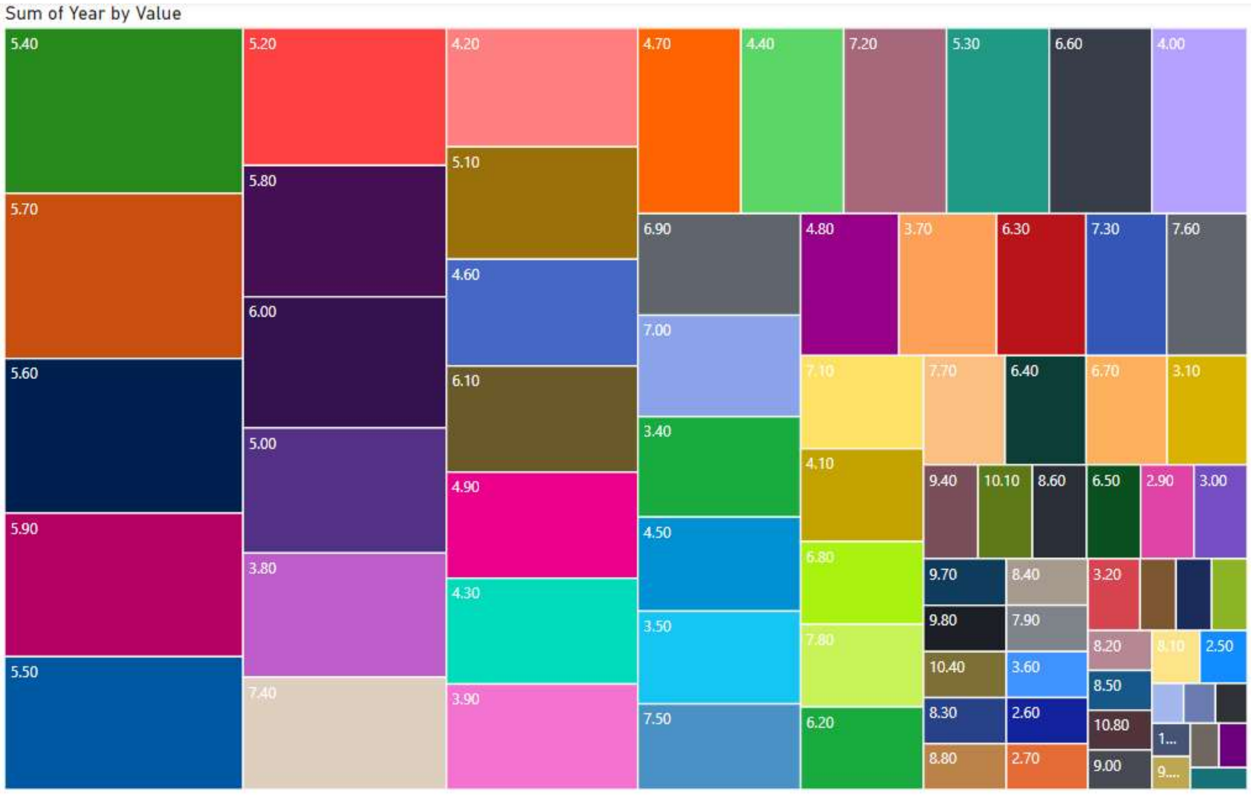
