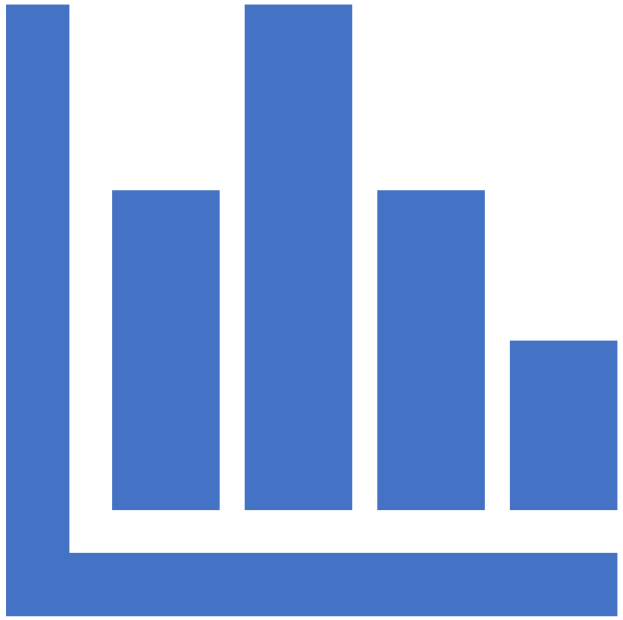


Business Intelligence Project

Vladomir Jungkman

Professor Bagley

ISM 4300

Data Overview

- Motor Fuel Taxes
- Obtained by Data.gov
  - Collected from unfiltered state reports

	state	month	year	tax type	value	fips state	numeric month	note	id
0	Delaware	July	2023	motor fuel	NaN	10	7	NaN	10_2023_7
1	Idaho	January	2021	motor fuel	22273106.0	16	1	NaN	16_2021_1
2	Alabama	August	2023	motor fuel	81898874.0	1	8	NaN	1_2023_8
3	Maine	February	2021	motor fuel	16310504.0	23	2	NaN	23_2021_2
4	Alabama	September	2023	motor fuel	85792362.0	1	9	NaN	1_2023_9
...	...	...	...	...	...	...	...	...	...
2749	Wisconsin	May	2023	motor fuel	77753040.0	55	5	NaN	55_2023_5
2750	Wyoming	April	2023	motor fuel	NaN	56	4	NaN	56_2023_4
2751	Wyoming	July	2023	motor fuel	NaN	56	7	NaN	56_2023_7
2752	Wyoming	June	2023	motor fuel	NaN	56	6	NaN	56_2023_6
2753	Wyoming	May	2023	motor fuel	NaN	56	5	NaN	56_2023_5

2754 rows × 9 columns

Business Questions to be Answered

Which regions and states pay the most motor fuel taxes?

Which month has the highest tax payment?

Which time of the year has higher taxes?

Is there a trend of increasing rates in Florida?

Cleaning Project Steps

Numeric and Null Checks

```
df.drop(['note','numeric month'],axis=1,inplace=True)
df['count'] = 1
pd.to_numeric(df['value'])
```

```
0              NaN
1       22273106.0
2       81898874.0
3       16310504.0
4       85792362.0
           ...
2749    77753040.0
2750           NaN
2751           NaN
2752           NaN
2753           NaN
Name: value, Length: 2754, dtype: float64
```

```
null=df[df.isnull().any(axis=1)]
#Seems to be that values are null, want to see if there are other values that are null
null.loc[(null['value'] >= 0) | (null['value'] < 0), 'test'] = 'True'
null.loc[null['test'] == 'True' ]
#This shows that all the null values are with the value column, because of this we cannot really change the nulls to 0
#This is because it skews the data, meaning we have to get rid of these null values
```

state	month	year	tax type	value	fips state	id	count	test

Cleaning Project Steps

Null Analysis

```
df2=df.dropna()
#Seems to be that about 1/5 (481/2273) of rows have null values of the tax cost
#May need to replace some null values with the average of most values to that correlated state
print("Total Number of null values:",len(null),"Total Number of rows from df without null values:",len(df2))
```

```
Total Number of null values: 481 Total Number of rows from df without null values: 2273
```

```
#Creating a count for merge between df of null values and df of filled values
df2count=df2.groupby(['state'])['count'].sum()
nullcount=null.groupby(['state'])['count'].sum()
df2count = df2count.to_frame()
nullcount = nullcount.to_frame()
nullcount['newcount']=nullcount['count']
nullcount=nullcount.drop(['count'],axis=1)
```

```
compare=nullcount.merge(df2count, how='left', on=['state'])
compare=compare.reset_index()
compare
#If the newcount(count of nulls) is 51 or 50, using the average does not make sense at all, meaning
#Cannot use Kansas, Montana,North Dakota, Rhode Island, South Dakota, Wyoming
```

