AMATH 582
Analysis of Predictive Climate Modeling Using SSPs and SVDs

Rachel Carroll, Shannon Dow

March, 2020

1 Abstract

The goal of this project is to develop a framework for analyzing Shared Socioeconomic Pathways (SSPs) using the Singular Value Decomposition (SVD). The SSP data provides extrapolated pollutant emissions data based on different possible narratives of the future. Using the predictive climate FaIR model, we can see how these emissions affect temperature. The purpose of using the SVD is to reduce the dimensionality and see dominant patterns and correlations of emission and temperature changes to compare the SSP scenarios.

2 Introduction and Overview

FaIR: The Finite Amplitude Impulse Response Model (FaIR):

FaIR is an emissions-based climate model that calculates mean atmospheric concentrations, radiative forcing, and temperature anomalies. The model uses emission data from 38 pollutants, including greenhouse gases, aerosols, and other agents. Specifically, the data comes from Representative Concentration Pathway (RCP) datasets which contain historical emission estimates as well as future projections across the pollutants. The RCP scenarios are based on estimated radiative forcing levels at year 2100. For example RCP 8.5 is a dataset that results in 8.5 radiative forcing level in 2100.

This model is valuable because it involves calculations specific to properties of the given pollutant. On the other hand, the model is somewhat limited to using only the RCP datasets given that it requires the data for all 38 pollutants. In this paper, we leverage the model and adjust it so it will work with other datasets. Moreover, we look at the calculated temperature annomolies of specific pollutants rather then average effects of a the 38 as a group. Rather than using the RCP data, this paper investigates the result of this model on Shared Socioeconomic Pathways data.

Shared Socioeconomic Pathway (SSP):
Shared Socioeconomic Pathways or SSPs are a reference framework recently developed to enhance climate research and modeling representing different ways the world could change in the future.

These SSP frameworks take into account factors such as population size, economic growth, technological advancements and examine their effects on emission levels and temperature. The matrix architecture used to describe the scenarios (see Figure 1) can help us understand the main qualities of the 5 narrative pathways. In general, challenges for mitigation represent difficulties that society as a whole faces in responding to climate issues. On the other axis, challenges for adaptation are societal, cultural, or actual environmental factors that make adaptation difficult or heighten the risk of not adapting. For this project, we well be focusing on three of the SSP narratives:

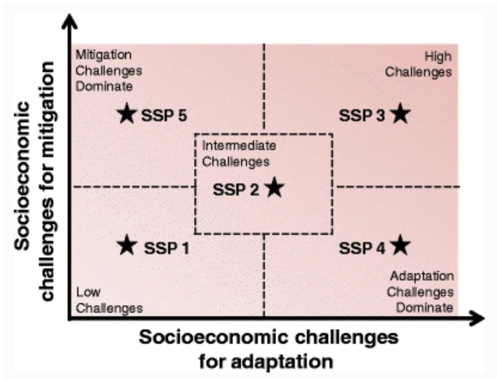


Figure 1: SSP Matrix Framework

- SSP1: An optimistic version of the future with low challenges for both adaptation as well as mitigation. This means that there are minimal barriers to sustainable innovation, despite low perceived risk. There is less inequality and a shift to the importance to the well-being of humanity.

- SSP3: The opposite of SSP 1 where the world development faces challenges to both adaptation and mitigation. This means that while humans are seeing the impacts of climate warming, there is limited being done to prevent or stop the problems. There are high amounts of inequality, nationalism, and environmental decay.

- SSP5: Another positive version of societal advancement. There are still large challenges to mitigation, so society makes limited efforts to innovate and become sustainable. Since challenges to advancement are low, the world does not see the negative environmental impacts of these policies directly.

These SSP frameworks are important in climate research since they allow for predictive modeling that can be used to understand what combinations of policies and reactions to certain socioeconomic factors are the best in preventing environmental degradation, increased emissions, and warming. We will be using the SVD to analyze the pollutants that have the greatest effect on the temperature in different SSP frameworks.

3 Theoretical Background

3.1 The Singular Value Decomposition

The Singular Value Decomposition (SVD) is a very valuable tool in computational data analysis because it extracts relevant correlations and variances across a data set, and in turn, highlights its significant behaviors. Given a matrix A, the SVD is defined as follows

$$A = U\Sigma V^*$$

where U and V are orthonormal and Σ is diagonal. To understand this representation, we will first consider how a matrix A behaves as a linear operator. Consider the standard linear equation $Ax = b$. This operation can be thought of as a projection of x onto the range of A. Physically, this occurs by rotating and stretching x, resulting in the transformation to the vector b.

3.2 Calculating the SVD

To calculate the SVD, while that will not be necessary in the algorithm since Python has its own SVD function, you first form the covariance matrix. Understanding this leads to better interpretability of the results. The covariance matrix looks like this:

$$C_A = \frac{1}{n-1}A^T A = \frac{1}{n-1}\begin{bmatrix} \sigma^2_{xa,xa}, \sigma^2_{xa,ya}, \cdots \\ \sigma^2_{ya,xa}, \sigma^2_{ya,ya} \cdots \\ \ddots, \ddots \end{bmatrix}$$

$$\begin{aligned} A^T A &= (U\Sigma V^*)^T (U\Sigma V^*) = V\Sigma^T U^* U\Sigma V^* = V\Sigma^2 V^* \\ &\implies A^T AV = V\Sigma^2 \end{aligned}$$

$$\begin{aligned} AA^T &= (U\Sigma V^*)(U\Sigma V^*)^T = U\Sigma V^* V\Sigma U^* = U\Sigma^2 U^* \\ &\implies A^T AU = U\Sigma^2 \end{aligned}$$

This shows that the columns of V are the eigenvectors of $A^T A$ and the columns of U are the eigenvectors of AA^T and the eigenvalues of the covariance matrix equal the squares of the singular values.

3.3 Interpretation and Computational Power of the SVD

The off-diagonal terms are the covariances between the columns. Large off diagonal entries represent a high-degree of redundancy, whereas a small off-diagonal coefficients means there is little redundancy in the data, i.e. they are statistically independent.

One of the main reasons the SVD is so powerful is that every matrix has a SVD. It has many applications since it diagonalizes the data into an optimalorthogonal coordinate system. This is possible since it uses two different bases (U and V) to represent the data.

When analyzing the matrices produced by the SVD, the main thing to interpret is the relative size of the diagonal elements of the matrix Σ, or the singular values of the matrix. Each large singular value represents the importance of the corresponding directions, or the first columns of U and V. Looking at the columns of U and V will also help us determine significant dynamics.

Temperature Calculations from Emissions:
The FaIR model performs the following calculations

For each pollutant i, let τ_i be the pollutants lifetime. Then the following equations are performed across the (annual) time series data

Concentration of Greenhouse Gasses (C_i):

$$\frac{dC_i}{dt} = E_i - \frac{C_i}{\tau}$$

Where E_i is the pollutant emissions in one year

Radiative Forcing (F_i):

$$F_i = \eta_i(C_i - C_{i,pi})$$

Where η_i is the pollutant's radiative efficiency parameter and $C_{i,pi}$ is the pre-industrial ear concentration of the pollutant (1850-1900).

Temperature (T_i):
Temperature is calculated from an impulse response model, where forcing is related to total temperature change in year t, T_t , by a two-time constant model.

$$T_{t,i} = T_{t-1,i}e^{\frac{1}{d_i}} \sum_{j=0}^{12} q_i \epsilon_j F_j \left(1 - e^{\frac{1}{d_i}}\right) \quad i = 1, 2$$

Where d_i refer to the fast and slow temperature changes due to response of forcing from the upper ocean and deep ocean respectively. q_i are parameters related to climate sensitivity. ϵ_j are parameters related to forcing-specific efficacies.

4 Algorithm Implementation and Development

Using the SVD, we can analyze the projected emissions data and compare the results from the three chosen SSP frameworks. The first step is to re-configure the data into a usable format. Once the reduced SVD has been calculated, plotting the singular values, especially on a log scale, will help determine how many relevant modes there are. Then, we can analyze the relevant modes to compare how the emission levels change over time in each of the different SSP frameworks.

Algorithm 1: Reformatting Data and Calculate SVD

Read in the CSV file and interpolate the emissions data for 2015-2100
Create 3 empty panda DataFrames for each of the SSPs
for Each SSP (1,3 and 5) **do**
  Create a column for each of the various pollutants and add them to the new DataFrame
end for
Take the SVD of each of the 3 DataFrames.

For each SSP, the resulting matrix that we take the SVD of has the form:

$$\begin{bmatrix} 'CO2' & 'CH4' & 'BC' & \dots & 'Pollutant' \\ 2015Em & 2015Em & 2015Em & \dots & YEAREm \\ 2016Em & 2016Em & 2016Em & \dots & YEAREm \\ \vdots & \vdots & \vdots & \dots & \vdots \end{bmatrix}$$

Once the emissions are analyzed, we can analyze the temperature in a similar manner. Once we run the SSP data through the FaIR model, we will have temperature data for the three SSPS that we are investigating for all of the pollutants. We then want to analyze how the emissions over time affect the dominant behavior of temperature using the SVD.