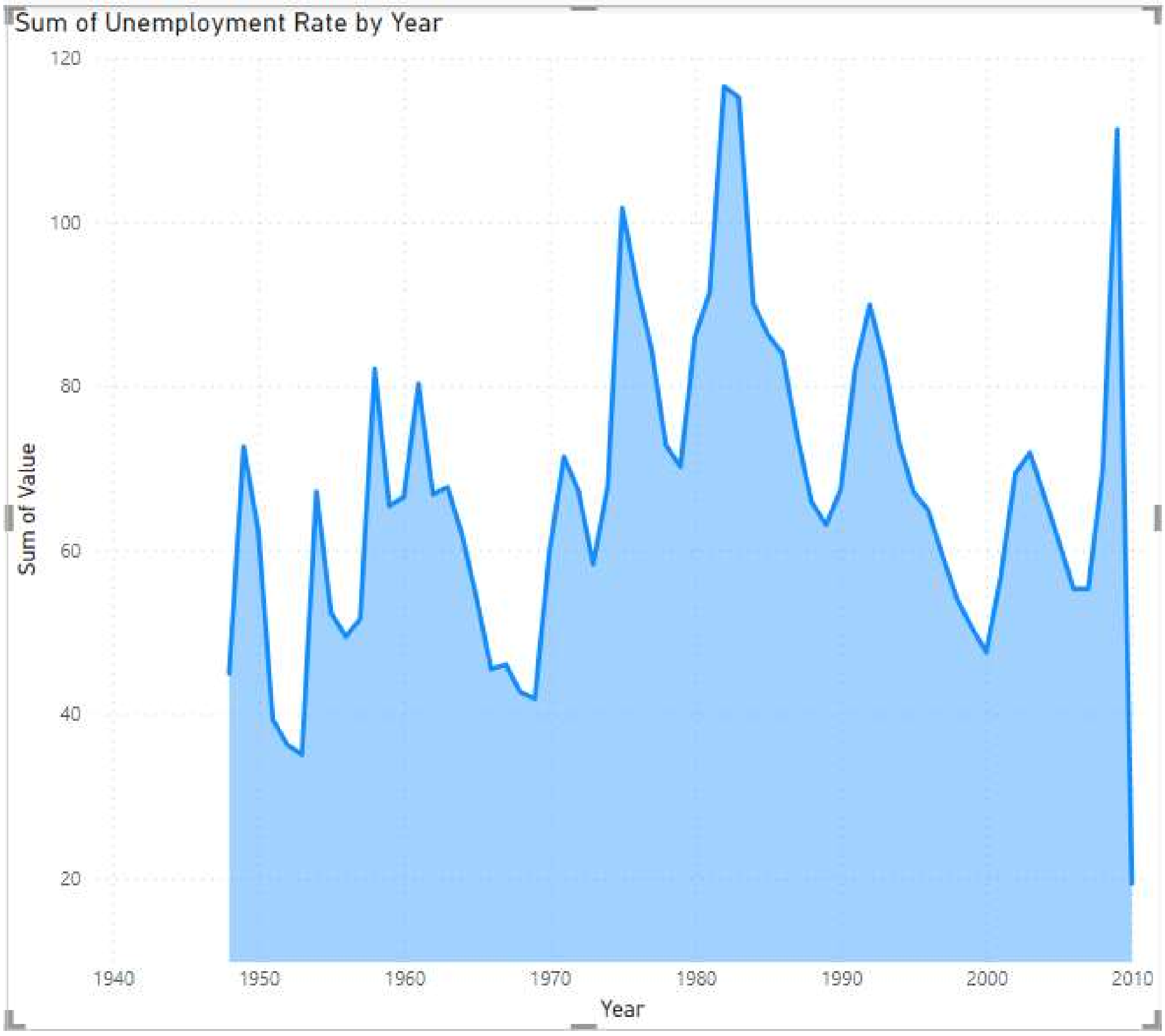
POWER BI