	state	newcount	count
0	Alaska	1	53.0
1	Connecticut	3	51.0
2	Delaware	43	11.0
3	District of Columbia	33	21.0
4	Florida	5	49.0
5	Idaho	3	51.0
6	Kansas	54	NaN
7	Maryland	4	50.0
8	Missouri	4	50.0
9	Montana	53	1.0
10	Nevada	41	13.0
11	New Jersey	3	51.0
12	New Mexico	4	50.0
13	North Dakota	54	NaN
14	Ohio	2	52.0
15	Rhode Island	53	1.0
16	South Dakota	54	NaN
17	Virginia	1	53.0
18	Washington	12	42.0
19	Wyoming	54	NaN

Cleaning Project Steps

Null Replacement

```
#Creates a clean data frame by getting rid of the states with null values of 50 or 51
cleandf = pd.DataFrame(columns=['state', 'month', 'year','tax type','value','fips state','id','count'])
removevalues = ['District of Columbia','Kansas','Montana','North Dakota','Rhode Island','South Dakota','Wyoming']
listnull=df['state'].tolist()
listnull=set(listnull)
for i in removevalues:
    listnull.remove(i)
for i in listnull:
    testdf=df.dropna()
    filldf=df.loc[df['state']==i]
    example=testdf.loc[df['state']==i]
    average=example['value'].mean()
    filldf=filldf.fillna(average)
    cleandf=cleandf.append(filldf)
    #Have this print to show that these values match the exponential values
    print(i)
    print(average)
```

Cleaning Project Steps

New Variables and Finalization

```
#the exponential value is used because of how big the values are, python automatically converts it
cleandf=cleandf.drop(['count','tax type'],axis=1)
```

```
#Make Region: Assigning state of each row to a region, state of null are included in this code but still will not have effect
#Got what state goes into what region from https://www.mappr.co/political-maps/us-regions-map/
cleandf.loc[(cleandf['state'] == 'Arkansas') | (cleandf['state'] == 'Alabama') | (cleandf['state'] == 'Georgia')\
            | (cleandf['state'] == 'Virginia') | (cleandf['state'] == 'Florida')| (cleandf['state'] == 'Kentucky')\
            | (cleandf['state'] == 'Louisiana')| (cleandf['state'] == 'Mississippi')| (cleandf['state'] == 'North Carolina') \
            | (cleandf['state'] == 'South Carolina')| (cleandf['state'] == 'Tennessee')| (cleandf['state'] == 'West Virginia')\
           ,'Region'] = 'Southeast'
cleandf.loc[(cleandf['state']=='Connecticut')| (cleandf['state'] == 'Delaware')| (cleandf['state'] == 'Maine') \
            | (cleandf['state'] == 'Maryland')| (cleandf['state'] == 'Massachusetts') | (cleandf['state'] == 'New Hampshire')\
            | (cleandf['state'] == 'New Jersey') | (cleandf['state'] == 'New York')| (cleandf['state'] == 'Pennsylvania') \
            | (cleandf['state'] == 'Rhode Island')| (cleandf['state'] == 'Vermont'),'Region']='Northeast'
cleandf.loc[(cleandf['state']=='Arizona') | (cleandf['state']=='New Mexico')| (cleandf['state']=='Oklahoma') \
            | (cleandf['state']=='Texas'),'Region']='Southwest'
cleandf.loc[(cleandf['state']=='Alaska')| (cleandf['state']=='California')| (cleandf['state']=='Colorado') \
             | (cleandf['state']=='Hawaii')| (cleandf['state']=='Idaho')| (cleandf['state']=='Montana') \
             | (cleandf['state']=='Nevada')| (cleandf['state']=='Oregon')| (cleandf['state']=='Utah') \
             | (cleandf['state']=='Washington')| (cleandf['state']=='Wyoming'),'Region']='West'
cleandf.loc[(cleandf['state']=='Illinois')| (cleandf['state']=='Indiana')| (cleandf['state']=='Iowa') \
            | (cleandf['state']=='Kansas')| (cleandf['state']=='Michigan')| (cleandf['state']=='Minnesota') \
            | (cleandf['state']=='Missouri')| (cleandf['state']=='Nebraska')| (cleandf['state']=='North Dakota') \
            | (cleandf['state']=='Ohio')| (cleandf['state']=='South Dakota')| (cleandf['state']=='Wisconsin') \
           ,'Region']='Midwest'
```

```
#Make Season: Add a season based on the month, will be used to see the most expensive seasons
cleandf.loc[(cleandf['month']=='December')|(cleandf['month']=='January')|(cleandf['month']=='February'),'Season']='Winter'
cleandf.loc[(cleandf['month']=='March')|(cleandf['month']=='April')|(cleandf['month']=='May'),'Season']='Spring'
cleandf.loc[(cleandf['month']=='June')|(cleandf['month']=='July')|(cleandf['month']=='August'),'Season']='Summer'
cleandf.loc[(cleandf['month']=='September')|(cleandf['month']=='October')|(cleandf['month']=='November'),'Season']='Fall'
```

```
#Do not really need the fips state since we already have the state name
cleandf=cleandf.drop(['fips state'],axis=1)
```

```
cleandf.to_csv("GovData.csv", sep=',', index=False, encoding='utf-8')
```