5 Computational Results

5.1 Emissions by Pollutant (Figure 2)

In examining the singular values for the emission data in each of the SSPs, there is one dominant singular value. However, upon examining the log plot, the first eight are potentially significant. The columns of U from the SVD represent how the emissions change in time. Since the first singular value is by far the most significant we will examine the first singular vectors (see Figure 2). Since most of the V values are negative, the actual pattern of emissions follow the negated U curves for most pollutants. As we would expect, SSP1 and SSP3 have vastly different dominant behavior over time. In SSP1, since the challenges to mitigation and adaptation are low, the world prioritizes sustainability and we expect emissions to trend down over time. Conversely, in SSP3, where challenges in both areas are high, emissions increase in general over time. In the SSP5 framework, sustainable innovation is not a priority, fossil fuel usage and emissions increase in the future. The columns of V are the coefficients of the column of U. The large up and down spikes seen in the plots of V are due to relative importance of each pollutant.

5.2 Emissions by Region (Figure 3)

In this case the V modes represent the larger spikes represent the significance of a pollutant emitted from a given region. The volatility of the V column shows that there is a wide range of emissions

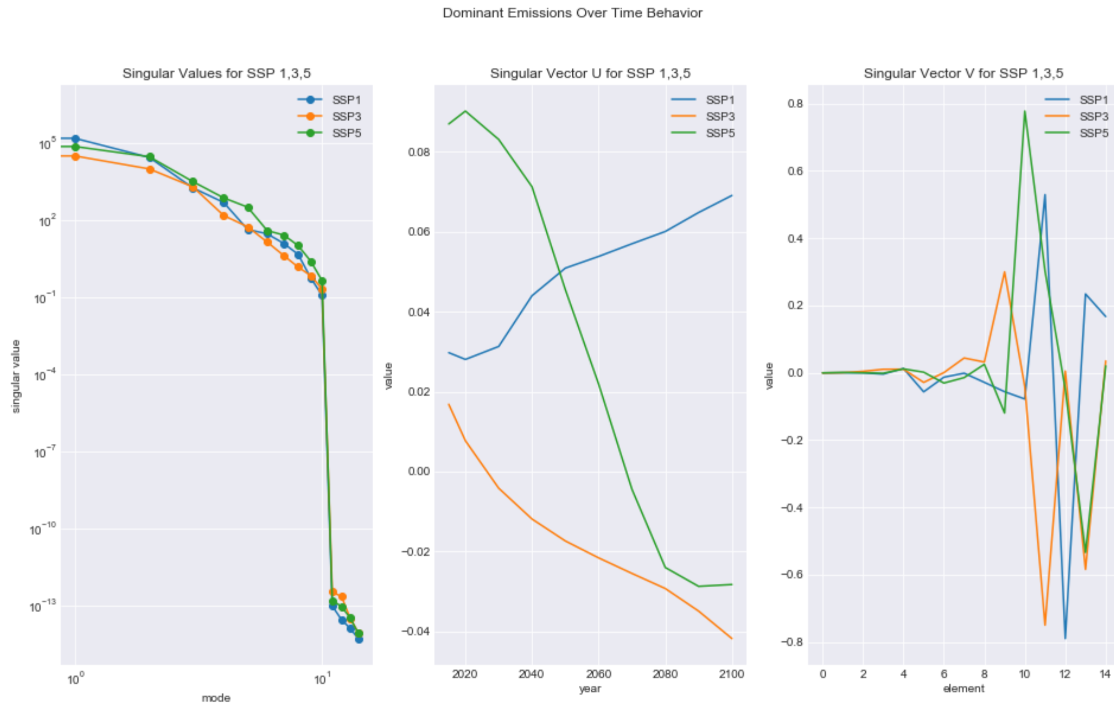


Figure 2: Emissions by Pollutant SVD Data

patterns in various regions. Since many of the values of V taper down near zero, there are relatively low changes in emission for many regions/pollutants.

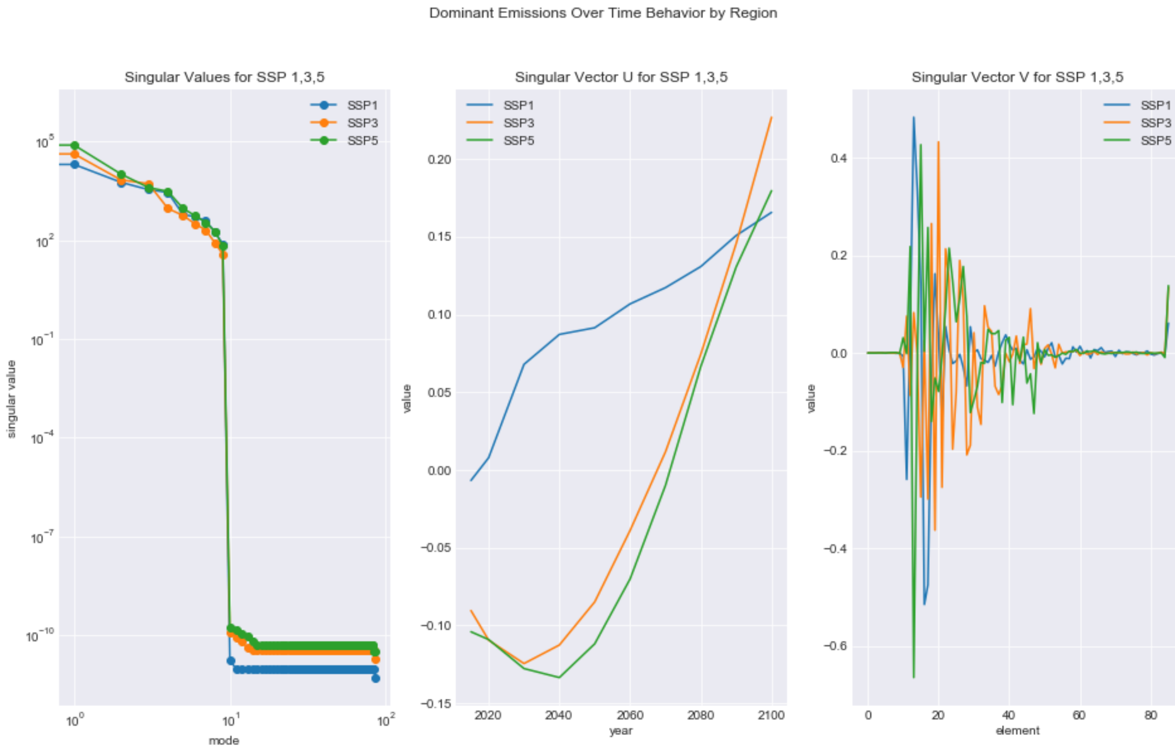


Figure 3: Emissions by Region SVD Data

5.3 Temperature (Figure 4)

As was the case in the emissions data, there is one dominant singular value. This means that the dominant behavior of the temperature can be described by the first mode. When analyzing the temperature, we expect that it would follow a similar pattern to the emissions data. Additionally, the shapes for SSP3 and SSP5 are more likely to look similar as in both of these scenarios there are high challenges to mitigation. It takes time for the temperature to be affected by the emission levels, and this is reflected in our data. For example, examining SSP1 curves in figures 4, the temperature begins to spike down a few years after the emission levels drop.

6 Summary and Conclusions

Using the SVD framework to analyze SSP data is an effective way to compare dominant behavior and trends for different SSP scenarios. It is also a helpful way to reduce the dimensionality of the data so it is easier to process and analyze. The SVD was especially useful for our data since we had one relatively dominant singular value in much of the analysis. As expected, a few of the more relevant pollutants are largely responsible for emissions and temperature changes.

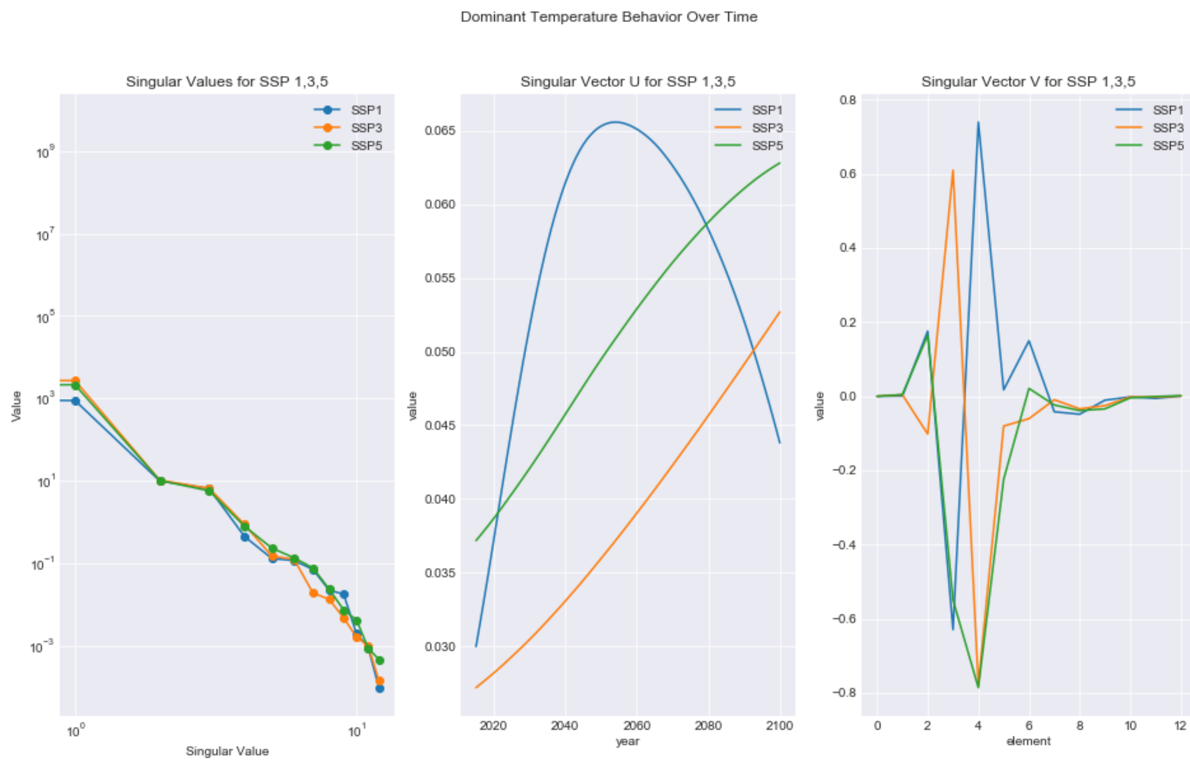


Figure 4: Temperature SVD Data

While in this project we focused mainly on the different pollutants and regions, it would be interesting to see what affects the industry has if we were to hold the rest of the parameters constant. Additionally, using machine learning, it would be interesting to test out different data and see if we could effectively classify that scenario into one of the five SSP narratives. The SVD would be useful as well in this type of classification. It would also be interesting to examine the other two scenarios SSP2 and SSP4 and look at other dynamics.

7 References

Hausfather, Zeke. “Explainer: How ’Shared Socioeconomic Pathways’ Explore Future Climate Change.” Carbon Brief, 21 Feb. 2019, www.carbonbrief.org/explainer-how-shared-socioeconomic-pathways-explore-future-climate-change.

Smith, C. J., Forster, P. M., Allen, M., Leach, N., Millar, R. J., Passerello, G. A., and Regayre, L. A.: FAIR v1.3: A simple emissions-based impulse response and carbon cycle model, Geosci. Model Dev. https://doi.org/10.5194/gmd-2017-266, 2018.

Millar, R. J., Nicholls, Z. R., Friedlingstein, P., and Allen, M. R.: A modified impulse-response representation of the global near-surface air temperature and atmospheric concentration response to carbon dioxide emissions, Atmos. Chem. Phys., 17, 7213-7228, https://doi.org/10.5194/acp-17-7213-2017, 2017.

Appendix A Python Functions

- `pd.read_csv` Reads in a CSV file and converts the data into a Python pandas DataFrame object.
- `.melt()` Can be used to un-pivot data and reshape it in a pandas DataFrame.
- `interpolate()` Uses the data from the CSV file that is for every 10 years and interpolates so there is emissions data for all of the years between based on the data patterns.
- `.shape()` This returns the shape of a pandas DataFrame, or the number of rows and columns
- `.head()` This gives a preview of the first few rows of a pandas DataFrame. You can give it a parameter to change the number of desired rows shown.
- `.mean()` This function calculates the average of a column in pandas DataFrame.
- `u,s,v =n p.linalg.svd(DataFrame, full_matrices = False` This takes in a pandas DataFrame object where all the values are numbers and returns the matrices u and v from the reduced SVD. It also returns a numpy array of the singular values. The False for full matrices denotes the reduced SVD.

Appendix B Jupyter Notebook:

B.1 GITHUB LINKS

Shannon:
https://github.com/shannondow/AMATH-582-Homeworks-2020
Rachel:
https://github.com/rachel-carroll/AMATH582

AMATH 582 Final Project

Rachel Carroll and Shannon Dow

March 19, 2020

Import Packages

In [26]:
```
%matplotlib inline

import pandas as pd
import numpy as np
from scipy import interpolate
import fair
from matplotlib import pyplot as plt

plt.style.use('seaborn-darkgrid')
plt.rcParams['figure.figsize'] = (16, 9)
```

Define functions

In [14]:
```python
def SSP_to_emissions_matrix1(df):
    pollutant_list=df.VARIABLE.unique()
    region_list=df.REGION.unique()
    ssp_list=df.SCENARIO.unique()
    years = np.array(df.YEAR.unique())
    emission_data = pd.DataFrame()
    emission_data['YEAR']=years
    for pollutant in pollutant_list:
        for region in region_list:
            for ssp in ssp_list:
                emission_vec = df.EMISSIONS[(df.VARIABLE==pollutant)&(df.RE
                emission_data[pollutant]= np.array(emission_vec)
    return emission_data

def SSP_to_emissions_matrix(df):
    pollutant_list=df.VARIABLE.unique()
    region_list=df.REGION.unique()
    ssp_list=df.SCENARIO.unique()
    years = np.array(df.YEAR.unique())
    emission_data = pd.DataFrame()
    emission_data['YEAR']=years
    for pollutant in pollutant_list:
        for region in region_list:
            for ssp in ssp_list:
                emission_vec = df.EMISSIONS[(df.VARIABLE==pollutant)&(df.RE
                emission_data[pollutant+'-'+region+'-'+ssp]= np.array(emiss
    return emission_data

def fair_format_emissions(df):
    fair_emissions_data = {'YEAR': df.YEAR,
                           'FossilCO2': df['CO2'],
                           'OtherCO2': 0,
                           'CH4': df['CH4'],
                           'N2O': df['N2O'],
                           'SOx': df['Sulfur'], #?
                           'CO': df['CO'],
                           'NMVOC': df['VOC'], #?
                           'NOx': df['NOx'] ,
                           'BC': df['BC'],
                           'OC': df['OC'],
                           'NH3':df['NH3'],
                           'CF4':  df['CF4'],
                           'C2F6':df['C2F6'],
                           'C6F14': 0,'HFC23': 0, 'HFC32': 0,'HFC43_10': 0,
                           'HFC125': 0,'HFC134a': 0,'HFC143a': 0,'HFC227ea': 0,'
                           'SF6': df['SF6'],
                           'CFC_11': 0,'CFC_12': 0,'CFC_113': 0,
                           'CFC_114': 0,'CFC_115': 0,'CARB_TET': 0,'MCF': 0,'HCF
                           'HCFC_141B': 0,'HCFC_142B': 0,'HALON1211': 0,'HALON12
                           'HALON1301': 0,'HALON2402': 0,'CH3BR': 0,'CH3CL': 0}
    fair_emissions_data = pd.DataFrame(fair_emissions_data)
    return(fair_emissions_data)

def fair_scm_by_pollutant(emissions_df,pollutant_list):
 #   fair_concentration = pd.DataFrame()
 #   fair_forcing = pd.DataFrame()
```

```
    fair_temp = pd.DataFrame()
    for pollutant in pollutant_list:
        pollutant_emissions = emissions_df.copy()
        for column in pollutant_emissions.columns:
            if column != 'YEAR' and column != pollutant:
                pollutant_emissions[column]=0
        pollutant_emissions = pd.DataFrame(pollutant_emissions)
        C, F, T = fair.forward.fair_scm(emissions=pollutant_emissions.value
#        fair_concentration[pollutant]=np.array(C)
#        fair_forcing[pollutant]=np.array(F)
        fair_temp[pollutant]=np.array(T)
        T=None
        F=None
        C=None
    return fair_temp
```

Section 1: Prepare Data

Read in all data from the SSP csv file and interpolate to turn decadal data to annual.

In [7]:
```
#Pull in SSP csv file (Chose Rach or Snoonz path)
#SSP_all_orig = pd.read_csv('/Users/shannondow/Desktop/ClimateModeling/SSP_
SSP_all_orig = pd.read_csv('/Users/rachelcarroll/Documents/ClimateModeling/

#add blank columns for missing years
SSP_all = SSP_all_orig.copy()
for i in range(2016,2100):
        if i % 10 != 0:
            SSP_all[i]=np.nan
SSP_all[1765]=0                     #these loops are not needed for emissions
for i in range(1766,2015):
            SSP_all[i]=np.nan
for i in range(2101,2501):
            SSP_all[i]=np.nan

#melt to unpivot the data so there is a column for year and emissions
SSP_all = SSP_all.melt(id_vars=['MODEL','SCENARIO','REGION','UNIT','VARIABL
#set all year values as integers
SSP_all.YEAR = [int(x) for x in SSP_all.YEAR]

#interpolate
SSP_all=SSP_all.sort_values(by=['REGION', 'UNIT','VARIABLE','SCENARIO','YEA
SSP_all.EMISSIONS=SSP_all.EMISSIONS.interpolate()
```

Select World data split into three SSP scenarios of interest and interpolate to fill in NaN rows.

In [8]:
```
SSP_world = SSP_all[(SSP_all.REGION == 'World')]
SSP_world=SSP_world.sort_values(by=['REGION', 'UNIT','VARIABLE','SCENARIO',
SSP_world['VARIABLE']=SSP_world['VARIABLE'].str.replace('CMIP6 Emissions',
SSP_world['VARIABLE']=SSP_world['VARIABLE'].str.replace('|', '')
SSP_world.loc[SSP_world['VARIABLE'].str.len()<7, 'TOTAL'] = 'True'
SSP_world = SSP_world[SSP_world.TOTAL == 'True']

# #Divide by SSP Scenarios of interest
SSP1_world_total_industry = SSP_world[(SSP_world.SCENARIO=='SSP1-19')]
SSP3_world_total_industry = SSP_world[(SSP_world.SCENARIO=='SSP3-70 (Baseli
SSP5_world_total_industry = SSP_world[(SSP_world.SCENARIO=='SSP5-85 (Baseli
```

Select regional data split into three SSP scenarios of interest and interpolate to fill in NaN rows.

In [9]:
```
SSP_regions = SSP_all[(SSP_all.REGION != 'World')&(SSP_all.YEAR >2014)&(SSP
SSP_regions=SSP_regions.sort_values(by=['REGION', 'VARIABLE','SCENARIO','YE
SSP_regions['VARIABLE']=SSP_regions['VARIABLE'].str.replace('CMIP6 Emission
SSP_regions['VARIABLE']=SSP_regions['VARIABLE'].str.replace('|', '')
SSP_regions.loc[SSP_regions['VARIABLE'].str.len()<7, 'TOTAL'] = 'True'
SSP_regions = SSP_regions[SSP_regions.TOTAL == 'True']
SSP_regions=SSP_regions.sort_values(by=['REGION', 'VARIABLE','SCENARIO','YE

SSP1_region_total_industry = SSP_regions[(SSP_regions.SCENARIO=='SSP1-19')]
SSP3_region_total_industry = SSP_regions[(SSP_regions.SCENARIO=='SSP3-70 (B
SSP5_region_total_industry = SSP_regions[(SSP_regions.SCENARIO=='SSP5-85 (B
```

Format emissions data into vectorized matrices

This function reformats the SSP data so there is a column for every pollutant/scenario/region combination in the dataframe that is fed into the funtion. The output file is in the format that can be SVD'd directly for the emissions analysis

In [15]:
```
SSP1_emission_data=SSP_to_emissions_matrix1(SSP1_world_total_industry)
SSP3_emission_data=SSP_to_emissions_matrix1(SSP3_world_total_industry)
SSP5_emission_data=SSP_to_emissions_matrix1(SSP5_world_total_industry)

SSP1_regions_emission=SSP_to_emissions_matrix(SSP1_region_total_industry)
SSP3_regions_emission=SSP_to_emissions_matrix(SSP3_region_total_industry)
SSP5_regions_emission=SSP_to_emissions_matrix(SSP5_region_total_industry)
```

Format emissions data for the FaIR model

(to match exact format of RCP data files)

In [16]:
```
ssp1_emissions_fair_data=fair_format_emissions(SSP1_emission_data)
ssp3_emissions_fair_data=fair_format_emissions(SSP3_emission_data)
ssp5_emissions_fair_data=fair_format_emissions(SSP5_emission_data)
```

Format emissions data for the FaIR model

In [18]:

```
# Run the FaIR model on the entire datasets
C_SSP1, F_SSP1, T_SSP1 = fair.forward.fair_scm(emissions=ssp1_emissions_fai
C_SSP3, F_SSP3, T_SSP3 = fair.forward.fair_scm(emissions=ssp3_emissions_fai
C_SSP5, F_SSP5, T_SSP5 = fair.forward.fair_scm(emissions=ssp5_emissions_fai
# Run the FaIR model one pollutant at a time
col_list=['FossilCO2', 'CH4', 'N2O', 'SOx', 'CO', 'NMVOC', 'NOx', 'BC', 'OC

SSP1_Temp=fair_scm_by_pollutant(ssp1_emissions_fair_data,col_list)
SSP3_Temp=fair_scm_by_pollutant(ssp3_emissions_fair_data,col_list)
SSP5_Temp=fair_scm_by_pollutant(ssp5_emissions_fair_data,col_list)
```

Section 2: Emissions - Singular Value Decmposition by Pollutant

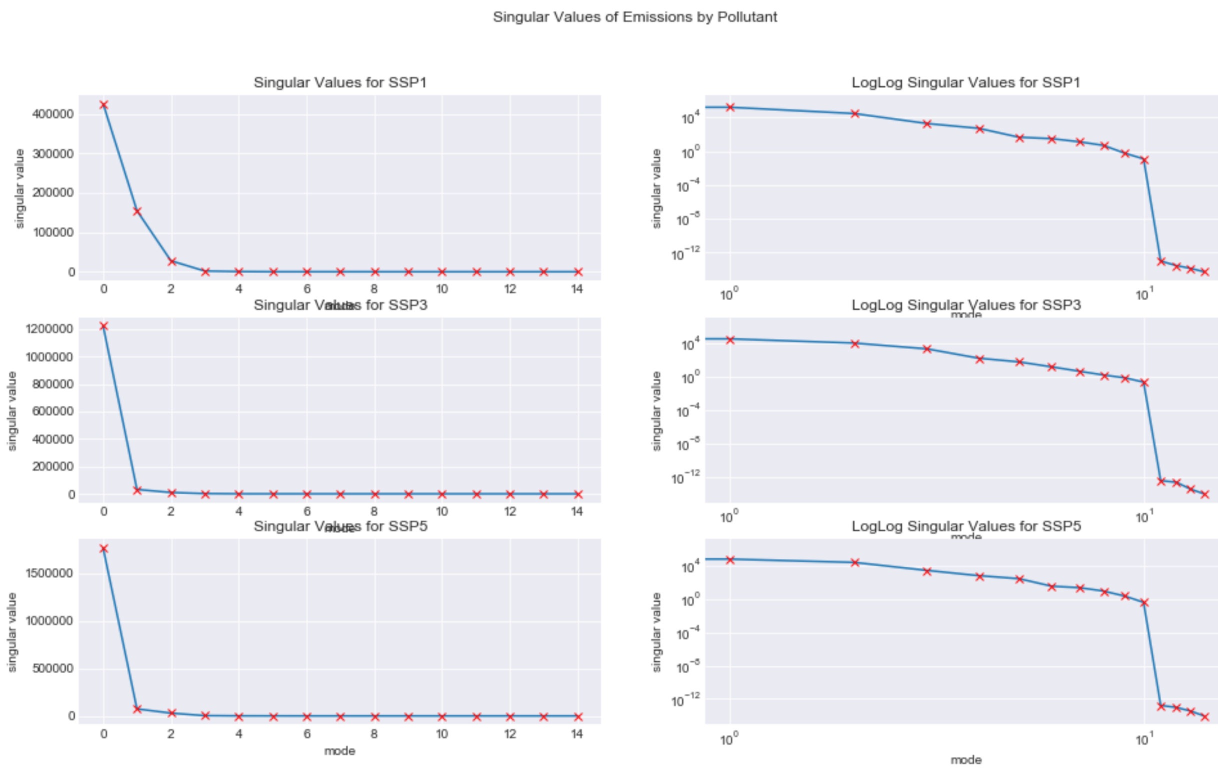
In [21]:

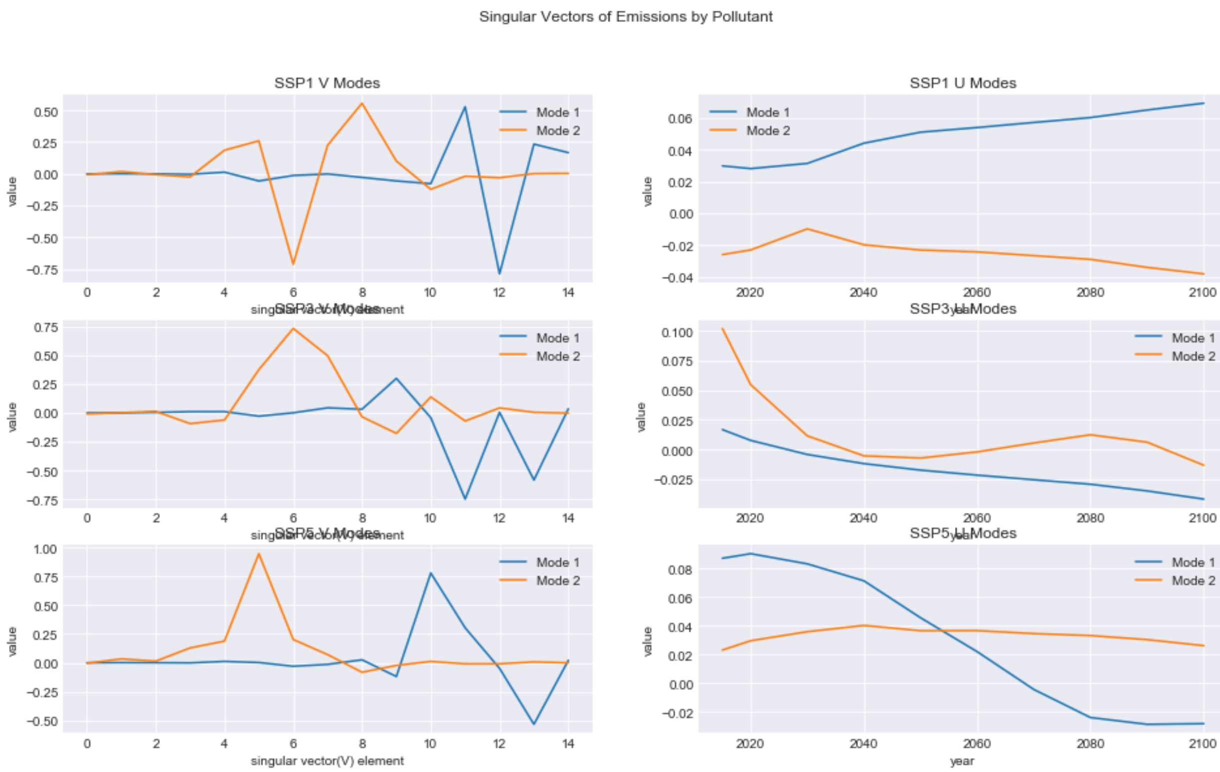
```
start_year=2015
end_year=2100
years=np.array(range(736))+1765
#create index for start and end year
end_indx=int(np.where(ssp1_emissions_fair_data.YEAR.values==end_year)[0])+1
start_indx=int(np.where(ssp1_emissions_fair_data.YEAR.values==start_year)[0

u1, s1, v1 = np.linalg.svd(SSP1_emission_data, full_matrices=False)
u3, s3, v3 = np.linalg.svd(SSP3_emission_data, full_matrices=False)
u5, s5, v5 = np.linalg.svd(SSP5_emission_data, full_matrices=False)
```

In [27]:
```
plt.figure(1)
plt.subplot(321)
plt.plot(s1)
plt.plot(s1,'rx')
plt.title('Singular Values for SSP1')
plt.xlabel('mode')
plt.ylabel('singular value')
plt.subplot(322)
plt.loglog(s1)
plt.loglog(s1,'rx')
plt.title('LogLog Singular Values for SSP1')
plt.xlabel('mode')
plt.ylabel('singular value')

plt.subplot(323)
plt.plot(s3)
plt.plot(s3,'rx')
plt.title('Singular Values for SSP3')
plt.xlabel('mode');plt.ylabel('singular value')
plt.subplot(324)
plt.loglog(s3)
plt.loglog(s3,'rx')
plt.title('LogLog Singular Values for SSP3')
plt.xlabel('mode')
plt.ylabel('singular value')

plt.subplot(325)
plt.plot(s5)
plt.plot(s5,'rx')
plt.title('Singular Values for SSP5')
plt.xlabel('mode')
plt.ylabel('singular value')
plt.subplot(326)
plt.loglog(s5)
plt.loglog(s5,'rx')
plt.title('LogLog Singular Values for SSP5')
plt.xlabel('mode')
plt.ylabel('singular value')
plt.suptitle('Singular Values of Emissions by Pollutant')

plt.figure(2)
plt.subplot(321)
plt.plot(v1[:,1],label='Mode 1')
plt.plot(v1[:,2],label='Mode 2')
plt.title('SSP1 V Modes');plt.xlabel('singular vector(V) element');plt.ylab
plt.subplot(322)
plt.plot(years[start_indx:end_indx],u1[start_indx:end_indx:,1],label='Mode
plt.plot(years[start_indx:end_indx],u1[start_indx:end_indx:,2],label='Mode
plt.title('SSP1 U Modes');plt.xlabel('year');plt.ylabel('value');plt.legend

plt.subplot(323)
plt.plot(v3[:,1],label='Mode 1')
plt.plot(v3[:,2],label='Mode 2')
plt.title('SSP3 V Modes');plt.xlabel('singular vector(V) element');plt.ylab
plt.subplot(324)
plt.plot(years[start_indx:end_indx],u3[start_indx:end_indx:,1],label='Mode
```

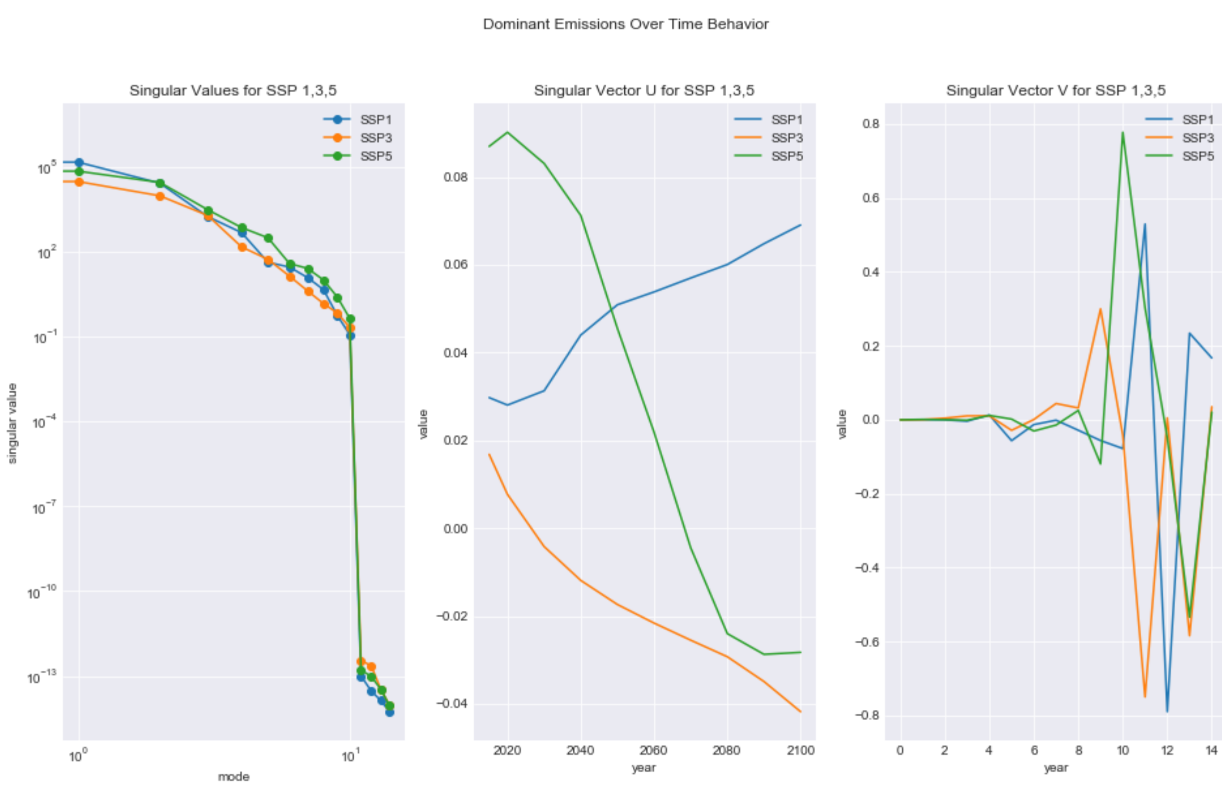
```
plt.plot(years[start_indx:end_indx],u3[start_indx:end_indx:,2],label='Mode 
plt.title('SSP3 U Modes');plt.xlabel('year');plt.ylabel('value');plt.legend

plt.subplot(325)
plt.plot(v5[:,1],label='Mode 1')
plt.plot(v5[:,2],label='Mode 2')
plt.title('SSP5 V Modes');plt.xlabel('singular vector(V) element');plt.ylab
plt.subplot(326)
plt.plot(years[start_indx:end_indx],u5[start_indx:end_indx:,1],label='Mode 
plt.plot(years[start_indx:end_indx],u5[start_indx:end_indx:,2],label='Mode 
plt.title('SSP5 U Modes');plt.xlabel('year');plt.ylabel('value');plt.legend
plt.suptitle('Singular Vectors of Emissions by Pollutant')
```

Out[27]: Text(0.5, 0.98, 'Singular Vectors of Emissions by Pollutant')