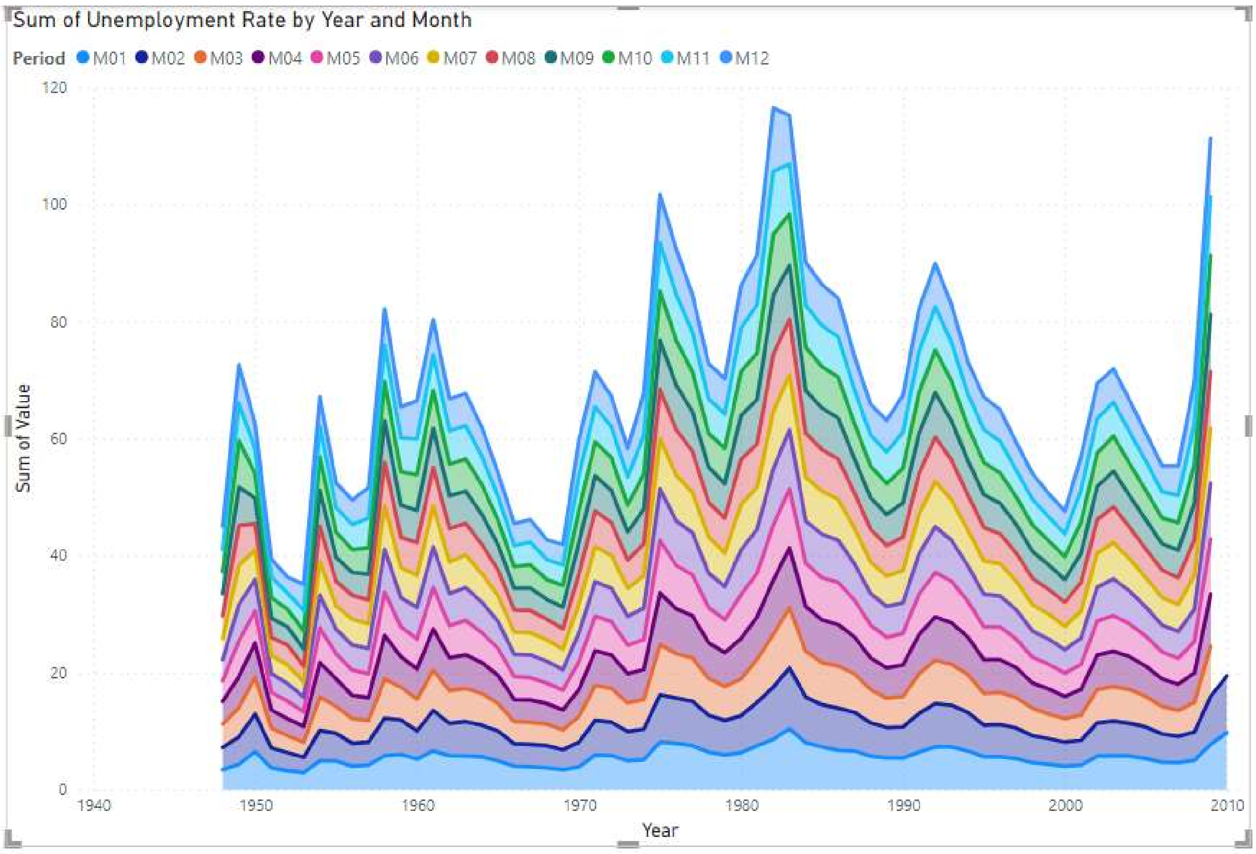
Tree Map



Area Chart


Sum of Unemployment Rate by Year
120
100
80
60
40
20
Sum of Value
1940
1950
1960
1970
1980
1990
2000
2010
Year