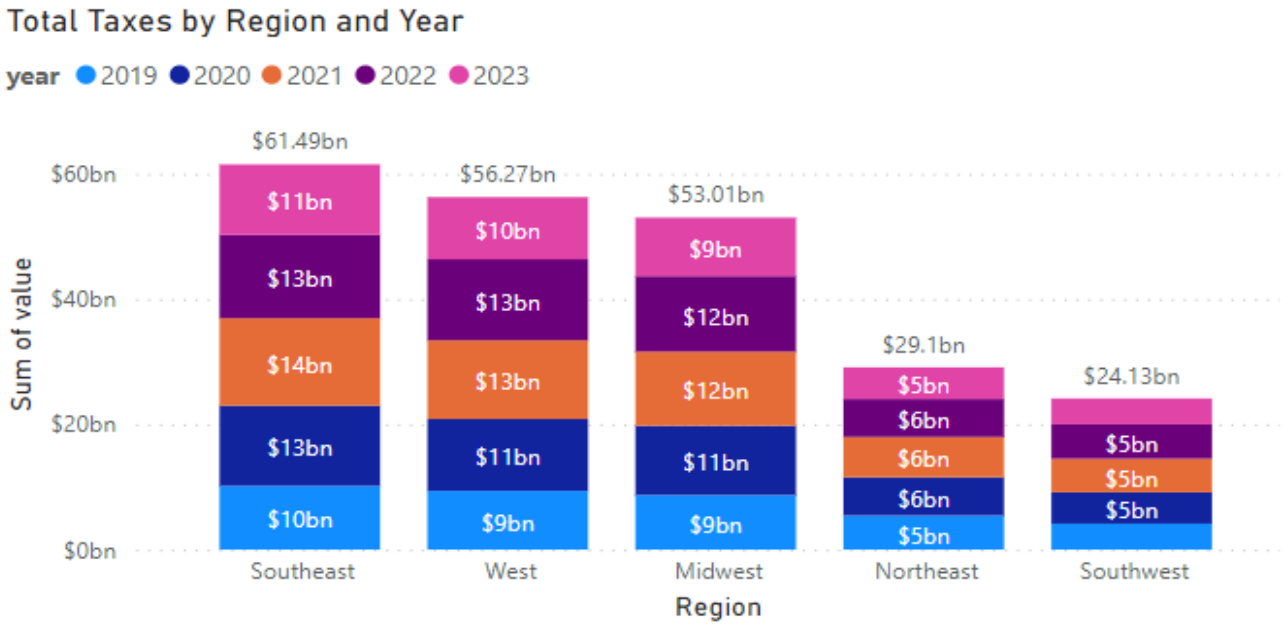
Cleaned DF

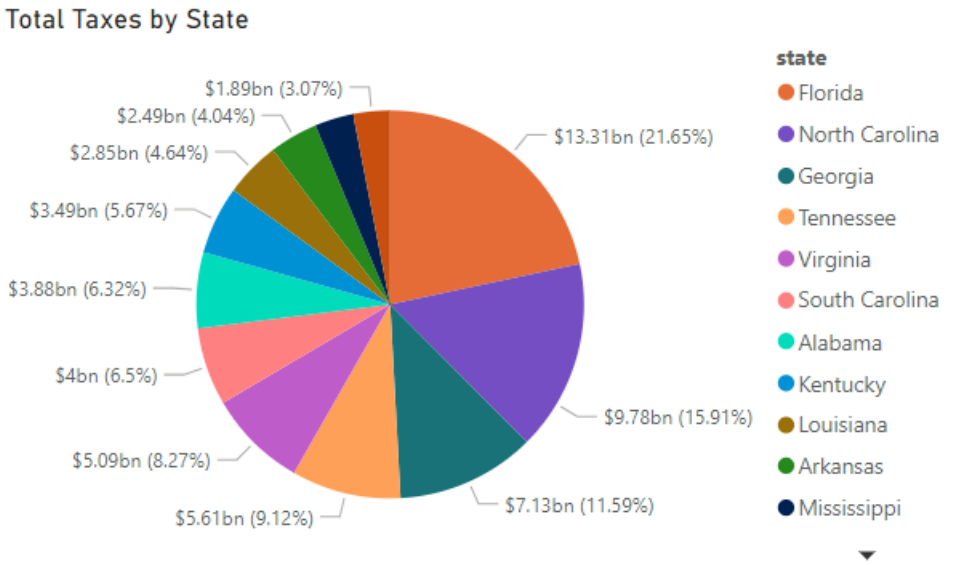
```
cleandf
```

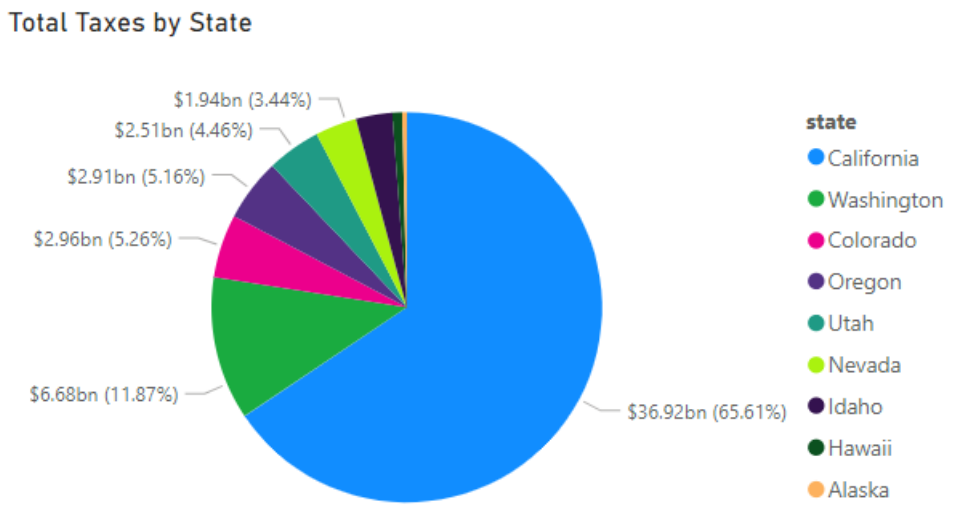
	state	month	year	value	id	Region	Season
12	California	August	2023	745427000.0	6_2023_8	West	Summer
13	California	September	2023	841505000.0	6_2023_9	West	Fall
254	California	April	2019	496996000.0	6_2019_4	West	Spring
255	California	April	2020	548318000.0	6_2020_4	West	Spring
256	California	April	2021	615441000.0	6_2021_4	West	Spring
...	...	...	...	...	...	...	...
2545	Vermont	March	2023	7790474.0	50_2023_3	Northeast	Spring
2730	Vermont	April	2023	6843191.0	50_2023_4	Northeast	Spring
2731	Vermont	July	2023	7578053.0	50_2023_7	Northeast	Summer
2732	Vermont	June	2023	7828634.0	50_2023_6	Northeast	Summer
2733	Vermont	May	2023	6495663.0	50_2023_5	Northeast	Spring

