Group_19_Project

December 14, 2023

1 PSTAT 100 Final Project: Greenhouse Gas Emission Analysis

Group 19: Alaina Liu, Elena Markova, Kris Hao, Min Kang For our project we have chosen to analyze the ClimateWatch historical emissions data: greenhouse gas emissions by U.S. state 1990-present.

2 Creating the Dataset

[1]:
```
# import necessary dependencies
import numpy as np
import pandas as pd
import altair as alt
alt.renderers.enable('mimetype')

# don't show warning messages
import warnings
warnings.filterwarnings('ignore')

# to merge datasets
from functools import reduce

# loess
import statsmodels.api as sm


# read in ch4 csv files
CH4_capita = pd.read_csv("ghg-emissions-ch4-capita.csv")
CH4_gdp = pd.read_csv("ghg-emissions-ch4-gdp.csv")
CH4_total = pd.read_csv("ghg-emissions-ch4-total.csv")

# read in co2 csv files
CO2_capita = pd.read_csv("ghg-emissions-co2-capita.csv")
CO2_gdp = pd.read_csv("ghg-emissions-co2-gdp.csv")
CO2_total = pd.read_csv("ghg-emissions-co2-total.csv")

# read in fgas csv files
fgas_capita = pd.read_csv("ghg-emissions-fgas-capita.csv")
```

```
fgas_gdp = pd.read_csv("ghg-emissions-fgas-gdp.csv")
fgas_total = pd.read_csv("ghg-emissions-fgas-total.csv")

# read in n2o csv files
N2O_capita = pd.read_csv("ghg-emissions-n2o-capita.csv")
N2O_gdp = pd.read_csv("ghg-emissions-n2o-gdp.csv")
N2O_total = pd.read_csv("ghg-emissions-n2o-total.csv")


# ch4: dropping the unit column and keeping iso, state, and unit. we melt the␣
↪year variable to turn them from a column per year into rows
CH4_capita_melt = pd.melt(CH4_capita, id_vars = ["iso", "Country/Region",␣
↪"unit"], var_name = "Year", value_name = "CH4pC").drop(columns = "unit")
CH4_gdp_melt = pd.melt(CH4_gdp, id_vars = ["iso", "Country/Region", "unit"],␣
↪var_name = "Year", value_name = "CH4pGDP").drop(columns = "unit")
CH4_total_melt = pd.melt(CH4_total, id_vars = ["iso", "Country/Region",␣
↪"unit"], var_name = "Year", value_name = "CH4TOT").drop(columns = "unit")

# co2: dropping the unit column and keeping iso, state, and unit. we melt the␣
↪year variable to turn them from a column per year into rows
CO2_capita_melt = pd.melt(CO2_capita, id_vars = ["iso", "Country/Region",␣
↪"unit"], var_name = "Year", value_name = "CO2pC").drop(columns = "unit")
CO2_gdp_melt = pd.melt(CO2_gdp, id_vars = ["iso", "Country/Region", "unit"],␣
↪var_name = "Year", value_name = "CO2pGDP").drop(columns = "unit")
CO2_total_melt = pd.melt(CO2_total, id_vars = ["iso", "Country/Region",␣
↪"unit"], var_name = "Year", value_name = "CO2TOT").drop(columns = "unit")

# fgas: dropping the unit column and keeping iso, state, and unit. we melt the␣
↪year variable to turn them from a column per year into rows
fgas_capita_melt = pd.melt(fgas_capita, id_vars = ["iso", "Country/Region",␣
↪"unit"], var_name = "Year", value_name = "FGaspC").drop(columns = "unit")
fgas_gdp_melt = pd.melt(fgas_gdp, id_vars = ["iso", "Country/Region", "unit"],␣
↪var_name = "Year", value_name = "FGaspGDP").drop(columns = "unit")
fgas_total_melt = pd.melt(fgas_total, id_vars = ["iso", "Country/Region",␣
↪"unit"], var_name = "Year", value_name = "FGasTOT").drop(columns = "unit")

# n2o: dropping the unit column and keeping iso, state, and unit. we melt the␣
↪year variable to turn them from a column per year into rows
N2O_capita_melt = pd.melt(N2O_capita, id_vars = ["iso", "Country/Region",␣
↪"unit"], var_name = "Year", value_name = "N2OpC").drop(columns = "unit")
N2O_gdp_melt = pd.melt(N2O_gdp, id_vars = ["iso", "Country/Region", "unit"],␣
↪var_name = "Year", value_name = "N2OpGDP").drop(columns = "unit")
N2O_total_melt = pd.melt(N2O_total, id_vars = ["iso", "Country/Region",␣
↪"unit"], var_name = "Year", value_name = "N2OTOT").drop(columns = "unit")
```

```
# creating list of all dataframes to prepare for merging
dfs = [CH4_capita_melt, CH4_gdp_melt, CH4_total_melt, CO2_capita_melt,␣
 ↪CO2_gdp_melt, CO2_total_melt, fgas_capita_melt, fgas_gdp_melt,␣
 ↪fgas_total_melt, N2O_capita_melt, N2O_gdp_melt, N2O_total_melt]

# using the reduce function to merge all the dataframes into one dataframe, on␣
 ↪iso, state, and year
ghg = reduce(lambda left, right: pd.merge(left, right, on = ["iso", "Country/
 ↪Region", "Year"], how = "left"), dfs)

# first 5 rows
ghg.head()


# renaming states column
ghg = ghg.rename(columns = {"Country/Region": "State"})

# remove 'USA.' from iso
ghg["iso"] = ghg["iso"].str.extract(r'USA\.(.*)')

# assigning regions
def get_region(state):
    if state in ['AK', 'CA', 'CO', 'HI', 'ID', 'MT', 'NV', 'OR', 'UT', 'WA',␣
 ↪'WY']:
        return 'West'
    elif state in ['AZ', 'NM', 'OK', 'TX']:
        return 'Southwest'
    elif state in ['IA', 'IL', 'IN', 'KS', 'MI', 'MN', 'MO', 'ND', 'NE', 'OH',␣
 ↪'SD', 'WI']:
        return 'Midwest'
    elif state in ['AL', 'AR', 'DC', 'DE', 'FL', 'GA', 'KY', 'LA', 'MD', 'MS',␣
 ↪'NC', 'SC', 'TN', 'VA', 'WV']:
        return 'Southeast'
    elif state in ['CT', 'MA', 'ME', 'NH', 'NJ', 'NY', 'PA', 'RI', 'VT']:
        return 'Northeast'
    else:
        return 'Other'
ghg["Region"] = ghg["iso"].apply(get_region)

# creating a column with the total greenhouse gas emissions
ghg["TOT"] = ghg["CH4TOT"] + ghg["CO2TOT"] + ghg["FGasTOT"] + ghg["N2OTOT"]

# creating a column with state gdp's (in trillions)
ghg["GDP"] = ((ghg["CH4TOT"] / ghg["CH4pGDP"]) + (ghg["CO2TOT"] /␣
 ↪ghg["CO2pGDP"]) + (ghg["FGasTOT"] / ghg["FGaspGDP"]) + (ghg["N2OTOT"] /␣
 ↪ghg["N2OpGDP"])) / 4
```

```python
# creating a column with state population (in millions)
ghg["Population"] = ((ghg["CH4TOT"] / ghg["CH4pC"]) + (ghg["CO2TOT"] /␣
↪ghg["CO2pC"]) + (ghg["FGasTOT"] / ghg["FGaspC"]) + (ghg["N2OTOT"] /␣
↪ghg["N2OpC"])) / 4

# final dataframe
ghg.head()

# checking for missing values - none
#ghg.isna().sum()
```

[1]:

```
  iso          State  Year  CH4pC  CH4pGDP  CH4TOT  CO2pC  CO2pGDP  CO2TOT  \
0  TX          Texas  1990   6.33   246.87  107.96  32.16  1254.40  548.54
1  CA     California  1990   2.15    71.05   64.37  10.42   344.40  312.06
2  PA   Pennsylvania  1990   3.13   128.85   37.23  18.61   766.75  221.56
3  OK       Oklahoma  1990  10.87   510.14   34.22  25.32  1188.46   79.73
4  WV  West Virginia  1990  20.10  1149.17   36.03  45.88  2623.43   82.24

   FGaspC  FGaspGDP  FGasTOT  N2OpC  N2OpGDP  N2OTOT     Region     TOT  \
0    0.26     10.31     4.51   3.23   126.05   55.12  Southwest  716.13
1    0.10      3.19     2.89   0.62    20.55   18.62       West  397.94
2    0.05      2.24     0.65   0.70    29.02    8.39  Northeast  267.83
3    0.20      9.62     0.65   3.71   174.05   11.68  Southwest  126.28
4    0.63     35.78     1.12   1.18    67.58    2.12  Southeast  121.51

        GDP  Population
0  0.437334   17.130763
1  0.906030   29.704992
2  0.289298   12.196428
3  0.067210    3.173814
4  0.031343    1.789857
```

3 Data Description

These data were collected by Climate Watch as part of their Historical Greenhouse Gas (GHG) Emissions data. The purpose of the dataset is to track trends of GHG by U.S. state 1990-2020. Our dataset contains 1581 observations with 19 variables. The observations are made up of all 50 U.S. States and District of Columbia with records for years 1990-2020 and we look at 4 GHG: Carbon Dioxide (CO2), Methane (CH4), Nitrous Oxide (N20), and Flourinated Gases (FGas). Variables `iso`, `State`, and `Year` designate the State Abbreviation, State Name, and Year of the data, respectively. Then for each GHG (CH4, CO2, FGas, N20) we have the GHG emission per state capita (i.e. `CH4pC`, `CO2pC`, `FGaspC`, `N2OpC`), GHG emission per State GDP (i.e. `CH4pGDP`, `CO2pGDP`, `FGaspGDP`, `N2OpGDP`), and total GHG emission (i.e. `CH4pTOT`, `CO2pTOT`, `FGaspTOT`, `N2OpTOT`). After that we have four remaining variables: `Region` specifies the US geographic region the state resides in, `TOT` shows the sum total of all 4 GHG emissions, `GDP` represents the state GDP of the specified year, and `Population` shows the state population of the specified year. To get `GDP/Population` we divide the total emissions by the perGDP/perCapita emissions for each gas and then take the average of each gas to get the `GDP/Population` for that state/year. All of the 'Total' measurements are measured in MtCO2e (Mega Tons of Carbon Dioxide Equivalent) units while the 'Per Capita' and 'Per GDP' measurements are measured in tCO2e (Tons of Carbon Dioxide Equivalent) per capita and tCO2e per million $ GDP, respectively. An important note is that this dataset does not contain any missing values, however, there are some outliers (big contributors to GHG emissions) that we will have to account for.

Name	Variable description	Type	Units of measurement
iso	State Abbrevation	Categorical	None
State	State Name	Categorical	None
Year	Year in which data was collected	Categorical	None
CH4pC	Methane emissions per State Capita	Numerical	tCO2e per capita
CH4pGDP	Methane emissions per State GDP	Numerical	tCO2e per million $ GDP
CH4TOT	Total Methane emissions	Numerical	MtCO2e
CO2pC	Carbon Dioxide emissions per State Capita	Numerical	tCO2e per capita
CO2pGDP	Carbon Dioxide emissions per State GDP	Numerical	tCO2e per million $ GDP
CO2TOT	Total Carbon Dioxide emissions	Numerical	MtCO2e
FGaspC	Flourinated Gases emissions per State Capita	Numerical	tCO2e per capita
FGaspGDP	Flourinated Gases emissions per State GDP	Numerical	tCO2e per million $ GDP
FGasTOT	Total Flourinated Gases emissions	Numerical	MtCO2e
N2OpC	Nitrous Oxide emissions per State Capita	Numerical	tCO2e per capita
N2OpGDP	Nitrous Oxide emissions per State GDP	Numerical	tCO2e per million $ GDP
N2OTOT	Total Nitrous Oxide emissions	Numerical	MtCO2e
Region	US Geographical Region the state is located in	Categorial	None

Name	Variable description	Type	Units of measurement
TOT	Sum Total GHG Emission for the year	Numerical	MtCO2e
GDP	State GDP for the year	Numerical	$ in trillions
Population	State Population for the year	Numerical	# of people in millions

4 Motivation

Greenhouse gases (GHG) pose an extraordinary threat to the future of human life on this planet. Getting control of the growth of GHG emissions is crucial to ensuring the sustainability and inhabitability of Earth. Through this analysis, we explore what factors are related to high levels of GHG emissions. Being aware of these factors can inform solutions to lower GHG emissions among states. With the data description in the previous step, we see that we have information for total gas emissions for all 4 GHG (Methane, Carbon Dioxide, Flourinated Gases, and Nitrous Oxide), State GDP, and State Population information for 1990-2020. Thinking about how GHG are emitted, we are expecting Population and GDP to have big influences on emissions but we would like to test our theory because it could be that external factors have even bigger influences on GHG emissions. This would be an ideal conclusion because that means that policies and community sentiment can actually have enough of an effect to turn around this problem we have with GHG. Our main question of interest is for each GHG, is Time, GDP, or Population the biggest contributor to emissions across U.S. geographical regions? While this question is fairly ambitious, we are hoping to get a concrete answer as one of those three factors (Time, GDP, Population), and then a further analysis with a bigger information/data set could potentially lead to more specific and influential answers.

5 Exploratory Data Analysis

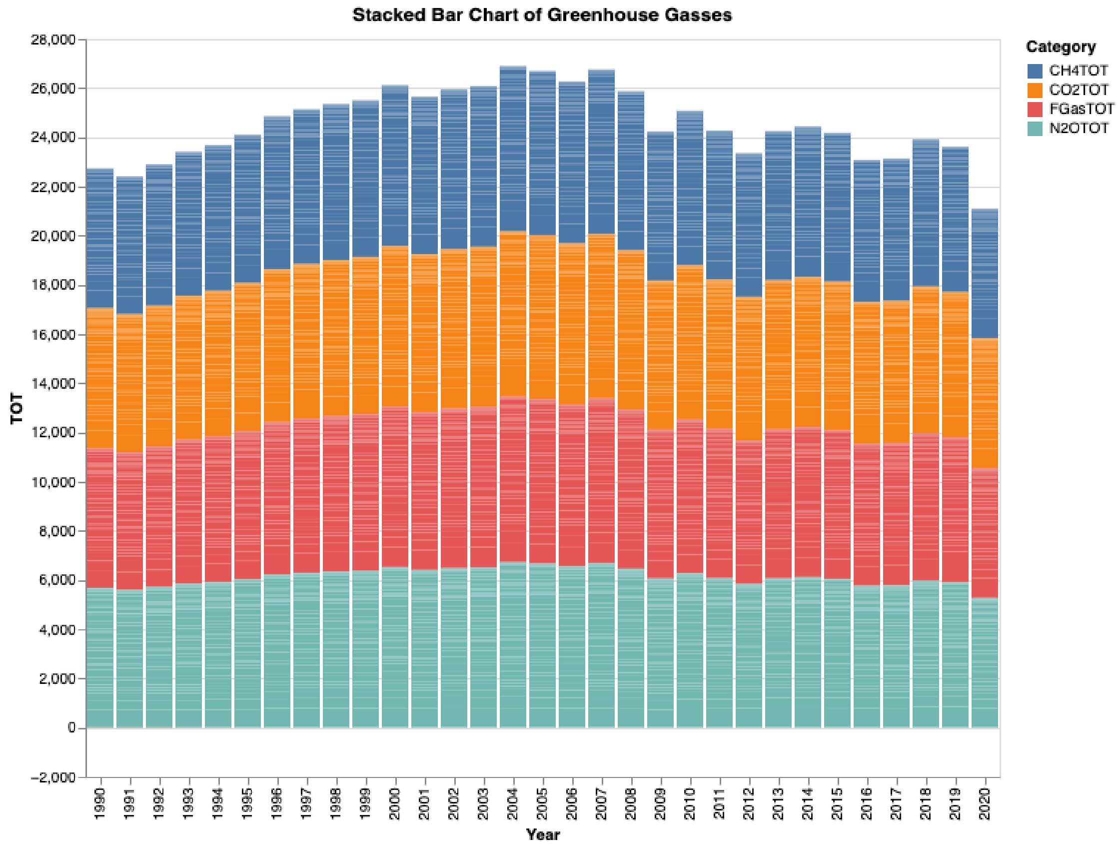
5.1 Amounts of Greenhouse Gases from 1990 to 2020 per Region

5.1.1 Total GHG

[2]:
```
# stack bar chart of total ghg emissions per year
ghg_stack = alt.Chart(ghg).mark_bar().encode(
    x = "Year",
    y = "TOT",
    color = alt.Color(
        "Category:N",
    scale = alt.Scale(
            domain = ["CH4TOT", "CO2TOT", "FGasTOT", "N2OTOT"])
    )
).transform_fold(
    ["CH4TOT", "CO2TOT", "FGasTOT", "N2OTOT"],
    as_ = ["Category", "Emission"]
).properties(
    height = 500,
```

```
    title = "Stacked Bar Chart of Greenhouse Gasses"
)
# display
ghg_stack
```

[2]:



We first wanted to look at the total levels of all four GHGs from 1990-2020, however, this plot isn't too helpful for our analysis so we decided to break the plots down into each gas to get more detail.

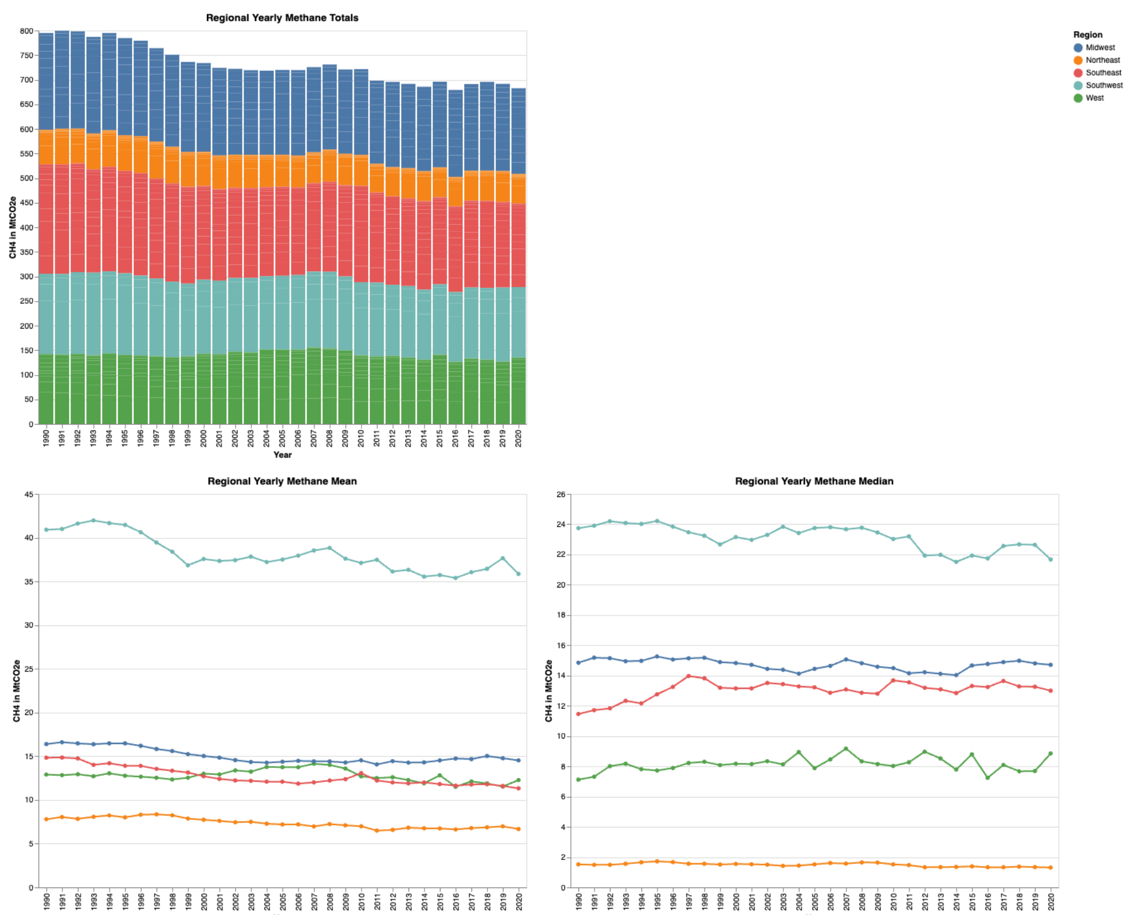
5.1.2 Methane

[3]:

```
# stack bar chart of yearly total ch4 emissions categorized by region
CH4_stack = alt.Chart(ghg).mark_bar().encode(
    x = alt.X("Year:O"),
    y = alt.Y("CH4TOT:Q", title = "CH4 in MtCO2e"),
    color = alt.Color("Region:N")
        ).properties(height = 500, title = "Regional Yearly Methane Totals")
# regional yearly mean ch4 emissions categorized by region
CH4_mean = alt.Chart(ghg).encode(
    x = alt.X("Year:O"),
    y = alt.Y("mean(CH4TOT):Q", title = "CH4 in MtCO2e"),
    color = alt.Color("Region:N")
```

```
).mark_line(point=True).properties(height = 500, title = "Regional Yearly␣
 ↪Methane Mean")
# regional yearly median ch4 emissions categorized by region
CH4_median = alt.Chart(ghg).encode(
    x = alt.X("Year:O"),
    y = alt.Y("median(CH4TOT):Q", title = "CH4 in MtCO2e"),
    color = alt.Color("Region:N")
).mark_line(point=True).properties(height = 500, title = "Regional Yearly␣
 ↪Methane Median")

CH4_stack & (CH4_mean | CH4_median)
```

[3]:



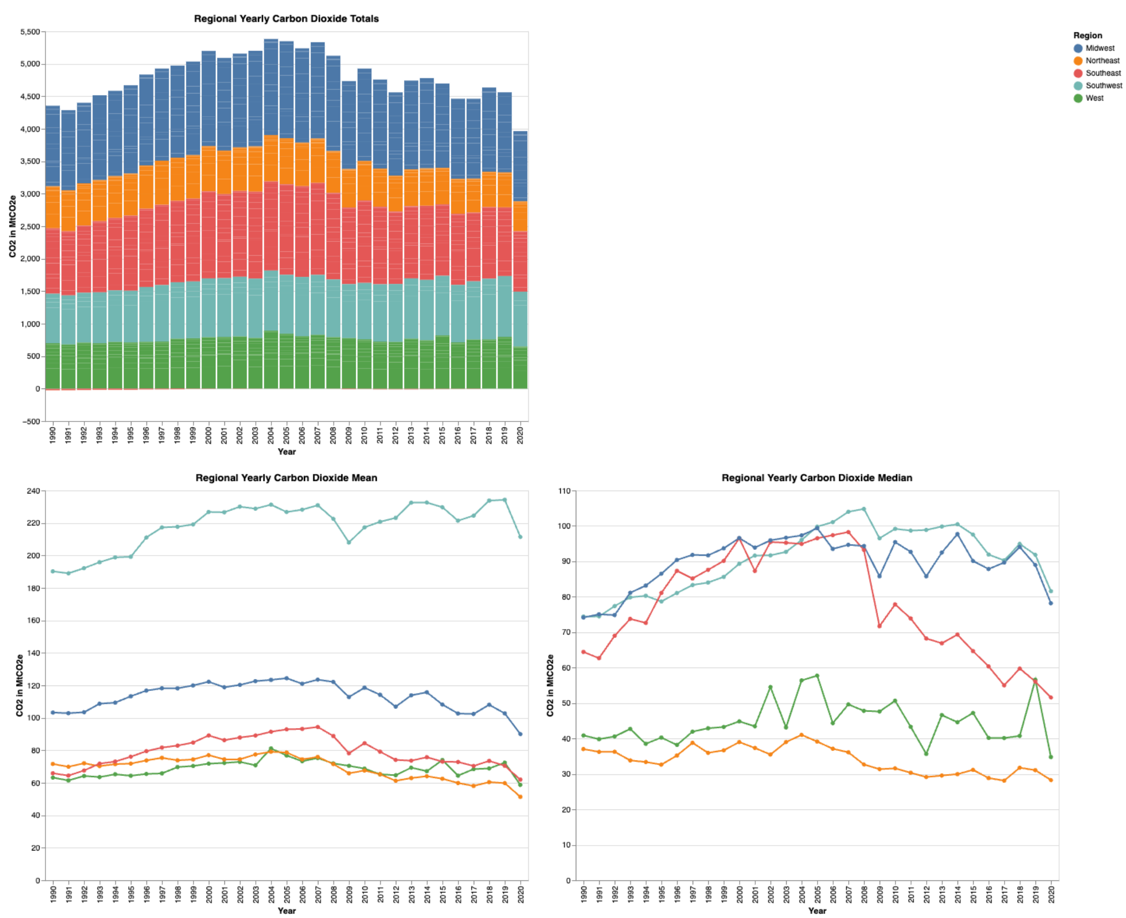
5.1.3 Carbon Dioxide

[4]:

```
# stack bar chart of yearly total co2 emissions categorized by region
CO2_stack = alt.Chart(ghg).mark_bar().encode(
    x = alt.X("Year:O"),
    y = alt.Y("CO2TOT:Q", title = "CO2 in MtCO2e"),
```

```
    color = alt.Color("Region:N")
        ).properties(height = 500, title = "Regional Yearly Carbon Dioxide␣
↪Totals")
# regional yearly mean co2 emissions categorized by region
CO2_mean = alt.Chart(ghg).encode(
    x = alt.X("Year:O"),
    y = alt.Y("mean(CO2TOT):Q", title = "CO2 in MtCO2e"),
    color = alt.Color("Region:N")
).mark_line(point=True).properties(height = 500, title = "Regional Yearly␣
↪Carbon Dioxide Mean")
# regional yearly median co2 emissions categorized by region
CO2_median = alt.Chart(ghg).encode(
    x = alt.X("Year:O"),
    y = alt.Y("median(CO2TOT):Q", title = "CO2 in MtCO2e"),
    color = alt.Color("Region:N")
).mark_line(point=True).properties(height = 500, title = "Regional Yearly␣
↪Carbon Dioxide Median")
# display
CO2_stack & (CO2_mean | CO2_median)
```

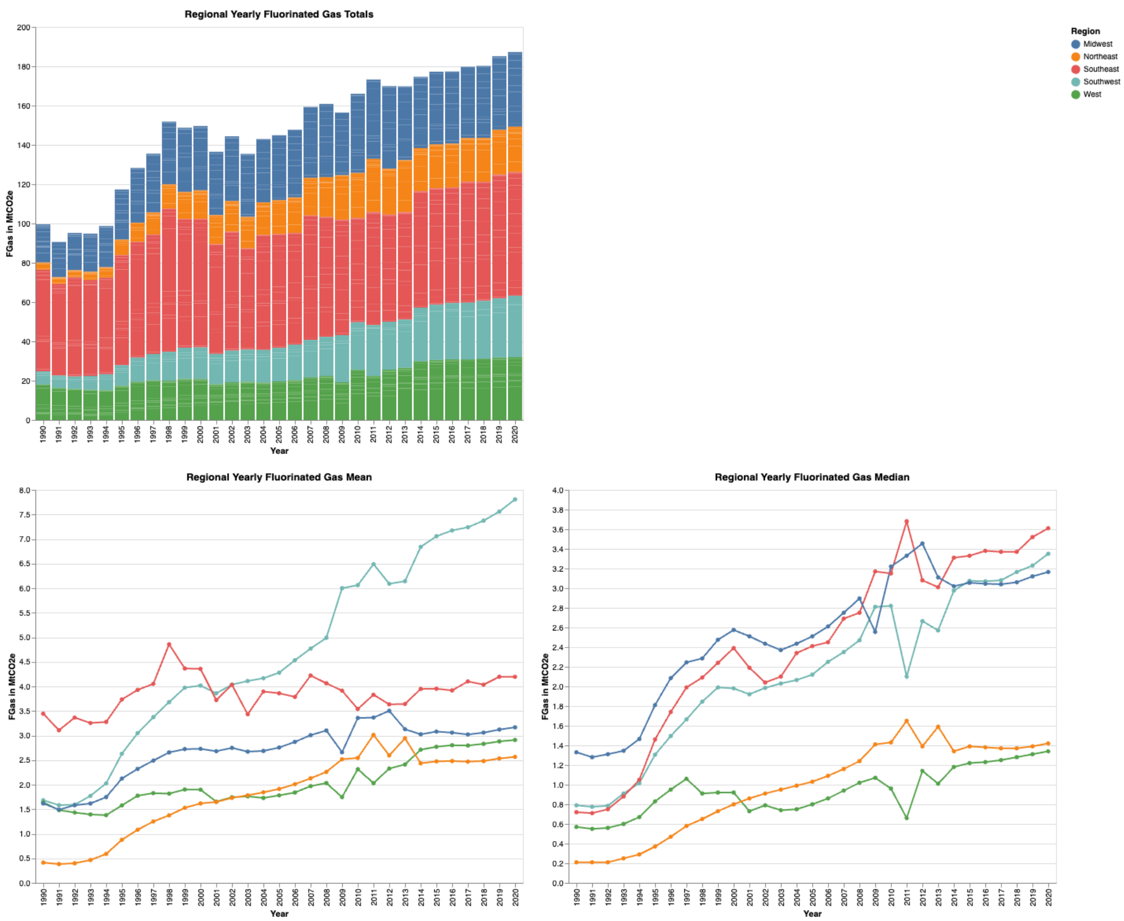
[4]:



5.1.4 Flourinated Gases

[5]:
```
# stack bar chart of yearly total fgas emissions categorized by region
FGas_stack = alt.Chart(ghg).mark_bar().encode(
    x = alt.X("Year:O"),
    y = alt.Y("FGasTOT:Q", title = "FGas in MtCO2e"),
    color = alt.Color("Region:N")
        ).properties(height = 500, title = "Regional Yearly Fluorinated Gas␣
↪Totals")
# regional yearly mean fgas emissions categorized by region
FGas_mean = alt.Chart(ghg).encode(
    x = alt.X("Year:O"),
    y = alt.Y("mean(FGasTOT):Q", title = "FGas in MtCO2e"),
    color = alt.Color("Region:N")
).mark_line(point=True).properties(height = 500, title = "Regional Yearly␣
↪Fluorinated Gas Mean")
# regional yearly median fgas emissions categorized by region
FGas_median = alt.Chart(ghg).encode(
    x = alt.X("Year:O"),
    y = alt.Y("median(FGasTOT):Q", title = "FGas in MtCO2e"),
    color = alt.Color("Region:N")
).mark_line(point=True).properties(height = 500, title = "Regional Yearly␣
↪Fluorinated Gas Median")
# display
FGas_stack & (FGas_mean | FGas_median)
```

[5]:

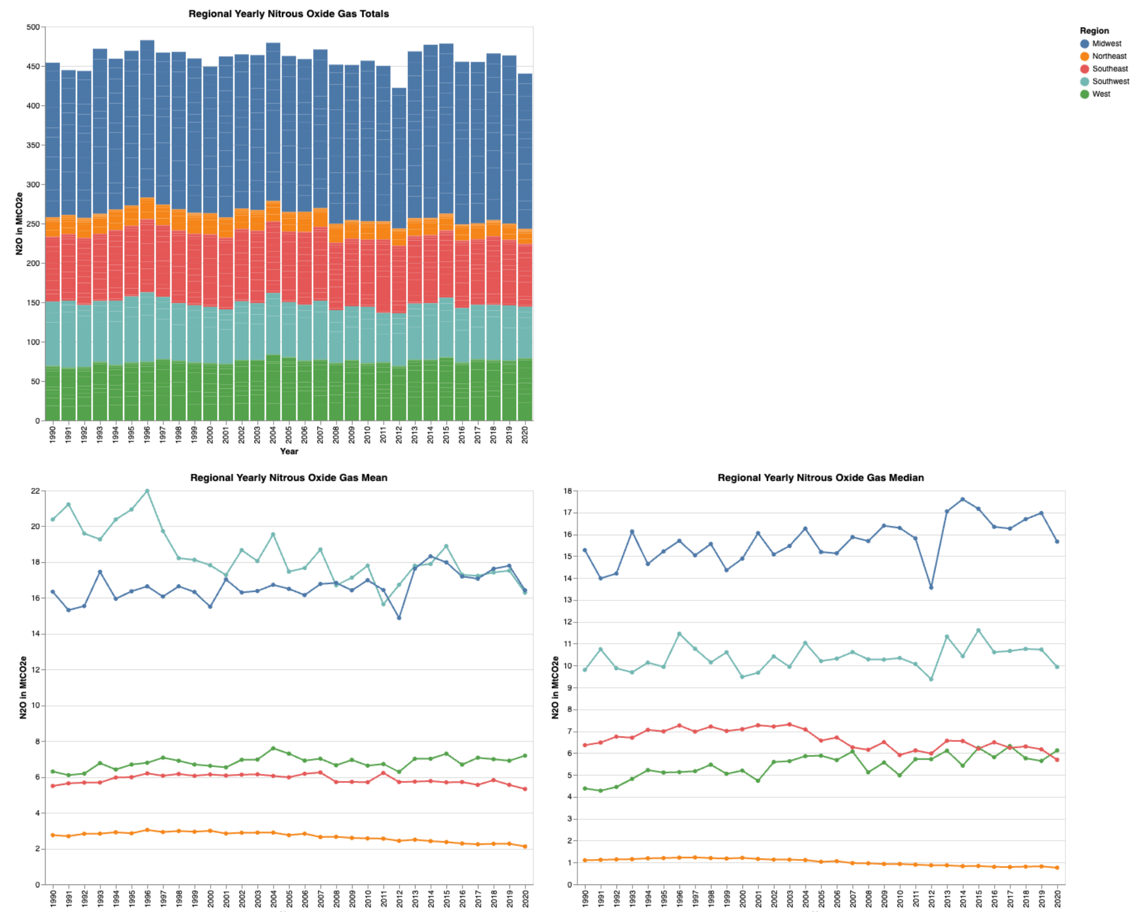


5.1.5 Nitrous Oxide

[6]:
```
# stack bar chart of yearly total no2 emissions categorized by region
N2O_stack = alt.Chart(ghg).mark_bar().encode(
    x = alt.X("Year:O"),
    y = alt.Y("N2OTOT:Q", title = "N2O in MtCO2e"),
    color = alt.Color("Region:N")
        ).properties(height = 500, title = "Regional Yearly Nitrous Oxide Gas␣
 ↪Totals")
# regional yearly mean no2 emissions categorized by region
N2O_mean = alt.Chart(ghg).encode(
    x = alt.X("Year:O"),
    y = alt.Y("mean(N2OTOT):Q", title = "N2O in MtCO2e"),
    color = alt.Color("Region:N")
).mark_line(point=True).properties(height = 500, title = "Regional Yearly␣
 ↪Nitrous Oxide Gas Mean")
# regional yearly median no2 emissions categorized by region
N2O_median = alt.Chart(ghg).encode(
```

```
    x = alt.X("Year:O"),
    y = alt.Y("median(N2OTOT):Q", title = "N2O in MtCO2e"),
    color = alt.Color("Region:N")
).mark_line(point=True).properties(height = 500, title = "Regional Yearly Nitrous Oxide Gas Median")
# display
N2O_stack & (N2O_mean | N2O_median)
```

[6]:



After breaking it down into each GHG, the most notable result we found was that while CH4 and NO2 stayed at relatively the same levels, CO2 has a bit of a parabolic shape - increasing at first and then decreasing starting in around 2005. FGas had the most noticeable trend of increasing over time with the Southwest region having a distinct increase compared to the other regions.