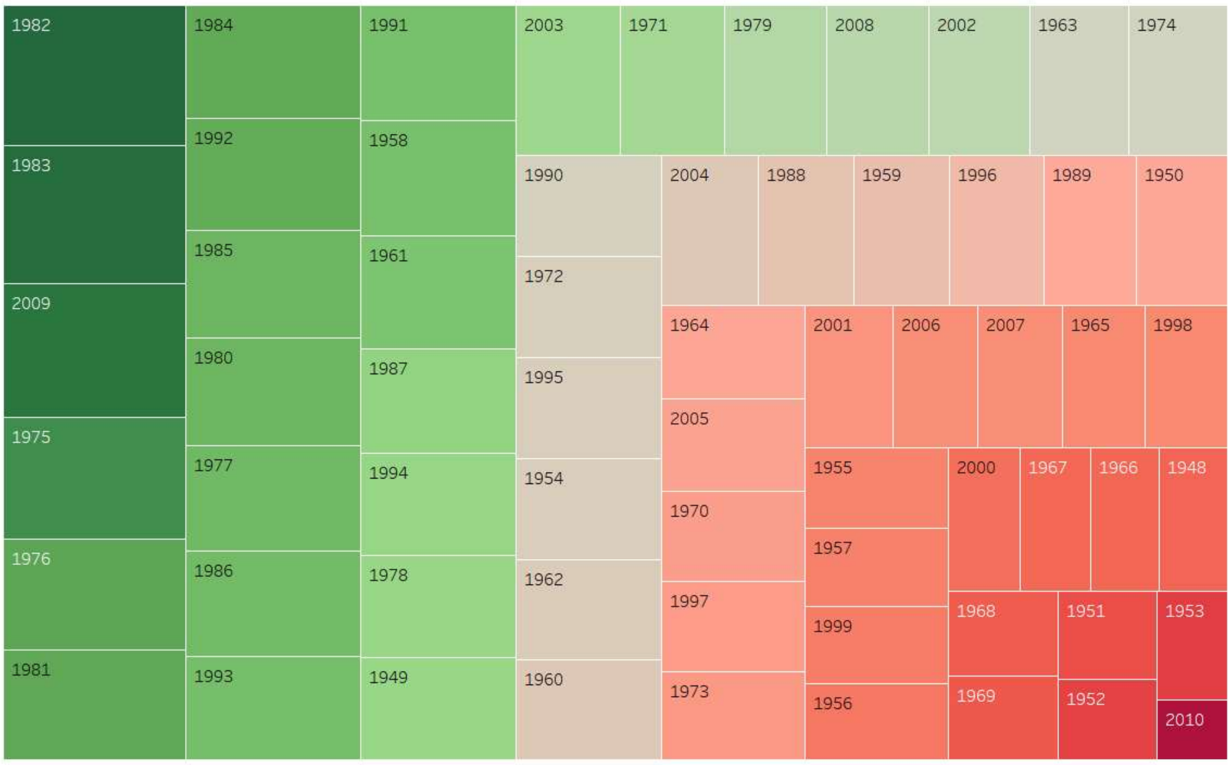
Stacked Area Chart



Tableau

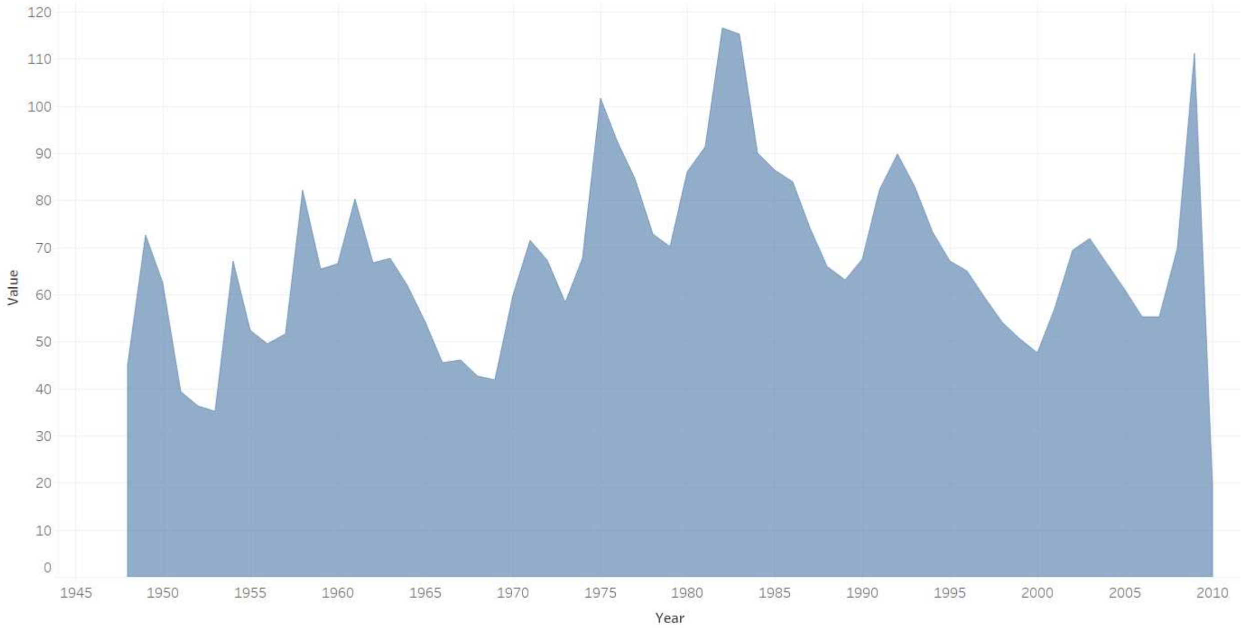
Tree Map

Unemployment rate summed up by year

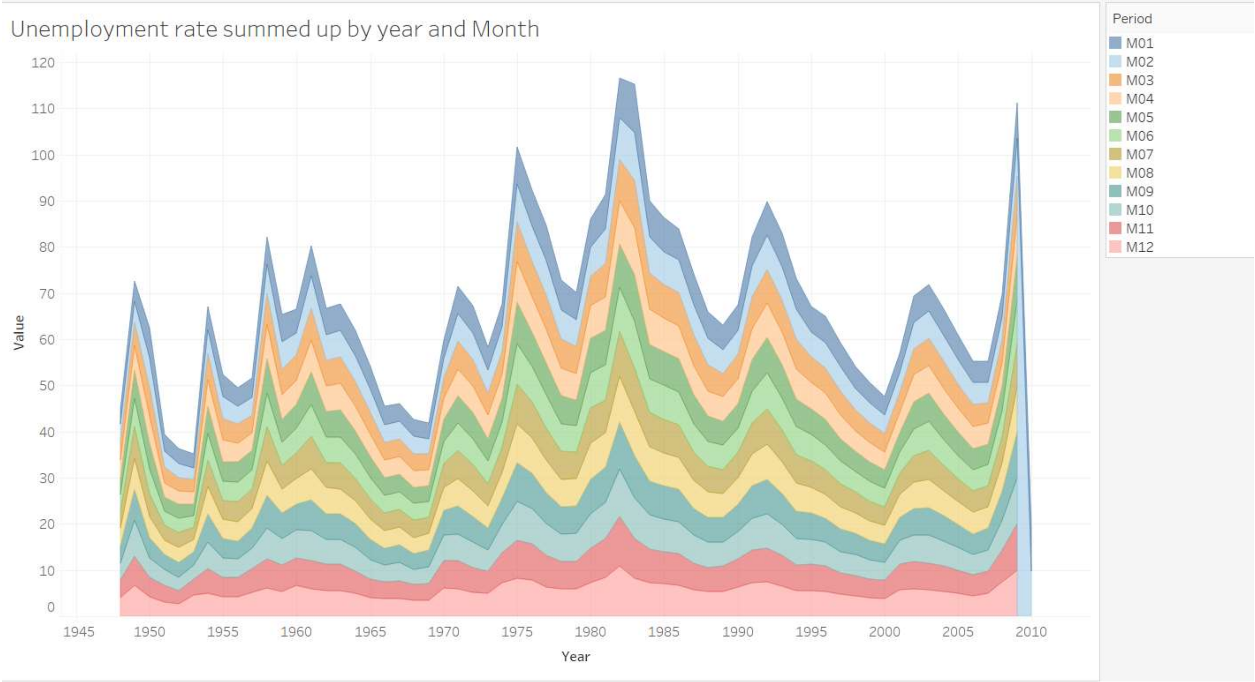


Area Chart

Unemployment rate summed up by year



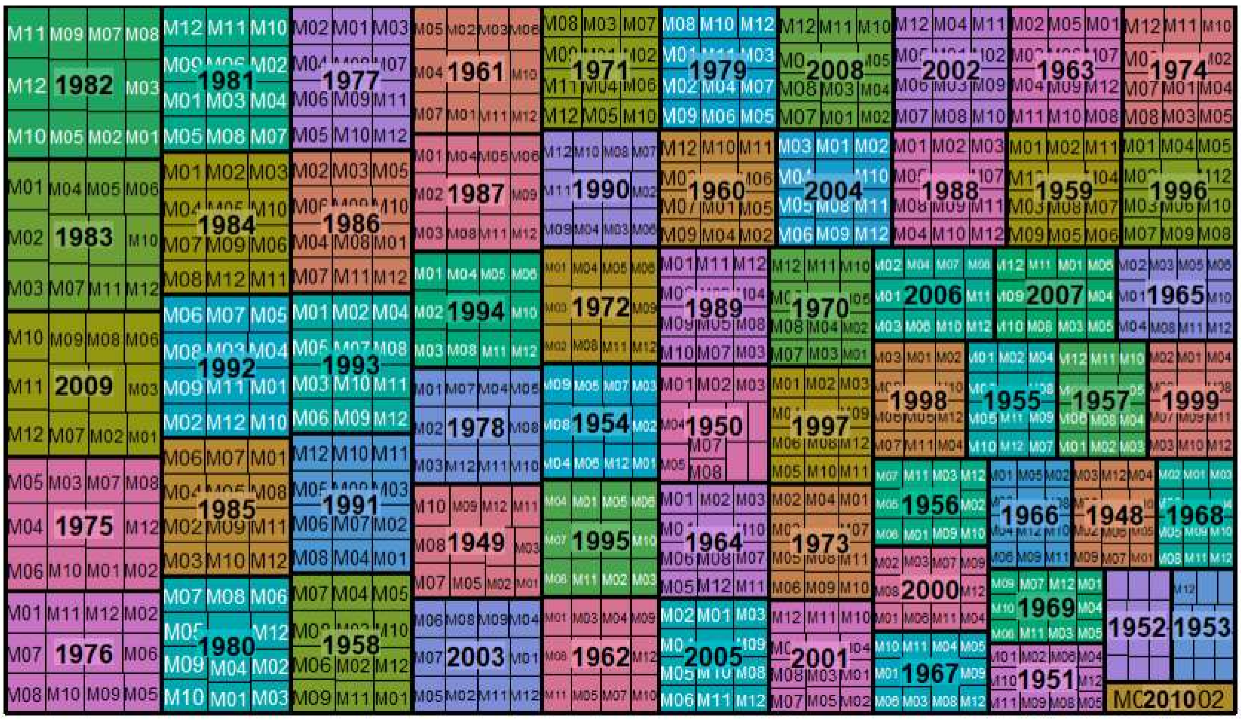
Stacked Area Chart