2376 rows × 7 columns

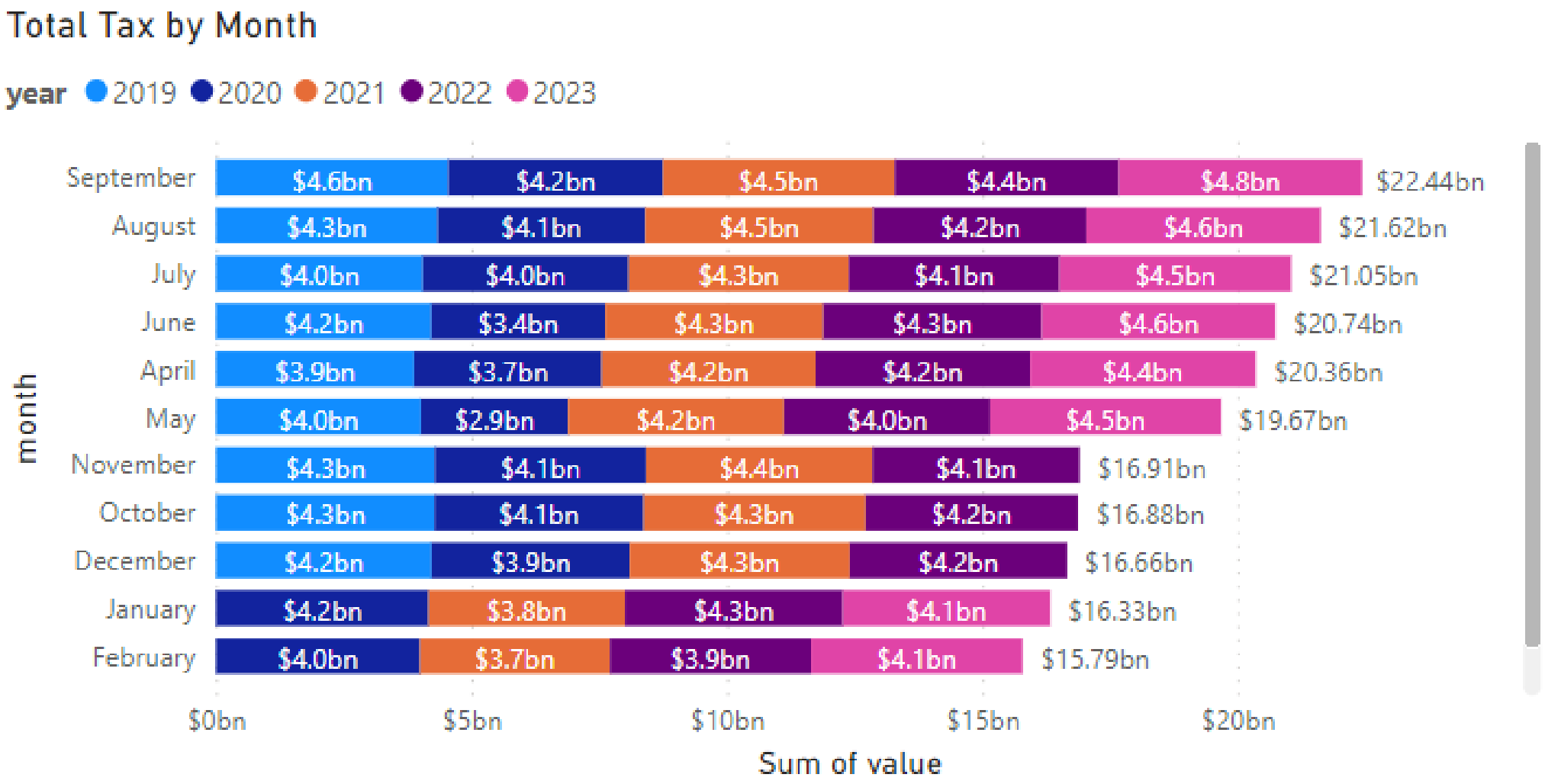
BI Q #1:Which regions and states pay the most taxes?