Singular Vectors of Emissions by Pollutant
SSP1 V Modes
SSP1 U Modes
SSP3 V Modes
SSP3 U Modes
SSP5 V Modes
SSP5 U Modes
Mode 1
Mode 2
value
singular vector(V) element
year

In [32]:
```
plt.subplot(131)
plt.loglog(s1,'o-',label = 'SSP1')
plt.loglog(s3,'o-',label = 'SSP3')
plt.loglog(s5,'o-',label = 'SSP5')
plt.title('Singular Values for SSP 1,3,5')
plt.xlabel('mode')
plt.ylabel('singular value')
plt.legend()

plt.subplot(132)
plt.plot(years[start_indx:end_indx], u1[start_indx:end_indx,1],label = 'SSP
plt.plot(years[start_indx:end_indx], u3[start_indx:end_indx,1],label = 'SSP
plt.plot(years[start_indx:end_indx], u5[start_indx:end_indx,1],label = 'SSP
plt.title('Singular Vector U for SSP 1,3,5')
plt.xlabel('year')
plt.ylabel('value')
plt.legend()

plt.subplot(133)
plt.plot(v1[:,1],label = 'SSP1')
plt.plot(v3[:,1],label = 'SSP3')
plt.plot(v5[:,1],label = 'SSP5')
plt.title('Singular Vector V for SSP 1,3,5')
plt.xlabel('year')
plt.ylabel('value')
plt.legend()

plt.suptitle("Dominant Emissions Over Time Behavior")

plt.savefig('dom_emissions.png')
```



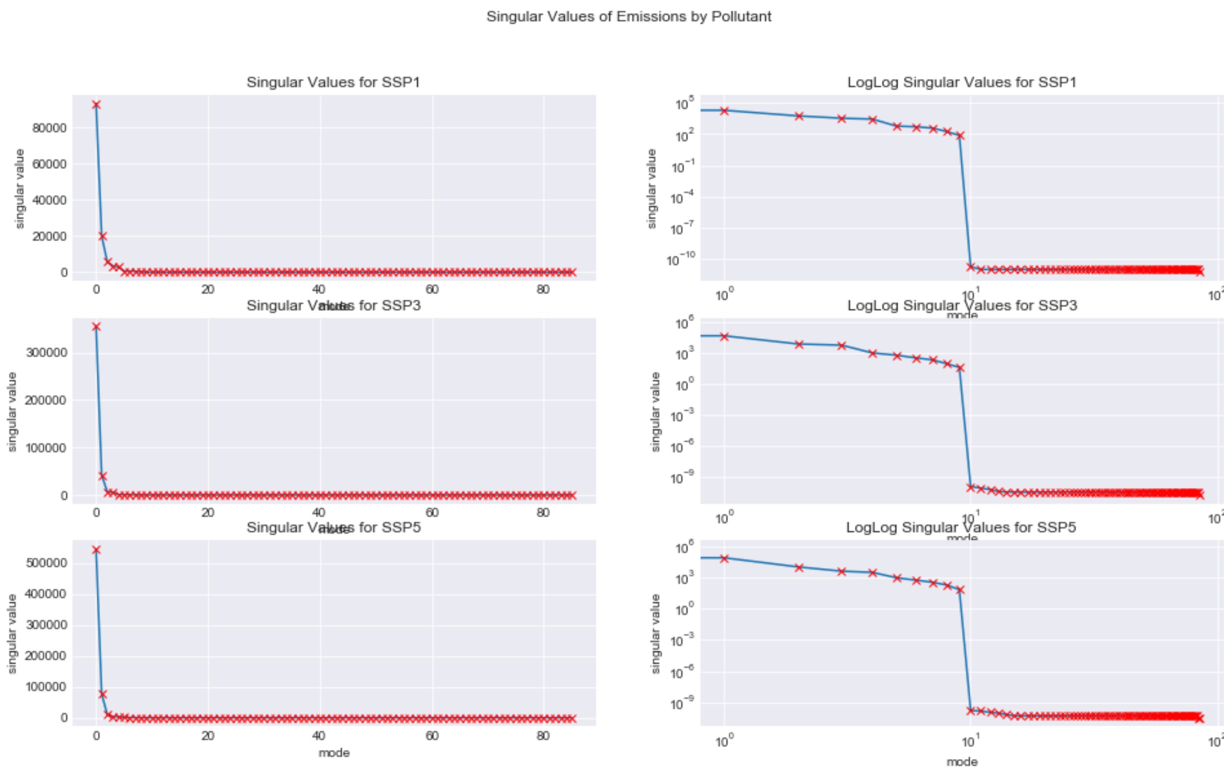
Section 3: Emissions - Singular Value Decmposition by Pollutant and

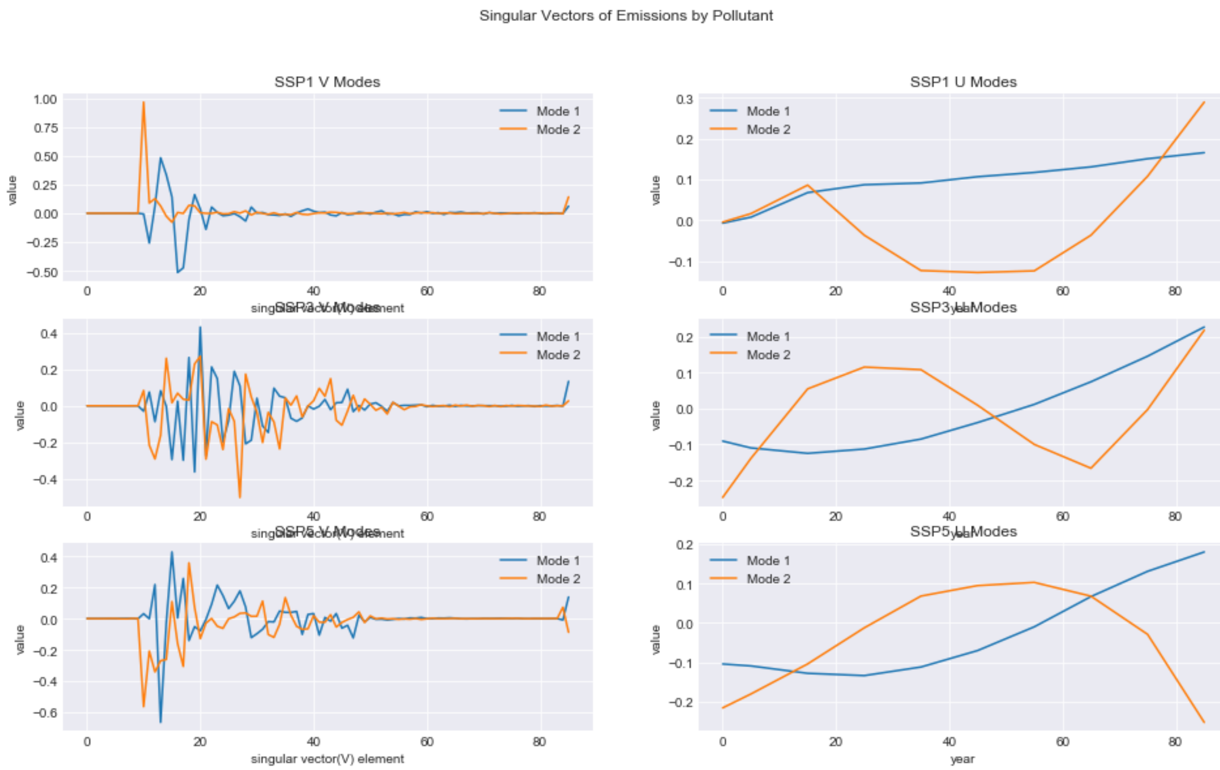
Region

In [33]:
```
ur1, sr1, vr1 = np.linalg.svd(SSP1_regions_emission, full_matrices=False)
ur3, sr3, vr3 = np.linalg.svd(SSP3_regions_emission, full_matrices=False)
ur5, sr5, vr5 = np.linalg.svd(SSP5_regions_emission, full_matrices=False)
```