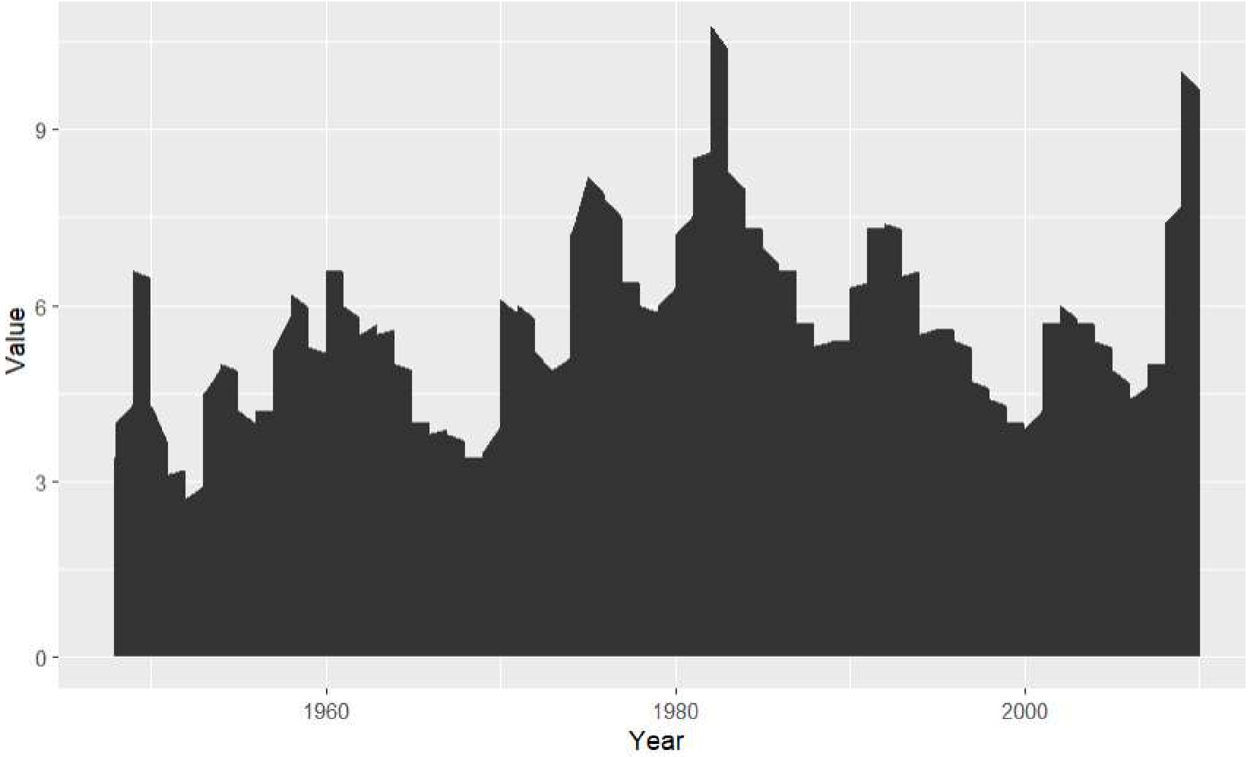
RSTUDIO

Treemap

Thought this was the coolest one! Really easy to setup the tree map to code by year and Month