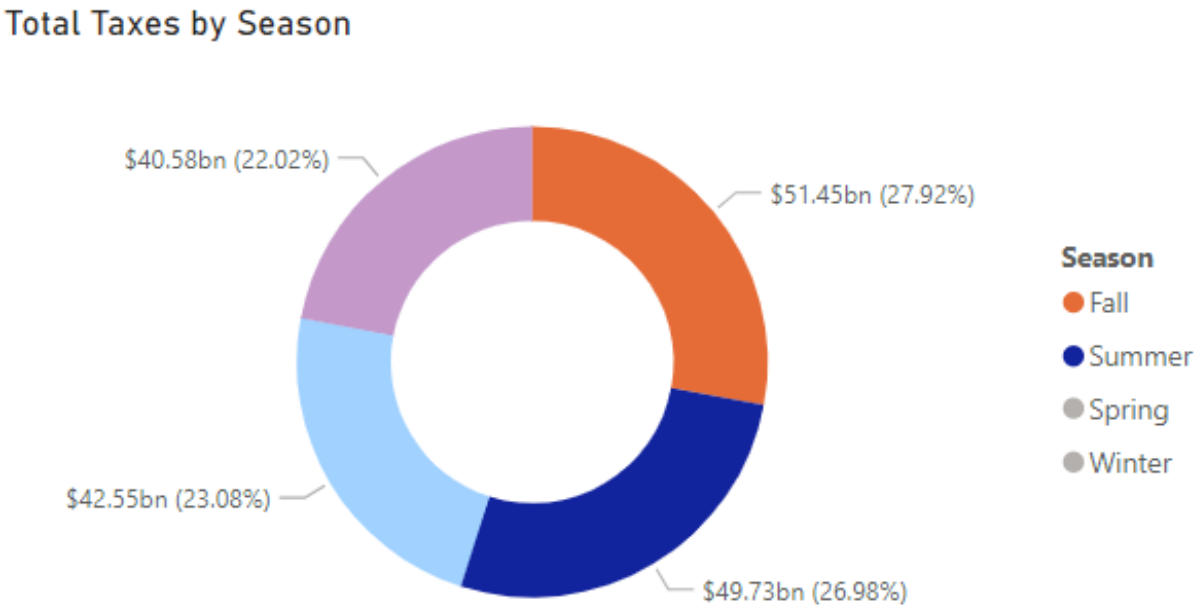


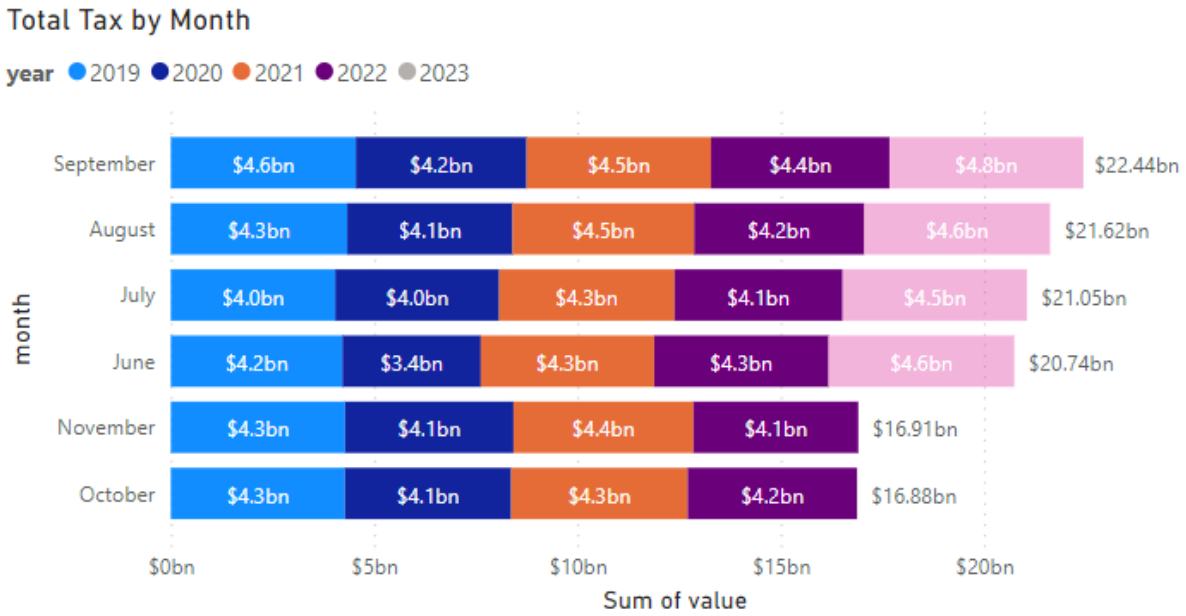


BI Q #2:Which month has the highest tax payment?

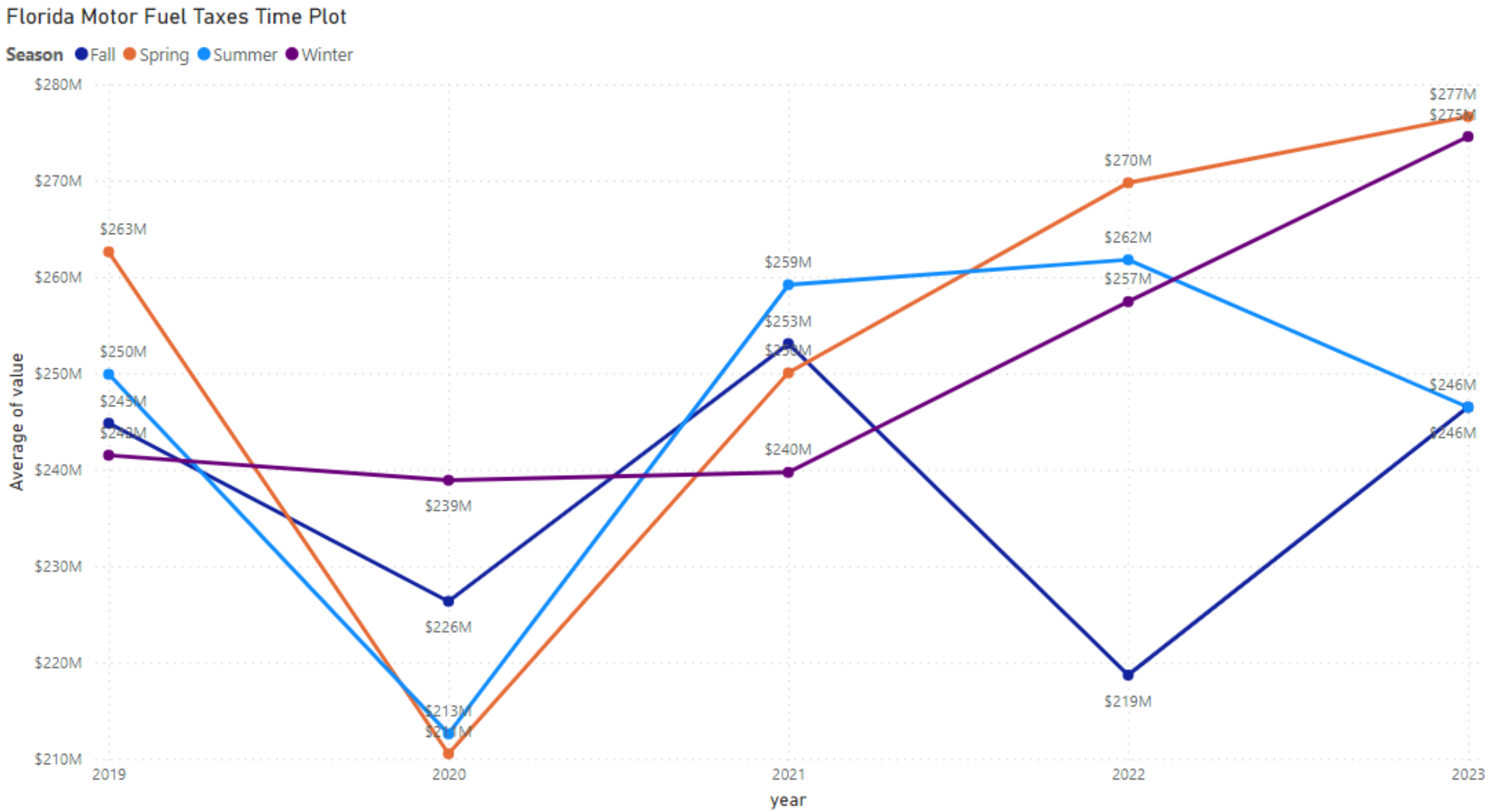


BI Q #3:Which time of the year has higher taxes?

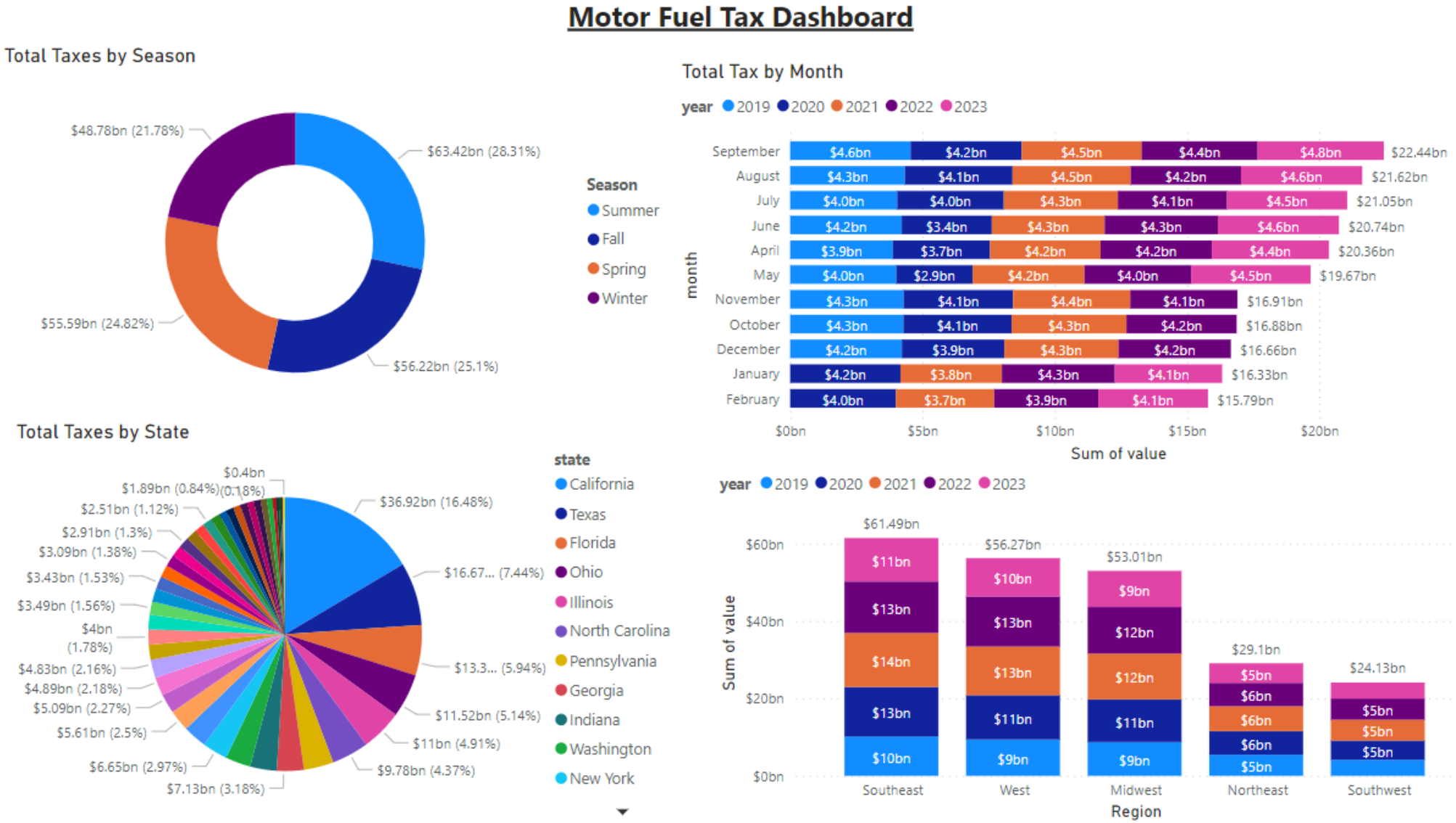




Is there a trend of increasing rates in Florida?



Dashboard


Motor Fuel Tax Dashboard
Total Taxes by Season
$48.78bn (21.78%)
$63.42bn (28.31%)
$55.59bn (24.82%)
$56.22bn (25.1%)
Season
Summer
Fall
Spring
Winter
Total Tax by Month
year 2019 2020 2021 2022 2023
September $4.6bn $4.2bn $4.5bn $4.4bn $4.8bn $22.44bn
August $4.3bn $4.1bn $4.5bn $4.2bn $4.6bn $21.62bn
July $4.0bn $4.0bn $4.3bn $4.1bn $4.5bn $21.05bn
June $4.2bn $3.4bn $4.3bn $4.3bn $4.6bn $20.74bn
April $3.9bn $3.7bn $4.2bn $4.2bn $4.4bn $20.36bn
May $4.0bn $2.9bn $4.2bn $4.0bn $4.5bn $19.67bn
November $4.3bn $4.1bn $4.4bn $4.1bn $16.91bn
October $4.3bn $4.1bn $4.3bn $4.2bn $16.88bn
December $4.2bn $3.9bn $4.3bn $4.2bn $16.66bn
January $4.2bn $3.8bn $4.3bn $4.1bn $16.33bn
February $4.0bn $3.7bn $3.9bn $4.1bn $15.79bn
month
$0bn $5bn $10bn $15bn $20bn
Sum of value
Total Taxes by State
$0.4bn
$1.89bn (0.84%)
$2.51bn (1.12%)
$2.91bn (1.3%)
$3.09bn (1.38%)
$3.43bn (1.53%)
$3.49bn (1.56%)
$4bn (1.78%)
$4.83bn (2.16%)
$4.89bn (2.18%)
$5.09bn (2.27%)
$5.61bn (2.5%)
$6.65bn (2.97%)
$7.13bn (3.18%)
$36.92bn (16.48%)
$16.67... (7.44%)
$13.3... (5.94%)
$11.52bn (5.14%)