In [35]:
```
plt.figure(1)
plt.subplot(321)
plt.plot(sr1)
plt.plot(sr1,'rx')
plt.title('Singular Values for SSP1')
plt.xlabel('mode')
plt.ylabel('singular value')
plt.subplot(322)
plt.loglog(sr1)
plt.loglog(sr1,'rx')
plt.title('LogLog Singular Values for SSP1')
plt.xlabel('mode')
plt.ylabel('singular value')

plt.subplot(323)
plt.plot(sr3)
plt.plot(sr3,'rx')
plt.title('Singular Values for SSP3')
plt.xlabel('mode');plt.ylabel('singular value')
plt.subplot(324)
plt.loglog(sr3)
plt.loglog(sr3,'rx')
plt.title('LogLog Singular Values for SSP3')
plt.xlabel('mode')
plt.ylabel('singular value')

plt.subplot(325)
plt.plot(sr5)
plt.plot(sr5,'rx')
plt.title('Singular Values for SSP5')
plt.xlabel('mode')
plt.ylabel('singular value')
plt.subplot(326)
plt.loglog(sr5)
plt.loglog(sr5,'rx')
plt.title('LogLog Singular Values for SSP5')
plt.xlabel('mode')
plt.ylabel('singular value')
plt.suptitle('Singular Values of Emissions by Pollutant')

plt.figure(2)
plt.subplot(321)
plt.plot(vr1[:,1],label='Mode 1')
plt.plot(vr1[:,2],label='Mode 2')
plt.title('SSP1 V Modes');plt.xlabel('singular vector(V) element');plt.ylab
plt.subplot(322)
plt.plot(ur1[:,1],label='Mode 1')
plt.plot(ur1[:,2],label='Mode 2')
plt.title('SSP1 U Modes');plt.xlabel('year');plt.ylabel('value');plt.legend

plt.subplot(323)
plt.plot(vr3[:,1],label='Mode 1')
plt.plot(vr3[:,2],label='Mode 2')
plt.title('SSP3 V Modes');plt.xlabel('singular vector(V) element');plt.ylab
plt.subplot(324)
plt.plot(ur3[:,1],label='Mode 1')
```

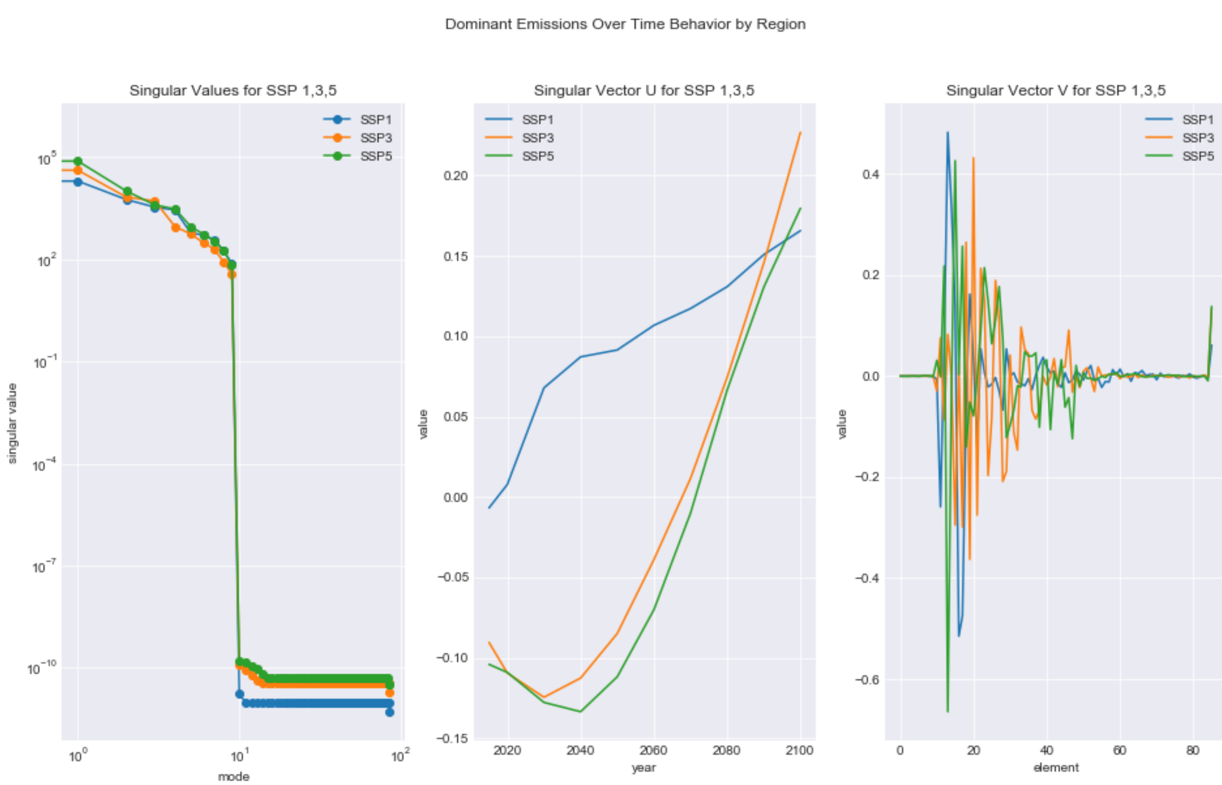
```
plt.plot(ur3[:,2],label='Mode 2')
plt.title('SSP3 U Modes');plt.xlabel('year');plt.ylabel('value');plt.legend

plt.subplot(325)
plt.plot(vr5[:,1],label='Mode 1')
plt.plot(vr5[:,2],label='Mode 2')
plt.title('SSP5 V Modes');plt.xlabel('singular vector(V) element');plt.ylab
plt.subplot(326)
plt.plot(ur5[:,1],label='Mode 1')
plt.plot(ur5[:,2],label='Mode 2')
plt.title('SSP5 U Modes');plt.xlabel('year');plt.ylabel('value');plt.legend
plt.suptitle('Singular Vectors of Emissions by Pollutant')

plt.savefig('dom_region_emissions.png')
```





Singular Vectors of Emissions by Pollutant
SSP1 V Modes
SSP1 U Modes
SSP3 V Modes
SSP3 U Modes
SSP5 V Modes
SSP5 U Modes
Mode 1
Mode 2
value
singular vector(V) element
year

In [39]:
```python
plt.subplot(131)
plt.loglog(sr1,'o-',label = 'SSP1')
plt.loglog(sr3,'o-',label = 'SSP3')
plt.loglog(sr5,'o-',label = 'SSP5')
plt.title('Singular Values for SSP 1,3,5')
plt.xlabel('mode')
plt.ylabel('singular value')
plt.legend()

plt.subplot(132)
plt.plot(years[start_indx:end_indx],ur1[:,1],label = 'SSP1')
plt.plot(years[start_indx:end_indx],ur3[:,1],label = 'SSP3')
plt.plot(years[start_indx:end_indx],ur5[:,1],label = 'SSP5')
plt.title('Singular Vector U for SSP 1,3,5')
plt.xlabel('year')
plt.ylabel('value')
plt.legend()

plt.subplot(133)
plt.plot(vr1[:,1],label = 'SSP1')
plt.plot(vr3[:,1],label = 'SSP3')
plt.plot(vr5[:,1],label = 'SSP5')
plt.title('Singular Vector V for SSP 1,3,5')
plt.xlabel('element')
plt.ylabel('value')
plt.legend()

plt.suptitle("Dominant Emissions Over Time Behavior by Region")

plt.savefig('dom_region_emissions2.png')
```



Section 4: Temperature - Singular Value Decomposition by

Pollutant

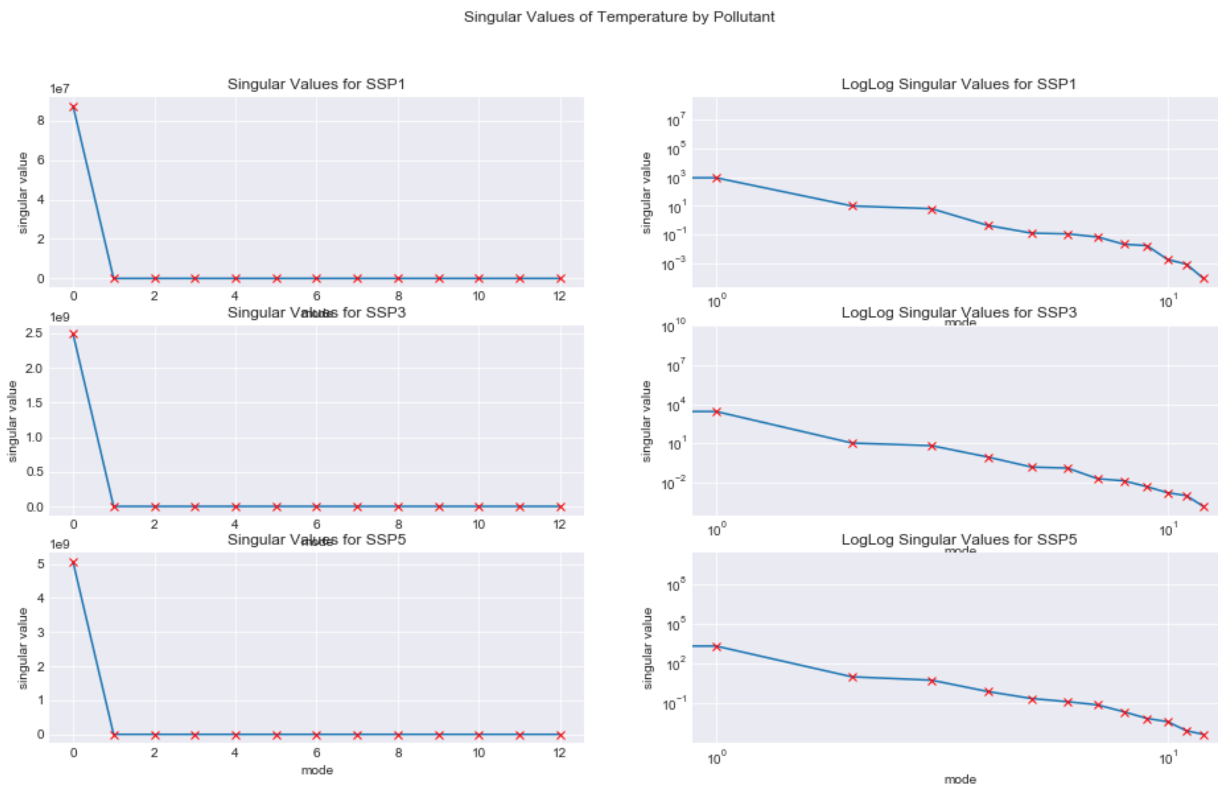
```
ut1, st1, vt1 = np.linalg.svd(SSP1_Temp, full_matrices=False)
ut3, st3, vt3 = np.linalg.svd(SSP3_Temp, full_matrices=False)
ut5, st5, vt5 = np.linalg.svd(SSP5_Temp, full_matrices=False)
```

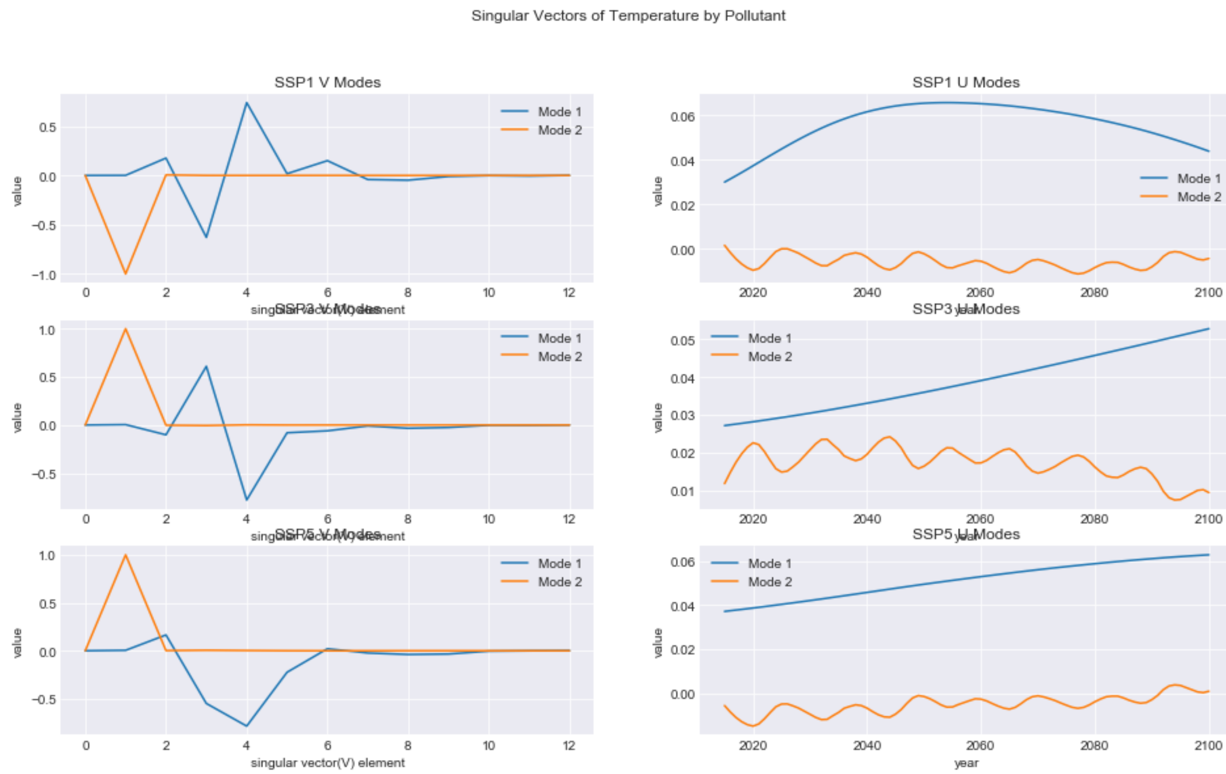
In [41]:

```
plt.figure(1)
plt.subplot(321)
plt.plot(st1)
plt.plot(st1,'rx')
plt.title('Singular Values for SSP1')
plt.xlabel('mode')
plt.ylabel('singular value')
plt.subplot(322)
plt.loglog(st1)
plt.loglog(st1,'rx')
plt.title('LogLog Singular Values for SSP1')
plt.xlabel('mode')
plt.ylabel('singular value')

plt.subplot(323)
plt.plot(st3)
plt.plot(st3,'rx')
plt.title('Singular Values for SSP3')
plt.xlabel('mode')
plt.ylabel('singular value')
plt.subplot(324)
plt.loglog(st3)
plt.loglog(st3,'rx')
plt.title('LogLog Singular Values for SSP3')
plt.xlabel('mode')
plt.ylabel('singular value')

plt.subplot(325)
plt.plot(st5)
plt.plot(st5,'rx')
plt.title('Singular Values for SSP5')
plt.xlabel('mode')
plt.ylabel('singular value')
plt.subplot(326)
plt.loglog(st5)
plt.loglog(st5,'rx')
plt.title('LogLog Singular Values for SSP5')
plt.xlabel('mode')
plt.ylabel('singular value')
plt.suptitle('Singular Values of Temperature by Pollutant')

plt.figure(2)
plt.subplot(321)
plt.plot(vt1[:,1],label='Mode 1')
plt.plot(vt1[:,2],label='Mode 2')
plt.title('SSP1 V Modes');plt.xlabel('singular vector(V) element');plt.ylab
plt.subplot(322)
plt.plot(years[start_indx:end_indx],ut1[start_indx:end_indx:,1],label='Mode
plt.plot(years[start_indx:end_indx],ut1[start_indx:end_indx:,2],label='Mode
plt.title('SSP1 U Modes');plt.xlabel('year');plt.ylabel('value');plt.legend

plt.subplot(323)
plt.plot(vt3[:,1],label='Mode 1')
plt.plot(vt3[:,2],label='Mode 2')
plt.title('SSP3 V Modes');plt.xlabel('singular vector(V) element');plt.ylab
plt.subplot(324)
```

```
plt.plot(years[start_indx:end_indx],ut3[start_indx:end_indx:,1],label='Mode
plt.plot(years[start_indx:end_indx],ut3[start_indx:end_indx:,2],label='Mode
plt.title('SSP3 U Modes');plt.xlabel('year');plt.ylabel('value');plt.legend

plt.subplot(325)
plt.plot(vt5[:,1],label='Mode 1')
plt.plot(vt5[:,2],label='Mode 2')
plt.title('SSP5 V Modes');plt.xlabel('singular vector(V) element');plt.ylab
plt.subplot(326)
plt.plot(years[start_indx:end_indx],ut5[start_indx:end_indx:,1],label='Mode
plt.plot(years[start_indx:end_indx],ut5[start_indx:end_indx:,2],label='Mode
plt.title('SSP5 U Modes');plt.xlabel('year');plt.ylabel('value');plt.legend
plt.suptitle("Singular Vectors of Temperature by Pollutant")
```

Out[41]: Text(0.5, 0.98, 'Singular Vectors of Temperature by Pollutant')




Singular Vectors of Temperature by Pollutant
SSP1 V Modes
SSP1 U Modes
SSP3 V Modes
SSP3 U Modes
SSP5 V Modes
SSP5 U Modes
Mode 1
Mode 2
value
singular vector(V) element
year

In [47]:

```
#dominant modes:
plt.subplot(131)
plt.loglog(st1,'o-',label = 'SSP1')
plt.loglog(st3,'o-',label = 'SSP3')
plt.loglog(st5,'o-',label = 'SSP5')
plt.title('Singular Values for SSP 1,3,5')
plt.xlabel('Singular Value')
plt.ylabel('Value')
plt.legend()
plt.suptitle("Dominant Temperature Over Time Behavior")

plt.subplot(132)
plt.plot(years[start_indx:end_indx], ut1[start_indx:end_indx,1],label = 'SS
plt.plot(years[start_indx:end_indx], ut3[start_indx:end_indx,1],label = 'SS
plt.plot(years[start_indx:end_indx], ut5[start_indx:end_indx,1],label = 'SS
plt.title('Singular Vector U for SSP 1,3,5')
plt.xlabel('year')
plt.ylabel('value')
plt.legend()

plt.subplot(133)
plt.plot(vt1[:,1],label = 'SSP1')
plt.plot(vt3[:,1],label = 'SSP3')
plt.plot(vt5[:,1],label = 'SSP5')
plt.title('Singular Vector V for SSP 1,3,5')
plt.xlabel('element')
plt.ylabel('value')
plt.legend()

plt.suptitle("Dominant Temperature Behavior Over Time")

plt.savefig('dom_temp.png')
```