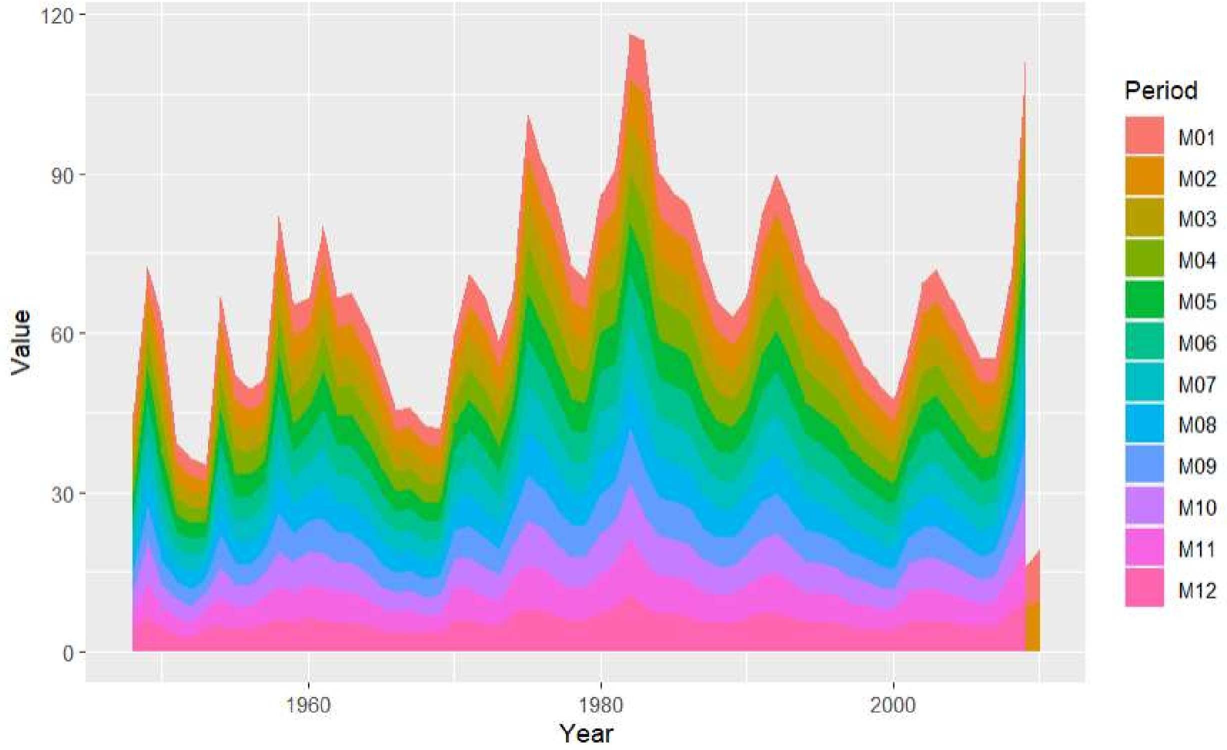


Area Chart



Stacked Area Chart

Also so cool looking in rstudio!



~~~
---
title: "week5"
output: html_document
date: "2024-01-15"
---

```{r setup, include=FALSE}
setwd("C:/Users/brean/OneDrive/Desktop/640/week5/breannaparkerdscc640week5")
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown


```{r cars}
library(readr)

data <- read.csv("unemployement_rate_1948_2010.csv")
data


```

## Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}
library(treemap)
#treemap
treemap(data,
            index= c("Year","Period"),
            vSize="Value",
            type="index"
            )

```

```{r pressure, echo=FALSE}

#area chart
library(ggplot2)
library(dplyr)

ggplot(data, aes(x=Year, y=Value)) +
    geom_area()

```

```{r pressure, echo=FALSE}
#stacked area chart
library(ggplot2)
library(dplyr)

ggplot(data, aes(x=Year, y=Value, fill=Period)) +
    geom_area()

```
~~~

In [44]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import squarify as sq
import matplotlib.dates as mdates
from matplotlib.dates import DateFormatter
```

In [2]:

```
#import data to dataframe
df=pd.read_csv('unemployement_rate_1948_2010.csv')
df.head()
```

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

In [61]:

