In [1]:
```
import pandas as pd
import numpy as np
import altair as alt
```

In [2]:
```
ghg = pd.read_csv('data/Annual_GHG.csv', encoding = 'latin-1')

ghg = ghg.loc[:,['Country', 'Industry','Gas_Type'\
                 ,'F2017','F2018','F2019','F2020','F2021']]
ghg.head(5)
```

Out[2]:

	Country	Industry	Gas_Type	F2017	F2018	F2019	F2020	F2021
0	Advanced Economies	Agriculture, Forestry and Fishing	Carbon dioxide	193.054238	191.720412	191.165538	187.134711	193.971754
1	Advanced Economies	Agriculture, Forestry and Fishing	Fluorinated gases	0.982652	0.851009	0.816072	0.778334	0.729931
2	Advanced Economies	Agriculture, Forestry and Fishing	Greenhouse gas	1380.725829	1388.771814	1386.321969	1352.443269	1345.263619
3	Advanced Economies	Agriculture, Forestry and Fishing	Methane	618.262461	620.189092	613.713837	611.371924	592.283622
4	Advanced Economies	Agriculture, Forestry and Fishing	Nitrous oxide	568.426479	576.011301	580.626521	553.158300	558.278313

Tidying

Each observation represents a region's emissions of a specific gas type in a given industry from 2017-2021. The data is already tidy, but for clarity's sake we will rename the `Country` column to `Region`.

In [3]:
```
ghg = ghg.rename(columns = {'Country': 'Region', 'Gas_Type': 'Gas Type'})
```

In [4]:
```
ghg = ghg[ghg['Region'] != 'Advanced Economies']
ghg = ghg[ghg['Region'] != 'Emerging and Developing Economies']
ghg = ghg[ghg['Region'] != 'G7']
ghg = ghg[ghg['Region'] != 'G20']
```

Next Steps

Now that we have a good idea about what our data looks like and what it means, we can begin to investigate some potnetial research questions.

In this project, there are two main questions we want to answer:

1. What is the relationship between industry and emissions? Which industries produce the most emissions, and which have increased/decreased their emissions over time?

2. How does gas type influence emissions over time? Do certain gases require/result in increased emissions? Do certain regions emit more of any specific gases?

Both of these questions will benefit from linear regression models in order to investigate the relationship between these variables over time. We will also further use visual analysis to look at the relationships between industry and emission as well as region and emission in regards to specific gases.

Result

In [5]:
```
ghg.groupby(['Industry']).sum()
```

```
C:\Users\29749\AppData\Local\Temp\ipykernel_13008\2227959074.py:1: FutureWarning: The default value of numeric_only in DataFr
ameGroupBy.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.
  ghg.groupby(['Industry']).sum()
```

Out[5]:

	F2017	F2018	F2019	F2020	F2021
Industry					
Agriculture, Forestry and Fishing	38181.345821	38392.521484	38413.370571	38727.287384	39213.042665
Construction	14877.244782	15172.587645	15544.285356	15503.260886	16250.010071
Electricity, Gas, Steam and Air Conditioning Supply	81694.270546	84240.705200	83534.016436	80688.271312	85120.772825
Manufacturing	57073.444403	58361.955057	59027.677650	58542.183986	60682.108343
Mining	19112.448003	19681.773850	20008.119467	19473.442237	20178.609955
Other Services Industries	18256.076010	18343.626381	18389.967461	17379.542298	18302.066947
Total Households	32206.591572	32198.768033	32221.320252	29468.538059	31119.448901
Total Industry and Households	296563.015538	302475.961891	303550.933730	293510.231613	305775.035922
Transportation and Storage	19306.299907	19890.011195	19931.930135	16924.147254	17922.807699
Water supply; sewerage, waste management and remediation activities	15855.294498	16194.013037	16480.246415	16803.558191	16986.168511

As we can see in the table above, `Electricity, Gas, Steam and Air Conditioning Supply` has the most emission for five years consecutively.

In [6]:

```
df = ghg[['Industry','F2017','F2018','F2019','F2020','F2021','Gas Type']].copy().reset_index().drop(columns = 'index')\
.rename(columns={'F2017': 2017,'F2018':2018,'F2019':2019,'F2020':2020,'F2021':2021})
df
```

Out[6]:

	Industry	2017	2018	2019	2020	2021	Gas Type
0	Agriculture, Forestry and Fishing	8.598177	8.885628	9.193573	9.519654	9.869859	Carbon dioxide
1	Agriculture, Forestry and Fishing	801.551149	820.263383	841.909644	859.337790	875.720444	Greenhouse gas
2	Agriculture, Forestry and Fishing	533.778860	548.656038	563.256921	574.852827	584.961495	Methane
3	Agriculture, Forestry and Fishing	259.174112	262.721717	269.459149	274.965308	280.889091	Nitrous oxide
4	Construction	93.987951	95.613220	96.690936	87.940823	93.301037	Carbon dioxide
...	...	...	...	...	...	...	...
934	Water supply; sewerage, waste management and r...	212.625849	219.993696	226.031515	227.326438	229.040646	Carbon dioxide
935	Water supply; sewerage, waste management and r...	12.641309	13.518851	13.771323	13.137126	12.133499	Fluorinated gases
936	Water supply; sewerage, waste management and r...	2642.549083	2699.002173	2746.707736	2800.593032	2831.028085	Greenhouse gas
937	Water supply; sewerage, waste management and r...	2284.968589	2331.721563	2371.492339	2422.875824	2450.209878	Methane
938	Water supply; sewerage, waste management and r...	132.313337	133.768063	135.412558	137.253644	139.644062	Nitrous oxide