5.2 Greenhouse Gas Trends per Region

Now that we've looked at each GHG separately and noticed that trends change among regions we decided to break down our analysis into 5 geographpical regions within the U.S. and we incorporated our GDP and Population data.

[7]:
```
# melting data for ease of plotting
ghg_melted = pd.melt(ghg, id_vars=["iso", "State", "Year", "Region", "GDP",␣
↪"Population"], value_vars=['CH4TOT', 'CO2TOT', 'FGasTOT', 'N2OTOT'],
                    var_name='Gas Type', value_name='Gas Total')
ghg_melted
```

[7]:
```
      iso                 State  Year     Region       GDP  Population  \
0      TX                 Texas  1990  Southwest  0.437334   17.130763
1      CA            California  1990       West  0.906030   29.704992
2      PA          Pennsylvania  1990  Northeast  0.289298   12.196428
3      OK              Oklahoma  1990  Southwest  0.067210    3.173814
4      WV         West Virginia  1990  Southeast  0.031343    1.789857
…     ..                    …     …          …         …           …
6319   NH         New Hampshire  2020  Northeast  0.076028    1.384524
6320   DE              Delaware  2020  Southeast  0.062597    0.994989
6321   HI                Hawaii  2020       West  0.070085    1.449887
6322   RI          Rhode Island  2020  Northeast  0.051596    1.106055
6323   DC  District of Columbia  2020  Southeast  0.122135    0.667361

     Gas Type  Gas Total
0      CH4TOT     107.96
1      CH4TOT      64.37
2      CH4TOT      37.23
3      CH4TOT      34.22
4      CH4TOT      36.03
…         …         …
6319   N2OTOT       0.35
6320   N2OTOT       0.48
6321   N2OTOT       0.58
6322   N2OTOT       0.20
6323   N2OTOT       0.08

[6324 rows x 8 columns]
```

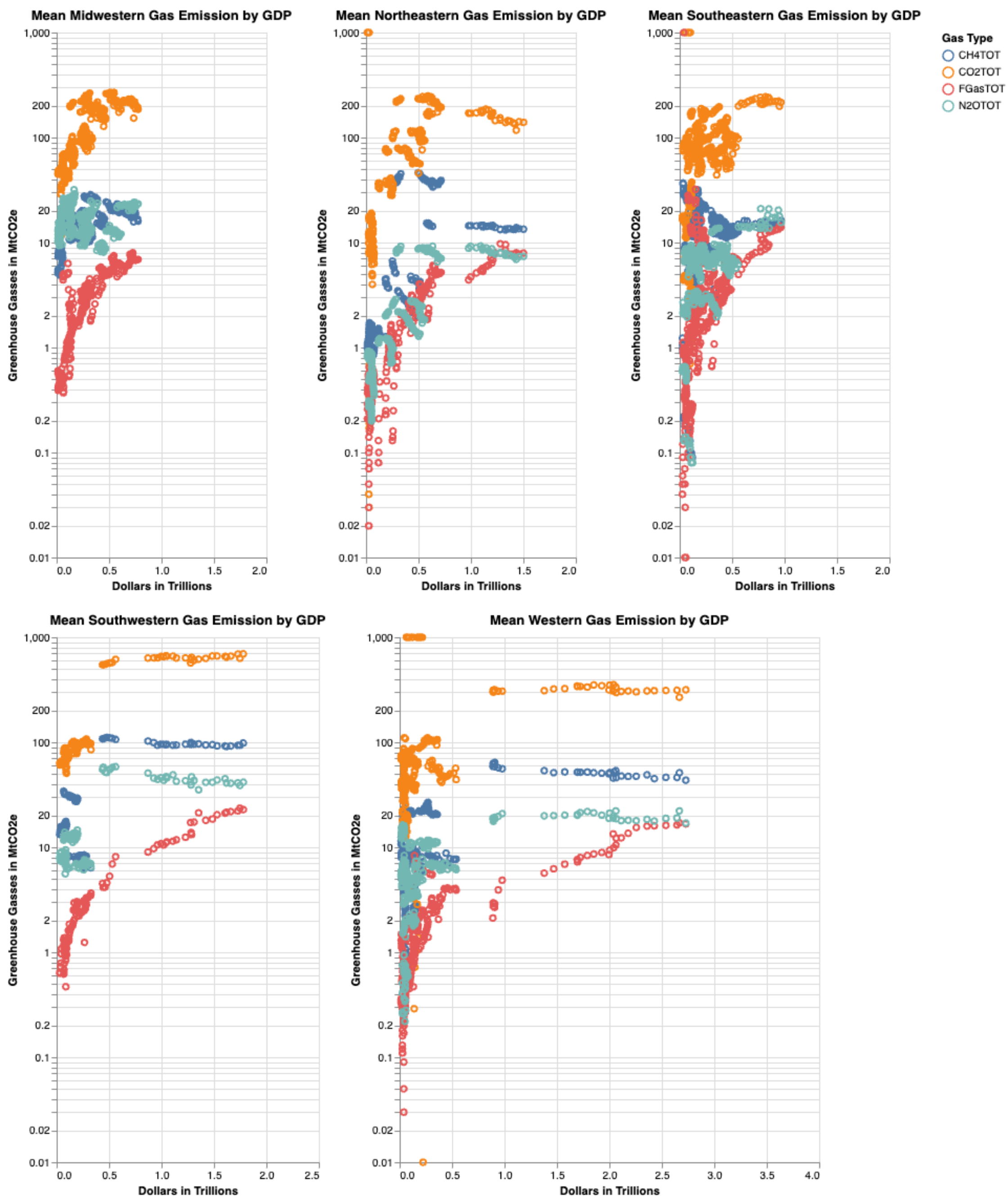
5.2.1 Trends by GDP

[8]:
```
# scatter plots of gdp against greenhouse gases for each region
gdp_mw = alt.Chart(ghg_melted[ghg_melted["Region"] == "Midwest"]).encode(
    x = alt.X("GDP:Q", scale = alt.Scale(domain = [0, 2]), title = "Dollars in␣
↪Trillions"),
    y = alt.Y("mean(Gas Total):Q", scale = alt.Scale(type = 'log', domain = [0.
↪01, 1000]), title = "Greenhouse Gasses in MtCO2e"),
    color = alt.Color("Gas Type:N")
).mark_point().properties(
    height = 500,
    width = 200,
    title = "Mean Midwestern Gas Emission by GDP"
```

```
)

gdp_ne = alt.Chart(ghg_melted[ghg_melted["Region"] == "Northeast"]).encode(
    x = alt.X("GDP:Q", scale = alt.Scale(domain = [0, 2]), title = "Dollars in␣
 ↪Trillions"),
    y = alt.Y("mean(Gas Total):Q", scale = alt.Scale(type = 'log', domain = [0.
 ↪01, 1000]), title = "Greenhouse Gasses in MtCO2e"),
    color = alt.Color("Gas Type:N")
).mark_point().properties(
    height = 500,
    width = 200,
    title = "Mean Northeastern Gas Emission by GDP"
)

gdp_se = alt.Chart(ghg_melted[ghg_melted["Region"] == "Southeast"]).encode(
    x = alt.X("GDP:Q", scale = alt.Scale(domain = [0, 2]), title = "Dollars in␣
 ↪Trillions"),
    y = alt.Y("mean(Gas Total):Q", scale = alt.Scale(type = 'log', domain = [0.
 ↪01, 1000]), title = "Greenhouse Gasses in MtCO2e"),
    color = alt.Color("Gas Type:N")
).mark_point().properties(
    height = 500,
    width = 200,
    title = "Mean Southeastern Gas Emission by GDP"
)

gdp_sw = alt.Chart(ghg_melted[ghg_melted["Region"] == "Southwest"]).encode(
    x = alt.X("GDP:Q", scale = alt.Scale(domain = [0, 2.5]), title = "Dollars␣
 ↪in Trillions"),
    y = alt.Y("mean(Gas Total):Q", scale = alt.Scale(type = 'log', domain = [0.
 ↪01, 1000]), title = "Greenhouse Gasses in MtCO2e"),
    color = alt.Color("Gas Type:N")
).mark_point().properties(
    height = 500,
    width = 250,
    title = "Mean Southwestern Gas Emission by GDP"
)

gdp_w = alt.Chart(ghg_melted[ghg_melted["Region"] == "West"]).encode(
    x = alt.X("GDP:Q", scale = alt.Scale(domain = [0, 4]), title = "Dollars in␣
 ↪Trillions"),
    y = alt.Y("mean(Gas Total):Q", scale = alt.Scale(type = 'log', domain = [0.
 ↪01, 1000]), title = "Greenhouse Gasses in MtCO2e"),
    color = alt.Color("Gas Type:N")
).mark_point().properties(
    height = 500,
```

```
    width = 400,
    title = "Mean Western Gas Emission by GDP"
)

(gdp_mw | gdp_ne | gdp_se) & (gdp_sw | gdp_w)
```

[8]:

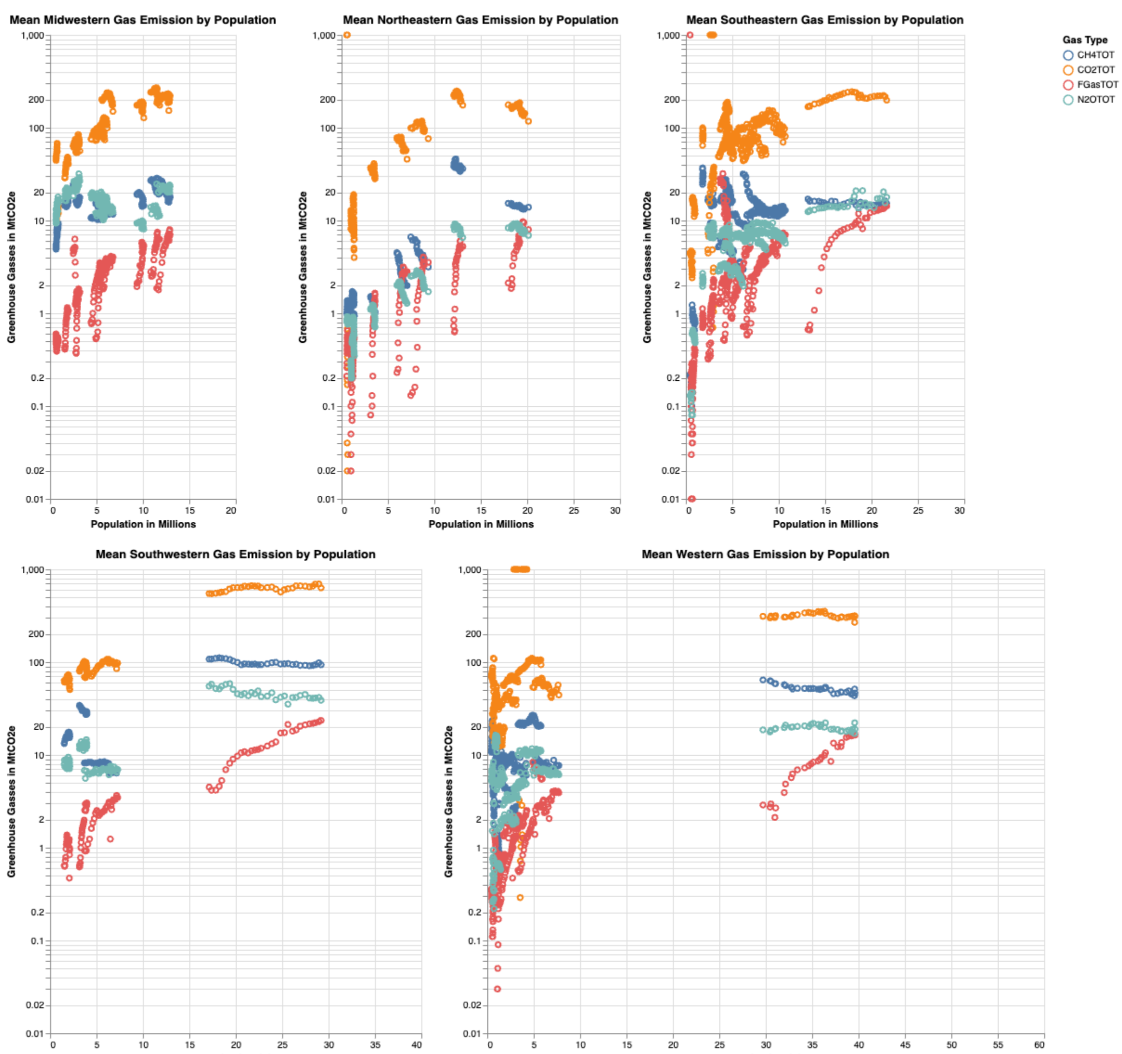


5.2.2 Trends by Population

[9]:
```
# scatter plots of capita against greenhouse gases for each region
pop_mw = alt.Chart(ghg_melted[ghg_melted["Region"] == "Midwest"]).encode(
    x = alt.X("Population:Q", scale = alt.Scale(domain = [0, 20]), title =␣
 ↪"Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", scale = alt.Scale(type = 'log', domain = [0.
 ↪01, 1000]), title = "Greenhouse Gasses in MtCO2e"),
    color = alt.Color("Gas Type:N")
).mark_point().properties(
    height = 500,
    width = 200,
    title = "Mean Midwestern Gas Emission by Population"
)

pop_ne = alt.Chart(ghg_melted[ghg_melted["Region"] == "Northeast"]).encode(
    x = alt.X("Population:Q", scale = alt.Scale(domain = [0, 30]), title =␣
 ↪"Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", scale = alt.Scale(type = 'log', domain = [0.
 ↪01, 1000]), title = "Greenhouse Gasses in MtCO2e"),
    color = alt.Color("Gas Type:N")
).mark_point().properties(
    height = 500,
    width = 300,
    title = "Mean Northeastern Gas Emission by Population"
)

pop_se = alt.Chart(ghg_melted[ghg_melted["Region"] == "Southeast"]).encode(
    x = alt.X("Population:Q", scale = alt.Scale(domain = [0, 30]), title =␣
 ↪"Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", scale = alt.Scale(type = 'log', domain = [0.
 ↪01, 1000]), title = "Greenhouse Gasses in MtCO2e"),
    color = alt.Color("Gas Type:N")
).mark_point().properties(
    height = 500,
    width = 300,
    title = "Mean Southeastern Gas Emission by Population"
)

pop_sw = alt.Chart(ghg_melted[ghg_melted["Region"] == "Southwest"]).encode(
    x = alt.X("Population:Q", scale = alt.Scale(domain = [0, 40]), title =␣
 ↪"Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", scale = alt.Scale(type = 'log', domain = [0.
 ↪01, 1000]), title = "Greenhouse Gasses in MtCO2e"),
    color = alt.Color("Gas Type:N")
).mark_point().properties(
    height = 500,
```

```python
    width = 400,
    title = "Mean Southwestern Gas Emission by Population"
)

pop_w = alt.Chart(ghg_melted[ghg_melted["Region"] == "West"]).encode(
    x = alt.X("Population:Q", scale = alt.Scale(domain = [0, 60]), title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", scale = alt.Scale(type = 'log', domain = [0.01, 1000]), title = "Greenhouse Gasses in MtCO2e"),
    color = alt.Color("Gas Type:N")
).mark_point().properties(
    height = 500,
    width = 600,
    title = "Mean Western Gas Emission by Population"
)

(pop_mw | pop_ne | pop_se) & (pop_sw | pop_w)
```

[9]:



Based on our Exploratory Data Anaysis, we found that there are differences among each GHG when we take into account GDP, Population, and Region. This leads us into our primary analysis where we investigate each gas within each region to find out whether time, gdp, or population was the biggest factor in GHG emissions.

6 Data Analysis for each GHG per Region

6.1 Midwest

[10]:
```python
# extract midwest region data
ghg_mw = ghg[ghg["Region"] == "Midwest"]

# Outlier Analysis ************************************************
# boxplot showing distributions of ch4 total emissions among states for each␣
↪year
mw_CH4_outliers = alt.Chart(ghg_mw).encode(
    x = alt.X("Year:O"),
    y = alt.Y("CH4TOT"),
    tooltip = ("State", "CH4TOT")
).mark_boxplot()

# boxplot showing distributions of co2 total emissions among states for each␣
↪year
mw_CO2_outliers = alt.Chart(ghg_mw).encode(
    x = alt.X("Year:O"),
    y = alt.Y("CO2TOT"),
    tooltip = ("State", "CO2TOT")
).mark_boxplot()

# boxplot showing distributions of fgas total emissions among states for each␣
↪year
mw_FGas_outliers = alt.Chart(ghg_mw).encode(
    x = alt.X("Year:O"),
    y = alt.Y("FGasTOT"),
    tooltip = ("State", "FGasTOT")
).mark_boxplot()

# boxplot showing distributions of n2o total emissions among states for each␣
↪year
mw_N2O_outliers = alt.Chart(ghg_mw).encode(
    x = alt.X("Year:O"),
    y = alt.Y("N2OTOT"),
    tooltip = ("State", "N2OTOT")
).mark_boxplot()
```

```
# Methane Analysis ************************************************
# plot mean ch4 emission vs. year
year_mw_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Midwest") &␣
↪(ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.
↪Scale(zero = False)),
    #color = alt.Color("Gas Type:N")
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Midwestern Methane Gas Emission by Year"
)

# plot mean ch4 emission vs. gdp
gdp_mw_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Midwest") &␣
↪(ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Midwestern Methane Gas Emission by GDP"
)

# plot mean ch4 emission vs. population
pop_mw_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Midwest") &␣
↪(ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Midwestern Methane gas Emission by Population"
)

# LOESS smoother
# extract midwestern ch4 data
mw_CH4 = ghg_melted[(ghg_melted['Region'] == 'Midwest') & (ghg_melted['Gas␣
↪Type'] == 'CH4TOT')]
```

```
# grid of gdp and population values
mw_grid_gdp = np.linspace(mw_CH4["GDP"].min(), mw_CH4["GDP"].max(), num = 100)
mw_grid_pop = np.linspace(mw_CH4["Population"].min(), mw_CH4["Population"].
  ↪max(), num = 100)

# fit loess smooth for gdp and population plots
gdp_mw_CH4_ls = sm.nonparametric.lowess(endog = mw_CH4["Gas Total"].values,
                                         exog = mw_CH4["GDP"].values,
                                         frac = 0.5,
                                         xvals = mw_grid_gdp)
pop_mw_CH4_ls = sm.nonparametric.lowess(endog = mw_CH4["Gas Total"].values,
                                         exog = mw_CH4["Population"].values,
                                         frac = 0.5,
                                         xvals = mw_grid_pop)

# store as dataframe
gdp_mw_CH4_df = pd.DataFrame({'GDP': mw_grid_gdp, 'Gas': gdp_mw_CH4_ls})
pop_mw_CH4_df = pd.DataFrame({'Population': mw_grid_pop, 'Gas': pop_mw_CH4_ls})


# loess smoother lines for gdp and population plots
gdp_mw_CH4_loess = alt.Chart(
    gdp_mw_CH4_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_mw_CH4_loess = alt.Chart(
    pop_mw_CH4_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# regression analysis
# create dummy and x/y variables for mlr
mw_CH4_indicators = pd.get_dummies(mw_CH4[['Year', 'Population', 'GDP']],
                           drop_first = True)
mw_CH4_x = sm.tools.add_constant(mw_CH4_indicators)
mw_CH4_y = mw_CH4['Gas Total']
mw_CH4_indicators.columns.values
```

```
# fit mlr model
mw_CH4_mlr = sm.OLS(endog = mw_CH4_y, exog = mw_CH4_x.astype(float))
mw_CH4_rslt = mw_CH4_mlr.fit()

# retreive estimates and std errors
mw_CH4_coef_tbl = pd.DataFrame({
    'estimate': mw_CH4_rslt.params.values,
    'standard error': np.sqrt(mw_CH4_rslt.cov_params().values.diagonal())},
    index = mw_CH4_x.columns
)
mw_CH4_coef_tbl.loc['error variance', 'estimate'] = mw_CH4_rslt.scale

# add column of exponentiated coefficients
mw_CH4_coef_tbl['exponentiated']=np.exp(mw_CH4_coef_tbl['estimate'])
mw_CH4_coef_tbl['exp_visual'] = mw_CH4_coef_tbl['exponentiated'].apply(lambda x: "{:.2f}".format(x))


# Carbon Dioxide Analysis ************************************************
# plot mean co2 emission vs. year
year_mw_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Midwest") & (ghg_melted["Gas Type"] == "CO2TOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.Scale(zero = False)),
    #color = alt.Color("Gas Type:N")
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Midwestern CO2 Gas Emission by Year"
)
# plot mean co2 emission vs. gdp
gdp_mw_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Midwest") & (ghg_melted["Gas Type"] == "CO2TOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Midwestern CO2 Gas Emission by GDP"
)
# plot mean co2 emission vs. population
pop_mw_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Midwest") & (ghg_melted["Gas Type"] == "CO2TOT")]).encode(
```

```
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Midwestern CO2 gas Emission by Population"
)

# LOESS smoother

# extract midwestern co2 data
mw_CO2 = ghg_melted[(ghg_melted['Region'] == 'Midwest') & (ghg_melted['Gas␣
↪Type'] == 'CO2TOT')]

# fit loess smooth for gdp and population plots
gdp_mw_CO2_ls = sm.nonparametric.lowess(endog = mw_CO2["Gas Total"].values,
                                    exog = mw_CO2["GDP"].values,
                                    frac = 0.5,
                                    xvals = mw_grid_gdp)
pop_mw_CO2_ls = sm.nonparametric.lowess(endog = mw_CO2["Gas Total"].values,
                                    exog = mw_CO2["Population"].values,
                                    frac = 0.5,
                                    xvals = mw_grid_pop)

# store as dataframe
gdp_mw_CO2_df = pd.DataFrame({'GDP': mw_grid_gdp, 'Gas': gdp_mw_CO2_ls})
pop_mw_CO2_df = pd.DataFrame({'Population': mw_grid_pop, 'Gas': pop_mw_CO2_ls})

# loess smoother lines for gdp and population plots
gdp_mw_CO2_loess = alt.Chart(
    gdp_mw_CO2_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_mw_CO2_loess = alt.Chart(
    pop_mw_CO2_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
```

```
# create dummy and x/y variables for mlr
mw_CO2_indicators = pd.get_dummies(mw_CO2[['Year', 'Population', 'GDP']],
                                   drop_first = True)
mw_CO2_x = sm.tools.add_constant(mw_CO2_indicators)
mw_CO2_y = mw_CO2['Gas Total']
mw_CO2_indicators.columns.values

# fit mlr model
mw_CO2_mlr = sm.OLS(endog = mw_CO2_y, exog = mw_CO2_x.astype(float))
mw_CO2_rslt = mw_CO2_mlr.fit()

# retreive estimates and std errors
mw_CO2_coef_tbl = pd.DataFrame({
     'estimate': mw_CO2_rslt.params.values,
     'standard error': np.sqrt(mw_CO2_rslt.cov_params().values.diagonal())},
     index = mw_CO2_x.columns
)
mw_CO2_coef_tbl.loc['error variance', 'estimate'] = mw_CO2_rslt.scale

# add column of exponentiated coefficients
mw_CO2_coef_tbl['exponentiated']=np.exp(mw_CO2_coef_tbl['estimate'])
mw_CO2_coef_tbl['exp_visual'] = mw_CO2_coef_tbl['exponentiated'].apply(lambda x:
  ↪ "{:.2f}".format(x))



# Flourinated Gases Analysis ************************************************
# plot mean fgas emission vs. year
year_mw_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Midwest") &␣
  ↪(ghg_melted["Gas Type"] == "FGasTOT")]).encode(
     x = alt.X("Year:O", title = "Year"),
     y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.
  ↪Scale(zero = False)),
     #color = alt.Color("Gas Type:N")
).mark_line().properties(
     height = 500,
     width = 300,
     title = "Mean Midwestern Flourinated Gas Emission by Year"
)
# plot mean fgas emission vs. gdp
gdp_mw_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Midwest") &␣
  ↪(ghg_melted["Gas Type"] == "FGasTOT")]).encode(
     x = alt.X("GDP:Q", title = "Dollars in Trillions"),
     y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.
  ↪Scale(zero = False)),
     color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
```

```python
    height = 500,
    width = 300,
    title = "Mean Midwestern Flourinated Gas Emission by GDP"
)
# plot mean fgas emission vs. population
pop_mw_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Midwest") & (ghg_melted["Gas Type"] == "FGasTOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Midwestern Flourinated Gas Emission by Population"
)

# LOESS smoother
# extract midwestern fgas data
mw_FGas = ghg_melted[(ghg_melted['Region'] == 'Midwest') & (ghg_melted['Gas Type'] == 'FGasTOT')]

# fit loess smooth for gdp and population plots
gdp_mw_FGas_ls = sm.nonparametric.lowess(endog = mw_FGas["Gas Total"].values,
                                    exog = mw_FGas["GDP"].values,
                                    frac = 0.5,
                                    xvals = mw_grid_gdp)
pop_mw_FGas_ls = sm.nonparametric.lowess(endog = mw_FGas["Gas Total"].values,
                                    exog = mw_FGas["Population"].values,
                                    frac = 0.5,
                                    xvals = mw_grid_pop)

# store as dataframe
gdp_mw_FGas_df = pd.DataFrame({'GDP': mw_grid_gdp, 'Gas': gdp_mw_FGas_ls})
pop_mw_FGas_df = pd.DataFrame({'Population': mw_grid_pop, 'Gas': pop_mw_FGas_ls})


# loess smoother lines for gdp and population plots
gdp_mw_FGas_loess = alt.Chart(
    gdp_mw_FGas_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
```

```
pop_mw_FGas_loess = alt.Chart(
    pop_mw_FGas_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# create dummy and x/y variables for mlr
mw_FGas_indicators = pd.get_dummies(mw_FGas[['Year', 'Population', 'GDP']],
                           drop_first = True)
mw_FGas_x = sm.tools.add_constant(mw_FGas_indicators)
mw_FGas_y = mw_FGas['Gas Total']
mw_FGas_indicators.columns.values

# fit mlr model
mw_FGas_mlr = sm.OLS(endog = mw_FGas_y, exog = mw_FGas_x.astype(float))
mw_FGas_rslt = mw_FGas_mlr.fit()

# retreive estimates and std errors
mw_FGas_coef_tbl = pd.DataFrame({
    'estimate': mw_FGas_rslt.params.values,
    'standard error': np.sqrt(mw_FGas_rslt.cov_params().values.diagonal())},
    index = mw_FGas_x.columns
)
mw_FGas_coef_tbl.loc['error variance', 'estimate'] = mw_FGas_rslt.scale

# add column of exponentiated coefficients
mw_FGas_coef_tbl['exponentiated']=np.exp(mw_FGas_coef_tbl['estimate'])
mw_FGas_coef_tbl['exp_visual'] = mw_FGas_coef_tbl['exponentiated'].apply(lambda␣
 ↪x: "{:.2f}".format(x))



# Nitrous Oxide Analysis ************************************************
# plot mean n2o emission vs. year
year_mw_N2O = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Midwest") &␣
 ↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Midwestern Nitrous Oxide Gas Emission by Year"
)
```

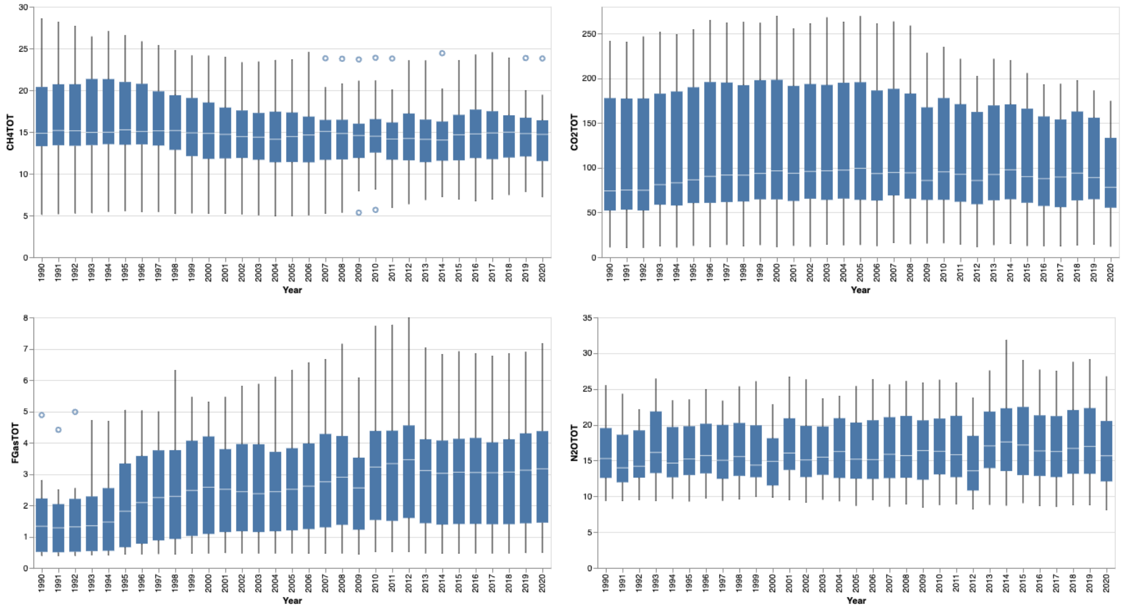
```
# plot mean n2o emission vs. gdp
gdp_mw_N2O = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Midwest") &␣
 ↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Midwestern Nitrous Oxide gas Emission by GDP"
)
# plot mean n2o emission vs. population
pop_mw_N2O = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Midwest") &␣
 ↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Midwestern Nitrous Oxide gas Emission by Population"
)

# LOESS smoother
# extract midwestern n2o data
mw_N2O = ghg_melted[(ghg_melted['Region'] == 'Midwest') & (ghg_melted['Gas␣
 ↪Type'] == 'N2OTOT')]

# fit loess smooth for gdp and population plots
gdp_mw_N2O_ls = sm.nonparametric.lowess(endog = mw_N2O["Gas Total"].values,
                                    exog = mw_N2O["GDP"].values,
                                    frac = 0.5,
                                    xvals = mw_grid_gdp)
pop_mw_N2O_ls = sm.nonparametric.lowess(endog = mw_N2O["Gas Total"].values,
                                    exog = mw_N2O["Population"].values,
                                    frac = 0.5,
                                    xvals = mw_grid_pop)

# store as dataframe
gdp_mw_N2O_df = pd.DataFrame({'GDP': mw_grid_gdp, 'Gas': gdp_mw_N2O_ls})
pop_mw_N2O_df = pd.DataFrame({'Population': mw_grid_pop, 'Gas': pop_mw_N2O_ls})


# loess smoother lines for gdp and population plots
gdp_mw_N2O_loess = alt.Chart(
```

```
    gdp_mw_N2O_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_mw_N2O_loess = alt.Chart(
    pop_mw_N2O_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# create dummy and x/y variables for mlr
mw_N2O_indicators = pd.get_dummies(mw_N2O[['Year', 'Population', 'GDP']],
                                   drop_first = True)
mw_N2O_x = sm.tools.add_constant(mw_N2O_indicators)
mw_N2O_y = mw_N2O['Gas Total']
mw_N2O_indicators.columns.values

# fit mlr model
mw_N2O_mlr = sm.OLS(endog = mw_N2O_y, exog = mw_N2O_x.astype(float))
mw_N2O_rslt = mw_N2O_mlr.fit()

# retreive estimates and std errors
mw_N2O_coef_tbl = pd.DataFrame({
    'estimate': mw_N2O_rslt.params.values,
    'standard error': np.sqrt(mw_N2O_rslt.cov_params().values.diagonal())},
    index = mw_N2O_x.columns
)
mw_N2O_coef_tbl.loc['error variance', 'estimate'] = mw_N2O_rslt.scale

# add column of exponentiated coefficients
mw_N2O_coef_tbl['exponentiated']=np.exp(mw_N2O_coef_tbl['estimate'])
mw_N2O_coef_tbl['exp_visual'] = mw_N2O_coef_tbl['exponentiated'].apply(lambda x:
 ↪ "{:.2f}".format(x))
```

6.1.1 Outlier Analysis

[11]:
```
(mw_CH4_outliers | mw_CO2_outliers) & (mw_FGas_outliers | mw_N2O_outliers)
```

[11]:

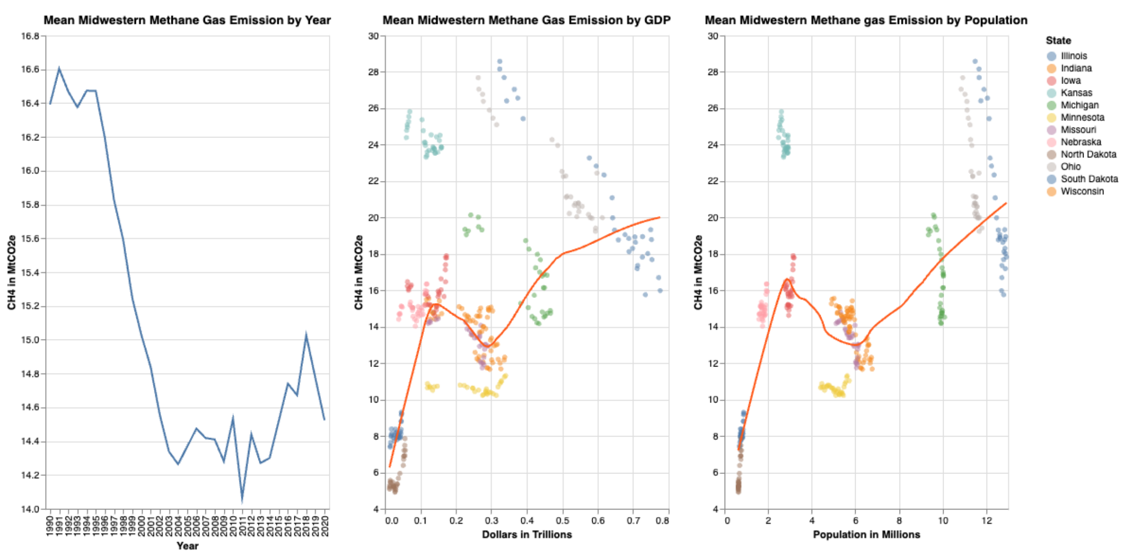


It seems that there aren't any consistent outliers in the Midwest Region. For Ch4, Kansas pops up as a high outlier from time to time in the second half of the time period and in 2009-2010 North Dakota was a low outlier. For FGas, Kansas once again was an outlier but only from 1990-1992. Looking at CO2 and NO2, there are no outliers at all, so looking at the Midwest region should show fairly consistent reportings without and skewing.

6.1.2 Methane

[12]:
```
year_mw_CH4 | gdp_mw_CH4 + gdp_mw_CH4_loess | pop_mw_CH4 + pop_mw_CH4_loess
```

[12]:



Looking at the first plot, we can see that there was a big decrease in CH4 emissions starting around 1995, but the scale shows it only going from about 16.5 to 14.5 MtCO2e which isn't much. THe GDP plot shows somewhat of a positive linear trend between CH4 emissions and GDP but there is some stacking and clusters that have negative trends so it doesn't seem very consistent. The Population graph, on the other hand, shows a positive linear trend with a fairly high correlation.

[13]:
```
mw_CH4_coef_tbl
```

[13]:

	estimate	standard error	exponentiated	exp_visual
const	10.326483	1.472801	3.053053e+04	30530.53
Population	1.760541	0.249249	5.815580e+00	5.82
GDP	-21.785324	5.356694	3.457436e-10	0.00
Year_1991	0.136040	1.889435	1.145728e+00	1.15
Year_1992	0.057686	1.889444	1.059382e+00	1.06
Year_1993	-0.086005	1.889449	9.175894e-01	0.92
Year_1994	0.136362	1.890033	1.146096e+00	1.15
Year_1995	0.138205	1.890299	1.148211e+00	1.15
Year_1996	-0.072690	1.891184	9.298888e-01	0.93
Year_1997	1.046065	1.942641	2.846427e+00	2.85
Year_1998	0.910544	1.950191	2.485673e+00	2.49
Year_1999	0.688793	1.959579	1.991311e+00	1.99
Year_2000	0.586126	1.969072	1.797013e+00	1.80
Year_2001	0.330679	1.965297	1.391913e+00	1.39
Year_2002	0.101900	1.970613	1.107273e+00	1.11
Year_2003	-0.017285	1.978766	9.828634e-01	0.98
Year_2004	0.034450	1.989607	1.035051e+00	1.04
Year_2005	0.239677	1.998490	1.270839e+00	1.27
Year_2006	0.368310	2.001739	1.445290e+00	1.45
Year_2007	0.323000	2.004603	1.381266e+00	1.38
Year_2008	0.221511	1.997969	1.247961e+00	1.25
Year_2009	-0.182433	1.977972	8.332403e-01	0.83
Year_2010	0.211527	1.989671	1.235563e+00	1.24
Year_2011	-0.156919	1.998557	8.547731e-01	0.85
Year_2012	0.294813	2.005940	1.342875e+00	1.34
Year_2013	0.156716	2.009899	1.169664e+00	1.17
Year_2014	0.322190	2.022218	1.380146e+00	1.38
Year_2015	0.602225	2.028512	1.826178e+00	1.83
Year_2016	0.853375	2.031977	2.347556e+00	2.35
Year_2017	0.824648	2.036534	2.281078e+00	2.28
Year_2018	1.301840	2.048480	3.676055e+00	3.68
Year_2019	1.107199	2.054624	3.025870e+00	3.03
Year_2020	0.534537	2.028486	1.706658e+00	1.71
error variance	21.419118	NaN	2.005417e+09	2005416505.61

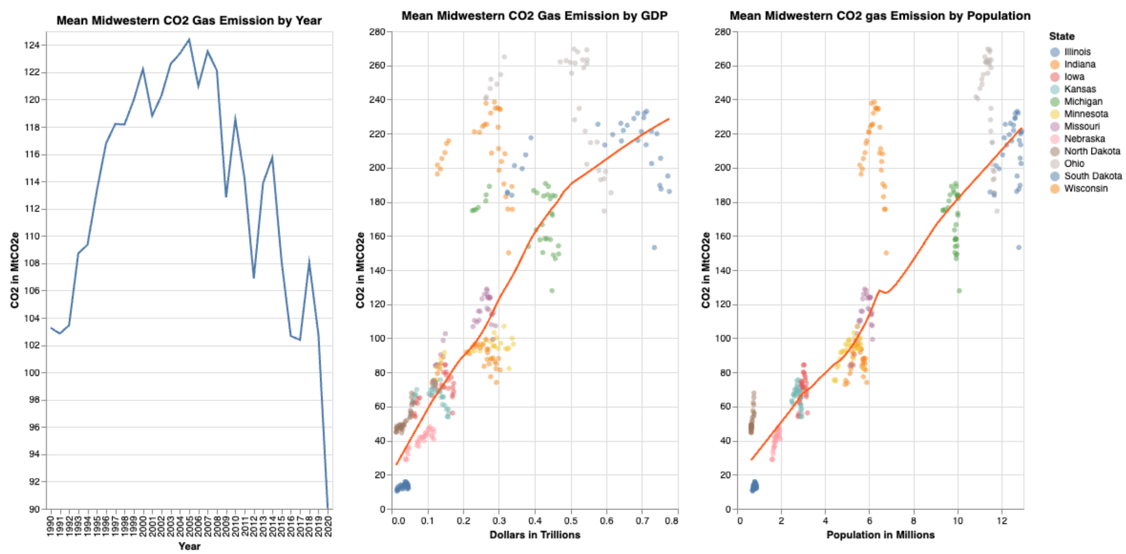
From fitting the multiple linear regression and extracting the exponentiated coefficients, we can again examine how each factor affects CH4 emissions, we can see that our conclusions from the graph above are fairly supported here. The exponentiated coefficient for GDP is almost 0, showing that GDP does not have much of an effect compared to the other factors. The coefficients for each year range from about 1-3 showing that there is a fair amount of influence. However, when we compare to the coefficient for Populaition of almost 6, we can see that there is a clear factor that exerts the most influence.

We can conclude that Population is the most significant factor contributing to CH4 emissions in the Midwest.

6.1.3 Carbon Dioxide

[14]:
```
year_mw_CO2 | gdp_mw_CO2 + gdp_mw_CO2_loess | pop_mw_CO2 + pop_mw_CO2_loess
```

[14]:



The year to year plot of carbon dioxide in the midwest follows a negative quadratic trend. The highest amount of carbon dioxide emission observed was in 2004. Both the GDP and population predictors follow a strong positive linear trend. We observe higher interstate variance in GDP than we do in population. However, for both GDP and population, it appears the more affluent/populous states will produce more carbon dioxide.

[15]:
```
mw_CO2_coef_tbl
```

[15]:

	estimate	standard error	exponentiated	exp_visual
const	2.168385	10.167153	8.744147e+00	8.74
Population	24.261232	1.720636	3.439689e+10	34396888867.96
GDP	-159.707443	36.978735	4.364543e-70	0.00
Year_1991	-1.463879	13.043294	2.313372e-01	0.23
Year_1992	-0.884611	13.043353	4.128748e-01	0.41
Year_1993	3.372216	13.043387	2.914303e+01	29.14
Year_1994	4.479327	13.047417	8.817535e+01	88.18
Year_1995	7.928297	13.049258	2.774699e+03	2774.70
Year_1996	11.493282	13.055366	9.805480e+04	98054.80
Year_1997	23.502726	13.410588	1.611032e+10	16110323612.51

```
Year_1998           23.763402       13.462704   2.090810e+10      20908098218.83
Year_1999           26.123963       13.527512   2.215607e+11     221560731534.23
Year_2000           28.893508       13.593048   3.534200e+12    3534199522921.45
Year_2001           24.780165       13.566988   5.779480e+10      57794800746.64
Year_2002           26.297595       13.603688   2.635728e+11     263572800496.52
Year_2003           29.130776       13.659968   4.480588e+12    4480588483275.43
Year_2004           30.570313       13.734804   1.890245e+13   18902448581468.15
Year_2005           32.141164       13.796127   9.093481e+13   90934810995337.36
Year_2006           28.673352       13.818555   2.835819e+12    2835819050952.41
Year_2007           30.924946       13.838326   2.694843e+13   26948430147870.68
Year_2008           28.677480       13.792528   2.847550e+12    2847549525346.39
Year_2009           17.177514       13.654489   2.884692e+07         28846921.92
Year_2010           23.752699       13.735249   2.068551e+10      20685505816.93
Year_2011           19.940085       13.796590   4.569503e+08        456950294.59
Year_2012           12.965407       13.847560   4.273708e+05           427370.75
Year_2013           19.949988       13.874889   4.614976e+08        461497647.46
Year_2014           22.711776       13.959932   7.304643e+09       7304642952.22
Year_2015           15.437995       14.003381   5.065660e+06          5065659.75
Year_2016           10.031475       14.027295   2.273076e+04            22730.76
Year_2017            9.882548       14.058759   1.958557e+04            19585.57
Year_2018           16.391553       14.141223   1.314501e+07         13145014.67
Year_2019           11.414526       14.183638   9.062864e+04            90628.64
Year_2020           -4.268706       14.003198   1.399988e-02                0.01
error variance    1020.732743             NaN            inf                 inf
```

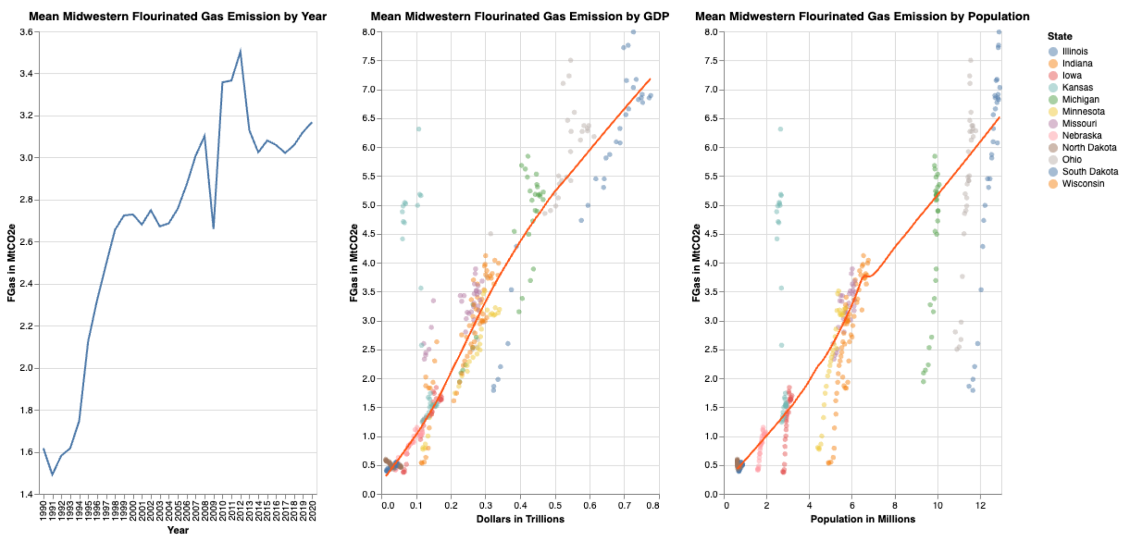
When looking at the coefficients of our year to year predictors for carbon dioxide in the midwest, our values range anywhere from 0 to 90934810995331. However, the coefficient for the yearly predictor stays consistently high. The coefficients for GDP and population are 0 and 34396888867.96 respectively. This suggests that GDP has very little influence on carbon dioxide production.

Thus, the year of carbon dioxide emissions is our strongest predictor of total carbon dioxide emissions in the midwest.

6.1.4 Flourinated Gas

[16]:
```
year_mw_FGas | gdp_mw_FGas + gdp_mw_FGas_loess | pop_mw_FGas + pop_mw_FGas_loess
```

[16]:



All three of our relevant predictors for flourinated gasses in the midwest follow a strong linear positive trend. For the yearly predictor, we observe a significant dip from around 2009. Although the GDP and population scatterplots look similar, we observe higher interstate variance for the GDP plot. However, the population plot has overall higher variance in regards to the main trend line.

[17]:
```
mw_FGas_coef_tbl
```

[17]:

	estimate	standard error	exponentiated	exp_visual
const	0.402481	0.280300	1.495531	1.50
Population	0.015318	0.047436	1.015436	1.02
GDP	9.136312	1.019473	9286.450880	9286.45
Year_1991	-0.127659	0.359593	0.880153	0.88
Year_1992	-0.089244	0.359594	0.914623	0.91
Year_1993	-0.076035	0.359595	0.926784	0.93
Year_1994	-0.027345	0.359706	0.973025	0.97
Year_1995	0.316856	0.359757	1.372805	1.37
Year_1996	0.455927	0.359926	1.577635	1.58
Year_1997	-0.018865	0.369719	0.981312	0.98
Year_1998	0.076778	0.371155	1.079802	1.08
Year_1999	0.069757	0.372942	1.072248	1.07
Year_2000	0.002204	0.374749	1.002206	1.00
Year_2001	-0.030224	0.374031	0.970228	0.97
Year_2002	-0.006969	0.375042	0.993055	0.99
Year_2003	-0.139564	0.376594	0.869737	0.87
Year_2004	-0.195554	0.378657	0.822379	0.82
Year_2005	-0.180738	0.380348	0.834654	0.83
Year_2006	-0.091539	0.380966	0.912526	0.91
Year_2007	0.016875	0.381511	1.017018	1.02
Year_2008	0.140298	0.380248	1.150617	1.15

Year_2009	-0.198841	0.376443	0.819680	0.82
Year_2010	0.425568	0.378669	1.530460	1.53
Year_2011	0.379352	0.380360	1.461337	1.46
Year_2012	0.470790	0.381766	1.601258	1.60
Year_2013	0.068300	0.382519	1.070687	1.07
Year_2014	-0.100681	0.384864	0.904222	0.90
Year_2015	-0.080165	0.386062	0.922964	0.92
Year_2016	-0.122374	0.386721	0.884818	0.88
Year_2017	-0.186447	0.387588	0.829902	0.83
Year_2018	-0.205380	0.389862	0.814338	0.81
Year_2019	-0.173196	0.391031	0.840972	0.84
Year_2020	-0.031226	0.386056	0.969256	0.97
error variance	0.775817	NaN	2.172366	2.17

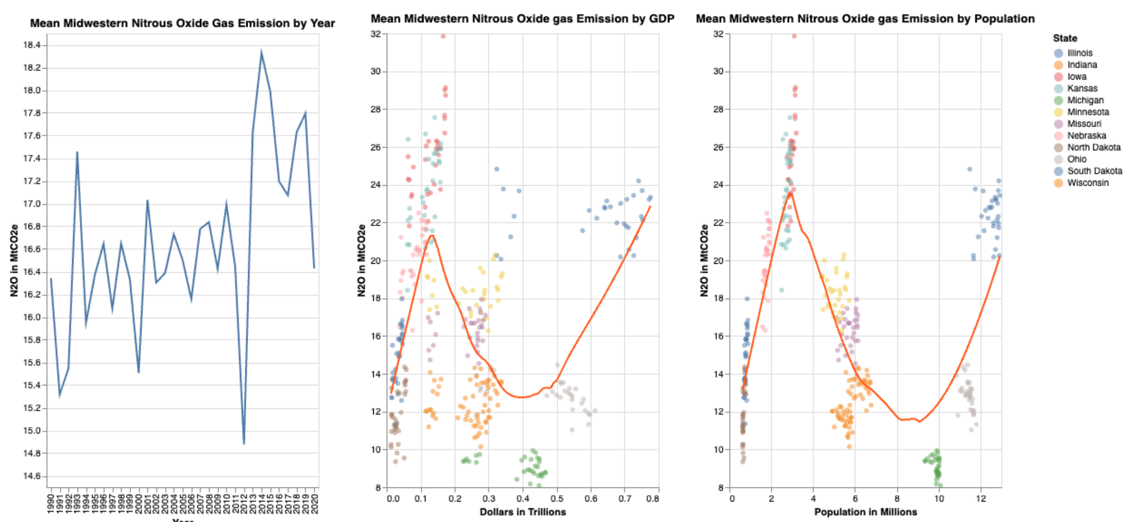
The yearly predictor coefficient for flourinated gasses in the midwest range from 0 to 1.60. This indicates that our time predictor may not be a strong contributor to flourinated gas emissions. The coefficient for our population predictor is 1.02, which also suggests weak contribution. However, the coefficient for our GDP predictor is 9286.45.

Thus, GDP seems to be our strongest predictor for flourinated gas emissions in the midwest.

6.1.5 Nitrous Oxide

[18]:
```
year_mw_N20 | gdp_mw_N20 + gdp_mw_N20_loess | pop_mw_N20 + pop_mw_N20_loess
```

[18]:



Our plot for Nitrous Oxide emissions over the years is highly random and almost looks like a white noise plot with a slight upward drift. Both the plots for GDP and Population also seem very scattered with no obvious pattern. The LOESS smoother gives us a line that increases, then decreases, then increases again for both plots, like a sideways z. Since these plots do not show a clear main contributer, we will need to look at the multiple linear regression model results.

[19]:
```
mw_N2O_coef_tbl
```

[19]:

	estimate	standard error	exponentiated	exp_visual
const	20.853549	1.627357	1.139151e+09	1139150850.83
Population	-1.736935	0.275405	1.760591e-01	0.18
GDP	33.317740	5.918824	2.949244e+14	294924365131648.38
Year_1991	-0.952478	2.087712	3.857839e-01	0.39
Year_1992	-0.845451	2.087722	4.293636e-01	0.43
Year_1993	1.087990	2.087727	2.968302e+00	2.97
Year_1994	-0.646344	2.088372	5.239578e-01	0.52
Year_1995	-0.272291	2.088667	7.616322e-01	0.76
Year_1996	-0.137372	2.089645	8.716461e-01	0.87
Year_1997	-3.006811	2.146502	4.944913e-02	0.05
Year_1998	-2.621495	2.154843	7.269411e-02	0.07
Year_1999	-3.158325	2.165216	4.249687e-02	0.04
Year_2000	-4.190707	2.175706	1.513558e-02	0.02
Year_2001	-2.575256	2.171535	7.613434e-02	0.08
Year_2002	-3.414339	2.177409	3.289816e-02	0.03
Year_2003	-3.502498	2.186417	3.012204e-02	0.03
Year_2004	-3.374552	2.198396	3.423344e-02	0.03
Year_2005	-3.767620	2.208211	2.310700e-02	0.02
Year_2006	-4.170577	2.211801	1.544334e-02	0.02
Year_2007	-3.599400	2.214965	2.734012e-02	0.03
Year_2008	-3.409594	2.207635	3.305462e-02	0.03
Year_2009	-3.416916	2.185540	3.281349e-02	0.03
Year_2010	-3.086544	2.198467	4.565947e-02	0.05
Year_2011	-3.804526	2.208285	2.226976e-02	0.02
Year_2012	-5.501148	2.216443	4.082083e-03	0.00
Year_2013	-2.810553	2.220818	6.017169e-02	0.06
Year_2014	-2.337884	2.234430	9.653165e-02	0.10
Year_2015	-2.784223	2.241384	6.177707e-02	0.06
Year_2016	-3.630512	2.245212	2.650260e-02	0.03
Year_2017	-3.825385	2.250248	2.181003e-02	0.02
Year_2018	-3.463837	2.263447	3.130940e-02	0.03
Year_2019	-3.394164	2.270236	3.356861e-02	0.03
Year_2020	-4.317874	2.241355	1.332818e-02	0.01
error variance	26.150429	NaN	2.275030e+11	227502986428.91

Our model shows that GDP is the strongest contributor to Nitrous Oxide emissions, with an exponentiated coefficient of 15 figures. All the other variables do not have high coefficients except Year_1993 with a coefficient of 2.97, signifying that something may have happened that year.

6.2 Southwest

[20]:
```
# extract southwest region data
ghg_sw = ghg[ghg["Region"] == "Southwest"]
```

```
# Outlier Analysis ************************************************
# boxplot showing distributions of ch4 total emissions among states for each year
sw_CH4_outliers = alt.Chart(ghg_sw).encode(
    x = alt.X("Year:O"),
    y = alt.Y("CH4TOT"),
    tooltip = ("State", "CH4TOT")
).mark_boxplot()

# boxplot showing distributions of co2 total emissions among states for each year
sw_CO2_outliers = alt.Chart(ghg_sw).encode(
    x = alt.X("Year:O"),
    y = alt.Y("CO2TOT"),
    tooltip = ("State", "CO2TOT")
).mark_boxplot()

# boxplot showing distributions of fgas total emissions among states for each year
sw_FGas_outliers = alt.Chart(ghg_sw).encode(
    x = alt.X("Year:O"),
    y = alt.Y("FGasTOT"),
    tooltip = ("State", "FGasTOT")
).mark_boxplot()

# boxplot showing distributions of n2o total emissions among states for each year
sw_N2O_outliers = alt.Chart(ghg_sw).encode(
    x = alt.X("Year:O"),
    y = alt.Y("N2OTOT"),
    tooltip = ("State", "N2OTOT")
).mark_boxplot()


# Methane Analysis ************************************************
# plot mean ch4 emission vs. year
year_sw_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southwest") & (ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.Scale(zero = False)),
    #color = alt.Color("Gas Type:N"),
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Southwestern Methane Gas Emission by Year"
```

```
)

# plot mean ch4 emission vs. gdp
gdp_sw_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southwest") & (ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southwestern Methane Gas Emission by GDP"
)

# plot mean ch4 emission vs. population
pop_sw_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southwest") & (ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southwestern Methane gas Emission by Population"
)

# LOESS smoother
# extract southwestern ch4 data
sw_CH4 = ghg_melted[(ghg_melted['Region'] == 'Southwest') & (ghg_melted['Gas Type'] == 'CH4TOT')]

# grid of gdp and population values
sw_grid_gdp = np.linspace(sw_CH4["GDP"].min(), sw_CH4["GDP"].max(), num = 100)
sw_grid_pop = np.linspace(sw_CH4["Population"].min(), sw_CH4["Population"].max(), num = 100)

# fit loess smooth for gdp and population plots
gdp_sw_CH4_ls = sm.nonparametric.lowess(endog = sw_CH4["Gas Total"].values,
                                    exog = sw_CH4["GDP"].values,
                                    frac = 0.5,
                                    xvals = sw_grid_gdp)
pop_sw_CH4_ls = sm.nonparametric.lowess(endog = sw_CH4["Gas Total"].values,
                                    exog = sw_CH4["Population"].values,
                                    frac = 0.5,
```

```
                                              xvals = sw_grid_pop)

# store as dataframe
gdp_sw_CH4_df = pd.DataFrame({'GDP': sw_grid_gdp, 'Gas': gdp_sw_CH4_ls})
pop_sw_CH4_df = pd.DataFrame({'Population': sw_grid_pop, 'Gas': pop_sw_CH4_ls})


# loess smoother lines for gdp and population plots
gdp_sw_CH4_loess = alt.Chart(
    gdp_sw_CH4_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_sw_CH4_loess = alt.Chart(
    pop_sw_CH4_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# plot for outlier Texas's GDP by Year
gdp_sw_CH4_Texas = alt.Chart(ghg).transform_filter(
    alt.FieldEqualPredicate(field = "State", equal = "Texas")
).encode(
    x = alt.X("Year:O"),
    y = alt.Y("GDP")
).mark_line().properties(title = "Texas GDP")

# regression analysis
# create dummy and x/y variables for mlr
sw_CH4_indicators = pd.get_dummies(sw_CH4[['Year', 'Population', 'GDP']],
                           drop_first = True)
sw_CH4_x = sm.tools.add_constant(sw_CH4_indicators)
sw_CH4_y = sw_CH4['Gas Total']
sw_CH4_indicators.columns.values

# fit mlr model
sw_CH4_mlr = sm.OLS(endog = sw_CH4_y, exog = sw_CH4_x.astype(float))
sw_CH4_rslt = sw_CH4_mlr.fit()

# retreive estimates and std errors
sw_CH4_coef_tbl = pd.DataFrame({
```

```
    'estimate': sw_CH4_rslt.params.values,
    'standard error': np.sqrt(sw_CH4_rslt.cov_params().values.diagonal())},
    index = sw_CH4_x.columns
)
sw_CH4_coef_tbl.loc['error variance', 'estimate'] = sw_CH4_rslt.scale

# add column of exponentiated coefficients
sw_CH4_coef_tbl['exponentiated']=np.exp(sw_CH4_coef_tbl['estimate'])
sw_CH4_coef_tbl['exp_visual'] = sw_CH4_coef_tbl['exponentiated'].apply(lambda x:
 ↪ "{:.2f}".format(x))



# Carbon Dioxide Analysis ************************************************
# plot mean co2 emission vs. year
year_sw_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southwest") &␣
 ↪(ghg_melted["Gas Type"] == "CO2TOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    #color = alt.Color("Gas Type:N")
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Southwestern CO2 Gas Emission by Year"
)
# plot mean co2 emission vs. gdp
gdp_sw_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southwest") &␣
 ↪(ghg_melted["Gas Type"] == "CO2TOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southwestern CO2 Gas Emission by GDP"
)
# plot mean co2 emission vs. population
pop_sw_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southwest") &␣
 ↪(ghg_melted["Gas Type"] == "CO2TOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
```

```
    width = 300,
    title = "Mean Southwestern CO2 gas Emission by Population"
)

# LOESS smoother

# extract midwestern co2 data
sw_CO2 = ghg_melted[(ghg_melted['Region'] == 'Southwest') & (ghg_melted['Gas␣
 ↪Type'] == 'CO2TOT')]

# fit loess smooth for gdp and population plots
gdp_sw_CO2_ls = sm.nonparametric.lowess(endog = sw_CO2["Gas Total"].values,
                                    exog = sw_CO2["GDP"].values,
                                    frac = 0.5,
                                    xvals = sw_grid_gdp)
pop_sw_CO2_ls = sm.nonparametric.lowess(endog = sw_CO2["Gas Total"].values,
                                    exog = sw_CO2["Population"].values,
                                    frac = 0.5,
                                    xvals = sw_grid_pop)

# store as dataframe
gdp_sw_CO2_df = pd.DataFrame({'GDP': sw_grid_gdp, 'Gas': gdp_sw_CO2_ls})
pop_sw_CO2_df = pd.DataFrame({'Population': sw_grid_pop, 'Gas': pop_sw_CO2_ls})

# loess smoother lines for gdp and population plots
gdp_sw_CO2_loess = alt.Chart(
    gdp_sw_CO2_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_sw_CO2_loess = alt.Chart(
    pop_sw_CO2_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# plot for outlier Texas's GDP by Year
gdp_sw_CO2_Texas = alt.Chart(ghg).transform_filter(
    alt.FieldEqualPredicate(field = "State", equal = "Texas")
).encode(
    x = alt.X("Year:O"),
```

```
    y = alt.Y("GDP")
).mark_line().properties(title = "Texas GDP")

# create dummy and x/y variables for mlr
sw_CO2_indicators = pd.get_dummies(sw_CO2[['Year', 'Population', 'GDP']],
                                   drop_first = True)
sw_CO2_x = sm.tools.add_constant(sw_CO2_indicators)
sw_CO2_y = sw_CO2['Gas Total']
sw_CO2_indicators.columns.values

# fit mlr model
sw_CO2_mlr = sm.OLS(endog = sw_CO2_y, exog = sw_CO2_x.astype(float))
sw_CO2_rslt = sw_CO2_mlr.fit()

# retreive estimates and std errors
sw_CO2_coef_tbl = pd.DataFrame({
    'estimate': sw_CO2_rslt.params.values,
    'standard error': np.sqrt(sw_CO2_rslt.cov_params().values.diagonal())},
    index = sw_CO2_x.columns
)
sw_CO2_coef_tbl.loc['error variance', 'estimate'] = sw_CO2_rslt.scale

# add column of exponentiated coefficients
sw_CO2_coef_tbl['exponentiated']=np.exp(sw_CO2_coef_tbl['estimate'])
sw_CO2_coef_tbl['exp_visual'] = sw_CO2_coef_tbl['exponentiated'].apply(lambda x:
 ↪ "{:.2f}".format(x))


# Flourinated Gases Analysis ************************************************
# plot mean fgas emission vs. year
year_sw_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southwest") &␣
 ↪(ghg_melted["Gas Type"] == "FGasTOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    #color = alt.Color("Gas Type:N")
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Southwestern Flourinated Gas Emission by Year"
)
# plot mean fgas emission vs. gdp
gdp_sw_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southwest") &␣
 ↪(ghg_melted["Gas Type"] == "FGasTOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
```

```
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southwestern Flourinated Gas Emission by GDP"
)
# plot mean fgas emission vs. population
pop_sw_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southwest") &␣
↪(ghg_melted["Gas Type"] == "FGasTOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southwestern Flourinated Gas Emission by Population"
)

# LOESS smoother
# extract midwestern fgas data
sw_FGas = ghg_melted[(ghg_melted['Region'] == 'Southwest') & (ghg_melted['Gas␣
↪Type'] == 'FGasTOT')]

# fit loess smooth for gdp and population plots
gdp_sw_FGas_ls = sm.nonparametric.lowess(endog = sw_FGas["Gas Total"].values,
                                    exog = sw_FGas["GDP"].values,
                                    frac = 0.5,
                                    xvals = sw_grid_gdp)
pop_sw_FGas_ls = sm.nonparametric.lowess(endog = sw_FGas["Gas Total"].values,
                                    exog = sw_FGas["Population"].values,
                                    frac = 0.5,
                                    xvals = sw_grid_pop)

# store as dataframe
gdp_sw_FGas_df = pd.DataFrame({'GDP': sw_grid_gdp, 'Gas': gdp_sw_FGas_ls})
pop_sw_FGas_df = pd.DataFrame({'Population': sw_grid_pop, 'Gas':␣
↪pop_sw_FGas_ls})


# loess smoother lines for gdp and population plots
gdp_sw_FGas_loess = alt.Chart(
    gdp_sw_FGas_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
```

```
    color = "#FF5919"
)
pop_sw_FGas_loess = alt.Chart(
    pop_sw_FGas_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# plot for outlier Texas's GDP by Year
gdp_sw_FGas_Texas = alt.Chart(ghg).transform_filter(
    alt.FieldEqualPredicate(field = "State", equal = "Texas")
).encode(
    x = alt.X("Year:O"),
    y = alt.Y("GDP")
).mark_line().properties(title = "Texas GDP")

# create dummy and x/y variables for mlr
sw_FGas_indicators = pd.get_dummies(sw_FGas[['Year', 'Population', 'GDP']],
                           drop_first = True)
sw_FGas_x = sm.tools.add_constant(sw_FGas_indicators)
sw_FGas_y = sw_FGas['Gas Total']
sw_FGas_indicators.columns.values

# fit mlr model
sw_FGas_mlr = sm.OLS(endog = sw_FGas_y, exog = sw_FGas_x.astype(float))
sw_FGas_rslt = sw_FGas_mlr.fit()

# retreive estimates and std errors
sw_FGas_coef_tbl = pd.DataFrame({
    'estimate': sw_FGas_rslt.params.values,
    'standard error': np.sqrt(sw_FGas_rslt.cov_params().values.diagonal())},
    index = sw_FGas_x.columns
)
sw_FGas_coef_tbl.loc['error variance', 'estimate'] = sw_FGas_rslt.scale

# add column of exponentiated coefficients
sw_FGas_coef_tbl['exponentiated']=np.exp(sw_FGas_coef_tbl['estimate'])
sw_FGas_coef_tbl['exp_visual'] = sw_FGas_coef_tbl['exponentiated'].apply(lambda␣
 ↪x: "{:.2f}".format(x))



# Nitrous Oxide Analysis ************************************************
# plot mean n2o emission vs. year
```

```
year_sw_N2O = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southwest") &␣
↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
↪Scale(zero = False)),
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Southwestern Nitrous Oxide Gas Emission by Year"
)
# plot mean n2o emission vs. gdp
gdp_sw_N2O = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southwest") &␣
↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southwestern Nitrous Oxide gas Emission by GDP"
)
# plot mean n2o emission vs. population
pop_sw_N2O = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southwest") &␣
↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southwestern Nitrous Oxide gas Emission by Population"
)

# LOESS smoother
# extract midwestern n2o data
sw_N2O = ghg_melted[(ghg_melted['Region'] == 'Southwest') & (ghg_melted['Gas␣
↪Type'] == 'N2OTOT')]

# fit loess smooth for gdp and population plots
gdp_sw_N2O_ls = sm.nonparametric.lowess(endog = sw_N2O["Gas Total"].values,
                                    exog = sw_N2O["GDP"].values,
                                    frac = 0.5,
                                    xvals = sw_grid_gdp)
pop_sw_N2O_ls = sm.nonparametric.lowess(endog = sw_N2O["Gas Total"].values,
```

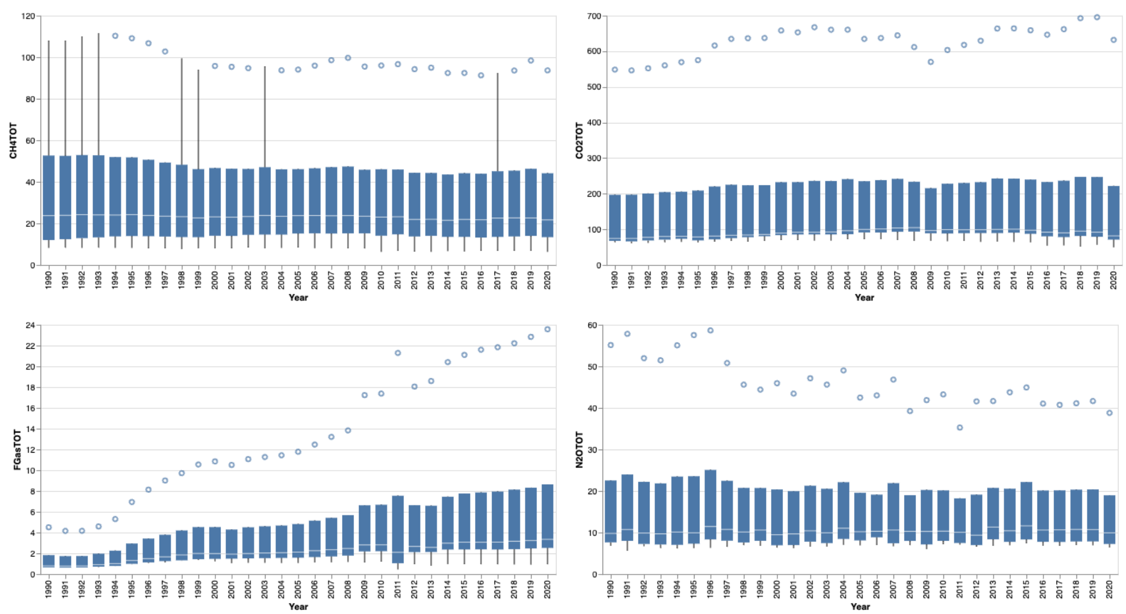
```
                                    exog = sw_N20["Population"].values,
                                    frac = 0.5,
                                    xvals = sw_grid_pop)

# store as dataframe
gdp_sw_N20_df = pd.DataFrame({'GDP': sw_grid_gdp, 'Gas': gdp_sw_N20_ls})
pop_sw_N20_df = pd.DataFrame({'Population': sw_grid_pop, 'Gas': pop_sw_N20_ls})


# loess smoother lines for gdp and population plots
gdp_sw_N20_loess = alt.Chart(
    gdp_sw_N20_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False)),
).mark_line(
    color = "#FF5919"
)
pop_sw_N20_loess = alt.Chart(
    pop_sw_N20_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# plot for outlier Texas's GDP by Year
gdp_sw_N20_Texas = alt.Chart(ghg).transform_filter(
    alt.FieldEqualPredicate(field = "State", equal = "Texas")
).encode(
    x = alt.X("Year:O"),
    y = alt.Y("GDP")
).mark_line().properties(title = "Texas GDP")

# create dummy and x/y variables for mlr
sw_N20_indicators = pd.get_dummies(sw_N20[['Year', 'Population', 'GDP']],
                           drop_first = True)
sw_N20_x = sm.tools.add_constant(sw_N20_indicators)
sw_N20_y = sw_N20['Gas Total']
sw_N20_indicators.columns.values

# fit mlr model
sw_N20_mlr = sm.OLS(endog = sw_N20_y, exog = sw_N20_x.astype(float))
sw_N20_rslt = sw_N20_mlr.fit()

# retreive estimates and std errors
```

```
sw_N2O_coef_tbl = pd.DataFrame({
    'estimate': sw_N2O_rslt.params.values,
    'standard error': np.sqrt(sw_N2O_rslt.cov_params().values.diagonal())},
    index = sw_N2O_x.columns
)
sw_N2O_coef_tbl.loc['error variance', 'estimate'] = sw_N2O_rslt.scale

# add column of exponentiated coefficients
sw_N2O_coef_tbl['exponentiated']=np.exp(sw_N2O_coef_tbl['estimate'])
sw_N2O_coef_tbl['exp_visual'] = sw_N2O_coef_tbl['exponentiated'].apply(lambda x:
 ↪ "{:.2f}".format(x))
```

6.2.1 Outlier Analysis

[21]:
```
sw_gases = (sw_CH4_outliers | sw_CO2_outliers) & (sw_FGas_outliers |␣
 ↪sw_N2O_outliers)
sw_gases
```

[21]:

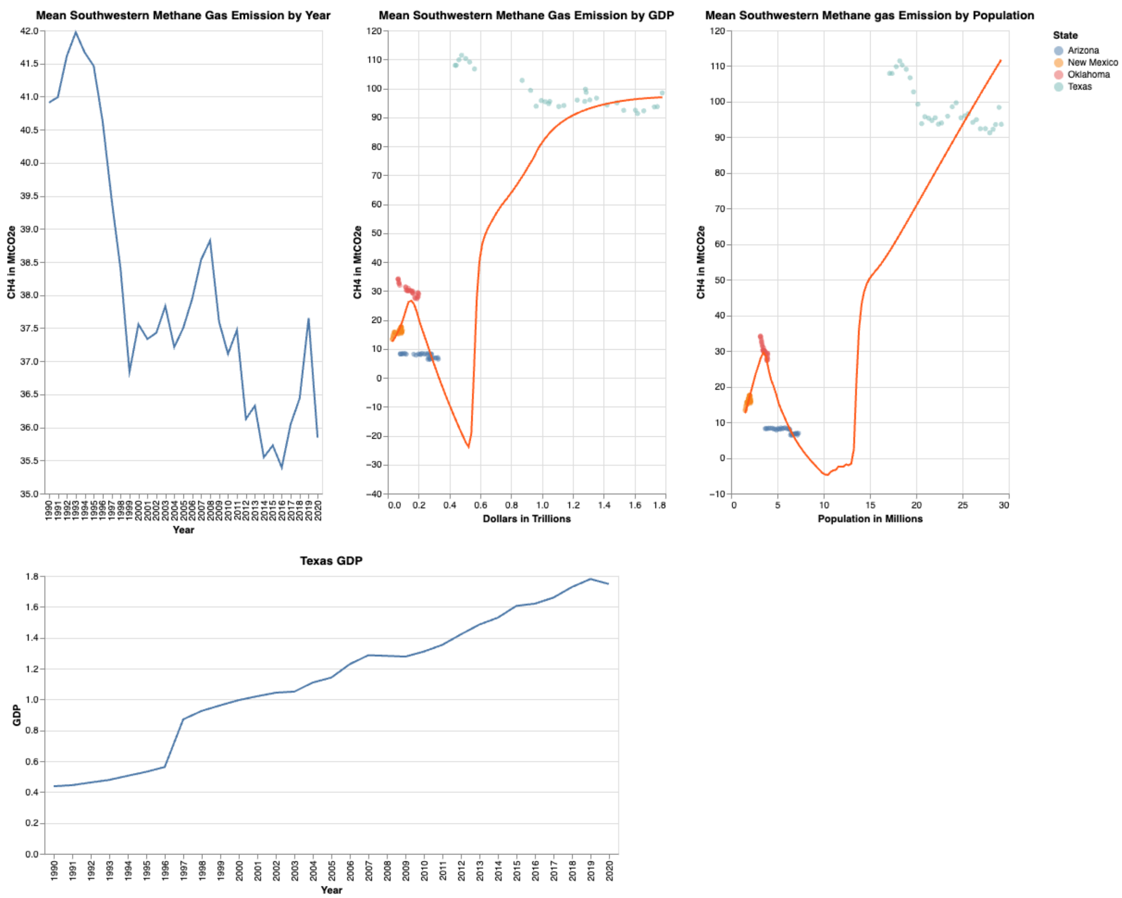


Here we see that for the Southwest region Texas is the main outlier for the methane, carbon dioxide, fluorinated, and nitrogen gases. The median of most of our gases stay stable, except for fluorinated gases increasing by around 3 MtCO2e. Interestingly, most of Texas' gas total stayed around the same as its 1990 value except for carbon dioxide which spiked from 550 to 630. We continue our analysis by analyzing this region's gases further, identifying outliers, GDP, and capita's influences on each gas.

6.2.2 Methane

[22]:
```
(year_sw_CH4 | gdp_sw_CH4 + gdp_sw_CH4_loess | pop_sw_CH4 + pop_sw_CH4_loess) &␣
↪gdp_sw_CH4_Texas
```

[22]:



We see that overtime methane's mean gas emissions in the Southwestern area decrease overtime with small spikes in 1993, 2008, and 2019. In addition, we see that lower GDP outliers have lower emission than the higher emission state Texas with a higher GDP and GDP range.

Interestingly when the Texas GDP methane emissions spikes near a high 0.5 GDP to a high 0.8 GDP (high 500 billion-high 800 billion USD) from 1996-1997, the mean emissions also spiked at the same time, indicating that Texas's influence was large and played into mean emission levels for the Southwest region.

With a high population, Texas has high emissions until 1993 then decreases in emissions till 1999 despite an increasing population. Overall, changes in Texas GDP doesn't influence emissions by a very notable amount

[23]:
```
sw_CH4_coef_tbl
```

[23]:

	estimate	standard error	exponentiated \
const	5.333213	7.768449	2.071022e+02
Population	7.074928	0.735721	1.181959e+03
GDP	-62.236066	13.857377	9.358794e-28
Year_1991	-0.400800	10.328730	6.697839e-01
Year_1992	-0.493798	10.328859	6.103037e-01
Year_1993	-0.816451	10.329080	4.419976e-01
Year_1994	-1.395148	10.329021	2.477963e-01
Year_1995	-2.266407	10.329268	1.036841e-01
Year_1996	-3.532005	10.329581	2.924621e-02
Year_1997	0.751133	10.419873	2.119399e+00
Year_1998	-0.364599	10.437835	6.944748e-01
Year_1999	-2.062240	10.452015	1.271688e-01
Year_2000	-1.467462	10.464995	2.305097e-01
Year_2001	-2.138843	10.468486	1.177911e-01
Year_2002	-2.427917	10.473367	8.822038e-02
Year_2003	-2.534777	10.473223	7.927937e-02
Year_2004	-2.855498	10.504836	5.752719e-02
Year_2005	-2.672904	10.523113	6.905144e-02
Year_2006	-2.148979	10.567115	1.166032e-01
Year_2007	-1.674529	10.594005	1.873964e-01
Year_2008	-2.500738	10.570548	8.202448e-02
Year_2009	-5.315564	10.528267	4.914505e-03
Year_2010	-6.251894	10.533230	1.926801e-03
Year_2011	-5.885943	10.554321	2.778225e-03
Year_2012	-7.041503	10.591816	8.748111e-04
Year_2013	-6.676835	10.624533	1.259759e-03
Year_2014	-7.431801	10.653577	5.921200e-04
Year_2015	-7.081188	10.701391	8.407735e-04
Year_2016	-8.173951	10.691746	2.819021e-04
Year_2017	-7.524827	10.718026	5.395219e-04
Year_2018	-6.505027	10.779431	1.495900e-03
Year_2019	-5.132900	10.823043	5.899429e-03
Year_2020	-7.840149	10.774826	3.936105e-04
error variance	213.363267	NaN	4.597158e+92

	exp_visual
const	207.10
Population	1181.96
GDP	0.00
Year_1991	0.67
Year_1992	0.61
Year_1993	0.44
Year_1994	0.25
Year_1995	0.10
Year_1996	0.03
Year_1997	2.12

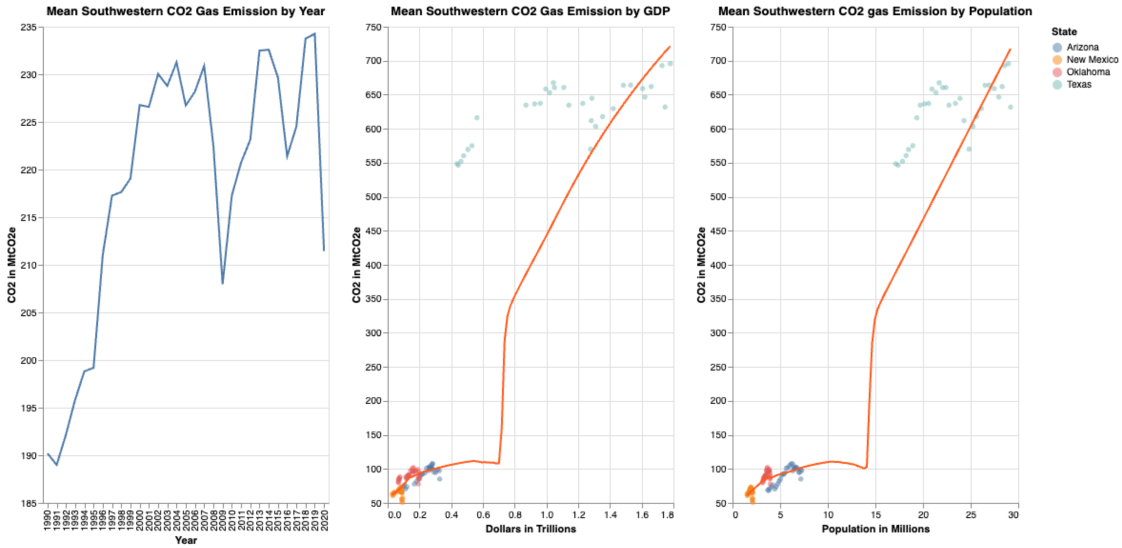
```
Year_1998                                                  0.69
Year_1999                                                  0.13
Year_2000                                                  0.23
Year_2001                                                  0.12
Year_2002                                                  0.09
Year_2003                                                  0.08
Year_2004                                                  0.06
Year_2005                                                  0.07
Year_2006                                                  0.12
Year_2007                                                  0.19
Year_2008                                                  0.08
Year_2009                                                  0.00
Year_2010                                                  0.00
Year_2011                                                  0.00
Year_2012                                                  0.00
Year_2013                                                  0.00
Year_2014                                                  0.00
Year_2015                                                  0.00
Year_2016                                                  0.00
Year_2017                                                  0.00
Year_2018                                                  0.00
Year_2019                                                  0.01
Year_2020                                                  0.00
error variance  4597158281580478961396830517453186159944478027…
```

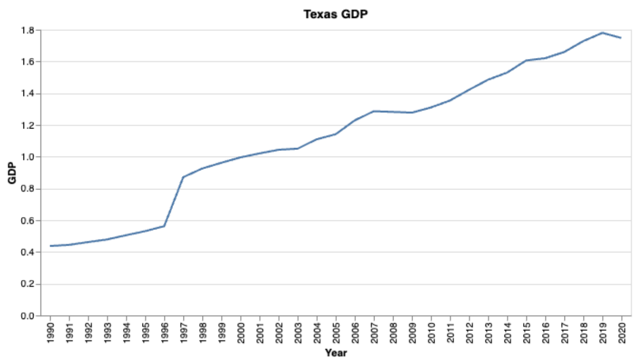
Fitting a linear regression model and extracting the exponentiated coefficients shows us that between GDP and Population, Population influences methane emissions more, with a exponentiated coefficient of 1181.96 while the coefficient for GDP is 0. We see years don't play a super signifcant role for methane since all our coefficents are close to each other.

6.2.3 Carbon Dioxide

[24]:
```
(year_sw_CO2 | gdp_sw_CO2 + gdp_sw_CO2_loess | pop_sw_CO2 + pop_sw_CO2_loess) &
 ↪gdp_sw_CO2_Texas
```

[24]:





We see that carbon dioxide's mean gas emissions in the Southwestern area increase overtime, there is a large spike from 1995-1997. In addition, we see that lower GDP outliers have much lower emissions than the higher emission and GDP states. Texas is again the highest emission state that drags up the mean emission.

Interestingly when the Texas GDP spikes near a high 0.5 GDP to a high 0.8 GDP (high 500 billion-high 800 billion USD) from 1996-1997, the mean GDP's emissions stay level. Overall there seems to be cyclical increasing trend in emissions for Texas by GDP or population since the outliers spike then dip then spike again.

With a high population, Texas has high emissions until 1993 then decreases in emissions till 1999 despite an linearly increasing population.

[25]:
```
sw_CO2_coef_tbl
```

[25]:

	estimate	standard error	exponentiated	\
const	-17.633944	23.846808	2.196218e-08	
Population	37.378119	2.258443	1.710452e+16	
GDP	-199.363314	42.537990	2.615851e-87	
Year_1991	-4.135617	31.706102	1.599279e-02	

Year_1992	-5.674154	31.706497	3.433574e-03
Year_1993	-6.478293	31.707176	1.536432e-03
Year_1994	-6.297508	31.706996	1.840886e-03
Year_1995	-10.620355	31.707753	2.441397e-05
Year_1996	-2.494307	31.708714	8.255365e-02
Year_1997	19.075459	31.985885	1.924715e+08
Year_1998	16.829669	32.041022	2.037195e+07
Year_1999	15.668630	32.084551	6.379691e+06
Year_2000	21.104957	32.124395	1.464759e+09
Year_2001	17.423859	32.135111	3.690507e+07
Year_2002	17.719669	32.150094	4.960834e+07
Year_2003	12.968047	32.149653	4.285005e+05
Year_2004	14.430952	32.246693	1.850472e+06
Year_2005	7.497854	32.302800	1.804167e+03
Year_2006	5.829630	32.437871	3.402327e+02
Year_2007	5.611796	32.520416	2.736352e+02
Year_2008	-8.643523	32.448409	1.762648e-04
Year_2009	-30.633804	32.318619	4.964873e-14
Year_2010	-24.846753	32.333855	1.618796e-11
Year_2011	-23.140167	32.398598	8.919764e-11
Year_2012	-22.407971	32.513696	1.854992e-10
Year_2013	-14.342735	32.614129	5.902407e-07
Year_2014	-16.145204	32.703285	9.732561e-08
Year_2015	-21.083212	32.850061	6.977139e-10
Year_2016	-33.954094	32.820451	1.794422e-15
Year_2017	-32.601874	32.901124	6.937231e-15
Year_2018	-22.742685	33.089619	1.327325e-10
Year_2019	-23.691946	33.223494	5.137105e-11
Year_2020	-49.910157	33.075482	2.110057e-22
error variance	2010.534362	NaN	inf

	exp_visual
const	0.00
Population	17104517795324408.00
GDP	0.00
Year_1991	0.02
Year_1992	0.00
Year_1993	0.00
Year_1994	0.00
Year_1995	0.00
Year_1996	0.08
Year_1997	192471490.85
Year_1998	20371946.40
Year_1999	6379691.36
Year_2000	1464759193.85
Year_2001	36905067.21
Year_2002	49608336.41

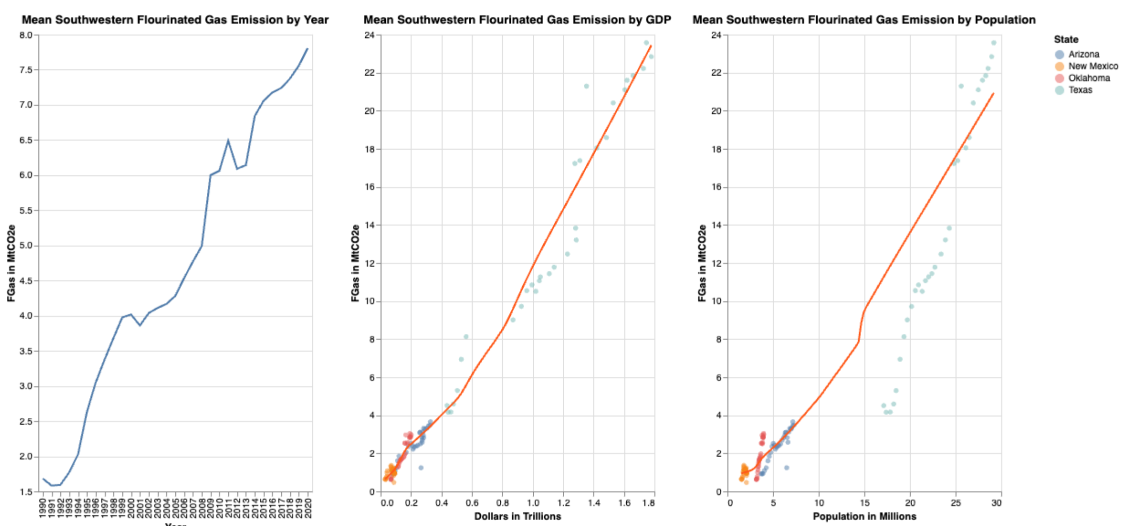
```
Year_2003                   428500.55
Year_2004                  1850472.44
Year_2005                     1804.17
Year_2006                      340.23
Year_2007                      273.64
Year_2008                        0.00
Year_2009                        0.00
Year_2010                        0.00
Year_2011                        0.00
Year_2012                        0.00
Year_2013                        0.00
Year_2014                        0.00
Year_2015                        0.00
Year_2016                        0.00
Year_2017                        0.00
Year_2018                        0.00
Year_2019                        0.00
Year_2020                        0.00
error variance                    inf
```

Fitting a linear regression model and extracting the exponentiated coefficients shows us that Population influences methane emissions much more with a exponentiated coefficient of 17104517795323800.00 compared to GDP's coefficient of 0. We see that earlier years are extremely significant for carbon dioxide, specifically from 1997 to 2007 there are high coefficients where asfrom 1991-1997 and 2008-2020 the coefficients are 0.

6.2.4 Fluorinated Gas

[26]:
```
year_sw_FGas | gdp_sw_FGas + gdp_sw_FGas_loess | pop_sw_FGas + pop_sw_FGas_loess
```

[26]:



Here we see that the average Southwestern fluorinated gas emissions increased linearly overtime, similarly by GDP and capita there was also linear growth in emissions as GDP and capita increased. Outliers fit our best fit line's trends of increasing GDP/capita over time as their emissions increased with a positive correlation.

[27]:
```
sw_FGas_coef_tbl
```

[27]:

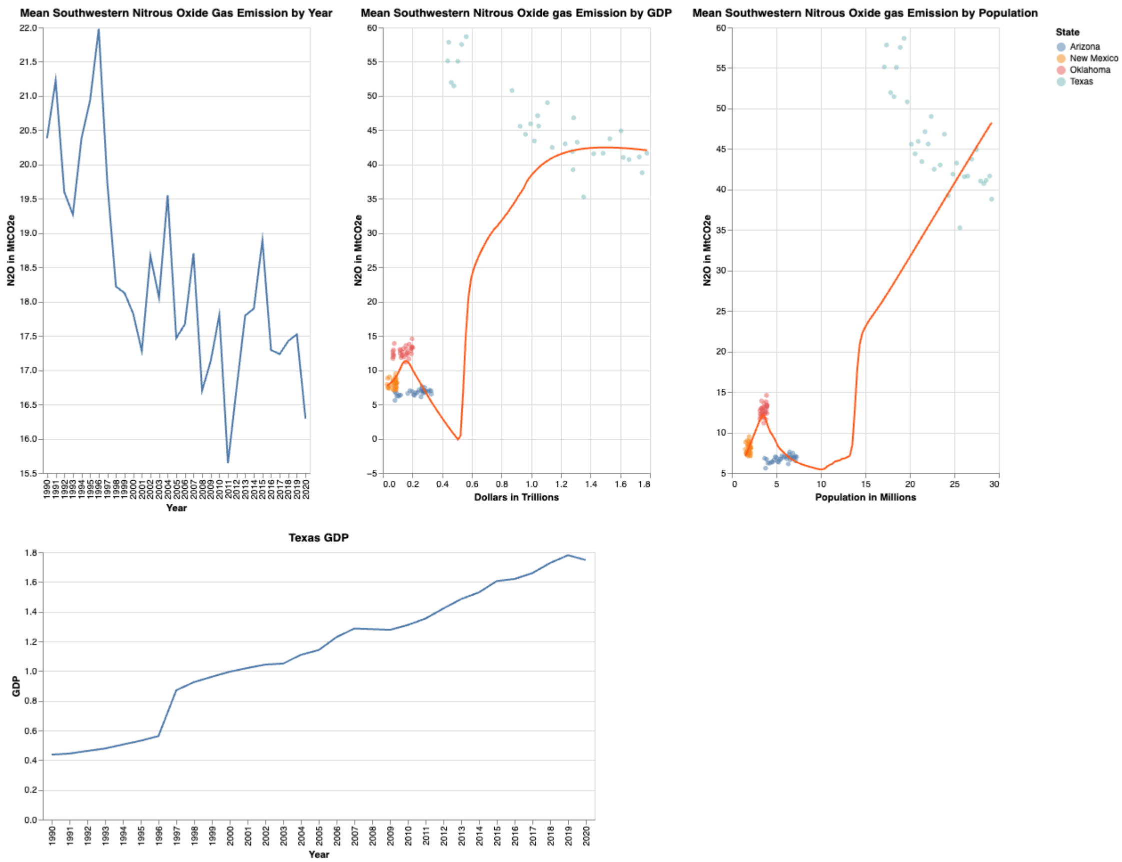
	estimate	standard error	exponentiated	exp_visual
const	0.523860	0.460047	1.688533e+00	1.69
Population	-0.206998	0.043569	8.130217e-01	0.81
GDP	16.132114	0.820632	1.014117e+07	10141165.60
Year_1991	-0.124931	0.611666	8.825582e-01	0.88
Year_1992	-0.198732	0.611674	8.197698e-01	0.82
Year_1993	-0.094125	0.611687	9.101688e-01	0.91
Year_1994	0.018011	0.611683	1.018174e+00	1.02
Year_1995	0.512289	0.611698	1.669108e+00	1.67
Year_1996	0.781490	0.611716	2.184725e+00	2.18
Year_1997	-0.543899	0.617064	5.804805e-01	0.58
Year_1998	-0.492236	0.618127	6.112582e-01	0.61
Year_1999	-0.397747	0.618967	6.718322e-01	0.67
Year_2000	-0.529543	0.619736	5.888739e-01	0.59
Year_2001	-0.796587	0.619942	4.508651e-01	0.45
Year_2002	-0.733620	0.620231	4.801677e-01	0.48
Year_2003	-0.736817	0.620223	4.786350e-01	0.48
Year_2004	-0.974359	0.622095	3.774343e-01	0.38
Year_2005	-1.061393	0.623177	3.459736e-01	0.35
Year_2006	-1.205403	0.625783	2.995713e-01	0.30
Year_2007	-1.218094	0.627376	2.957933e-01	0.30
Year_2008	-0.964835	0.625986	3.810461e-01	0.38
Year_2009	0.196886	0.623483	1.217606e+00	1.22
Year_2010	0.145184	0.623777	1.156252e+00	1.16
Year_2011	0.376431	0.625026	1.457075e+00	1.46
Year_2012	-0.324038	0.627246	7.232229e-01	0.72
Year_2013	-0.515969	0.629183	5.969221e-01	0.60
Year_2014	-0.042077	0.630903	9.587955e-01	0.96
Year_2015	-0.152618	0.633735	8.584575e-01	0.86
Year_2016	-0.078454	0.633164	9.245447e-01	0.92
Year_2017	-0.202106	0.634720	8.170082e-01	0.82
Year_2018	-0.388696	0.638357	6.779402e-01	0.68
Year_2019	-0.459478	0.640939	6.316134e-01	0.63
Year_2020	-0.036635	0.638084	9.640279e-01	0.96
error variance	0.748264	NaN	2.113327e+00	2.11

Fitting a linear regression model and extracting the exponentiated coefficients interestingly, our population had a low exponentiated coefficient of 0.81 whereas GDP had a high exponentiated coefficient of 10141165.60, so GDP heavily influences fluorinated gas quantities in the Southwest region. Our year coefficent values are all very similar so time isn't a large influence.

6.2.5 Nitrous Oxide

[28]:
```
(year_sw_N20 | gdp_sw_N20 + gdp_sw_N20_loess | pop_sw_N20 + pop_sw_N20_loess) &
gdp_sw_N20_Texas
```

[28]:



We can see that our mean nitrous oxide levels decrease overtime, with a peak spike in emissions in 1996 and a massive dip in 2011. Again, in 1996 our Texas GDP spiked from a high 0.5 GDP to a high 0.8 GDP (high 500 billion-high 800 billion USD). During this spike time period for Texas the Southwestern's average N20 levels spiked to the peak of 22 MtCO2e. This means that an increase or spike in Texas's GDP was at the same time as the whole region's massive increase in emissions. In addition, for Texas as capita and GDP increased our outliers mostly showed a slight decreasing trend in N2O emissions.

[29]:
```
sw_N20_coef_tbl
```

[29]:

	estimate	standard error	exponentiated	exp_visual
const	1.426037	2.880891	4.162174e+00	4.16
Population	4.025515	0.272839	5.600914e+01	56.01
GDP	-43.770435	5.138940	9.789070e-20	0.00
Year_1991	0.591452	3.830358	1.806610e+00	1.81
Year_1992	-1.380599	3.830406	2.514278e-01	0.25

```
Year_1993       -2.043313        3.830488   1.295986e-01                0.13
Year_1994       -0.997039        3.830466   3.689704e-01                0.37
Year_1995       -0.749290        3.830558   4.727020e-01                0.47
Year_1996        0.145027        3.830674   1.156071e+00                1.16
Year_1997        1.875208        3.864159   6.522177e+00                6.52
Year_1998        0.477164        3.870820   1.611498e+00                1.61
Year_1999        0.421940        3.876078   1.524917e+00                1.52
Year_2000        0.154637        3.880892   1.167234e+00                1.17
Year_2001       -0.576613        3.882186   5.617978e-01                0.56
Year_2002        0.668317        3.883996   1.950951e+00                1.95
Year_2003       -0.179701        3.883943   8.355198e-01                0.84
Year_2004        1.653923        3.895666   5.227448e+00                5.23
Year_2005       -0.373806        3.902445   6.881104e-01                0.69
Year_2006        0.106613        3.918762   1.112503e+00                1.11
Year_2007        1.220098        3.928734   3.387520e+00                3.39
Year_2008       -1.413978        3.920035   2.431741e-01                0.24
Year_2009       -1.950013        3.904356   1.422722e-01                0.14
Year_2010       -1.461014        3.906196   2.320009e-01                0.23
Year_2011       -3.504478        3.914018   3.006246e-02                0.03
Year_2012       -2.133044        3.927922   1.184761e-01                0.12
Year_2013       -0.834393        3.940056   4.341379e-01                0.43
Year_2014       -0.591347        3.950826   5.535809e-01                0.55
Year_2015        0.691921        3.968558   1.997549e+00                2.00
Year_2016       -1.303471        3.964981   2.715875e-01                0.27
Year_2017       -1.255402        3.974727   2.849612e-01                0.28
Year_2018       -0.528599        3.997499   5.894299e-01                0.59
Year_2019       -0.194980        4.013672   8.228509e-01                0.82
Year_2020       -2.023963        3.995791   1.321307e-01                0.13
error variance  29.343008             NaN   5.539959e+12  5539958595958.60
```

Fitting a linear regression model and extracting the exponentiated coefficients interestingly, compared to GDP's coefficient of 0 our population had a relatively high exponentiated coefficient of 56.01 meaning that it influenced nitrous oxide's quantities in the Southwest region. Our year coefficent values are all quite similar so time isn't a large influence.

6.3 Northeast

[30]:
```
# extract northeast region data
ghg_ne = ghg[ghg["Region"] == "Northeast"]

# Outlier Analysis ************************************************
# boxplot showing distributions of ch4 total emissions among states for each␣
↪year
ne_CH4_outliers = alt.Chart(ghg_ne).encode(
    x = alt.X("Year:O"),
    y = alt.Y("CH4TOT"),
    tooltip = ("State", "CH4TOT")
```

```
).mark_boxplot()

# boxplot showing distributions of co2 total emissions among states for each␣
↪year
ne_CO2_outliers = alt.Chart(ghg_ne).encode(
    x = alt.X("Year:O"),
    y = alt.Y("CO2TOT"),
    tooltip = ("State", "CO2TOT")
).mark_boxplot()

# boxplot showing distributions of fgas total emissions among states for each␣
↪year

ne_FGas_outliers = alt.Chart(ghg_ne).encode(
    x = alt.X("Year:O"),
    y = alt.Y("FGasTOT"),
    tooltip = ("State", "FGasTOT")
).mark_boxplot()

# boxplot showing distributions of n2o total emissions among states for each␣
↪year
ne_N2O_outliers = alt.Chart(ghg_ne).encode(
    x = alt.X("Year:O"),
    y = alt.Y("N2OTOT"),
    tooltip = ("State", "N2OTOT")
).mark_boxplot()


# Methane Analysis ************************************************
# plot mean ch4 emission vs. year
year_ne_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Northeast") &␣
↪(ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.
↪Scale(zero = False)),
    #color = alt.Color("Gas Type:N")
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Northeastern Methane Gas Emission by Year"
)

# plot mean ch4 emission vs. gdp
gdp_ne_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Northeast") &␣
↪(ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
```

```
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    tooltip = ("State", "Year", "Gas Total"),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Northeastern Methane Gas Emission by GDP"
)

# plot mean ch4 emission vs. population
pop_ne_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Northeast") &␣
 ↪(ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    tooltip = ("State", "Year", "Gas Total"),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Northeastern Methane gas Emission by Population"
)

# LOESS smoother
# extract northeastern ch4 data
ne_CH4 = ghg_melted[(ghg_melted['Region'] == 'Northeast') & (ghg_melted['Gas␣
 ↪Type'] == 'CH4TOT')]

# grid of gdp and population values
ne_grid_gdp = np.linspace(ne_CH4["GDP"].min(), ne_CH4["GDP"].max(), num = 100)
ne_grid_pop = np.linspace(ne_CH4["Population"].min(), ne_CH4["Population"].
 ↪max(), num = 100)

# fit loess smooth for gdp and population plots
gdp_ne_CH4_ls = sm.nonparametric.lowess(endog = ne_CH4["Gas Total"].values,
                                    exog = ne_CH4["GDP"].values,
                                    frac = 0.5,
                                    xvals = ne_grid_gdp)
pop_ne_CH4_ls = sm.nonparametric.lowess(endog = ne_CH4["Gas Total"].values,
                                    exog = ne_CH4["Population"].values,
                                    frac = 0.5,
                                    xvals = ne_grid_pop)

# store as dataframe
gdp_ne_CH4_df = pd.DataFrame({'GDP': ne_grid_gdp, 'Gas': gdp_ne_CH4_ls})
pop_ne_CH4_df = pd.DataFrame({'Population': ne_grid_pop, 'Gas': pop_ne_CH4_ls})
```

```
# loess smoother lines for gdp and population plots
gdp_ne_CH4_loess = alt.Chart(
    gdp_ne_CH4_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_ne_CH4_loess = alt.Chart(
    pop_ne_CH4_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# year vs gdp and year vs population
ne_gdp_year = alt.Chart(ne_CH4).encode(
    x = alt.X("Year:O"),
    y = alt.Y("GDP"),
    color = alt.Color("State:N")
).mark_line()

ne_pop_year = alt.Chart(ne_CH4).encode(
    x = alt.X("Year:O"),
    y = alt.Y("Population"),
    color = alt.Color("State:N")
).mark_line()

# # create dummy and x/y variables for mlr
ne_CH4_indicators = pd.get_dummies(ne_CH4[['Year', 'Population', 'GDP']],
                           drop_first = True)
ne_CH4_x = sm.tools.add_constant(ne_CH4_indicators)
ne_CH4_y = ne_CH4['Gas Total']
ne_CH4_indicators.columns.values

# fit mlr model
ne_CH4_mlr = sm.OLS(endog = ne_CH4_y, exog = ne_CH4_x.astype(float))
ne_CH4_rslt = ne_CH4_mlr.fit()

# retreive estimates and std errors
ne_CH4_coef_tbl = pd.DataFrame({
    'estimate': ne_CH4_rslt.params.values,
    'standard error': np.sqrt(ne_CH4_rslt.cov_params().values.diagonal())},
```

```
    index = ne_CH4_x.columns
)
ne_CH4_coef_tbl.loc['error variance', 'estimate'] = ne_CH4_rslt.scale

# add column of exponentiated coefficients
ne_CH4_coef_tbl['exponentiated']=np.exp(ne_CH4_coef_tbl['estimate'])
ne_CH4_coef_tbl['exp_visual'] = ne_CH4_coef_tbl['exponentiated'].apply(lambda x:
 ↪ "{:.2f}".format(x))
# create dummy and x/y variables for mlr
ne_CH4_indicators = pd.get_dummies(ne_CH4[['Year', 'Population', 'GDP']],
                           drop_first = True)
ne_CH4_x = sm.tools.add_constant(ne_CH4_indicators)
ne_CH4_y = ne_CH4['Gas Total']
ne_CH4_indicators.columns.values

# fit mlr model
ne_CH4_mlr = sm.OLS(endog = ne_CH4_y, exog = ne_CH4_x.astype(float))
ne_CH4_rslt = ne_CH4_mlr.fit()

# retreive estimates and std errors
ne_CH4_coef_tbl = pd.DataFrame({
    'estimate': ne_CH4_rslt.params.values,
    'standard error': np.sqrt(ne_CH4_rslt.cov_params().values.diagonal())},
    index = ne_CH4_x.columns
)
ne_CH4_coef_tbl.loc['error variance', 'estimate'] = ne_CH4_rslt.scale

# add column of exponentiated coefficients
ne_CH4_coef_tbl['exponentiated']=np.exp(ne_CH4_coef_tbl['estimate'])
ne_CH4_coef_tbl['exp_visual'] = ne_CH4_coef_tbl['exponentiated'].apply(lambda x:
 ↪ "{:.2f}".format(x))

# Carbon Dioxide Analysis ***********************************************
# plot mean co2 emission vs. year
year_ne_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Northeast") &␣
 ↪(ghg_melted["Gas Type"] == "CO2TOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    #color = alt.Color("Gas Type:N")
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Northeastern CO2 Gas Emission by Year"
)

# plot mean co2 emission vs. gdp
```

```
gdp_ne_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Northeast") &␣
 ↪(ghg_melted["Gas Type"] == "CO2TOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Northeastern CO2 Gas Emission by GDP"
)

# plot mean co2 emission vs. population
pop_ne_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Northeast") &␣
 ↪(ghg_melted["Gas Type"] == "CO2TOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Northeastern CO2 gas Emission by Population"
)

# LOESS smoother
# extract northeastern co2 data
ne_CO2 = ghg_melted[(ghg_melted['Region'] == 'Northeast') & (ghg_melted['Gas␣
 ↪Type'] == 'CO2TOT')]

# fit loess smooth for gdp and population plots
gdp_ne_CO2_ls = sm.nonparametric.lowess(endog = ne_CO2["Gas Total"].values,
                                    exog = ne_CO2["GDP"].values,
                                    frac = 0.5,
                                    xvals = ne_grid_gdp)
pop_ne_CO2_ls = sm.nonparametric.lowess(endog = ne_CO2["Gas Total"].values,
                                    exog = ne_CO2["Population"].values,
                                    frac = 0.5,
                                    xvals = ne_grid_pop)

# store as dataframe
gdp_ne_CO2_df = pd.DataFrame({'GDP': ne_grid_gdp, 'Gas': gdp_ne_CO2_ls})
pop_ne_CO2_df = pd.DataFrame({'Population': ne_grid_pop, 'Gas': pop_ne_CO2_ls})

# loess smoother lines for gdp and population plots
gdp_ne_CO2_loess = alt.Chart(
    gdp_ne_CO2_df
```

```
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_ne_CO2_loess = alt.Chart(
    pop_ne_CO2_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# create dummy and x/y variables for mlr
ne_CO2_indicators = pd.get_dummies(ne_CO2[['Year', 'Population', 'GDP']],
                           drop_first = True)
ne_CO2_x = sm.tools.add_constant(ne_CO2_indicators)
ne_CO2_y = ne_CO2['Gas Total']
ne_CO2_indicators.columns.values

# fit mlr model
ne_CO2_mlr = sm.OLS(endog = ne_CO2_y, exog = ne_CO2_x.astype(float))
ne_CO2_rslt = ne_CO2_mlr.fit()

# retreive estimates and std errors
ne_CO2_coef_tbl = pd.DataFrame({
    'estimate': ne_CO2_rslt.params.values,
    'standard error': np.sqrt(ne_CO2_rslt.cov_params().values.diagonal())},
    index = ne_CO2_x.columns
)
ne_CO2_coef_tbl.loc['error variance', 'estimate'] = ne_CO2_rslt.scale

# add column of exponentiated coefficients
ne_CO2_coef_tbl['exponentiated']=np.exp(ne_CO2_coef_tbl['estimate'])
ne_CO2_coef_tbl['exp_visual'] = ne_CO2_coef_tbl['exponentiated'].apply(lambda x:
 ↪ "{:.2f}".format(x))

# Flourinated Gases Analysis ************************************************
# plot mean fgas emission vs. year
year_ne_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Northeast") &␣
 ↪(ghg_melted["Gas Type"] == "FGasTOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    #color = alt.Color("Gas Type:N")
```

```python
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Northeastern Flourinated Gas Emission by Year"
)

# plot mean fgas emission vs. gdp
gdp_ne_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Northeast") &
 ↪(ghg_melted["Gas Type"] == "FGasTOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Northeastern FLourinated Gas Emission by GDP"
)

# plot mean fgas emission vs. population
pop_ne_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Northeast") &
 ↪(ghg_melted["Gas Type"] == "FGasTOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Northeastern Flourinated Gas Emission by Population"
)

# LOESS smoother
# extract norheastern fgas data
ne_FGas = ghg_melted[(ghg_melted['Region'] == 'Northeast') & (ghg_melted['Gas
 ↪Type'] == 'FGasTOT')]

# fit loess smooth for gdp and population plots
gdp_ne_FGas_ls = sm.nonparametric.lowess(endog = ne_FGas["Gas Total"].values,
                                    exog = ne_FGas["GDP"].values,
                                    frac = 0.5,
                                    xvals = ne_grid_gdp)
pop_ne_FGas_ls = sm.nonparametric.lowess(endog = ne_FGas["Gas Total"].values,
                                    exog = ne_FGas["Population"].values,
                                    frac = 0.5,
                                    xvals = ne_grid_pop)
```

```
# store as dataframe
gdp_ne_FGas_df = pd.DataFrame({'GDP': ne_grid_gdp, 'Gas': gdp_ne_FGas_ls})
pop_ne_FGas_df = pd.DataFrame({'Population': ne_grid_pop, 'Gas':␣
↪pop_ne_FGas_ls})

# loess smoother lines for gdp and population plots
gdp_ne_FGas_loess = alt.Chart(
    gdp_ne_FGas_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_ne_FGas_loess = alt.Chart(
    pop_ne_FGas_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# regression analysis
# create dummy and x/y variables for mlr
ne_FGas_indicators = pd.get_dummies(ne_FGas[['Year', 'Population', 'GDP']],
                                    drop_first = True)
ne_FGas_x = sm.tools.add_constant(ne_FGas_indicators)
ne_FGas_y = ne_FGas['Gas Total']
ne_FGas_indicators.columns.values

# fit mlr model
ne_FGas_mlr = sm.OLS(endog = ne_FGas_y, exog = ne_FGas_x.astype(float))
ne_FGas_rslt = ne_FGas_mlr.fit()

# retreive estimates and std errors
ne_FGas_coef_tbl = pd.DataFrame({
    'estimate': ne_FGas_rslt.params.values,
    'standard error': np.sqrt(ne_FGas_rslt.cov_params().values.diagonal())},
    index = ne_FGas_x.columns
)
ne_FGas_coef_tbl.loc['error variance', 'estimate'] = ne_FGas_rslt.scale

# add column of exponentiated coefficients
ne_FGas_coef_tbl['exponentiated']=np.exp(ne_FGas_coef_tbl['estimate'])
ne_FGas_coef_tbl['exp_visual'] = ne_FGas_coef_tbl['exponentiated'].apply(lambda␣
↪x: "{:.2f}".format(x))
```

```
# Nitrous Oxide Analysis *************************************************
# plot mean n2o emission vs. year
year_ne_N2O = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Northeast") &␣
↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
↪Scale(zero = False)),
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Northeastern Nitrous Oxide Gas Emission by Year"
)

# plot mean n2o emission vs. gdp
gdp_ne_N2O = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Northeast") &␣
↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Northeastern Nitrous Oxide gas Emission by GDP"
)

# plot mean n2o emission vs. population
pop_ne_N2O = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Northeast") &␣
↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Northeastern Nitrous Oxide gas Emission by Population"
)

# LOESS smoother
# extract northeastern n2o data
ne_N2O = ghg_melted[(ghg_melted['Region'] == 'Northeast') & (ghg_melted['Gas␣
↪Type'] == 'N2OTOT')]
```

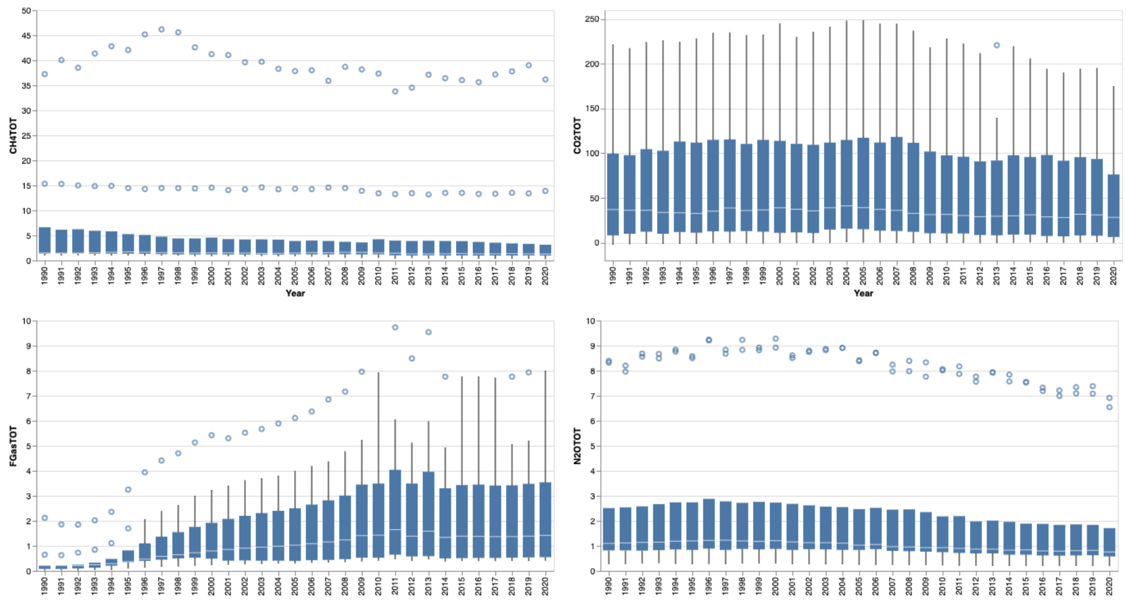
```
# fit loess smooth for gdp and population plots
gdp_ne_N2O_ls = sm.nonparametric.lowess(endog = ne_N2O["Gas Total"].values,
                                    exog = ne_N2O["GDP"].values,
                                    frac = 0.5,
                                    xvals = ne_grid_gdp)
pop_ne_N2O_ls = sm.nonparametric.lowess(endog = ne_N2O["Gas Total"].values,
                                    exog = ne_N2O["Population"].values,
                                    frac = 0.5,
                                    xvals = ne_grid_pop)

# store as dataframe
gdp_ne_N2O_df = pd.DataFrame({'GDP': ne_grid_gdp, 'Gas': gdp_ne_N2O_ls})
pop_ne_N2O_df = pd.DataFrame({'Population': ne_grid_pop, 'Gas': pop_ne_N2O_ls})

# loess smoother lines for gdp and population plots
gdp_ne_N2O_loess = alt.Chart(
    gdp_ne_N2O_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_ne_N2O_loess = alt.Chart(
    pop_ne_N2O_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# regression analysis
# create dummy and x/y variables for mlr
ne_N2O_indicators = pd.get_dummies(ne_N2O[['Year', 'Population', 'GDP']],
                           drop_first = True)
ne_N2O_x = sm.tools.add_constant(ne_N2O_indicators)
ne_N2O_y = ne_N2O['Gas Total']
ne_N2O_indicators.columns.values

# fit mlr model
ne_N2O_mlr = sm.OLS(endog = ne_N2O_y, exog = ne_N2O_x.astype(float))
ne_N2O_rslt = ne_N2O_mlr.fit()

# retreive estimates and std errors
ne_N2O_coef_tbl = pd.DataFrame({
    'estimate': ne_N2O_rslt.params.values,
```

```
    'standard error': np.sqrt(ne_N2O_rslt.cov_params().values.diagonal())},
    index = ne_N2O_x.columns
)

ne_N2O_coef_tbl.loc['error variance', 'estimate'] = ne_N2O_rslt.scale

# add column of exponentiated coefficients
ne_N2O_coef_tbl['exponentiated']=np.exp(ne_N2O_coef_tbl['estimate'])
ne_N2O_coef_tbl['exp_visual'] = ne_N2O_coef_tbl['exponentiated'].apply(lambda x:
 ↪ "{:.2f}".format(x))
```

6.3.1 Outlier Analysis

[31]:
```
(ne_CH4_outliers | ne_CO2_outliers) & (ne_FGas_outliers | ne_N2O_outliers)
```

[31]:

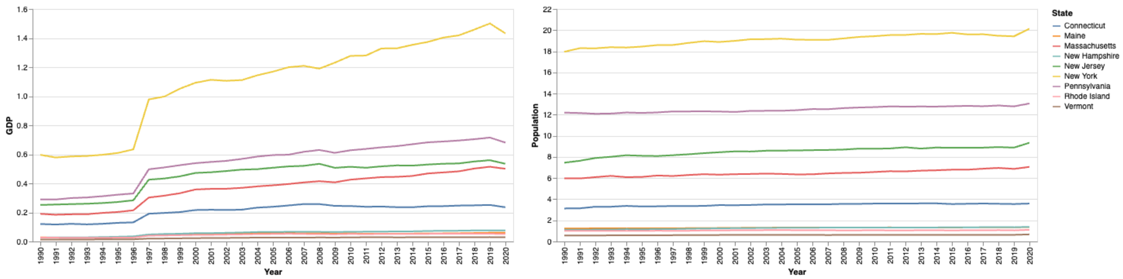


We see that there are a few notable outliers in the Northeastern region. For both Methane and Nitrous Oxide emissions, New York and Pennsylvania are consistent outliers from 1990 through 2020. For Fluorinated Gases, New York is an outlier for most of the years, while Pennsylvania was an outlier from 1990 to 1995 only. There are no outliers in the plot of Carbon Dioxide except for Pennsylvania in 2013.

6.3.2 Correlation between Time and GDP and Time and Population

[32]:
```
ne_gdp_year | ne_pop_year
```

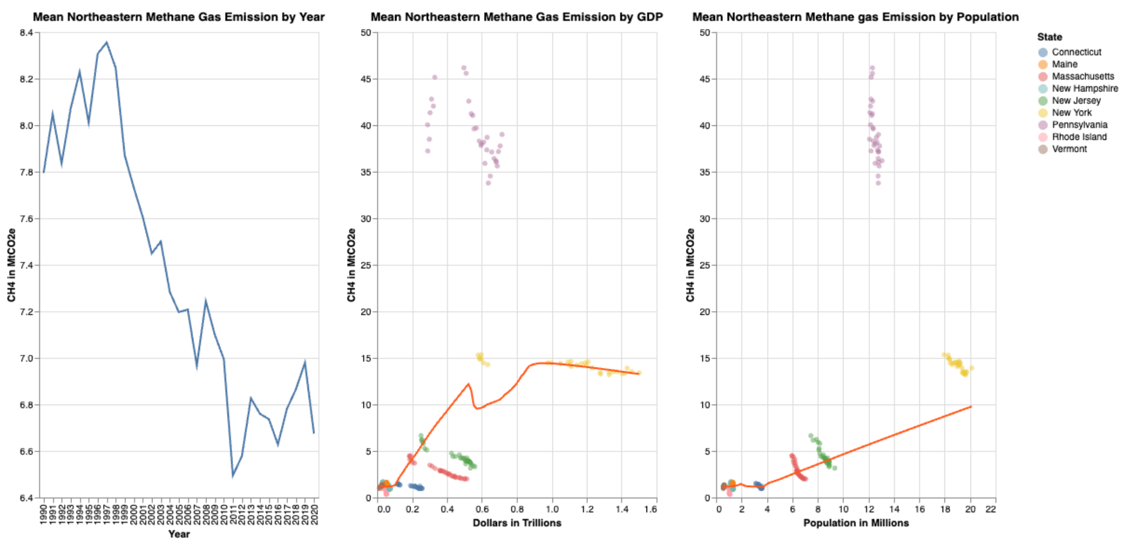
[32]:



6.3.3 Methane

[33]:
```
year_ne_CH4 | gdp_ne_CH4 + gdp_ne_CH4_loess | pop_ne_CH4 + pop_ne_CH4_loess
```

[33]:



Here, we plotted graphs of the amount of CH4 emissions by year, GDP, and population. Our first plot of emissions by year shows a clear downward trend time passes. It is worth noticing that in our plots for emissions against GDP and population, there is a very clear group of outlier points belonging to Pennsylvania. Let's take a closer look:

In the plot above of GDP against Year, we looked at the GDP of the states over time. We see that there is a positive linear trend: as the year increased, GDP also increased. We also see that there is a spike in GDP between the years of 1996 and 1997. Going back to our graph of average methane gas emissions against GDP, let's look at the points for Pennsylvania, marked in purple. We see that the mean emission peaks between a GDP of 0.6 and 1.0 (600 billion-1 trillion USD), which corresponds to in between the years of 1996 and 1997. After that, the GDP decreases significantly. This evidently matches up with our plot of the mean methane emissions over the years. After 1997, the mean methane amount steadily decreases too. We can generalize this to the other states as well: as time and gdp have a positive relationship, we see that each state (grouped by color) in the plot of CH4 emissions against GDP shows that emissions decrease as gdp/time increases, which lines up with our findings.

The right plot above of Population against Year shows each state's population over the years, and while there is a positive linear trend, it is not a drastic one. In our plot of CH4 emissions against population, we do see a positive trend: as state population increases, CH4 emissions also increases. Unlike where GDP and year had a clear positive correlation, our plots of population show that it affects the amount of CH4 emissions without much regard to year.

[34]:
```
ne_CH4_coef_tbl
```

[34]:

	estimate	standard error	exponentiated \
const	-6.355862	2.842299	1.736538e-03
Population	4.026119	0.291730	5.604298e+01
GDP	-49.573846	5.100507	2.953605e-22
Year_1991	-0.111824	3.841307	8.942017e-01
Year_1992	-0.382746	3.841306	6.819863e-01
Year_1993	-0.275069	3.841380	7.595199e-01
Year_1994	-0.003972	3.841274	9.960363e-01
Year_1995	0.053549	3.841214	1.055009e+00
Year_1996	0.544048	3.841349	1.722967e+00
Year_1997	5.173851	3.874255	1.765936e+02
Year_1998	5.233345	3.877451	1.874186e+02
Year_1999	5.291210	3.884740	1.985835e+02
Year_2000	5.826891	3.895837	3.393021e+02
Year_2001	5.839459	3.898822	3.435934e+02
Year_2002	5.619383	3.898528	2.757191e+02
Year_2003	5.803390	3.901458	3.314211e+02
Year_2004	6.033400	3.909918	4.171309e+02
Year_2005	6.300353	3.916709	5.447644e+02
Year_2006	6.607053	3.923130	7.402980e+02
Year_2007	6.613160	3.928678	7.448328e+02
Year_2008	6.800172	3.927935	8.980018e+02
Year_2009	6.359818	3.922623	5.781409e+02
Year_2010	6.682232	3.932420	7.980983e+02
Year_2011	6.114619	3.931794	4.524238e+02
Year_2012	6.596299	3.941091	7.323800e+02
Year_2013	6.894244	3.942432	9.865799e+02
Year_2014	7.008264	3.947118	1.105734e+03
Year_2015	7.302312	3.955209	1.483727e+03
Year_2016	7.512785	3.962892	1.831308e+03
Year_2017	7.818890	3.967219	2.487144e+03
Year_2018	8.355482	3.979328	4.253436e+03
Year_2019	9.057124	3.994106	8.579444e+03
Year_2020	7.044125	3.956354	1.146105e+03
error variance	66.396186	NaN	6.846953e+28

exp_visual

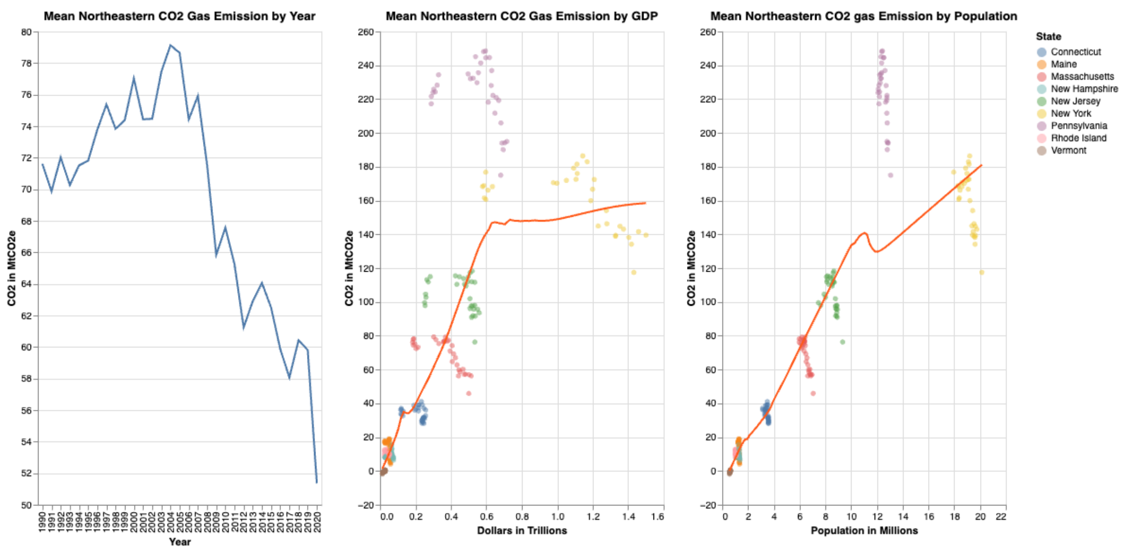
```
const                                  0.00
Population                            56.04
GDP                                    0.00
Year_1991                              0.89
Year_1992                              0.68
Year_1993                              0.76
Year_1994                              1.00
Year_1995                              1.06
Year_1996                              1.72
Year_1997                            176.59
Year_1998                            187.42
Year_1999                            198.58
Year_2000                            339.30
Year_2001                            343.59
Year_2002                            275.72
Year_2003                            331.42
Year_2004                            417.13
Year_2005                            544.76
Year_2006                            740.30
Year_2007                            744.83
Year_2008                            898.00
Year_2009                            578.14
Year_2010                            798.10
Year_2011                            452.42
Year_2012                            732.38
Year_2013                            986.58
Year_2014                           1105.73
Year_2015                           1483.73
Year_2016                           1831.31
Year_2017                           2487.14
Year_2018                           4253.44
Year_2019                           8579.44
Year_2020                           1146.11
error variance   684695300022141930264172953360.00
```

Our MLR model shows that Population is a big contributor to Methane emissions, with an exponentiated coefficient of 56.04. Furthermore, years 1997 and onward affect emissions by a lot, with coefficients 176 and up, compared to years 1990 to 1996 which do not appear to contribute to emissions.

6.3.4 Carbon Dioxide

[35]:
```
year_ne_CO2 | gdp_ne_CO2 + gdp_ne_CO2_loess | pop_ne_CO2 + pop_ne_CO2_loess
```

[35]:



The emissions against time plot shows a negative quadratic trend. In both the GDP and Population against emission plots, we see an upward trend in CO2 emission as GDP/Population increases. Along with that, we can see that within each state there seems to be a negative quadratic trend in the GDP and population plots, similar to the CO2 emissions against time plot. From the graphs, it seems that all three factors contribute to CO2 emissions.

[36]:
```
ne_CO2_coef_tbl
```

[36]:

	estimate	standard error	exponentiated	exp_visual
const	-19.039916	10.354294	5.383558e-09	0.00
Population	22.520932	1.062752	6.035545e+09	6035545145.76
GDP	-210.198705	18.580784	5.150651e-92	0.00
Year_1991	-3.588790	13.993607	2.763175e-02	0.03
Year_1992	-1.975972	13.993606	1.386265e-01	0.14
Year_1993	-4.508459	13.993876	1.101542e-02	0.01
Year_1994	-2.902534	13.993490	5.488397e-02	0.05
Year_1995	-1.413512	13.993271	2.432874e-01	0.24
Year_1996	1.149973	13.993760	3.158107e+00	3.16
Year_1997	22.094268	14.113634	3.939298e+09	3939298345.35
Year_1998	21.008844	14.125278	1.330532e+09	1330531648.64
Year_1999	23.132771	14.151830	1.112845e+10	11128452333.79
Year_2000	28.596065	14.192258	2.624904e+12	2624903873596.35
Year_2001	26.468471	14.203131	3.126878e+11	312687794518.91
Year_2002	26.036101	14.202060	2.029247e+11	202924712720.11
Year_2003	29.446509	14.212734	6.144071e+12	6144070599656.62
Year_2004	32.999658	14.243554	2.145701e+14	214570147914323.81
Year_2005	34.040892	14.268292	6.078149e+14	607814899148466.25
Year_2006	31.003143	14.291682	2.914029e+13	29140287053354.40
Year_2007	33.491778	14.311896	3.509897e+14	350989703665368.75

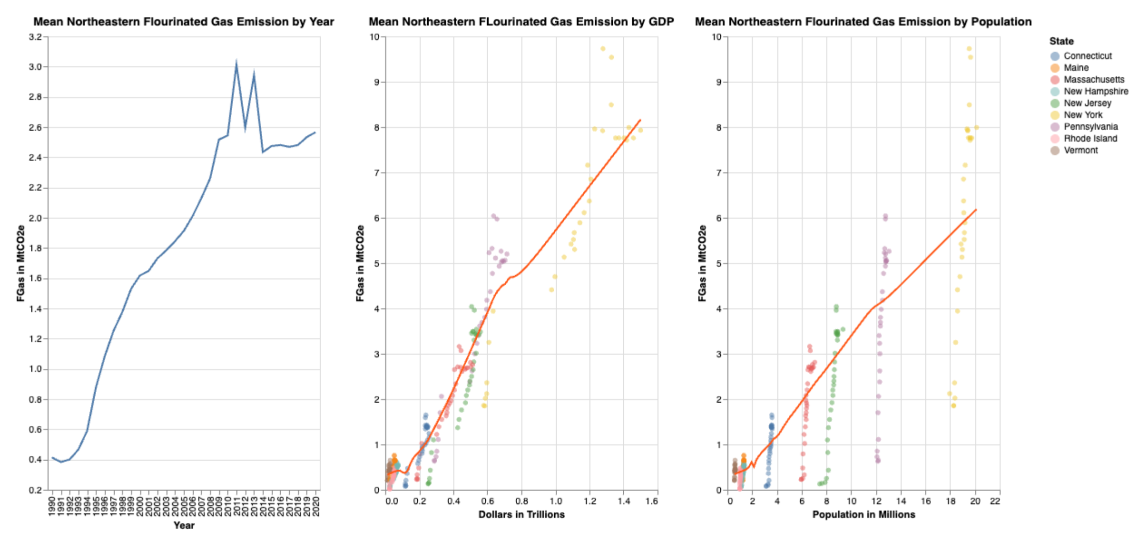
Year_2008	28.527087	14.309189	2.449947e+12	2449946866108.65
Year_2009	21.370502	14.289837	1.910254e+09	1910254225.28
Year_2010	24.783136	14.325525	5.796678e+10	57966781207.91
Year_2011	22.012184	14.323248	3.628858e+09	3628858363.65
Year_2012	19.641314	14.357115	3.389335e+08	338933489.70
Year_2013	21.478030	14.362000	2.127111e+09	2127110500.89
Year_2014	23.311143	14.379071	1.330150e+10	13301500058.23
Year_2015	23.021912	14.408544	9.960690e+09	9960689735.52
Year_2016	21.791097	14.436535	2.909065e+09	2909064521.45
Year_2017	20.563134	14.452297	8.520312e+08	852031233.68
Year_2018	24.806725	14.496410	5.935043e+10	59350433890.09
Year_2019	26.881002	14.550246	4.723574e+11	472357401592.18
Year_2020	10.069314	14.412717	2.360737e+04	23607.37
error variance	881.139214	NaN	inf	inf

Fitting a linear regression model and extracting the exponentiated coefficients shows us that between GDP and Population, Population influences CO2 emissions more, with a coefficient of 6035545145.76 while the coefficient for GDP is 0. We also see how the different years play a significant role after 1997, with coefficients in the millions.

6.3.5 Fluorinated Gases

[37]:
```
year_ne_FGas | gdp_ne_FGas + gdp_ne_FGas_loess | pop_ne_FGas + pop_ne_FGas_loess
```

[37]:



Our plot of Fluorinated Gases against time shows a clear upward trend. Our plot for Fluorinated Gases against GDP does so as well with a very clear linear pattern with points around the LOESS smoothing line. Our plot for Fluorinated Gases against Population also shows an upward trend, with the highest emission values increasing as population increases, however at each population level there is a lot of variance in emissions. Our color coded graph clearly shows that this is because each state's population does not have a drastic increase over the years, yet the FGas emissions do. Therefore, for Fluorinated Gases emissions in the Northeastern region, Time and GDP are the biggest contributers.

[38]:
```
ne_FGas_coef_tbl
```

[38]:

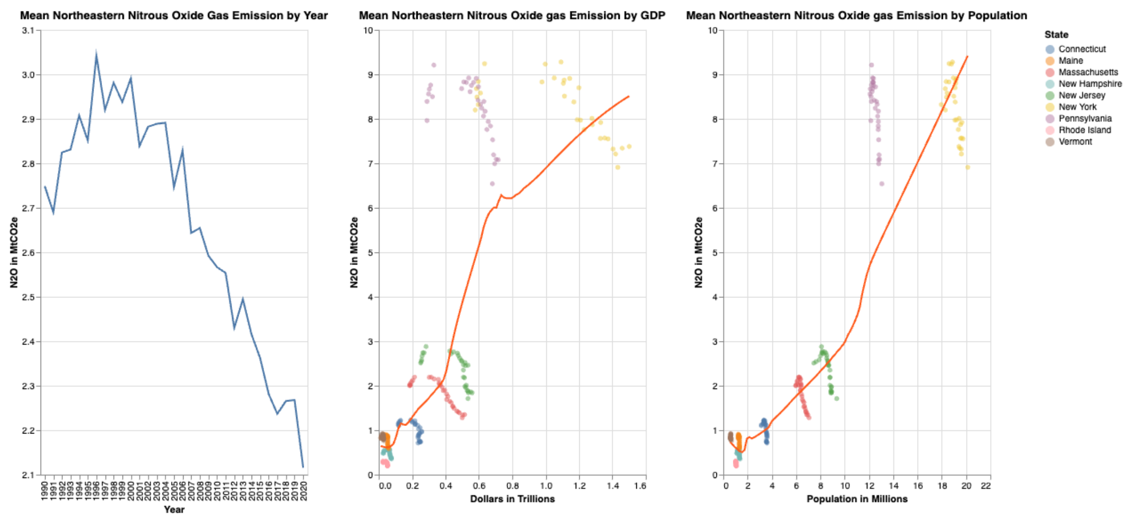
	estimate	standard error	exponentiated	exp_visual
const	-0.492804	0.167819	0.610911	0.61
Population	-0.016721	0.017225	0.983418	0.98
GDP	5.834761	0.301151	341.982997	341.98
Year_1991	-0.012101	0.226804	0.987972	0.99
Year_1992	-0.012741	0.226804	0.987339	0.99
Year_1993	0.048736	0.226808	1.049943	1.05
Year_1994	0.149674	0.226802	1.161455	1.16
Year_1995	0.406778	0.226798	1.501971	1.50
Year_1996	0.570216	0.226806	1.768650	1.77
Year_1997	0.191889	0.228749	1.211536	1.21
Year_1998	0.274604	0.228938	1.316009	1.32
Year_1999	0.355446	0.229368	1.426817	1.43
Year_2000	0.361899	0.230023	1.436053	1.44
Year_2001	0.364404	0.230200	1.439655	1.44
Year_2002	0.439248	0.230182	1.551540	1.55
Year_2003	0.464944	0.230355	1.591925	1.59
Year_2004	0.470618	0.230855	1.600983	1.60
Year_2005	0.499446	0.231256	1.647809	1.65
Year_2006	0.556379	0.231635	1.744344	1.74
Year_2007	0.641357	0.231962	1.899057	1.90
Year_2008	0.764185	0.231919	2.147243	2.15
Year_2009	1.040422	0.231605	2.830410	2.83
Year_2010	1.007560	0.232183	2.738911	2.74
Year_2011	1.471094	0.232146	4.353995	4.35
Year_2012	1.001126	0.232695	2.721346	2.72
Year_2013	1.341151	0.232774	3.823442	3.82
Year_2014	0.803200	0.233051	2.232674	2.23
Year_2015	0.799463	0.233529	2.224347	2.22
Year_2016	0.772135	0.233983	2.164381	2.16
Year_2017	0.736138	0.234238	2.087857	2.09
Year_2018	0.692141	0.234953	1.997989	2.00
Year_2019	0.692312	0.235826	1.998330	2.00
Year_2020	0.831478	0.233596	2.296711	2.30
error variance	0.231465	NaN	1.260445	1.26

Our multiple linear regression analysis model supports our findings from the graphs, where wesaw that GDP and year contribute the most to Fluorinated Gases emissions. GDP especially has a high exponentiated coefficient value of 341.98, and coefficients of each year range from 0.99 to 4.35. Meanwhile, The coefficient for Population is 0.98, suggesting minimal contribution.

6.3.6 Nitrous Oxide

[39]:
```
year_ne_N20 | gdp_ne_N20 + gdp_ne_N20_loess | pop_ne_N20 + pop_ne_N20_loess
```

[39]:



Similar to our graphs for CO2, we see that there is a downward quadratic trend in Nitrous Oxide emissions over the years, as well as a positive linear trend in N2O emissions against both GDP and population. Again, since time and GDP in the northeast region have a positive linear correlation, we can see the same quadratic trend within each individual state in the emissions against GDP plot. From the plots it seems that all three variables contribute to NItrous Oxide emissions.

[40]:
```
ne_N20_coef_tbl
```

[40]:

	estimate	standard error	exponentiated	exp_visual
const	-0.787580	0.351882	0.454944	0.45
Population	0.826232	0.036117	2.284694	2.28
GDP	-6.483265	0.631452	0.001529	0.00
Year_1991	-0.122855	0.475561	0.884392	0.88
Year_1992	-0.012194	0.475561	0.987881	0.99
Year_1993	-0.034792	0.475570	0.965807	0.97
Year_1994	0.050028	0.475557	1.051301	1.05
Year_1995	0.030499	0.475550	1.030969	1.03
Year_1996	0.235263	0.475566	1.265242	1.27
Year_1997	0.706856	0.479640	2.027606	2.03
Year_1998	0.775854	0.480036	2.172447	2.17
Year_1999	0.775123	0.480938	2.170859	2.17
Year_2000	0.914452	0.482312	2.495408	2.50
Year_2001	0.773157	0.482682	2.166596	2.17
Year_2002	0.797740	0.482645	2.220517	2.22
Year_2003	0.814651	0.483008	2.258388	2.26
Year_2004	0.873412	0.484055	2.395069	2.40

```
Year_2005        0.776965        0.484896       2.174862        2.17
Year_2006        0.893875        0.485691       2.444583        2.44
Year_2007        0.737495        0.486378       2.090692        2.09
Year_2008        0.724533        0.486286       2.063767        2.06
Year_2009        0.613609        0.485628       1.847085        1.85
Year_2010        0.637061        0.486841       1.890915        1.89
Year_2011        0.606157        0.486764       1.833372        1.83
Year_2012        0.531442        0.487915       1.701383        1.70
Year_2013        0.600276        0.488081       1.822621        1.82
Year_2014        0.540635        0.488661       1.717096        1.72
Year_2015        0.524625        0.489662       1.689825        1.69
Year_2016        0.487635        0.490614       1.628461        1.63
Year_2017        0.459372        0.491149       1.583080        1.58
Year_2018        0.545425        0.492648       1.725342        1.73
Year_2019        0.635625        0.494478       1.888203        1.89
Year_2020        0.198467        0.489804       1.219531        1.22
error variance   1.017649             NaN       2.766681        2.77
```

From our multiple linear regression model, our exponentiated coefficients show that Population contributes to Nitrous Oxide emissions. Time also does, with the coefficients taking on values ranging from 0.88 to 2.44. The coefficient for GDP is very close to zero, so the trend we saw in our plot of Nitrous Oxide emissions against GDP may be because of time's influence, as time and GDP have a positive relationship.

6.4 Southeast

[41]:

```
# extract southeast region data
ghg_se = ghg[ghg["Region"] == "Southeast"]

# Outlier Analysis ************************************************
# boxplot showing distributions of ch4 total emissions among states for each␣
↪year
se_CH4_outliers = alt.Chart(ghg_se).encode(
    x = alt.X("Year:O"),
    y = alt.Y("CH4TOT"),
    tooltip = ("State", "CH4TOT")
).mark_boxplot()

# boxplot showing distributions of co2 total emissions among states for each␣
↪year
se_CO2_outliers = alt.Chart(ghg_se).encode(
    x = alt.X("Year:O"),
    y = alt.Y("CO2TOT"),
    tooltip = ("State", "CO2TOT")
).mark_boxplot()
```

```
# boxplot showing distributions of fgas total emissions among states for each␣
↪year
se_FGas_outliers = alt.Chart(ghg_se).encode(
    x = alt.X("Year:O"),
    y = alt.Y("FGasTOT"),
    tooltip = ("State", "FGasTOT")
).mark_boxplot()

# boxplot showing distributions of n2o total emissions among states for each␣
↪year
se_N2O_outliers = alt.Chart(ghg_se).encode(
    x = alt.X("Year:O"),
    y = alt.Y("N2OTOT"),
    tooltip = ("State", "N2OTOT")
).mark_boxplot()


# Methane Analysis ************************************************
# plot mean ch4 emission vs. year
year_se_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southeast") &␣
↪(ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.
↪Scale(zero = False)),
    #color = alt.Color("Gas Type:N")
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Southeastern Methane Gas Emission by Year"
)

# plot mean ch4 emission vs. gdp
gdp_se_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southeast") &␣
↪(ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southeastern Methane Gas Emission by GDP"
)

# plot mean ch4 emission vs. population
```

```
pop_se_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southeast") & (ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southeastern Methane gas Emission by Population"
)

# LOESS smoother
# extract southeastern ch4 data
se_CH4 = ghg_melted[(ghg_melted['Region'] == 'Southeast') & (ghg_melted['Gas Type'] == 'CH4TOT')]

# grid of gdp and population values
se_grid_gdp = np.linspace(se_CH4["GDP"].min(), se_CH4["GDP"].max(), num = 100)
se_grid_pop = np.linspace(se_CH4["Population"].min(), se_CH4["Population"].max(), num = 100)

# fit loess smooth for gdp and population plots
gdp_se_CH4_ls = sm.nonparametric.lowess(endog = se_CH4["Gas Total"].values,
                                    exog = se_CH4["GDP"].values,
                                    frac = 0.5,
                                    xvals = se_grid_gdp)
pop_se_CH4_ls = sm.nonparametric.lowess(endog = se_CH4["Gas Total"].values,
                                    exog = se_CH4["Population"].values,
                                    frac = 0.5,
                                    xvals = se_grid_pop)

# store as dataframe
gdp_se_CH4_df = pd.DataFrame({'GDP': se_grid_gdp, 'Gas': gdp_se_CH4_ls})
pop_se_CH4_df = pd.DataFrame({'Population': se_grid_pop, 'Gas': pop_se_CH4_ls})


# loess smoother lines for gdp and population plots
gdp_se_CH4_loess = alt.Chart(
    gdp_se_CH4_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_se_CH4_loess = alt.Chart(
```

```
    pop_se_CH4_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# regression analysis
# create dummy and x/y variables for mlr
se_CH4_indicators = pd.get_dummies(se_CH4[['Year', 'Population', 'GDP']],
                           drop_first = True)
se_CH4_x = sm.tools.add_constant(se_CH4_indicators)
se_CH4_y = se_CH4['Gas Total']
se_CH4_indicators.columns.values

# fit mlr model
se_CH4_mlr = sm.OLS(endog = se_CH4_y, exog = se_CH4_x.astype(float))
se_CH4_rslt = se_CH4_mlr.fit()

# retreive estimates and std errors
se_CH4_coef_tbl = pd.DataFrame({
    'estimate': se_CH4_rslt.params.values,
    'standard error': np.sqrt(se_CH4_rslt.cov_params().values.diagonal())},
    index = se_CH4_x.columns
)
se_CH4_coef_tbl.loc['error variance', 'estimate'] = se_CH4_rslt.scale

# add column of exponentiated coefficients
se_CH4_coef_tbl['exponentiated']=np.exp(se_CH4_coef_tbl['estimate'])
se_CH4_coef_tbl['exp_visual'] = se_CH4_coef_tbl['exponentiated'].apply(lambda x:
 ↪ "{:.2f}".format(x))


# Carbon Dioxide Analysis ************************************************
# plot mean co2 emission vs. year
year_se_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southeast") &␣
 ↪(ghg_melted["Gas Type"] == "CO2TOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    #color = alt.Color("Gas Type:N")
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Southeastern CO2 Gas Emission by Year"
)
```

```
# plot mean co2 emission vs. gdp
gdp_se_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southeast") &␣
 ↪(ghg_melted["Gas Type"] == "CO2TOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southeastern CO2 Gas Emission by GDP"
)
# plot mean co2 emission vs. population
pop_se_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southeast") &␣
 ↪(ghg_melted["Gas Type"] == "CO2TOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southeastern CO2 gas Emission by Population"
)

# LOESS smoother

# extract southeastern co2 data
se_CO2 = ghg_melted[(ghg_melted['Region'] == 'Southeast') & (ghg_melted['Gas␣
 ↪Type'] == 'CO2TOT')]

# fit loess smooth for gdp and population plots
gdp_se_CO2_ls = sm.nonparametric.lowess(endog = se_CO2["Gas Total"].values,
                                    exog = se_CO2["GDP"].values,
                                    frac = 0.5,
                                    xvals = se_grid_gdp)
pop_se_CO2_ls = sm.nonparametric.lowess(endog = se_CO2["Gas Total"].values,
                                    exog = se_CO2["Population"].values,
                                    frac = 0.5,
                                    xvals = se_grid_pop)

# store as dataframe
gdp_se_CO2_df = pd.DataFrame({'GDP': se_grid_gdp, 'Gas': gdp_se_CO2_ls})
pop_se_CO2_df = pd.DataFrame({'Population': se_grid_pop, 'Gas': pop_se_CO2_ls})

# loess smoother lines for gdp and population plots
gdp_se_CO2_loess = alt.Chart(
```

```
    gdp_se_CO2_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_se_CO2_loess = alt.Chart(
    pop_se_CO2_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# create dummy and x/y variables for mlr
se_CO2_indicators = pd.get_dummies(se_CO2[['Year', 'Population', 'GDP']],
                            drop_first = True)
se_CO2_x = sm.tools.add_constant(se_CO2_indicators)
se_CO2_y = se_CO2['Gas Total']
se_CO2_indicators.columns.values

# fit mlr model
se_CO2_mlr = sm.OLS(endog = se_CO2_y, exog = se_CO2_x.astype(float))
se_CO2_rslt = se_CO2_mlr.fit()

# retreive estimates and std errors
se_CO2_coef_tbl = pd.DataFrame({
    'estimate': se_CO2_rslt.params.values,
    'standard error': np.sqrt(se_CO2_rslt.cov_params().values.diagonal())},
    index = se_CO2_x.columns
)
se_CO2_coef_tbl.loc['error variance', 'estimate'] = se_CO2_rslt.scale

# add column of exponentiated coefficients
se_CO2_coef_tbl['exponentiated']=np.exp(se_CO2_coef_tbl['estimate'])
se_CO2_coef_tbl['exp_visual'] = se_CO2_coef_tbl['exponentiated'].apply(lambda x:
 ↪ "{:.2f}".format(x))


# Flourinated Gases Analysis ************************************************
# plot mean fgas emission vs. year
year_se_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southeast") &␣
 ↪(ghg_melted["Gas Type"] == "FGasTOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
```

```
    y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.Scale(zero = False)),
    #color = alt.Color("Gas Type:N")
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Southeastern Flourinated Gas Emission by Year"
)
# plot mean fgas emission vs. gdp
gdp_se_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southeast") & (ghg_melted["Gas Type"] == "FGasTOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southeastern Flourinated Gas Emission by GDP"
)
# plot mean fgas emission vs. population
pop_se_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southeast") & (ghg_melted["Gas Type"] == "FGasTOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southeastern Flourinated Gas Emission by Population"
)

# LOESS smoother
# extract southeastern fgas data
se_FGas = ghg_melted[(ghg_melted['Region'] == 'Southeast') & (ghg_melted['Gas Type'] == 'FGasTOT')]

# fit loess smooth for gdp and population plots
gdp_se_FGas_ls = sm.nonparametric.lowess(endog = se_FGas["Gas Total"].values,
                                    exog = se_FGas["GDP"].values,
                                    frac = 0.5,
                                    xvals = se_grid_gdp)
pop_se_FGas_ls = sm.nonparametric.lowess(endog = se_FGas["Gas Total"].values,
                                    exog = se_FGas["Population"].values,
                                    frac = 0.5,
```

```
                                     xvals = se_grid_pop)

# store as dataframe
gdp_se_FGas_df = pd.DataFrame({'GDP': se_grid_gdp, 'Gas': gdp_se_FGas_ls})
pop_se_FGas_df = pd.DataFrame({'Population': se_grid_pop, 'Gas':␣
 ↪pop_se_FGas_ls})


# loess smoother lines for gdp and population plots
gdp_se_FGas_loess = alt.Chart(
    gdp_se_FGas_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_se_FGas_loess = alt.Chart(
    pop_se_FGas_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# create dummy and x/y variables for mlr
se_FGas_indicators = pd.get_dummies(se_FGas[['Year', 'Population', 'GDP']],
                            drop_first = True)
se_FGas_x = sm.tools.add_constant(se_FGas_indicators)
se_FGas_y = se_FGas['Gas Total']
se_FGas_indicators.columns.values

# fit mlr model
se_FGas_mlr = sm.OLS(endog = se_FGas_y, exog = se_FGas_x.astype(float))
se_FGas_rslt = se_FGas_mlr.fit()

# retreive estimates and std errors
se_FGas_coef_tbl = pd.DataFrame({
    'estimate': se_FGas_rslt.params.values,
    'standard error': np.sqrt(se_FGas_rslt.cov_params().values.diagonal())},
    index = se_FGas_x.columns
)
se_FGas_coef_tbl.loc['error variance', 'estimate'] = se_FGas_rslt.scale

# add column of exponentiated coefficients
se_FGas_coef_tbl['exponentiated']=np.exp(se_FGas_coef_tbl['estimate'])
```

```
se_FGas_coef_tbl['exp_visual'] = se_FGas_coef_tbl['exponentiated'].apply(lambda␣
↪x: "{:.2f}".format(x))



# Nitrous Oxide Analysis ************************************************
# plot mean n2o emission vs. year
year_se_N2O = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southeast") &␣
↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
↪Scale(zero = False)),
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Southeastern Nitrous Oxide Gas Emission by Year"
)
# plot mean n2o emission vs. gdp
gdp_se_N2O = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southeast") &␣
↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southeastern Nitrous Oxide gas Emission by GDP"
)
# plot mean n2o emission vs. population
pop_se_N2O = alt.Chart(ghg_melted[(ghg_melted["Region"] == "Southeast") &␣
↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Southeastern Nitrous Oxide gas Emission by Population"
)

# LOESS smoother
# extract southeastern n2o data
se_N2O = ghg_melted[(ghg_melted['Region'] == 'Southeast') & (ghg_melted['Gas␣
↪Type'] == 'N2OTOT')]
```

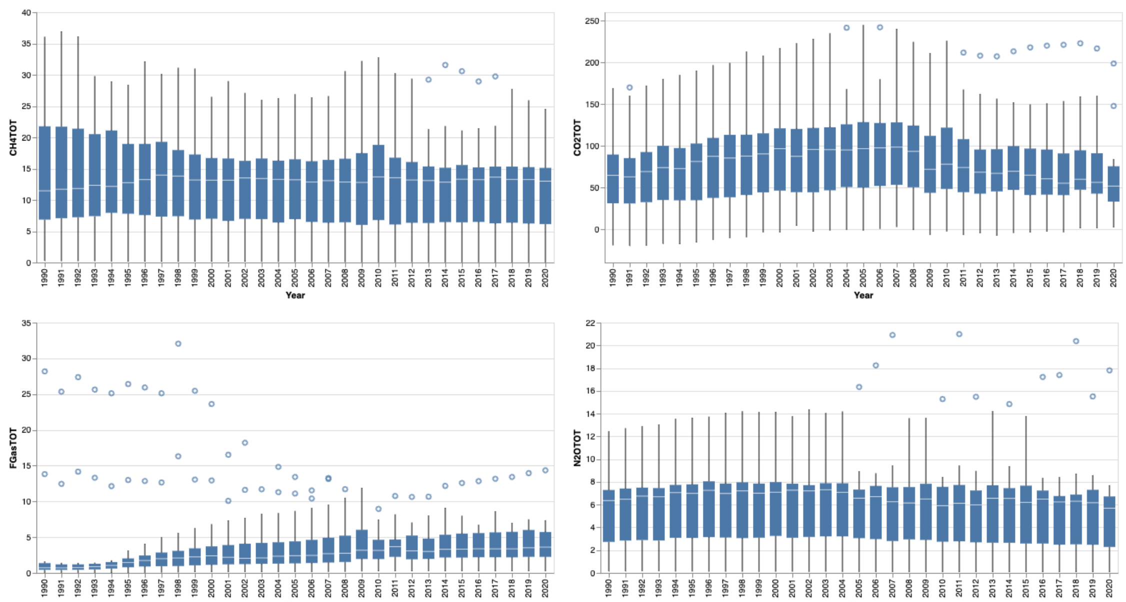
```python
# fit loess smooth for gdp and population plots
gdp_se_N20_ls = sm.nonparametric.lowess(endog = se_N20["Gas Total"].values,
                                        exog = se_N20["GDP"].values,
                                        frac = 0.5,
                                        xvals = se_grid_gdp)
pop_se_N20_ls = sm.nonparametric.lowess(endog = se_N20["Gas Total"].values,
                                        exog = se_N20["Population"].values,
                                        frac = 0.5,
                                        xvals = se_grid_pop)

# store as dataframe
gdp_se_N20_df = pd.DataFrame({'GDP': se_grid_gdp, 'Gas': gdp_se_N20_ls})
pop_se_N20_df = pd.DataFrame({'Population': se_grid_pop, 'Gas': pop_se_N20_ls})


# loess smoother lines for gdp and population plots
gdp_se_N20_loess = alt.Chart(
    gdp_se_N20_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_se_N20_loess = alt.Chart(
    pop_se_N20_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# create dummy and x/y variables for mlr
se_N20_indicators = pd.get_dummies(se_N20[['Year', 'Population', 'GDP']],
                                   drop_first = True)
se_N20_x = sm.tools.add_constant(se_N20_indicators)
se_N20_y = se_N20['Gas Total']
se_N20_indicators.columns.values

# fit mlr model
se_N20_mlr = sm.OLS(endog = se_N20_y, exog = se_N20_x.astype(float))
se_N20_rslt = se_N20_mlr.fit()

# retreive estimates and std errors
se_N20_coef_tbl = pd.DataFrame({
```

```
    'estimate': se_N2O_rslt.params.values,
    'standard error': np.sqrt(se_N2O_rslt.cov_params().values.diagonal())},
    index = se_N2O_x.columns
)
se_N2O_coef_tbl.loc['error variance', 'estimate'] = se_N2O_rslt.scale

# add column of exponentiated coefficients
se_N2O_coef_tbl['exponentiated']=np.exp(se_N2O_coef_tbl['estimate'])
se_N2O_coef_tbl['exp_visual'] = se_N2O_coef_tbl['exponentiated'].apply(lambda x:
 ↪ "{:.2f}".format(x))
```

6.4.1 Outlier Analysis

[42]:
```
(se_CH4_outliers | se_CO2_outliers) & (se_FGas_outliers | se_N2O_outliers)
```

[42]:

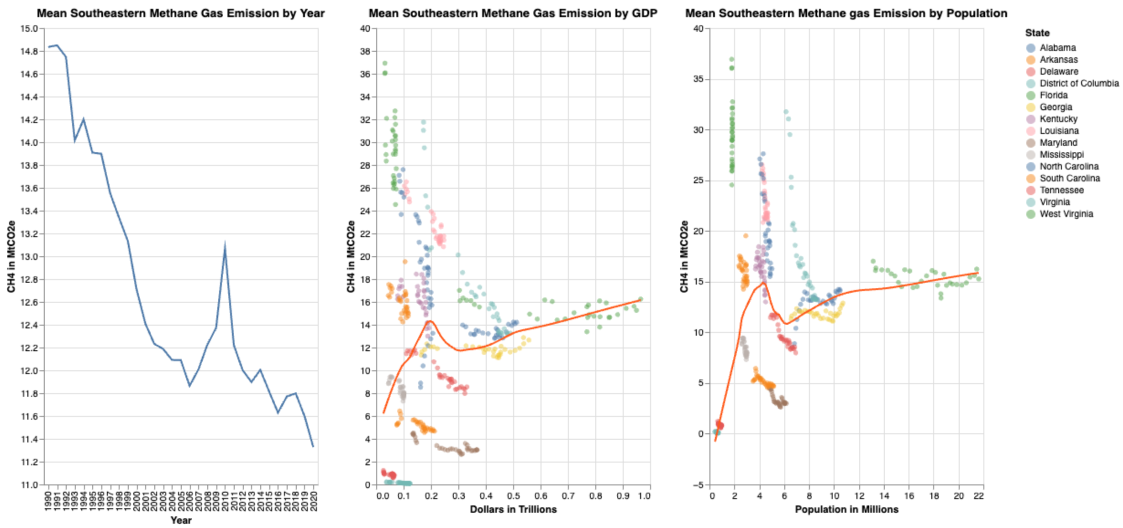


It seems that there are not any consistent outliers for any GHG in the Southeastern region. For Methane, West Virginia is a slight outlier for a 5 year stretch. For Carbon Dioxide, Florida is an outlier inconsistently until 2011 where it is a consistent outlier through 2020 and Louisiana joins as an outlier in 2020 too. For Flourinated Gases, Louisiana and Kentucky are consistent outliers (with Kentucky having significantly higher emissions) until 2008 where it switches and Florida becomes an outlier until 2020. For Nitrous Oxide, Florida is sporadically an outlier.

6.4.2 Methane

[43]:
```
year_se_CH4 | gdp_se_CH4 + gdp_se_CH4_loess | pop_se_CH4 + pop_se_CH4_loess
```

[43]:



For both the GDP and Population graphs it seems that the data is very stacked so even though there is a slight positive linear trend, it doesn't seem to follow the data that well. It seems that even with relatively similar GDPs, different states have wildly varying CH4 emissions. However, if we look at the graph that plots mean emisisons by year, we can see that over time the emissions have done down almost steadily with a spike in 2010 as the only exception. Here we can conclude that time has been the most influential factor in emissions instead of GDP and Population. Unfortunately, with our dataset we are limited in what we can conclude past this. It seems though that there were external factors influencing CH4 emissions instead of a correlation with GDP or Population.

[44]:
```
se_CH4_coef_tbl
```

[44]:

	estimate	standard error	exponentiated	\
const	9.904266	2.231389	2.001557e+04	
Population	2.308494	0.362015	1.005927e+01	
GDP	-48.851815	8.637237	6.080320e-22	
Year_1991	-0.101336	2.978563	9.036297e-01	
Year_1992	-0.129762	2.978539	8.783043e-01	
Year_1993	-0.901998	2.978578	4.057581e-01	
Year_1994	-0.575401	2.978761	5.624794e-01	
Year_1995	-0.816459	2.978974	4.419940e-01	
Year_1996	-0.745637	2.979351	4.744319e-01	
Year_1997	1.880356	3.036331	6.555839e+00	
Year_1998	1.950792	3.047428	7.034255e+00	
Year_1999	2.065672	3.060744	7.890595e+00	
Year_2000	1.710681	3.064952	5.532730e+00	
Year_2001	1.468322	3.068209	4.341943e+00	
Year_2002	1.415678	3.074545	4.119280e+00	
Year_2003	1.624978	3.086850	5.078305e+00	
Year_2004	1.881331	3.105241	6.562236e+00	
Year_2005	2.235377	3.125108	9.350002e+00	

Year_2006	2.174975	3.135220	8.801964e+00
Year_2007	2.193252	3.129962	8.964319e+00
Year_2008	2.195836	3.120164	8.987508e+00
Year_2009	1.844081	3.095919	6.322285e+00
Year_2010	2.660487	3.102914	1.430325e+01
Year_2011	1.803807	3.103472	6.072720e+00
Year_2012	1.574557	3.104125	4.828602e+00
Year_2013	1.507660	3.106795	4.516148e+00
Year_2014	1.726938	3.113622	5.623410e+00
Year_2015	1.748149	3.125839	5.743963e+00
Year_2016	1.707051	3.134730	5.512679e+00
Year_2017	2.040171	3.146475	7.691921e+00
Year_2018	2.235620	3.157332	9.352279e+00
Year_2019	2.241916	3.170355	9.411344e+00
Year_2020	1.515526	3.145024	4.551814e+00
error variance	66.536434	NaN	7.877822e+28

	exp_visual
const	20015.57
Population	10.06
GDP	0.00
Year_1991	0.90
Year_1992	0.88
Year_1993	0.41
Year_1994	0.56
Year_1995	0.44
Year_1996	0.47
Year_1997	6.56
Year_1998	7.03
Year_1999	7.89
Year_2000	5.53
Year_2001	4.34
Year_2002	4.12
Year_2003	5.08
Year_2004	6.56
Year_2005	9.35
Year_2006	8.80
Year_2007	8.96
Year_2008	8.99
Year_2009	6.32
Year_2010	14.30
Year_2011	6.07
Year_2012	4.83
Year_2013	4.52
Year_2014	5.62
Year_2015	5.74
Year_2016	5.51

```
Year_2017                                7.69
Year_2018                                9.35
Year_2019                                9.41
Year_2020                                4.55
error variance  787782172786778263293075128 32.00
```

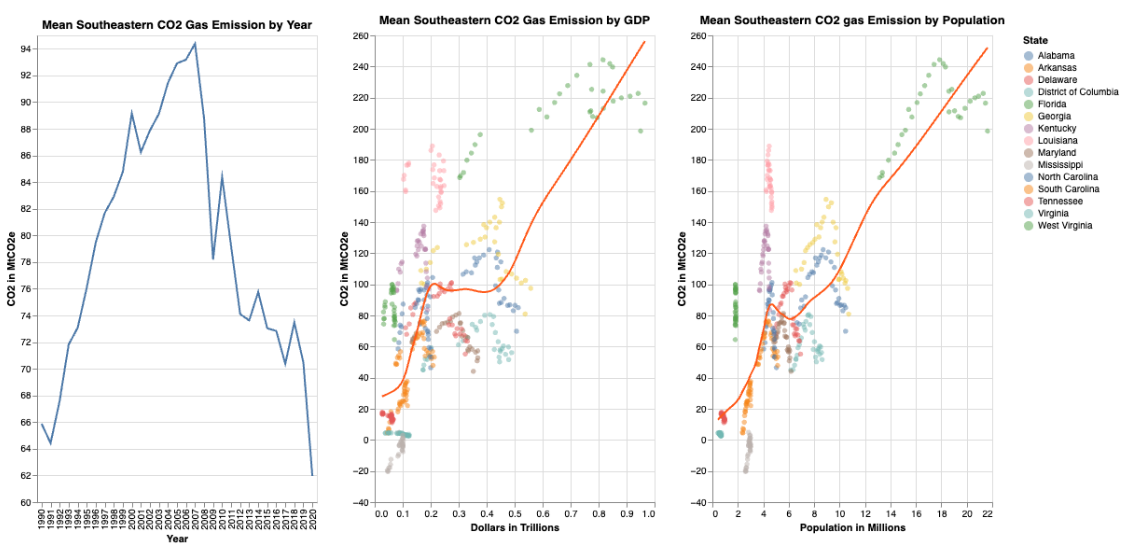
From fitting the multiple linear regression and extracting the exponentiated coefficients, we can again examine how each factor affects CH4 emissions, and here we have confirmation with the coeffients of GDP being almost 0 and Population having a fairly low effect. However, over the years there is a very wide range of coefficients from about 0.5-14.5 which demonstrates that different years have had significant affects on Ch4 emissions. With this data we could look into specific years that have significant influence over the mlr fit and investigate what caused those changes in CH4 emissions.

Time is the most significant factor in Methane emissions for the Southeast.

6.4.3 Carbon Dioxide

[45]:
```
year_se_CO2 | gdp_se_CO2 + gdp_se_CO2_loess | pop_se_CO2 + pop_se_CO2_loess
```

[45]:



Once again for Carbon Dioxide, we have a fairly stacked graph for GDP where there are wide ranges of CO2 emissions for the same level of GDP which means that GDP is not as big a factor here. The population graph has a fairly consistent positive trend that more closely follows the pattern so that has some effect. However, there is a big variation in CO2 emissions over time, with emissions going from 64 MtCO2e in 1990 to 95 MtCO2e in 2008 and back down to 62 MtCO2e in 2020. This means that something external significantly caused CO2 emissions to rise in 1990 and then severly decrease starting in 2008.

[46]:
```
se_CO2_coef_tbl
```

[46]:

	estimate	standard error	exponentiated	exp_visual
const	10.657922	10.882649	4.252817e+04	42528.17
Population	15.781981	1.765573	7.145408e+06	7145408.27
GDP	-130.074963	42.124436	3.229692e-57	0.00
Year_1991	-2.422758	14.526670	8.867668e-02	0.09
Year_1992	0.215009	14.526553	1.239873e+00	1.24
Year_1993	3.627146	14.526741	3.760536e+01	37.61
Year_1994	4.496733	14.527634	8.972351e+01	89.72
Year_1995	6.963095	14.528674	1.056899e+03	1056.90
Year_1996	9.939272	14.530514	2.072865e+04	20728.65
Year_1997	19.354125	14.808407	2.543253e+08	254325279.16
Year_1998	20.679527	14.862528	9.572041e+08	957204144.97
Year_1999	22.801264	14.927472	7.988464e+09	7988463813.74
Year_2000	26.587362	14.947994	3.521639e+11	352163853280.88
Year_2001	23.330353	14.963879	1.355950e+10	13559496170.16
Year_2002	24.668751	14.994780	5.170142e+10	51701420855.99
Year_2003	26.068418	15.054794	2.095898e+11	209589790910.82
Year_2004	28.563487	15.144489	2.540767e+12	2540766676582.72
Year_2005	30.217360	15.241382	1.328106e+13	13281063937428.64
Year_2006	30.359720	15.290699	1.531295e+13	15312950781239.66
Year_2007	30.517836	15.265052	1.793610e+13	17936096845010.26
Year_2008	23.660460	15.217266	1.886284e+10	18862836627.48
Year_2009	11.303200	15.099021	8.108068e+04	81080.68
Year_2010	17.208151	15.133138	2.974437e+07	29744374.64
Year_2011	11.515713	15.135861	1.002792e+05	100279.16
Year_2012	5.893483	15.139045	3.626661e+02	362.67
Year_2013	5.130176	15.152067	1.690468e+02	169.05
Year_2014	6.999789	15.185360	1.096402e+03	1096.40
Year_2015	4.294197	15.244946	7.327334e+01	73.27
Year_2016	3.958107	15.288306	5.235814e+01	52.36
Year_2017	1.525773	15.345588	4.598698e+00	4.60
Year_2018	4.572004	15.398539	9.673776e+01	96.74
Year_2019	1.694646	15.462055	5.444716e+00	5.44
Year_2020	-8.381347	15.338514	2.291012e-04	0.00
error variance	1582.625110	NaN	inf	inf

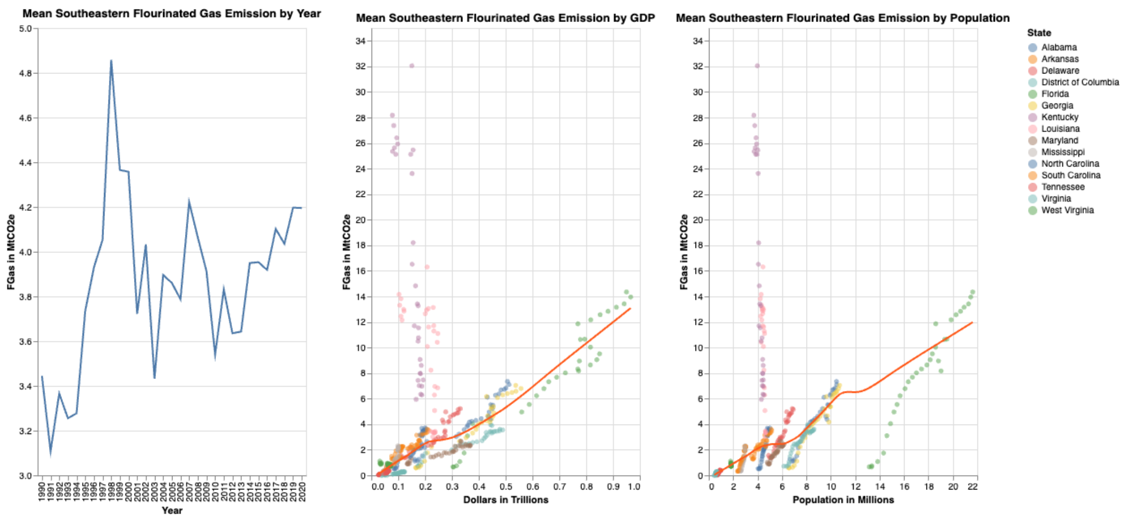
From fitting the multiple linear regression and extracting the exponentiated coefficients, we can again examine how each factor affects CO2 emissions, and we have exact confirmation of what we saw in the graphs. For GDP, the coefficient is close to 0 which shows that GDP had very little effect on emissions. For Population, there is a fairly significant effect with a coefficient of 7145408 which should be noted as we saw in the graphs. However, the coefficients of the years vary massively, going from almost 0 to 13 digits. This means that time was a very big factor for this GHG in the Southeast.

Time is the most significant factor in Carbon Dioxide emissions for the Southeast.

6.4.4 Flourinated Gas

[47]:
```
year_se_FGas | gdp_se_FGas + gdp_se_FGas_loess | pop_se_FGas + pop_se_FGas_loess
```

[47]:



For Flourinated Gases though we have a different story, the mean emissions do not change much or consistently over the time period. The Population graph does show a fairly consistent positive linear trend, but there is still some stacking for each level of Population. The most significant graph here is the graph of GDP. With the exception of a few states, there is a very strong correlation between higher GDP and higher FGas emissions.

[48]:
```
se_FGas_coef_tbl
```

[48]:

	estimate	standard error	exponentiated	exp_visual
const	1.763331	1.316915	5.831831e+00	5.83
Population	0.352067	0.213653	1.422004e+00	1.42
GDP	1.361811	5.097499	3.903254e+00	3.90
Year_1991	-0.362647	1.757880	6.958317e-01	0.70
Year_1992	-0.137236	1.757865	8.717644e-01	0.87
Year_1993	-0.277872	1.757888	7.573937e-01	0.76
Year_1994	-0.292910	1.757996	7.460892e-01	0.75
Year_1995	0.135168	1.758122	1.144729e+00	1.14
Year_1996	0.298759	1.758345	1.348185e+00	1.35
Year_1997	0.309169	1.791973	1.362293e+00	1.36
Year_1998	1.078072	1.798522	2.939007e+00	2.94
Year_1999	0.551124	1.806381	1.735202e+00	1.74
Year_2000	0.506720	1.808864	1.659839e+00	1.66
Year_2001	-0.151744	1.810786	8.592086e-01	0.86
Year_2002	0.128888	1.814526	1.137563e+00	1.14
Year_2003	-0.500824	1.821788	6.060312e-01	0.61
Year_2004	-0.080533	1.832642	9.226245e-01	0.92

Year_2005	-0.158327	1.844367	8.535708e-01	0.85
Year_2006	-0.261062	1.850335	7.702334e-01	0.77
Year_2007	0.146252	1.847232	1.157488e+00	1.16
Year_2008	-0.036080	1.841449	9.645629e-01	0.96
Year_2009	-0.190592	1.827140	8.264701e-01	0.83
Year_2010	-0.592916	1.831269	5.527132e-01	0.55
Year_2011	-0.321983	1.831598	7.247107e-01	0.72
Year_2012	-0.540104	1.831983	5.826879e-01	0.58
Year_2013	-0.551010	1.833559	5.763676e-01	0.58
Year_2014	-0.271657	1.837588	7.621159e-01	0.76
Year_2015	-0.297293	1.844799	7.428261e-01	0.74
Year_2016	-0.358249	1.850046	6.988987e-01	0.70
Year_2017	-0.200906	1.856977	8.179893e-01	0.82
Year_2018	-0.294563	1.863385	7.448572e-01	0.74
Year_2019	-0.156026	1.871071	8.555373e-01	0.86
Year_2020	-0.160546	1.856121	8.516790e-01	0.85
error variance	23.175238	NaN	1.161122e+10	11611218981.84

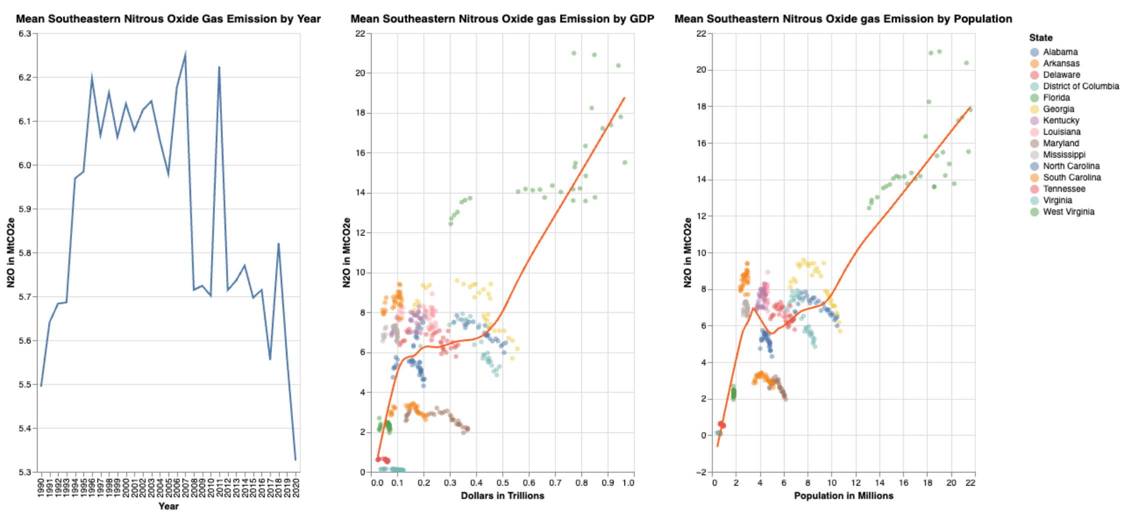
From fitting the multiple linear regression and extracting the exponentiated coefficients, we can again examine how each factor affects FGas emissions, and we can see confirmation of what we saw in the graphs. There is little variation among the coefficients for each year showing that time is not a big factor for emissions. While Population has an effect, the coefficient for GDP is double that amount showing that GDP is the most significant factor for FGas.

GDP is the most significant factor in Flourinated Gases emissions for the Southeast.

6.4.5 Nitrous Oxide

[49]:
```
year_se_N2O | gdp_se_N2O + gdp_se_N2O_loess | pop_se_N2O + pop_se_N2O_loess
```

[49]:



So, again with Nitrous Oxide we see that clustered and stacked pattern amongst the GDP and

Population graphs. This means that while there is some influence of these 2 factors and they follow a positive linear trend, they most likely will not be the most influential factor for NO2 emisisons. In our time graph, on the other hand, there is a similar trend to the FGas time graph because something caused mean NO2 emisisons to increase in 1990 and dramatically decrease from 2008-2020. The difference here is that there are a lot of volatile spikes and dips.

[50]:
```
se_N2O_coef_tbl
```

[50]:

	estimate	standard error	exponentiated	exp_visual
const	0.746303	0.587846	2.109188e+00	2.11
Population	1.633853	0.095371	5.123576e+00	5.12
GDP	-22.651195	2.275426	1.454491e-10	0.00
Year_1991	0.054251	0.784684	1.055749e+00	1.06
Year_1992	0.089376	0.784678	1.093491e+00	1.09
Year_1993	0.031316	0.784688	1.031811e+00	1.03
Year_1994	0.334694	0.784736	1.397513e+00	1.40
Year_1995	0.335415	0.784792	1.398521e+00	1.40
Year_1996	0.547124	0.784892	1.728275e+00	1.73
Year_1997	1.753454	0.799902	5.774514e+00	5.77
Year_1998	1.944440	0.802826	6.989715e+00	6.99
Year_1999	1.956938	0.806334	7.077619e+00	7.08
Year_2000	2.019617	0.807442	7.535438e+00	7.54
Year_2001	1.956887	0.808300	7.077258e+00	7.08
Year_2002	2.024762	0.809970	7.574310e+00	7.57
Year_2003	2.132227	0.813211	8.433624e+00	8.43
Year_2004	2.161653	0.818056	8.685480e+00	8.69
Year_2005	2.206050	0.823290	9.079776e+00	9.08
Year_2006	2.445293	0.825954	1.153393e+01	11.53
Year_2007	2.418912	0.824569	1.123363e+01	11.23
Year_2008	1.744650	0.821988	5.723897e+00	5.72
Year_2009	1.497205	0.815600	4.469180e+00	4.47
Year_2010	1.494110	0.817443	4.455368e+00	4.46
Year_2011	1.986057	0.817590	7.286743e+00	7.29
Year_2012	1.441821	0.817762	4.228388e+00	4.23
Year_2013	1.457744	0.818466	4.296254e+00	4.30
Year_2014	1.510913	0.820264	4.530865e+00	4.53
Year_2015	1.503711	0.823483	4.498352e+00	4.50
Year_2016	1.558122	0.825825	4.749892e+00	4.75
Year_2017	1.459114	0.828919	4.302148e+00	4.30
Year_2018	1.773552	0.831779	5.891744e+00	5.89
Year_2019	1.579235	0.835210	4.851244e+00	4.85
Year_2020	1.117667	0.828537	3.057713e+00	3.06
error variance	4.617803	NaN	1.012713e+02	101.27

From fitting the multiple linear regression and extracting the exponentiated coefficients, we can again examine how each factor affects NO2 emissions, we have a similar conclusion to our graphs. The GDP coefficient is almost 0 which aligns with our clustered and stacked graph. Population has a bigger effect here, however the year coefficients range from 1 to 11 which suggests that time has a bigger influence on NO2 emissions.

Time is the most significant factor in Nitrous Oxide emissions for the Southeast.

6.5 West

[51]:
```
# create a dataframe to isolate the West region
ghg_w = ghg[ghg["Region"] == "West"]

# create boxplots for each gas in the west to look for outliers over years
w_CH4_outliers = alt.Chart(ghg_w).encode(
    x = alt.X("Year:O"),
    y = alt.Y("CH4TOT"),
    tooltip = ("State", "CH4TOT")
).mark_boxplot()

w_CO2_outliers = alt.Chart(ghg_w).encode(
    x = alt.X("Year:O"),
    y = alt.Y("CO2TOT"),
    tooltip = ("State", "CO2TOT")
).mark_boxplot()

w_FGas_outliers = alt.Chart(ghg_w).encode(
    x = alt.X("Year:O"),
    y = alt.Y("FGasTOT"),
    tooltip = ("State", "FGasTOT")
).mark_boxplot()

w_N2O_outliers = alt.Chart(ghg_w).encode(
    x = alt.X("Year:O"),
    y = alt.Y("N2OTOT"),
    tooltip = ("State", "N2OTOT")
).mark_boxplot()

# methane

# methane against year
year_w_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "West") &␣
↪(ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.
↪Scale(zero = False)),
    #color = alt.Color("Gas Type:N")
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Western Methane Gas Emission by Year"
)
```

```python
# methane against gdp
gdp_w_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "West") &␣
↪(ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Western Methane Gas Emission by GDP"
)

# methane against population
pop_w_CH4 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "West") &␣
↪(ghg_melted["Gas Type"] == "CH4TOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "CH4 in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Western Methane gas Emission by Population"
)


# methane dataframe
w_CH4 = ghg_melted[(ghg_melted['Region'] == 'West') & (ghg_melted['Gas Type']␣
↪== 'CH4TOT') & np.isfinite(ghg_melted['Population'])]

# grid of gdp and population values
w_grid_gdp = np.linspace(w_CH4["GDP"].min(), w_CH4["GDP"].max(), num = 100)
w_grid_pop = np.linspace(w_CH4["Population"].min(), w_CH4["Population"].max(),␣
↪num = 100)

# fit loess smooth
gdp_w_CH4_ls = sm.nonparametric.lowess(endog = w_CH4["Gas Total"].values,
                                    exog = w_CH4["GDP"].values,
                                    frac = 0.5,
                                    xvals = w_grid_gdp)
pop_w_CH4_ls = sm.nonparametric.lowess(endog = w_CH4["Gas Total"].values,
                                    exog = w_CH4["Population"].values,
                                    frac = 0.5,
                                    xvals = w_grid_pop)
```

```python
# store as dataframe
gdp_w_CH4_df = pd.DataFrame({'GDP': w_grid_gdp, 'Gas': gdp_w_CH4_ls})
pop_w_CH4_df = pd.DataFrame({'Population': w_grid_pop, 'Gas': pop_w_CH4_ls})

# loess smoother lines
gdp_w_CH4_loess = alt.Chart(
    gdp_w_CH4_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_w_CH4_loess = alt.Chart(
    pop_w_CH4_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# methane regression analysis
# create dummy and x/y variables for mlr
w_CH4_indicators = pd.get_dummies(w_CH4[['Year', 'Population', 'GDP']],
                           drop_first = True)
w_CH4_x = sm.tools.add_constant(w_CH4_indicators)
w_CH4_y = w_CH4['Gas Total']
w_CH4_indicators.columns.values

# fit mlr model
w_CH4_mlr = sm.OLS(endog = w_CH4_y, exog = w_CH4_x.astype(float))
w_CH4_rslt = w_CH4_mlr.fit()

# retreive estimates and std errors
w_CH4_coef_tbl = pd.DataFrame({
    'estimate': w_CH4_rslt.params.values,
    'standard error': np.sqrt(w_CH4_rslt.cov_params().values.diagonal())},
    index = w_CH4_x.columns
)
w_CH4_coef_tbl.loc['error variance', 'estimate'] = w_CH4_rslt.scale

# add column of exponentiated coefficients
w_CH4_coef_tbl['exponentiated']=np.exp(w_CH4_coef_tbl['estimate'])
w_CH4_coef_tbl['exp_visual'] = w_CH4_coef_tbl['exponentiated'].apply(lambda x:␣
↪"{:.2f}".format(x))
```

```python
# carbon dioxide

# carbon dioxide against year
year_w_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "West") &␣
↪(ghg_melted["Gas Type"] == "CO2TOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.
↪Scale(zero = False)),
    #color = alt.Color("Gas Type:N")
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Western CO2 Gas Emission by Year"
)

# carbon dioxide against GDP
gdp_w_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "West") &␣
↪(ghg_melted["Gas Type"] == "CO2TOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Western CO2 Gas Emission by GDP"
)

# carbon dioxide against population
pop_w_CO2 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "West") &␣
↪(ghg_melted["Gas Type"] == "CO2TOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "CO2 in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Western CO2 gas Emission by Population"
)

# carbon dioxide dataframe
w_CO2 = ghg_melted[(ghg_melted['Region'] == 'West') & (ghg_melted['Gas Type']␣
↪== 'CO2TOT')]
```

```
# fit loess smooth
gdp_w_CO2_ls = sm.nonparametric.lowess(endog = w_CO2["Gas Total"].values,
                                        exog = w_CO2["GDP"].values,
                                        frac = 0.5,
                                        xvals = w_grid_gdp)
pop_w_CO2_ls = sm.nonparametric.lowess(endog = w_CO2["Gas Total"].values,
                                        exog = w_CO2["Population"].values,
                                        frac = 0.5,
                                        xvals = w_grid_pop)

# store as dataframe
gdp_w_CO2_df = pd.DataFrame({'GDP': w_grid_gdp, 'Gas': gdp_w_CO2_ls})
pop_w_CO2_df = pd.DataFrame({'Population': w_grid_pop, 'Gas': pop_w_CO2_ls})

# loess smoother lines
gdp_w_CO2_loess = alt.Chart(
    gdp_w_CO2_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_w_CO2_loess = alt.Chart(
    pop_w_CO2_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# carbon dioxide regression analysis
# create dummy and x/y variables for mlr
w_CO2_indicators = pd.get_dummies(w_CO2[['Year', 'Population', 'GDP']],
                               drop_first = True)
w_CO2_x = sm.tools.add_constant(w_CO2_indicators)

# handle infinite cases
w_CO2_x = w_CO2_x.replace([np.inf, -np.inf], np.nan)
w_CO2_x = w_CO2_x.dropna()
w_CO2_y = w_CO2['Gas Total']
w_CO2_y = w_CO2_y.reindex(w_CO2_x.index)
w_CO2_indicators.columns.values

# fit mlr model
w_CO2_mlr = sm.OLS(endog = w_CO2_y, exog = w_CO2_x.astype(float))
```

```
w_CO2_rslt = w_CO2_mlr.fit()

# retreive estimates and std errors
w_CO2_coef_tbl = pd.DataFrame({
    'estimate': w_CO2_rslt.params.values,
    'standard error': np.sqrt(w_CO2_rslt.cov_params().values.diagonal())},
    index = w_CO2_x.columns
)
w_CO2_coef_tbl.loc['error variance', 'estimate'] = w_CO2_rslt.scale

# add column of exponentiated coefficients
w_CO2_coef_tbl['exponentiated']=np.exp(w_CO2_coef_tbl['estimate'])
w_CO2_coef_tbl['exp_visual'] = w_CO2_coef_tbl['exponentiated'].apply(lambda x:␣
↪"{:.2f}".format(x))

# flourinated gas

# flourinated gas against year
year_w_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "West") &␣
↪(ghg_melted["Gas Type"] == "FGasTOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.
↪Scale(zero = False)),
    #color = alt.Color("Gas Type:N")
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Western Flourinated Gas Emission by Year"
)

# flourinated gas against GDP
gdp_w_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "West") &␣
↪(ghg_melted["Gas Type"] == "FGasTOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
    y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.
↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Western Flourinated Gas Emission by GDP"
)

# flourinated gas against population
pop_w_FGas = alt.Chart(ghg_melted[(ghg_melted["Region"] == "West") &␣
↪(ghg_melted["Gas Type"] == "FGasTOT")]).encode(
```

```
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "FGas in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Western Flourinated Gas Emission by Population"
)

# flourinated gas dataframe
w_FGas = ghg_melted[(ghg_melted['Region'] == 'West') & (ghg_melted['Gas Type']␣
 ↪== 'FGasTOT')]

# fit loess smooth
gdp_w_FGas_ls = sm.nonparametric.lowess(endog = w_FGas["Gas Total"].values,
                                        exog = w_FGas["GDP"].values,
                                        frac = 0.5,
                                        xvals = w_grid_gdp)
pop_w_FGas_ls = sm.nonparametric.lowess(endog = w_FGas["Gas Total"].values,
                                        exog = w_FGas["Population"].values,
                                        frac = 0.5,
                                        xvals = w_grid_pop)

# store as dataframe
gdp_w_FGas_df = pd.DataFrame({'GDP': w_grid_gdp, 'Gas': gdp_w_FGas_ls})
pop_w_FGas_df = pd.DataFrame({'Population': w_grid_pop, 'Gas': pop_w_FGas_ls})

# loess smoother lines
gdp_w_FGas_loess = alt.Chart(
    gdp_w_FGas_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)
pop_w_FGas_loess = alt.Chart(
    pop_w_FGas_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# flourinated gas regression analysis
```

```
# create dummy and x/y variables for mlr
w_FGas_indicators = pd.get_dummies(w_FGas[['Year', 'Population', 'GDP']],
                                   drop_first = True)
w_FGas_x = sm.tools.add_constant(w_FGas_indicators)
# handle infinite cases
w_FGas_x = w_FGas_x.replace([np.inf, -np.inf], np.nan)
w_FGas_x = w_FGas_x.dropna()
w_FGas_y = w_FGas['Gas Total']
w_FGas_y = w_FGas_y.reindex(w_FGas_x.index)
w_FGas_indicators.columns.values

# fit mlr model
w_FGas_mlr = sm.OLS(endog = w_FGas_y, exog = w_FGas_x.astype(float))
w_FGas_rslt = w_FGas_mlr.fit()

# retreive estimates and std errors
w_FGas_coef_tbl = pd.DataFrame({
    'estimate': w_FGas_rslt.params.values,
    'standard error': np.sqrt(w_FGas_rslt.cov_params().values.diagonal())},
    index = w_FGas_x.columns
)
w_FGas_coef_tbl.loc['error variance', 'estimate'] = w_FGas_rslt.scale

# add column of exponentiated coefficients
w_FGas_coef_tbl['exponentiated']=np.exp(w_FGas_coef_tbl['estimate'])
w_FGas_coef_tbl['exp_visual'] = w_FGas_coef_tbl['exponentiated'].apply(lambda x:
 ↪ "{:.2f}".format(x))

# nitrous oxide

# nitrous oxide against year
year_w_N20 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "West") &␣
 ↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("Year:O", title = "Year"),
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
).mark_line().properties(
    height = 500,
    width = 300,
    title = "Mean Western Nitrous Oxide Gas Emission by Year"
)

# nitrous oxide against GDP
gdp_w_N20 = alt.Chart(ghg_melted[(ghg_melted["Region"] == "West") &␣
 ↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("GDP:Q", title = "Dollars in Trillions"),
```

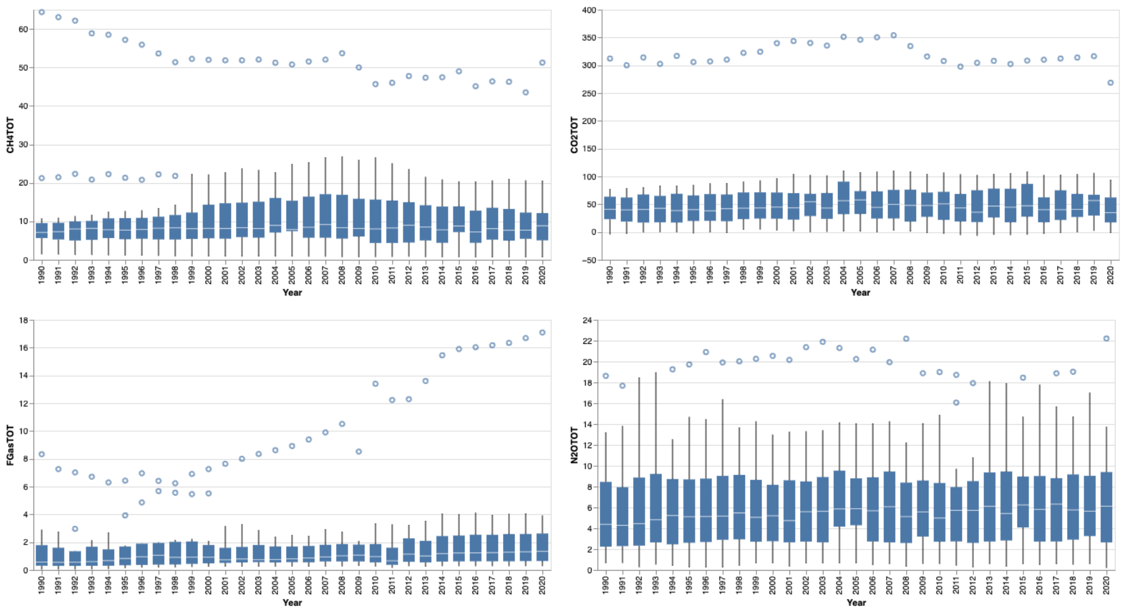
```
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Western Nitrous Oxide gas Emission by GDP"
)

# nitrous oxide against population
pop_w_N2O = alt.Chart(ghg_melted[(ghg_melted["Region"] == "West") &␣
 ↪(ghg_melted["Gas Type"] == "N2OTOT")]).encode(
    x = alt.X("Population:Q", title = "Population in Millions"),
    y = alt.Y("mean(Gas Total):Q", title = "N2O in MtCO2e", scale = alt.
 ↪Scale(zero = False)),
    color = alt.Color("State")
).mark_circle(opacity = 0.5).properties(
    height = 500,
    width = 300,
    title = "Mean Western Nitrous Oxide gas Emission by Population"
)

# nitrous oxide dataframe
w_N2O = ghg_melted[(ghg_melted['Region'] == 'West') & (ghg_melted['Gas Type']␣
 ↪== 'N2OTOT')]

# fit loess smooth
gdp_w_N2O_ls = sm.nonparametric.lowess(endog = w_N2O["Gas Total"].values,
                                    exog = w_N2O["GDP"].values,
                                    frac = 0.5,
                                    xvals = w_grid_gdp)
pop_w_N2O_ls = sm.nonparametric.lowess(endog = w_N2O["Gas Total"].values,
                                    exog = w_N2O["Population"].values,
                                    frac = 0.5,
                                    xvals = w_grid_pop)

# store as dataframe
gdp_w_N2O_df = pd.DataFrame({'GDP': w_grid_gdp, 'Gas': gdp_w_N2O_ls})
pop_w_N2O_df = pd.DataFrame({'Population': w_grid_pop, 'Gas': pop_w_N2O_ls})

# loess smoother lines
gdp_w_N2O_loess = alt.Chart(
    gdp_w_N2O_df
).encode(
    x = alt.X("GDP"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
```

```
    color = "#FF5919"
)
pop_w_N20_loess = alt.Chart(
    pop_w_N20_df
).encode(
    x = alt.X("Population"),
    y = alt.Y("Gas", scale = alt.Scale(zero = False))
).mark_line(
    color = "#FF5919"
)

# nitrous oxide regression analysis
# create dummy and x/y variables for mlr
w_N20_indicators = pd.get_dummies(w_N20[['Year', 'Population', 'GDP']],
                                   drop_first = True)
w_N20_x = sm.tools.add_constant(w_N20_indicators)
# handle infinite cases
w_N20_x = w_N20_x.replace([np.inf, -np.inf], np.nan)
w_N20_x = w_N20_x.dropna()
w_N20_y = w_N20['Gas Total']
w_N20_y = w_N20_y.reindex(w_N20_x.index)
w_N20_indicators.columns.values

# fit mlr model
w_N20_mlr = sm.OLS(endog = w_N20_y, exog = w_N20_x.astype(float))
w_N20_rslt = w_N20_mlr.fit()

# retreive estimates and std errors
w_N20_coef_tbl = pd.DataFrame({
    'estimate': w_N20_rslt.params.values,
    'standard error': np.sqrt(w_N20_rslt.cov_params().values.diagonal())},
    index = w_N20_x.columns
)
w_N20_coef_tbl.loc['error variance', 'estimate'] = w_N20_rslt.scale

# add column of exponentiated coefficients
w_N20_coef_tbl['exponentiated']=np.exp(w_N20_coef_tbl['estimate'])
w_N20_coef_tbl['exp_visual'] = w_N20_coef_tbl['exponentiated'].apply(lambda x:␣
↪"{:.2f}".format(x))
```

6.5.1 Outlier Analysis

[52]:
```
(w_CH4_outliers | w_CO2_outliers) & (w_FGas_outliers | w_N20_outliers)
```

[52]:

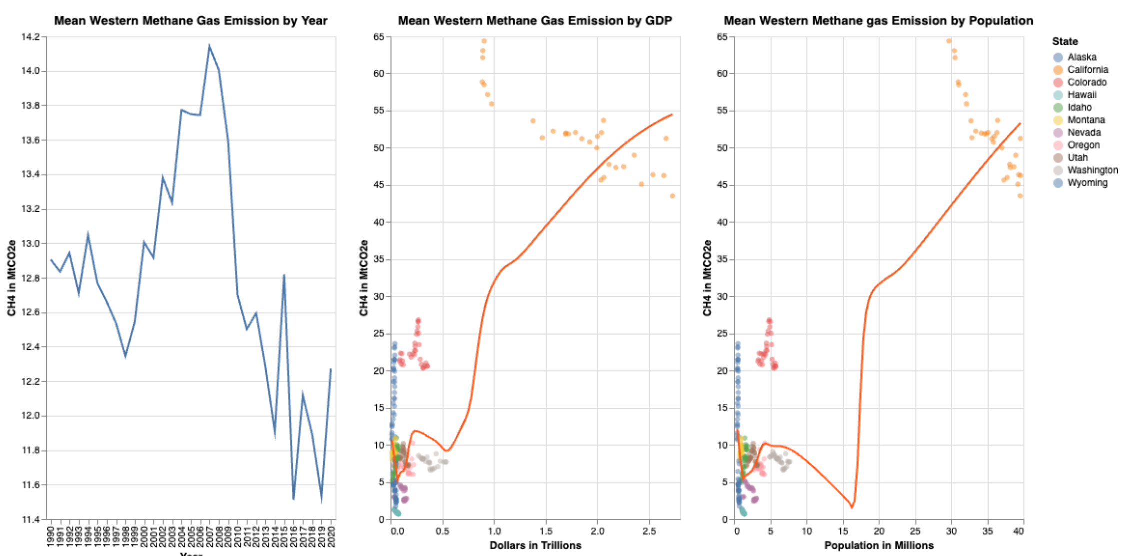


For methane, California is a strong outlier. Colorado was an outlier from 1990 to 1998. For carbon dioxide, California is the sole outlier. For the flourinated gasses, California is a strong outlier. Washington was an outlier from 1990 to 2000. Finally, for nitrous oxide, California is a strong outlier. Montana was an outlier in 2011. Looking at these plots, we can expect California to be an important factor for our total gas emmissions for all the gasses we are interested in.

6.5.2 Methane

[53]:
```
year_w_CH4 | gdp_w_CH4 + gdp_w_CH4_loess | pop_w_CH4 + pop_w_CH4_loess
```

[53]:



The total methane gas emission from year to year follows a negative quadratic trend. The emissions spike from around 2006 to 2007. The trend itself has a moderate amount of noise. The emissions against GDP and emissions against population look very similar to one another. Each of the individual state follow a similar trend shape to the yearly plot. All of the states are also at considerably lower GDP and population size than California, so they are compressed to the left. Due to California, the plots end up resembling a positive linear trend.

[54]:
```
w_CH4_coef_tbl
```

[54]:

	estimate	standard error	exponentiated \
const	5.505120	1.953528	2.459480e+02
Population	2.281475	0.156596	9.791107e+00
GDP	-19.586147	2.939663	3.117772e-09
Year_1991	-0.338817	2.723781	7.126132e-01
Year_1992	-0.319677	2.723808	7.263838e-01
Year_1993	-0.709162	2.723887	4.920563e-01
Year_1994	-0.415219	2.723860	6.601960e-01
Year_1995	-0.887880	2.723925	4.115273e-01
Year_1996	-0.992210	2.723835	3.707566e-01
Year_1997	-0.060632	2.728315	9.411698e-01
Year_1998	-0.119496	2.730192	8.873677e-01
Year_1999	0.172176	2.732420	1.187887e+00
Year_2000	0.687311	2.734809	1.988362e+00
Year_2001	0.425203	2.733772	1.529902e+00
Year_2002	0.843190	2.734148	2.323769e+00
Year_2003	0.760691	2.735776	2.139754e+00
Year_2004	1.313579	2.737437	3.719462e+00
Year_2005	1.414317	2.740594	4.113678e+00
Year_2006	1.498975	2.743462	4.477099e+00
Year_2007	1.885796	2.744997	6.591602e+00
Year_2008	1.718258	2.745438	5.574810e+00
Year_2009	0.913229	2.740557	2.492356e+00
Year_2010	0.009733	2.741522	1.009780e+00
Year_2011	-0.259348	2.741901	7.715541e-01
Year_2012	-0.174150	2.743108	8.401707e-01
Year_2013	-0.398269	2.745507	6.714814e-01
Year_2014	-0.738926	2.748027	4.776265e-01
Year_2015	0.318287	2.752627	1.374770e+00
Year_2016	-0.954458	2.755408	3.850208e-01
Year_2017	-0.146650	2.761453	8.635959e-01
Year_2018	0.155751	2.838348	1.168535e+00
Year_2019	-0.355630	2.774330	7.007316e-01
Year_2020	0.156028	2.769201	1.168859e+00
error variance	40.802398	NaN	5.251175e+17

	exp_visual
const	245.95

```
Population                    9.79
GDP                           0.00
Year_1991                     0.71
Year_1992                     0.73
Year_1993                     0.49
Year_1994                     0.66
Year_1995                     0.41
Year_1996                     0.37
Year_1997                     0.94
Year_1998                     0.89
Year_1999                     1.19
Year_2000                     1.99
Year_2001                     1.53
Year_2002                     2.32
Year_2003                     2.14
Year_2004                     3.72
Year_2005                     4.11
Year_2006                     4.48
Year_2007                     6.59
Year_2008                     5.57
Year_2009                     2.49
Year_2010                     1.01
Year_2011                     0.77
Year_2012                     0.84
Year_2013                     0.67
Year_2014                     0.48
Year_2015                     1.37
Year_2016                     0.39
Year_2017                     0.86
Year_2018                     1.17
Year_2019                     0.70
Year_2020                     1.17
error variance  525117468708176576.00
```

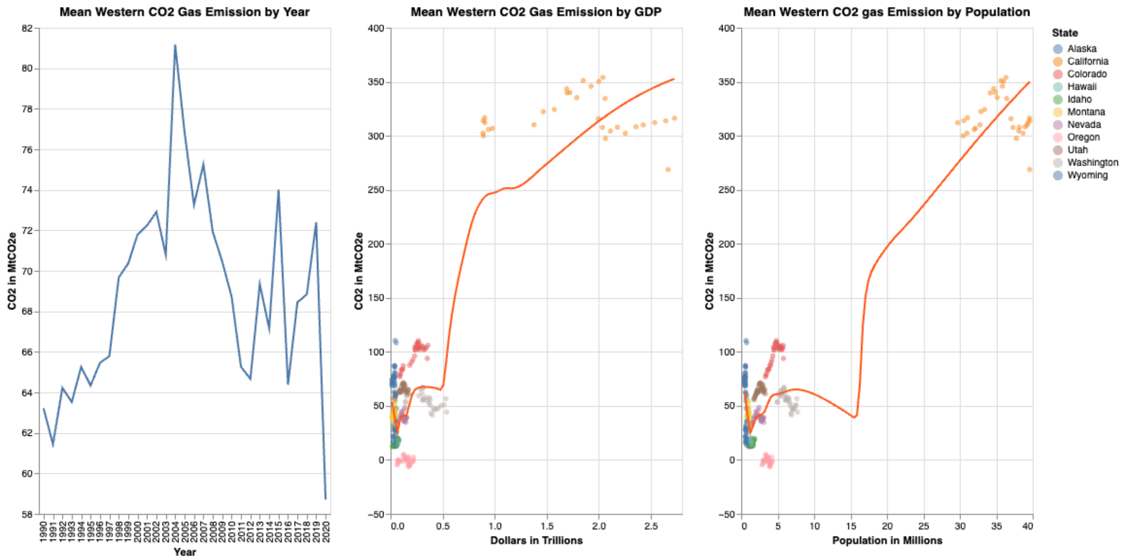
When looking at the coefficients from our multiple linear regression analysis, population has a factor of 9.79 and GDP has a factor of 0. The coefficients for year range from around 0 to 7. Considering we observed that the emissions for methane spikes at certain years, this is likely why certain years could be weighted so highly.

Methane emissions in the West are most strongly predicted by population size of the state.

6.5.3 Carbon Dioxide

[55]:
```
year_w_CO2 | gdp_w_CO2 + gdp_w_CO2_loess | pop_w_CO2 + pop_w_CO2_loess
```

[55]:



Similar to methane, carbon dioxide emissions follow a negative quadratic trend from year to year. The highest spikes of emissions were from around 2004 to 2005. When excluding California, we can see for the GDP and population plots that all the other states follow a similar negative quadratic trend. However, as GDP and population grow, California largely increases total emmissions and the overall trend becomes positively linear.

[56]:
```
w_CO2_coef_tbl
```

[56]:

	estimate	standard error	exponentiated	exp_visual
const	23.477278	9.348392	1.570553e+10	15705525221.90
Population	10.550455	0.749375	3.819480e+04	38194.80
GDP	-47.001938	14.067431	3.866498e-21	0.00
Year_1991	-2.992024	13.034353	5.018578e-02	0.05
Year_1992	-0.703386	13.034478	4.949065e-01	0.49
Year_1993	-2.205543	13.034857	1.101906e-01	0.11
Year_1994	-0.827963	13.034727	4.369385e-01	0.44
Year_1995	-2.883839	13.035039	5.591967e-02	0.06
Year_1996	-2.039135	13.034608	1.301412e-01	0.13
Year_1997	0.362436	13.056048	1.436825e+00	1.44
Year_1998	4.307317	13.065032	7.424102e+01	74.24
Year_1999	4.783710	13.075691	1.195470e+02	119.55
Year_2000	5.813479	13.087126	3.347817e+02	334.78
Year_2001	5.471597	13.082160	2.378398e+02	237.84
Year_2002	5.744866	13.083960	3.125818e+02	312.58
Year_2003	3.541679	13.091750	3.452485e+01	34.52
Year_2004	13.570361	13.099702	7.825873e+05	782587.28
Year_2005	9.246173	13.114808	1.036483e+04	10364.83
Year_2006	5.579090	13.128531	2.648306e+02	264.83
Year_2007	7.177668	13.135879	1.309850e+03	1309.85

Year_2008	3.542193	13.137988	3.454259e+01	34.54
Year_2009	0.708735	13.114632	2.031419e+00	2.03
Year_2010	-1.332456	13.119248	2.638286e-01	0.26
Year_2011	-5.318222	13.121063	4.901460e-03	0.00
Year_2012	-6.246157	13.126840	1.937887e-03	0.00
Year_2013	-1.503941	13.138318	2.222524e-01	0.22
Year_2014	-4.001599	13.150375	1.828638e-02	0.02
Year_2015	2.882191	13.172388	1.785335e+01	17.85
Year_2016	-7.011465	13.185697	9.014866e-04	0.00
Year_2017	-2.696823	13.214628	6.741939e-02	0.07
Year_2018	3.424190	13.582600	3.069776e+01	30.70
Year_2019	1.730699	13.276248	5.644600e+00	5.64
Year_2020	-12.678854	13.251703	3.116333e-06	0.00
error variance	934.372713	NaN	inf	inf

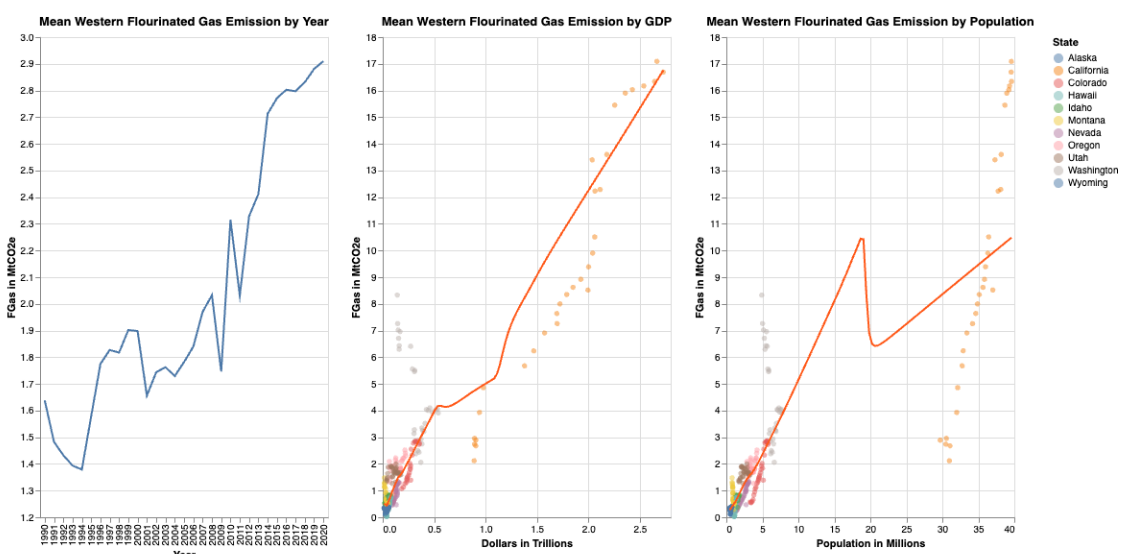
The yearly coefficients of our multiple linear regression analysis ranges from 0 all the way to 782587. The coefficients for GDP and population are 0 and 38194.80 respectively. The biggest contributor for carbon dioxide emissions in the west from 2004 to 2005 are the years itself.

However, the total emissions of carbon dioxide in the west is most consistently predicted by population.

6.5.4 Flourinated Gas

[57]:
```
year_w_FGas | gdp_w_FGas + gdp_w_FGas_loess | pop_w_FGas + pop_w_FGas_loess
```

[57]:



The total emissions for flourinated gasses follow similar trends for the year, GDP, and population plots. All of them appear to be positively linear, however, the population plot also demonstrates some positively cubic qualities as well. California does not seem to deviate the trends too drastically for the flourinated gasses despite still being an outlier state. Washington appears to produce a considerable amount more flourinated gas polution per gdp and capita.

[58]:
```
w_FGas_coef_tbl
```

[58]:

	estimate	standard error	exponentiated	exp_visual
const	1.335310	0.350081	3.801173	3.80
Population	-0.185997	0.028063	0.830276	0.83
GDP	8.756051	0.526800	6348.990814	6348.99
Year_1991	-0.129942	0.488113	0.878146	0.88
Year_1992	-0.185463	0.488118	0.830720	0.83
Year_1993	-0.224615	0.488132	0.798824	0.80
Year_1994	-0.262720	0.488127	0.768957	0.77
Year_1995	-0.086787	0.488139	0.916873	0.92
Year_1996	0.061396	0.488123	1.063320	1.06
Year_1997	-0.434665	0.488926	0.647481	0.65
Year_1998	-0.546595	0.489262	0.578918	0.58
Year_1999	-0.573397	0.489661	0.563607	0.56
Year_2000	-0.688251	0.490090	0.502454	0.50
Year_2001	-0.913879	0.489904	0.400966	0.40
Year_2002	-0.853373	0.489971	0.425976	0.43
Year_2003	-0.899854	0.490263	0.406629	0.41
Year_2004	-1.004502	0.490560	0.366227	0.37
Year_2005	-1.056676	0.491126	0.347609	0.35
Year_2006	-1.088320	0.491640	0.336782	0.34
Year_2007	-1.014737	0.491915	0.362498	0.36
Year_2008	-0.973696	0.491994	0.377684	0.38
Year_2009	-1.157476	0.491120	0.314278	0.31
Year_2010	-0.627881	0.491292	0.533721	0.53
Year_2011	-0.939946	0.491360	0.390649	0.39
Year_2012	-0.688859	0.491577	0.502149	0.50
Year_2013	-0.672443	0.492007	0.510460	0.51
Year_2014	-0.449515	0.492458	0.637938	0.64
Year_2015	-0.506747	0.493282	0.602452	0.60
Year_2016	-0.549260	0.493781	0.577377	0.58
Year_2017	-0.681278	0.494864	0.505970	0.51
Year_2018	-0.846690	0.508644	0.428832	0.43
Year_2019	-0.837441	0.497172	0.432817	0.43
Year_2020	-0.733548	0.496253	0.480202	0.48
error variance	1.310335	NaN	3.707417	3.71

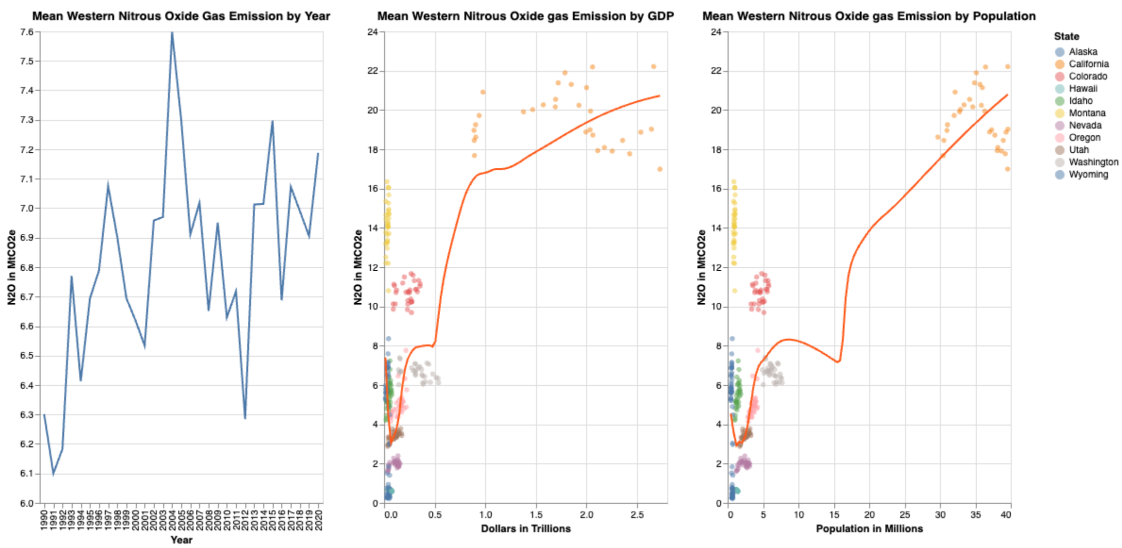
The yearly coefficients of our multiple linear regression analysis for flourinated gasses range from 0 to 1.06. The coefficient for GDP is 6348.99, and the coefficient for population is 0.83. Thus, the contributions of time and population seem to be fairly negligible.

As a result, is it strongly indicated that flourinated gas emissions in the west is most strongly predicted by GDP.

6.5.5 Nitrous Oxide

[59]:
```
year_w_N2O | gdp_w_N2O + gdp_w_N2O_loess | pop_w_N2O + pop_w_N2O_loess
```

[59]:



They yearly plot of nitrous oxide emissions in the west looks to be a strongly noisy positive linear trend. For the GDP and population plots, we first observe a negative quadratic trend when excluding California. However, the emissions begin to follow a positive linear trend once the GDP and population grow to California levels. The most important factor is not made immediately apparent.

[60]:
```
w_N2O_coef_tbl
```

[60]:

	estimate	standard error	exponentiated	exp_visual
const	4.092491	1.236514	5.988891e+01	59.89
Population	0.624554	0.099120	1.867412e+00	1.87
GDP	-3.919267	1.860702	1.985564e-02	0.02
Year_1991	-0.273509	1.724056	7.607054e-01	0.76
Year_1992	-0.217732	1.724073	8.043410e-01	0.80
Year_1993	0.324041	1.724123	1.382704e+00	1.38
Year_1994	-0.050288	1.724106	9.509558e-01	0.95
Year_1995	0.168613	1.724147	1.183662e+00	1.18
Year_1996	0.256468	1.724090	1.292358e+00	1.29
Year_1997	0.738643	1.726926	2.093094e+00	2.09
Year_1998	0.582787	1.728114	1.791022e+00	1.79
Year_1999	0.379945	1.729524	1.462203e+00	1.46
Year_2000	0.297553	1.731037	1.346560e+00	1.35
Year_2001	0.165339	1.730380	1.179793e+00	1.18
Year_2002	0.571062	1.730618	1.770146e+00	1.77
Year_2003	0.587673	1.731648	1.799795e+00	1.80
Year_2004	1.206078	1.732700	3.340360e+00	3.34
Year_2005	0.923116	1.734698	2.517121e+00	2.52

Year_2006	0.542075	1.736513	1.719572e+00	1.72
Year_2007	0.634948	1.737485	1.886925e+00	1.89
Year_2008	0.254642	1.737764	1.289999e+00	1.29
Year_2009	0.460828	1.734675	1.585387e+00	1.59
Year_2010	0.127400	1.735285	1.135871e+00	1.14
Year_2011	0.190901	1.735525	1.210339e+00	1.21
Year_2012	-0.254489	1.736290	7.753123e-01	0.78
Year_2013	0.486652	1.737808	1.626861e+00	1.63
Year_2014	0.482065	1.739403	1.619415e+00	1.62
Year_2015	0.783465	1.742314	2.189044e+00	2.19
Year_2016	0.168772	1.744075	1.183850e+00	1.18
Year_2017	0.586653	1.747901	1.797961e+00	1.80
Year_2018	0.597362	1.796573	1.817319e+00	1.82
Year_2019	0.480574	1.756052	1.617002e+00	1.62
Year_2020	0.711849	1.752805	2.037755e+00	2.04
error variance	16.347226	NaN	1.257506e+07	12575059.37

The yearly coefficients of our multiple linear regression analysis of nitrous oxide in the west range from 0 to 3.34. This suggests that time may not be our strongest predictor. However, the coefficients for GDP and population are 0.02 and 1.87 respectively. Thus, although population may be our most significant predictor, it is ambiguous whether any of these three metrics are strongly responsible for nitrous oxide emissions in the West.

Therefore, we cannot strongly assert any of our three predictors: year, GDP, or population, are most responsible for nitrous oxide emissions in the west.

7 Conclusion

Midwest:

- Methane: Population
- Carbon Dioxide: Population and Time
- Flourinated Gas: GDP
- Nitrous Oxide: GDP
- Overall: From our analysis, greenhouse gas emissions in the Midwest region were mainly influenced by Population and Time. More specifically, Population was a big contributor for CH_4 and CO_2 emissions, and GDP was a big contributor for FGas and N_2O emissions. In our plots of emission against GDP and emission against population, the trends looked very similar for each gas. Time was a significant contributor only for CO_2.

Southwest:

- Methane: Population
- Carbon Dioxide: Population and Time
- Flourinated Gas: GDP
- Nitrous Oxide: Population

- Overall: Methane and nitrous oxide emissions were relatively influenced by population, whereas fluorinated gas was heavily influenced by GDP and carbon dioxide was heavily influenced by population for 10 years from 1997 to 2007. Overall population did play a part for most of our gases but had the most influence coupled with time for CO2. GDP influenced flourinated gas emissions which could tie into our initial EDA which found that fluorinated gas increased in emissions over time, and could've also been influenced by GDP. Texas, our outlier state dragged the emission average up but also contributed to the Southwestern's region GDP and mostly decreased or stabilized its emissions as GDP or capita increased to higher values.

Northeast:

- Methane: Population and time
- Carbon Dioxide: Population and time
- Flourinated Gas: GDP and time
- Nitrous Oxide: Population and time
- Overall: Time is the biggest contributor for all four gases, and population is also a contributor for ch4, co2, and n2o. The three showed similar trends in our plots. While we did see that there were positive trends between gas emissions and GDP, we noticed that in the northeast region, time and GDP were positive correlated. Therefore, the trends we were seeing in our GDP graphs were likely explained by the hidden time factor and not GDP itself. However, for Fluorinated Gases, GDP plays a significant role in contributing to emission amounts.

Southeast:

- Methane: Time and Population
- Carbon Dioxide: Time and Population
- Flourinated Gas: GDP
- Nitrous Oxide: Time and Population
- Overall: It seems that time is the biggest contibutor for 3 of the gases, with Population as a close second, but for Flourinated Gases it is GDP. For CH4, CO2, and NO2, there were a lot of spikes year over year, and for most of the gases the GDP and Population plots didn't really have any strong linear trends, but instead very stacked and clustered graphs. Therefore, this means that there must've been some hidden factors whether it be social, political, or environmental that led to the changes in GHG emissions. For Flourinated Gases on the other hand, there was a very strong positive correlation between emissions and GDP thus showing that high GDP correlates with high Flourinated Gas emissions.

West:

- Methane: Population
- Carbon Dioxide: Population
- Flourinated Gas: GDP
- Nitrous Oxide: Unclear

- Overall: For the western region of the United States, only the flourinated gasses seems to be strongly effected by GDP. This suggests that there exists some sucessful industry that results in flourinated gas production as either a product or byproduct. Methane and carbon dioxide both correlate with population most strongly out of our three interested predictors. However, both of these plots may benefit from doing anaylsis without California affecting outcomes so heavily. Nitrous oxide is the most elusive to analyze amongst our gasses. None of our predictors of were able to adequately explain its trend. Looking at other predictors and cofactors may be necessary to understand it better.

We began this project with the intention of understanding whether GHG emisions were strongly tied to GDP, Population, or something else entirely. Seeing as GDP and Population are factors that are difficult to change and thus lower GHG emisions, we were hoping that we would find evidence to the contrary. Some of our findings do in fact come to this conclusion, however, some have contradicted others. Looking at the results from our analyses, it seems that some of our results vary from region to region - this may mean that some states will require different approaches in their quests to lower GHG emissions. Across the board, though, we find that high Flourinated Gas emissions are very closely tied to high GDP and high Carbon Dioxide emissions are closely tied to high Population. Taking these results into consideration, we recognize the limit of our dataset and analysis but hope that this can lead to further investigation. It seems that time was indeed a big factor due to the fact that there would be many sporadic spikes and dips in GHG emissions year over year across both regions and gases. Further investigation could perhaps take into account other data about those years and see if any other external factors contribute to GHG emissions so that eventually we could see a decrease in nationwide GHG emissions and save our atmospere.