```
#Treemap
label=df["Year"]
sq.plot(df['Value'], label = label).set(title='Unemployment Rate by Month')
plt.axis('off')
```

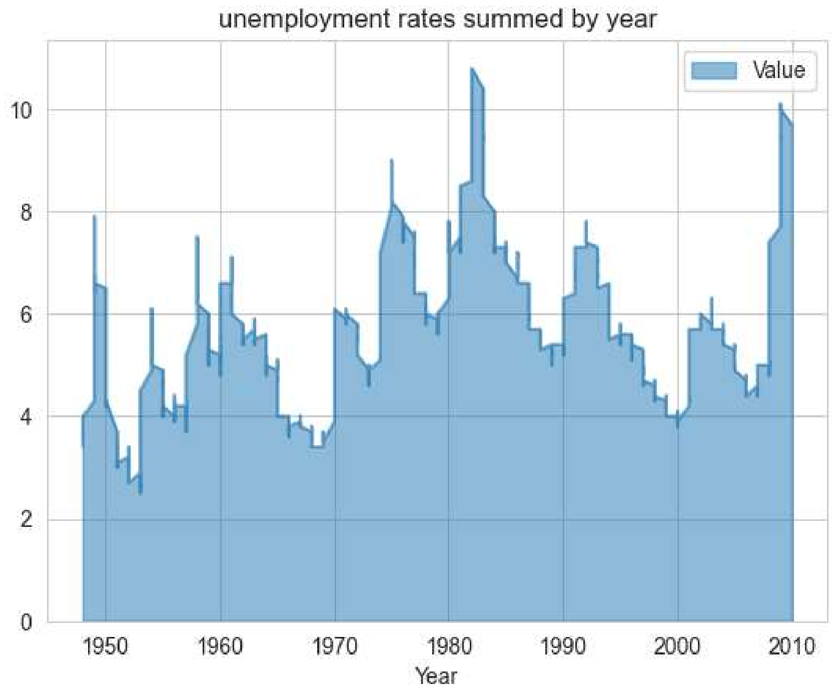
```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[61], line 4
      2 label=df["Year"]
      3 sq.plot(df['Value'], label = label).set(title='Unemployment Rate by Month')
----> 4 plt.axis('off')

AttributeError: 'list' object has no attribute 'axis'
```

Unemployment Rate by Month
100
80
60
40
20
0
0
20
40
60
80
100

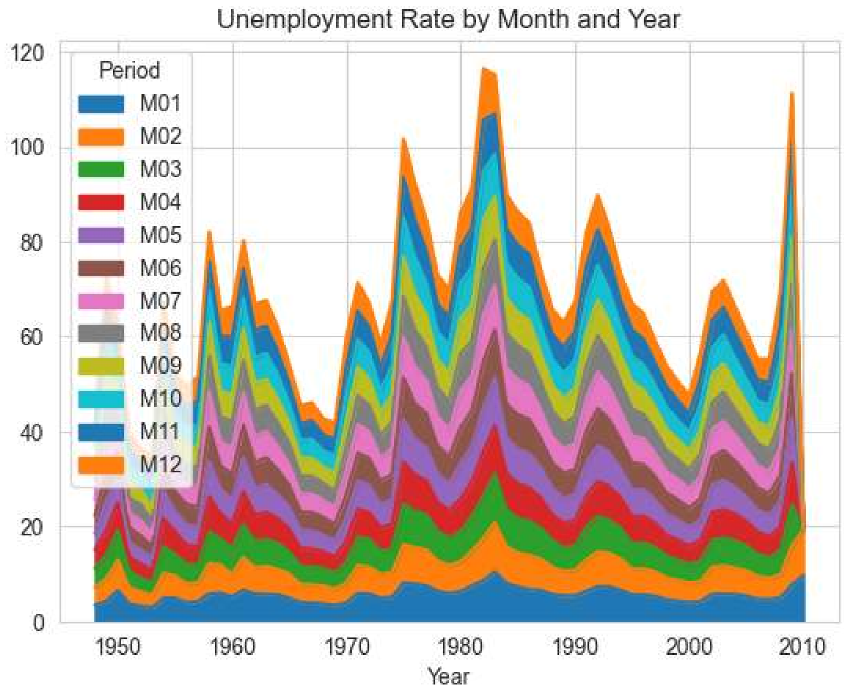
```
In [28]: #stacked area chart
         df.plot.area(x = 'Year', y = 'Value', stacked=False)
         plt.title("unemployment rates summed by year")
```

```
Out[28]: Text(0.5, 1.0, 'unemployment rates summed by year')
```



```
In [59]: #Stacked Area Chart
         df_unstack = df.groupby(['Year', 'Period']).sum().unstack()
         plt = df_unstack.plot(kind='area',y='Value', stacked = True).set(title ="Unemployment Rate by Month and Yea
```

```
C:\Users\brean\AppData\Local\Temp\ipykernel_39368\4239037196.py:2: FutureWarning: The default value of num
eric_only in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will default to False.
Either specify numeric_only or select only columns which should be valid for the function.
  df_unstack = df.groupby(['Year', 'Period']).sum().unstack()
```



```
In [ ]:
```