMPLOYMENT RATE")
plt.title('UNEMPLOYMENT RATE FROM 1948 TO 2010")
plt.show()
```



In [10]:
```
# spending_08.to_csv(r"/Users/aaronbrown/Documents/Classwork/DSC 640 - Data Presentation and Visualization/Data/expenditures_2008.csv", index=True)
```

References

Choosing Colormaps in Matplotlib:

https://matplotlib.org/stable/tutorials/colors/colormaps.html

List of named colors in matplotlib:

https://matplotlib.org/stable/gallery/color/named_colors.html

Seaborn Styling, Color:

https://www.codecademy.com/article/seaborn-design-ii

Treemaps in Python using Squarify

https://www.geeksforgeeks.org/treemaps-in-python-using-squarify/

How to Change the Position of a Legend in Matplotlib

https://www.statology.org/matplotlib-legend-position/

In [ ]:
```

```