$11bn (4.91%)
$9.78bn (4.37%)
state
California
Texas
Florida
Ohio
Illinois
North Carolina
Pennsylvania
Georgia
Indiana
Washington
New York
year 2019 2020 2021 2022 2023
$61.49bn
$56.27bn
$53.01bn
$29.1bn
$24.13bn
$60bn
$40bn
$20bn
$0bn
Sum of value
Southeast
West
Midwest
Northeast
Southwest
Region

References

- “State FIPS Code Listing.” Www.mercercountypa.gov, www.mercercountypa.gov/dps/state_fips_code_listing.htm.
- “Sales Tax Collections by State.” Data.gov, Bureau of Transportation Statistics, 9 Nov. 2023, catalog.data.gov/dataset/sales-tax-collections-by-state. Accessed 13 Nov. 2023.
- “What Is Motor Fuel Tax? - CountyOffice.org.” Www.youtube.com, www.youtube.com/watch?v=otV4r--tv20. Accessed 13 Nov. 2023.
- “Florida Dept. Of Revenue - Home.” Floridarevenue.com, floridarevenue.com/MotorFuel/Pages/default.aspx. Accessed 13 Nov. 2023.
- “Gas Taxes Just Went up in These 9 States – Forbes Advisor.” Www.forbes.com, www.forbes.com/advisor/personal-finance/gas-tax-increases/. Accessed 13 Nov. 2023.
- “5 US Regions Map and Facts | Mappr.” Www.mappr.co, 27 Dec. 2021, www.mappr.co/political-maps/us-regions-map/.
- “AAA Gas Prices.” AAA Gas Prices, gasprices.aaa.com/?state=FL.