939 rows × 7 columns

In [7]:

```
df = df.melt(id_vars = ['Industry','Gas Type'], value_vars = [2017,2018,2019,2020,2021],\
        var_name = 'Year', value_name = 'Measurement')
```

In [8]:

```
df
```

Out[8]:

	Industry	Gas Type	Year	Measurement
0	Agriculture, Forestry and Fishing	Carbon dioxide	2017	8.598177
1	Agriculture, Forestry and Fishing	Greenhouse gas	2017	801.551149
2	Agriculture, Forestry and Fishing	Methane	2017	533.778860
3	Agriculture, Forestry and Fishing	Nitrous oxide	2017	259.174112
4	Construction	Carbon dioxide	2017	93.987951
...	...	...	...	...
4690	Water supply; sewerage, waste management and r...	Carbon dioxide	2021	229.040646
4691	Water supply; sewerage, waste management and r...	Fluorinated gases	2021	12.133499
4692	Water supply; sewerage, waste management and r...	Greenhouse gas	2021	2831.028085
4693	Water supply; sewerage, waste management and r...	Methane	2021	2450.209878
4694	Water supply; sewerage, waste management and r...	Nitrous oxide	2021	139.644062

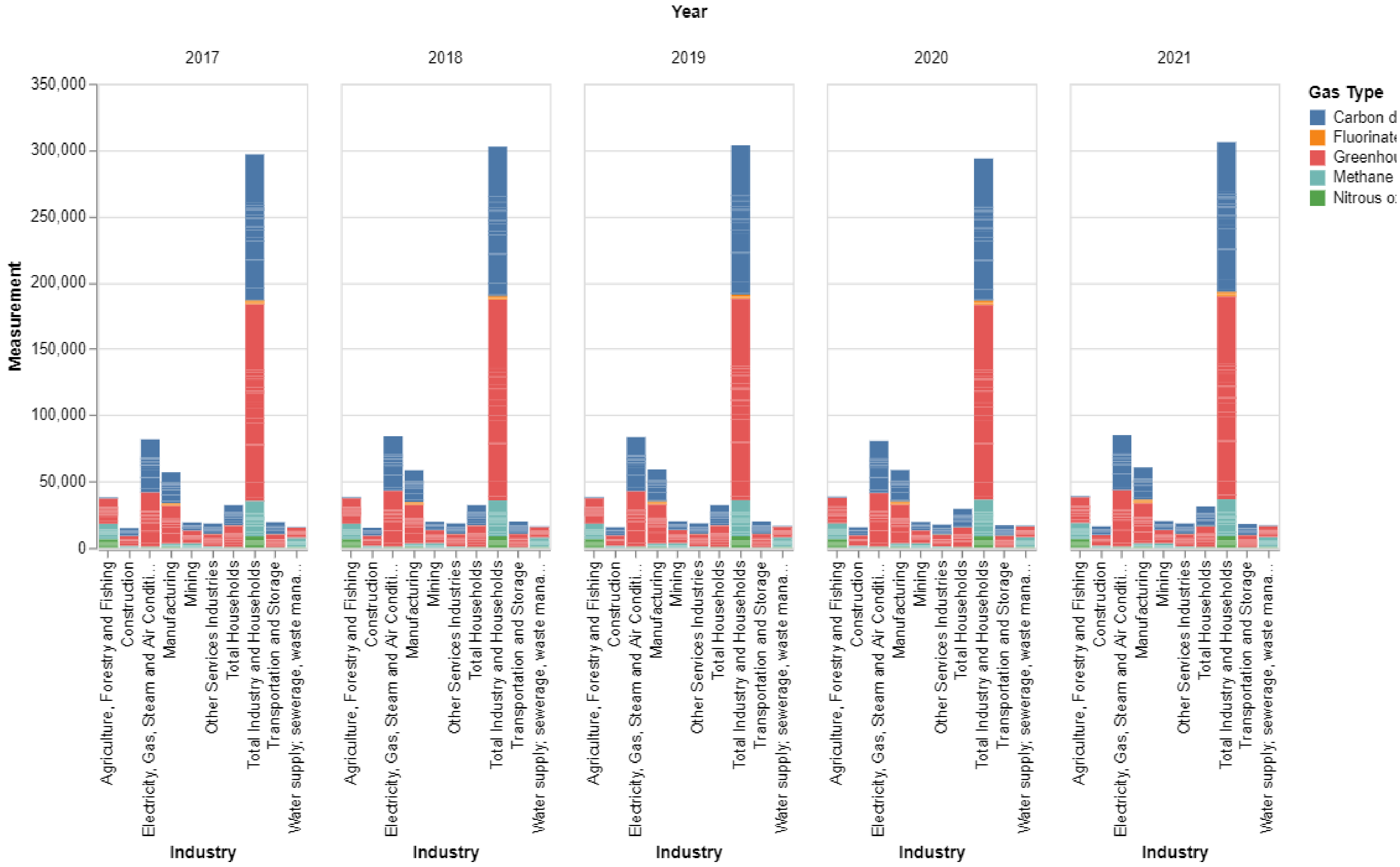
4695 rows × 4 columns

In [9]:

```
bar = alt.Chart(df).mark_bar().encode(
    x = 'Industry:N',
    y = 'Measurement:Q',
    color = 'Gas Type'
).properties(
width = 135,
height = 300).facet(column = 'Year')
bar
```

```
C:\Users\29749\AppData\Local\Programs\Python\Python310\lib\site-packages\altair\utils\core.py:317: FutureWarning: iteritems i
s deprecated and will be removed in a future version. Use .items instead.
  for col_name, dtype in df.dtypes.iteritems():
```

Out[9]:



In [ ]: