```
#####################################################################################
 Here is the code for the Bayesian isotope mixing model used in this study including source signature
calculations.

import arviz as az
import corner
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats as stats
import pandas as pd
import pymc as pm
import pymc.sampling_jax
import jaxlib

###read in data here
obs_rawdata_df = pd.read_excel('###')
obs_data_df = obs_rawdata_df.set_index('sampleID')
obs_data_df

#Configurations
NUM_MCMC_CHAINS = 10
LENGTH_MCMC_CHAIN = 40000
BURNIN_LENGTH = int(0.25*LENGTH_MCMC_CHAIN) # discard this number of initial steps while
calculating the statisics of f

all_factor_marginal_stats_df_arr = []
modeled_val_df_arr = []
samples_closet_to_median_in_5D = []
idata_arr = []

obs_data_for_pymc_df = obs_data_df.iloc[0:10]
d34S = obs_data_df['d34S']
d18O = obs_data_df['O18']
D17O = obs_data_df['O17']
SOR = obs_data_df['SOR']
temp = obs_data_df['CTC_temp_3m_C']

### d18O(SO4) source signatures
# gas-phase chemistry
d18O_H2O_vapor = (temp*0.403)-25.946
d18O_OH = (0.71*d18O_H2O_vapor)+16.5

#aqueous-phase chemistry
d18O_H2O_liquid = (temp*0.4315) - 17.224
d18O_SIV = (d18O_H2O_liquid*0.9447) + 7.7837

d18O_NO2 = (d18O_SIV*0.75) + (d18O_H2O_liquid*0.25)
```

```
d18O_TMI = (d18O_SIV*0.75) + (23.5*0.25) #d18O(O2) = 23.5
d18O_H2O2 = (d18O_SIV*0.5) + (35*0.5) #d18O(H2O2) = 35
d18O_HOOH = (d18O_SIV*0.5) + (23.5*0.5) #d18O(O2(aq))= 23.5
d18O_O3 = (d18O_SIV*0.75) + (130*0.25) #d18O(O3) = 130
d18O_primary = 23.5

### D17O(SO4) source signatures
D17O_primary = -0.34 #Unit: per mille
D17O_O3 = 9.8 #Unit: per mille
D17O_H2O2 =   0.81
D17O_HOOH =   -0.17
D17O_OH = 0
D17O_TMI = -0.085
D17O_NO2 = 0
### d34S(SO4) source signatures
#reference Harris et al. 2012 and 2013
# e = epsilon values

e_OH = (-0.004*temp) + 10.6 + 1
e_H2O2 = (-0.085*temp) + 16.51 + 1
e_O3 = (-0.085*temp) + 16.51 + 1
e_TMI = (-0.237*temp) + -5.039 +1
e_NO2 = 1
d34S_SO4_primary = 4.7
d34S_emission = 4.7

obs_data_for_pymc_df = obs_data_df.iloc[0:63]
for i_sample_id in obs_data_for_pymc_df.index:
  i_d18O = obs_data_df.loc[i_sample_id, 'O18']
  i_D17O = obs_data_df.loc[i_sample_id, 'O17']
  i_d34S =obs_data_df.loc[i_sample_id, 'd34S']
  i_SO4_obs = obs_data_df.loc[i_sample_id, 'SO4_umol']
  i_SO2_obs = obs_data_df.loc[i_sample_id, 'SO2_umol']
  i_e_TMI = obs_data_df.loc[i_sample_id, 'e_TMI']
  i_e_O3 = obs_data_df.loc[i_sample_id, 'e_O3']
  i_e_H2O2 = obs_data_df.loc[i_sample_id, 'e_H2O2']
  i_e_OH = obs_data_df.loc[i_sample_id, 'e_OH']
  i_d18O_TMI = obs_data_df.loc[i_sample_id, 'd18O_TMI']
  i_d18O_O3 = obs_data_df.loc[i_sample_id, 'd18O_O3']
  i_d18O_H2O2 = obs_data_df.loc[i_sample_id, 'd18O_H2O2']
  i_d18O_OH = obs_data_df.loc[i_sample_id, 'd18O_OH']
  i_d18O_NO2 = obs_data_df.loc[i_sample_id, 'd18O_NO2']
  i_obs = np.array([i_D17O, i_d18O, i_d34S]) # order: D17O, d18O

  with pm.Model() as model:
    # this is where you weigh the importance of different pathways.
    #Its useful when two signatures are very similar
    #but you have good reason to believe one is more dominant than another
```

```
    #(pH, photolysis rates, O3 obs, HYSPLIT, etc)
    f = pm.Dirichlet('f', a=[1,1,1,1,1,1]) # Using equal weight prior

    f_primary = pm.Deterministic("f_primary", f[0])
    f_OH = pm.Deterministic("f_OH", f[1])
    #f_HOOH_pm = pm.Deterministic("f_HOOH_pm", f[2])
    f_H2O2 = pm.Deterministic("f_H2O2", f[2])
    f_TMI = pm.Deterministic("f_TMI", f[3])
    f_O3 = pm.Deterministic("f_O3", f[4])
    f_NO2 = pm.Deterministic("f_NO2", f[5])

    # Statistics of the observations
    ###these are the analytical errors // can also expand to get a larger range of uncertainty
    obs_D17O_std = 0.16 #Unit: per mille
    obs_d18O_std =  0.8 #Unit: per mille
    obs_d34S_std = 1.0
    obs_D17O_var = obs_D17O_std**2
    obs_d18O_var = obs_d18O_std**2
    obs_d34S_var= obs_d34S_std**2
    obs_D17O_d18O_correlation = 0.0
    obs_D17O_d34S_correlation = 0.0
    obs_d18O_d34S_correlation = 0.0

    obs_D17O_d18O_cov = obs_D17O_std*obs_d18O_std*obs_D17O_d18O_correlation
    obs_D17O_d34S_cov = obs_D17O_std*obs_d34S_std* obs_D17O_d34S_correlation
    obs_d18O_d34S_cov = obs_d18O_std*obs_d34S_std*obs_d18O_d34S_correlation

    # Main dish :-)
    model_D17O = pm.Deterministic("model_D17O", f_primary*D17O_primary + f_O3*D17O_O3 +
f_H2O2*D17O_H2O2 + f_OH*D17O_OH + f_NO2*D17O_NO2 +f_TMI*D17O_TMI )
    model_d18O = pm.Deterministic("model_d18O", f_primary*d18O_primary + f_O3*i_d18O_O3 +
f_H2O2*i_d18O_H2O2 + f_OH*i_d18O_OH +f_NO2*i_d18O_NO2 + f_TMI*i_d18O_TMI)
    ###we calculate SOR_2nd since the fraction of primary sulfate is large and independent of chemical
fractionation
    ###this is not necessary in secondary sulfate-dominated regimes
    SO4_secondary_umol = i_SO4_obs*(1-f_primary)
    SOR_2nd = SO4_secondary_umol  / (SO4_secondary_umol + i_SO2_obs)
    model_d34S_epsilon = pm.Deterministic("model_d34S_epsilon", f_primary*4.7 + (1-f_primary)*(-
1*((d34S_emission- (i_e_O3*f_O3/(1-f_primary)+ (i_e_H2O2*f_H2O2/(1-f_primary))+
(i_e_OH*f_OH/(1-f_primary)) + (e_NO2*f_NO2/(1-f_primary))+ (i_e_TMI*f_TMI/(1-
f_primary))*(np.log(SOR_2nd)*((1-SOR_2nd)/SOR_2nd)))))))

    # Likelihood function
    obs = pm.MvNormal("obs", observed = i_obs,
                    mu=[model_D17O, model_d18O,model_d34S_epsilon],
                    cov=np.array([[obs_D17O_var,obs_D17O_d18O_cov,obs_D17O_d34S_cov],
                            [obs_D17O_d18O_cov,obs_d18O_var,obs_d18O_d34S_cov],
                            [obs_D17O_d34S_cov,obs_d18O_d34S_cov,obs_d34S_var]]))
```

```
    idata_Dirichlet = pm.sampling_jax.sample_numpyro_nuts(draws=LENGTH_MCMC_CHAIN,
                                          chains=NUM_MCMC_CHAINS,
                                          tune=4000)

  # Create the corner plot
  corner.corner(idata_Dirichlet, var_names=['f_primary', 'f_OH', 'f_H2O2', 'f_TMI', 'f_O3', 'f_NO2'])
  plt.suptitle("{}'s corner plot".format(i_sample_id), fontsize=30)
  plt.show()

  # Save the statistics for each sample for later

  i_all_factor_df = idata_Dirichlet['posterior']['f'].isel(draw=slice(BURNIN_LENGTH,
None)).stack({'sample':['chain','draw']}).to_dataframe()['f'].unstack('f_dim_0')
  i_all_factor_df.rows = ['f_primary', 'f_OH', 'f_H2O2', 'f_TMI', 'f_O3', 'f_NO2']
  i_all_factor_marginal_stats_df = pd.DataFrame()

  for i_factor in i_all_factor_df.columns:

    i_all_factor_marginal_stats_df.loc['2.5th percentile',i_factor] =
i_all_factor_df[i_factor].quantile(0.025)
    i_all_factor_marginal_stats_df.loc['25th percentile',i_factor]  = i_all_factor_df[i_factor].quantile(0.25)
    i_all_factor_marginal_stats_df.loc['Median',i_factor]  = i_all_factor_df[i_factor].quantile(0.5)
    i_all_factor_marginal_stats_df.loc['75th percentile',i_factor]  = i_all_factor_df[i_factor].quantile(0.75)
    i_all_factor_marginal_stats_df.loc['97.5th percentile',i_factor]  =
i_all_factor_df[i_factor].quantile(0.975)

  all_factor_marginal_stats_df_arr += [i_all_factor_marginal_stats_df]

  i_model_vals_df = idata_Dirichlet.posterior[['model_D17O', 'model_d18O',
'model_d34S_epsilon']].isel(draw=slice(BURNIN_LENGTH,
None)).stack({'sample':['chain','draw']}).to_dataframe()
  i_modeled_val_stats_df = pd.DataFrame()

  for i_model_val_type in ['model_D17O', 'model_d18O', 'model_d34S_epsilon']:

    i_modeled_val_stats_df.loc['2.5th percentile',i_model_val_type] =
i_model_vals_df[i_model_val_type].quantile(0.025)
    i_modeled_val_stats_df.loc['25th percentile',i_model_val_type]  =
i_model_vals_df[i_model_val_type].quantile(0.25)
    i_modeled_val_stats_df.loc['Median',i_model_val_type]  =
i_model_vals_df[i_model_val_type].quantile(0.5)
    i_modeled_val_stats_df.loc['75th percentile',i_model_val_type]  =
i_model_vals_df[i_model_val_type].quantile(0.75)
    i_modeled_val_stats_df.loc['97.5th percentile',i_model_val_type]  =
i_model_vals_df[i_model_val_type].quantile(0.975)

  modeled_val_df_arr += [i_modeled_val_stats_df]
```

```
  idata_arr += [idata_Dirichlet]

  ################################################################################
  #### Output statistics
  all_factor_marginal_stats_df = pd.concat(all_factor_marginal_stats_df_arr, axis=0,
keys=obs_data_for_pymc_df.index)
modeled_vals_df = pd.concat(modeled_val_df_arr, axis=0, keys=obs_data_for_pymc_df.index)
modeled_D17O_stats_df = modeled_vals_df.loc[:,'model_D17O'].unstack()
modeled_d18O_stats_df = modeled_vals_df.loc[:,'model_d18O'].unstack()
modeled_d34S_stats_df = modeled_vals_df.loc[:, 'model_d34S_epsilon'].unstack()

all_factor_marginal_stats_df = pd.concat(all_factor_marginal_stats_df_arr, axis=0,
keys=obs_data_for_pymc_df.index)
modeled_vals_df = pd.concat(modeled_val_df_arr, axis=0, keys=obs_data_for_pymc_df.index)
modeled_D17O_stats_df = modeled_vals_df.loc[:,'model_D17O'].unstack()
modeled_d18O_stats_df = modeled_vals_df.loc[:, 'model_d18O'].unstack()
modeled_d34S_stats_df = modeled_vals_df.loc[:,'model_d34S_epsilon'].unstack()

f_primary_marginal_stats_df = all_factor_marginal_stats_df.loc[:,0].unstack()
f_OH_marginal_stats_df = all_factor_marginal_stats_df.loc[:,1].unstack()
f_HOOH_marginal_stats_df = all_factor_marginal_stats_df.loc[:,2].unstack()
f_TMI_marginal_stats_df = all_factor_marginal_stats_df.loc[:,3].unstack()
f_O3_marginal_stats_df = all_factor_marginal_stats_df.loc[:,4].unstack()
f_NO2_marginal_stats_df = all_factor_marginal_stats_df.loc[:,5].unstack()

f_primary_marginal_stats_df = all_factor_marginal_stats_df.loc[:,0].unstack()
Primary = f_primary_marginal_stats_df['97.5th percentile']
print(*Primary)

f_OH_marginal_stats_df = all_factor_marginal_stats_df.loc[:,1].unstack()
OH = f_OH_marginal_stats_df['97.5th percentile']
print(*OH)

f_H2O2_marginal_stats_df = all_factor_marginal_stats_df.loc[:,2].unstack()
H2O2 = f_H2O2_marginal_stats_df['97.5th percentile']
print(*H2O2)

f_TMI_marginal_stats_df = all_factor_marginal_stats_df.loc[:,3].unstack()
TMI = f_TMI_marginal_stats_df['97.5th percentile']
print(*TMI)

f_O3_marginal_stats_df = all_factor_marginal_stats_df.loc[:,4].unstack()
O3 = f_O3_marginal_stats_df['97.5th percentile']
print(*O3)

f_NO2_marginal_stats_df = all_factor_marginal_stats_df.loc[:,5].unstack()
NO2 = f_NO2_marginal_stats_df['97.5th percentile']
```

```
print(*NO2)


f_primary_marginal_stats_df = all_factor_marginal_stats_df.loc[:,0].unstack()
Primary = f_primary_marginal_stats_df['2.5th percentile']
print(*Primary)

f_OH_marginal_stats_df = all_factor_marginal_stats_df.loc[:,1].unstack()
OH = f_OH_marginal_stats_df['2.5th percentile']
print(*OH)

f_H2O2_marginal_stats_df = all_factor_marginal_stats_df.loc[:,2].unstack()
H2O2 = f_H2O2_marginal_stats_df['2.5th percentile']
print(*H2O2)

f_TMI_marginal_stats_df = all_factor_marginal_stats_df.loc[:,3].unstack()
TMI = f_TMI_marginal_stats_df['2.5th percentile']
print(*TMI)

f_O3_marginal_stats_df = all_factor_marginal_stats_df.loc[:,4].unstack()
O3 = f_O3_marginal_stats_df['2.5th percentile']
print(*O3)

f_NO2_marginal_stats_df = all_factor_marginal_stats_df.loc[:,5].unstack()
NO2 = f_NO2_marginal_stats_df['2.5th percentile']
print(*NO2)

####### compare model output with observations
all_factor_marginal_stats_df = pd.concat(all_factor_marginal_stats_df_arr, axis=0,
keys=obs_data_for_pymc_df.index)
modeled_vals_df = pd.concat(modeled_val_df_arr, axis=0, keys=obs_data_for_pymc_df.index)
modeled_D17O_stats_df = modeled_vals_df.loc[:,'model_D17O'].unstack()
modeled_d18O_stats_df = modeled_vals_df.loc[:,'model_d18O'].unstack()
#modeled_d34S_stats_df = modeled_vals_df.loc[:, 'model_d34S'].unstack()
modeled_d34S_stats_df = modeled_vals_df.loc[:, 'model_d34S_epsilon'].unstack()

all_factor_marginal_stats_df = pd.concat(all_factor_marginal_stats_df_arr, axis=0,
keys=obs_data_for_pymc_df.index)
modeled_vals_df = pd.concat(modeled_val_df_arr, axis=0, keys=obs_data_for_pymc_df.index)
modeled_D17O_stats_df = modeled_vals_df.loc[:,'model_D17O'].unstack()
modeled_d18O_stats_df = modeled_vals_df.loc[:, 'model_d18O'].unstack()
#modeled_d34S_stats_df = modeled_vals_df.loc[:, 'model_d34S'].unstack()
modeled_d34S_stats_df = modeled_vals_df.loc[:,'model_d34S_epsilon'].unstack()

f_primary_marginal_stats_df = all_factor_marginal_stats_df.loc[:,0].unstack()
f_O3_marginal_stats_df = all_factor_marginal_stats_df.loc[:,4].unstack()
#f_H2O2_marginal_stats_df = all_factor_marginal_stats_df.loc[:,2].unstack()
f_OH_marginal_stats_df = all_factor_marginal_stats_df.loc[:,1].unstack()
```

```
f_TMI_marginal_stats_df = all_factor_marginal_stats_df.loc[:,3].unstack()
f_NO2_marginal_stats_df = all_factor_marginal_stats_df.loc[:,5].unstack()
f_HOOH_marginal_stats_df = all_factor_marginal_stats_df.loc[:,2].unstack()
i_SO4_obs = obs_data_df.loc[i_sample_id, 'SO4_umol']
i_SO2_obs = obs_data_df.loc[i_sample_id, 'SO2_umol']
i_d34S =obs_data_df.loc[i_sample_id, 'd34S']

plt.figure(figsize=(5,5), dpi=300)
plt.scatter(y=modeled_D17O_stats_df['Median'],
        x=obs_data_df.loc[obs_data_for_pymc_df.index,'O17'],
        marker='x', color='k')

plt.plot(np.arange(-100,100),
      np.arange(-100,100),
      linestyle='--', alpha=0.5, color='k')
plt.gca().set_aspect('equal')

plt.xlim(-0.3,3)
plt.xlabel('$\Delta^{17}$O Observations(‰)')

plt.ylim(-0.3,3)
plt.ylabel('MCMC Median $\Delta^{17}$O (‰)')
plt.title('$\Delta^{17}$O')
plt.show()


plt.figure(figsize=(5,5),  dpi=300)
plt.scatter(y=modeled_d18O_stats_df['Median'],
        x=obs_data_df.loc[obs_data_for_pymc_df.index,'O18'],
        marker='x', color='k')

plt.plot(np.arange(-100,100),
      np.arange(-100,100),
      linestyle='--', alpha=0.5, color='k')
plt.gca().set_aspect('equal')

plt.xlim(0,25)
plt.ylim(0,25)
plt.xlabel('$\delta^{18}$O Observations (‰)')
plt.ylabel('MCMC Median $\delta^{18}$O (‰)')
plt.title(' $\delta^{18}$O')

plt.show()

plt.figure(figsize=(5,5),  dpi=300)
plt.scatter(y=modeled_d34S_stats_df['Median'],
        x=obs_data_df.loc[obs_data_for_pymc_df.index,'d34S'],
        marker='x', color='k')
```

```
plt.plot(np.arange(-100,100),
        np.arange(-100,100),
        linestyle='--', alpha=0.5, color='k')
plt.gca().set_aspect('equal')

plt.xlim(3.3,10)
plt.ylim(3,10)
#ax.text(4, 9, 'R$^$ = 0.68', color = 'k', fontsize = 22)

plt.xlabel('$\delta^{34}$S Observations (‰)')
plt.ylabel('MCMC Median $\delta^{34}$S (‰)')
plt.title('$\delta^{34}$S')

plt.show()

plt.figure(figsize=(10,5), dpi=300)

plt.plot(modeled_D17O_stats_df.index,
        modeled_D17O_stats_df['Median'],
        label="MCMC samples' median",
        linewidth=3, alpha=.7, color='darkgoldenrod')
plt.fill_between(x=modeled_D17O_stats_df.index,
              y1=modeled_D17O_stats_df['2.5th percentile'],
              y2=modeled_D17O_stats_df['97.5th percentile'],
              label="MCMC samples' 2.5 to 97.5th percentiles",
              linewidth=2, alpha=.25, color='goldenrod')

plt.plot(obs_data_for_pymc_df.index,
        obs_data_df.loc[obs_data_for_pymc_df.index,'O17'],
        label="Observations",
        color='k', linewidth=2, alpha=.7)

plt.tick_params(rotation=90, axis='x')
plt.ylabel(('$\Delta^{17}$O ‰'))

plt.legend()
plt.title('$\Delta^{17}$O', fontsize=18)
plt.show()


# Plot d18O

plt.figure(figsize=(10,5), dpi=300)

plt.plot(modeled_d18O_stats_df.index,
        modeled_d18O_stats_df['Median'],
        label="MCMC samples' median",
```

```
        linewidth=3, alpha=.75, color='purple')
plt.fill_between(x=modeled_d18O_stats_df.index,
                y1=modeled_d18O_stats_df['2.5th percentile'],
                y2=modeled_d18O_stats_df['97.5th percentile'],
                label="MCMC samples' 2.5 to 97.5th percentiles",
                linewidth=2, alpha=.1, color='indigo')

plt.plot(obs_data_for_pymc_df.index,
        obs_data_df.loc[obs_data_for_pymc_df.index,'O18'],
        label="Observations",
        color='k', linewidth=2, alpha=.6)

plt.tick_params(rotation=90, axis='x')
plt.ylabel(('$\delta^{18}$O ‰'))

plt.legend()
plt.title(' $\delta^{18}$O',fontsize=18)
plt.show()


# Plot d34S_2nd
plt.figure(figsize=(10,5), dpi=300)

plt.plot(modeled_d34S_stats_df.index,
        modeled_d34S_stats_df['Median'],
        label="MCMC samples' median",
        linewidth=3, alpha=.75, color='b')
plt.fill_between(x=modeled_d34S_stats_df.index,
                y1=modeled_d34S_stats_df['2.5th percentile'],
                y2=modeled_d34S_stats_df['97.5th percentile'],
                label="MCMC samples' 2.5 to 97.5th percentiles",
                linewidth=2, alpha=.1, color='b')

plt.plot(obs_data_for_pymc_df.index,
        obs_data_df.loc[obs_data_for_pymc_df.index,'d34S'],
        label="Observations",
        color='k', linewidth=2, alpha=.6)

plt.tick_params(rotation=90, axis='x')
plt.ylabel(('$\delta^{34}$S ‰'))

plt.legend()
plt.title('$\delta^{34}$S',fontsize=18)
plt.show()

d34S_2nd = modeled_d34S_stats_df['Median']
d34S_lower = modeled_d34S_stats_df['2.5th percentile']
d34S_upper = modeled_d34S_stats_df['97.5th percentile']
```

```
f_primary = f_primary_marginal_stats_df['Median']

d34S_MCMC = (f_primary*4.5) + (d34S_2nd*(1-f_primary))
d34S_MCMC_lower = (f_primary*4.5) + (d34S_lower *(1-f_primary))
d34S_MCMC_upper = (f_primary*4.5) + (d34S_upper *(1-f_primary))

plt.figure(figsize=(10,5), dpi=300)

plt.plot(modeled_d34S_stats_df.index,
        d34S_MCMC,
        label="MCMC samples' median",
        linewidth=3, alpha=.75, color='b')
plt.fill_between(x=modeled_d34S_stats_df.index,
              y1=d34S_MCMC_lower ,
              y2=d34S_MCMC_upper,
              label="MCMC samples' 2.5 to 97.5th percentiles",
              linewidth=2, alpha=.1, color='b')

plt.plot(obs_data_for_pymc_df.index,
        obs_data_df.loc[obs_data_for_pymc_df.index,'d34S'],
        label="Observations",
        color='k', linewidth=2, alpha=.6)

plt.tick_params(rotation=90, axis='x')
plt.ylabel(('$\delta^{34}$S ‰'))

plt.legend()
plt.title('$\delta^{34}$S',fontsize=18)
plt.show()

###bar chart

import seaborn as sns
from sklearn import preprocessing
import numpy as np

all_factor_marginal_stats_df = pd.concat(all_factor_marginal_stats_df_arr, axis=0,
keys=obs_data_for_pymc_df.index)

modeled_vals_df = pd.concat(modeled_val_df_arr, axis=0, keys=obs_data_for_pymc_df.index)

f_primary_marginal_stats_df = all_factor_marginal_stats_df.loc[:,0].unstack()
f_O3_marginal_stats_df = all_factor_marginal_stats_df.loc[:,4].unstack()
f_H2O2_marginal_stats_df = all_factor_marginal_stats_df.loc[:,2].unstack()
f_OH_marginal_stats_df = all_factor_marginal_stats_df.loc[:,1].unstack()
f_TMI_marginal_stats_df = all_factor_marginal_stats_df.loc[:,3].unstack()
f_NO2_marginal_stats_df = all_factor_marginal_stats_df.loc[:,5].unstack()
```

```
all_factor_marginal_stats_df = pd.concat(all_factor_marginal_stats_df_arr, axis=0,
keys=obs_data_for_pymc_df.index)

labels = f_primary_marginal_stats_df.index
labels = f_primary_marginal_stats_df.index


j=20
f_primary_marginal_stats_df = all_factor_marginal_stats_df.loc[:,0].unstack()
Primary = f_primary_marginal_stats_df['Median']

f_OH_marginal_stats_df = all_factor_marginal_stats_df.loc[:,1].unstack()
OH = f_OH_marginal_stats_df['Median']

f_H2O2_marginal_stats_df = all_factor_marginal_stats_df.loc[:,2].unstack()
H2O2 = f_H2O2_marginal_stats_df['Median']

f_TMI_marginal_stats_df = all_factor_marginal_stats_df.loc[:,3].unstack()
TMI = f_TMI_marginal_stats_df['Median']

f_O3_marginal_stats_df = all_factor_marginal_stats_df.loc[:,4].unstack()
O3 = f_O3_marginal_stats_df['Median']

f_NO2_marginal_stats_df = all_factor_marginal_stats_df.loc[:,5].unstack()
NO2 = f_NO2_marginal_stats_df['Median']


fig, ax = plt.subplots(figsize=(20,8), dpi=500)
width = 0.88

#pal = sns.color_palette("Set1")
#rescale = lambda y: (y - np.min(y)) / (np.max(y) - np.min(y))
Norm = 1/(Primary+OH+O3+TMI+H2O2+NO2)
O3 = Norm*O3
OH=Norm*OH
TMI=Norm*TMI
H2O2=Norm*H2O2
Primary =Norm*Primary
NO2 = Norm*NO2
#triplet = Norm*triplet

NO2_plt = ax.bar(labels[0:43], NO2[0:43], width, color='teal',alpha=0.6,label='NO$_2$')
O3_plt =ax.bar(labels[0:43], O3[0:43], width, bottom = NO2[0:43], color='gold', alpha=
0.8,label='O$_3$')
OH_plt=ax.bar(labels[0:43], OH[0:43], width, bottom=O3[0:43]+NO2[0:43],
color='darkorange',alpha=0.8, label='OH')
TMI_plt=ax.bar(labels[0:43], TMI[0:43], width, bottom=O3[0:43]+OH[0:43]+NO2[0:43], color=
'mediumvioletred',alpha=0.8,label='TMI')
```

```
H2O2_plt =ax.bar(labels[0:43], H2O2[0:43], width,
bottom=O3[0:43]+TMI[0:43]+OH[0:43]+NO2[0:43],color='mediumblue', alpha=0.5, label='H$_2$O$_2$')
primary_plt = ax.bar(labels[0:43], Primary[0:43], width,
bottom=OH[0:43]+TMI[0:43]+H2O2[0:43]+O3[0:43]+NO2[0:43],color='midnightblue',alpha=0.8,label='Pr
imary')
#NO2_plt = ax.bar(labels, NO2, width,
bottom=OH+TMI+H2O2+O3+Primary+NO2,color='teal',alpha=0.6,label='NO2')
#triplet_plt = ax.bar(labels, triplet, width,
bottom=OH+TMI+H2O2+O3+Primary+NO2,color='mediumblue',alpha=0.15,label='triplet')

ax.set_xticks(f_primary_marginal_stats_df.index[0:43])
ax.set_xlabel('Sample ID',fontsize=16)

ax.tick_params(axis = 'y', labelsize=16, rotation=0)
plt.tick_params(rotation=90, axis='x',labelsize=16)

ax.set_ylabel('Fraction of PM0.7 Sulfate',fontsize=20)
ax.set_title('Sources and Formation of Fairbanks Sulfate',fontsize=24)
ax.legend(prop={'size': 16},loc='upper center', bbox_to_anchor=(0.5, -0.3),
        fancybox=True, shadow=True, ncol=5)

for r1, r2, r3, r4, r5,r6 in zip(NO2_plt,O3_plt, OH_plt, TMI_plt, H2O2_plt,primary_plt):
    h1 = r1.get_height()
    h2 = r2.get_height()
    h3 = r3.get_height()
    h4 =r4.get_height()
    h5 = r5.get_height()
    h6 = r6.get_height()
    plt.text(r1.get_x() + r1.get_width() / 2., h1 / 2., "{}".format(np.round(h1, decimals=2)), ha="center",
va="center", color="white", fontsize=8, fontweight="bold")
    plt.text(r2.get_x() + r2.get_width() / 2., (h1 + h2/2), "{}".format(np.round(h2, decimals=2)),
ha="center", va="center", color="midnightblue", fontsize=8, fontweight="bold")
    plt.text(r3.get_x() + r3.get_width() / 2., (h1 + h2+h3/2), "{}".format(np.round(h3, decimals=2)),
ha="center", va="center", color="white", fontsize=8, fontweight="bold")
    plt.text(r4.get_x() + r4.get_width() / 2., (h1 + h2+h3+h4/2), "{}".format(np.round(h4, decimals=2)),
ha="center", va="center", color="white", fontsize=8, fontweight="bold")
    plt.text(r5.get_x() + r5.get_width() / 2., (h1 + h2+h3+h4+h5/2), "{}".format(np.round(h5, decimals=2)),
ha="center", va="center", color="white", fontsize=8, fontweight="bold")
    plt.text(r6.get_x() + r6.get_width() / 2., (h1 + h2+h3+h4+h5+h6/2), "{}".format(np.round(h6,
decimals=2)), ha="center", va="center", color="white", fontsize=8, fontweight="bold")

plt.show()

###time series with 95% confidence interval
import matplotlib.dates as mdates
all_factor_marginal_stats_df = pd.concat(all_factor_marginal_stats_df_arr, axis=0,
keys=obs_data_for_pymc_df.index)
```

```
all_factor_marginal_stats_df = pd.concat(all_factor_marginal_stats_df_arr, axis=0,
keys=obs_data_for_pymc_df.index)
modeled_vals_df = pd.concat(modeled_val_df_arr, axis=0, keys=obs_data_for_pymc_df.index[0:43])
modeled_D17O_stats_df = modeled_vals_df.loc[:,'model_D17O'].unstack()
modeled_d18O_stats_df = modeled_vals_df.loc[:, 'model_d18O'].unstack()
modeled_d34S_stats_df = modeled_vals_df.loc[:,'model_d34S_epsilon'].unstack()

f_primary_marginal_stats_df = all_factor_marginal_stats_df.loc[:,0].unstack()
f_O3_marginal_stats_df = all_factor_marginal_stats_df.loc[:,4].unstack()
f_H2O2_marginal_stats_df = all_factor_marginal_stats_df.loc[:,2].unstack()
f_OH_marginal_stats_df = all_factor_marginal_stats_df.loc[:,1].unstack()
f_TMI_marginal_stats_df = all_factor_marginal_stats_df.loc[:,3].unstack()
f_NO2_marginal_stats_df = all_factor_marginal_stats_df.loc[:,5].unstack()

modeled_vals_df = pd.concat(modeled_val_df_arr, axis=0, keys=obs_data_for_pymc_df.index)
modeled_D17O_stats_df = modeled_vals_df.loc[:,'model_D17O'].unstack()
modeled_d18O_stats_df = modeled_vals_df.loc[:,'model_d18O'].unstack()

fig, ax = plt.subplots(figsize=(15,5), dpi=300)

plt.plot(f_primary_marginal_stats_df.index[0:43],
        f_primary_marginal_stats_df['Median'][0:43],
        label="Primary median",
        linewidth=4, alpha=.9, color='midnightblue')
plt.fill_between(f_primary_marginal_stats_df.index[0:43],
               y1=f_primary_marginal_stats_df['2.5th percentile'][0:43],
               y2=f_primary_marginal_stats_df['97.5th percentile'][0:43],
               #label="Primary 25 to 75th percentiles",
               linewidth=2, alpha=.05, color='midnightblue')

plt.fill_between(f_primary_marginal_stats_df.index[0:43],
              y1=f_O3_marginal_stats_df['2.5th percentile'][0:43],
              y2=f_O3_marginal_stats_df['97.5th percentile'][0:43],
              #label="O$_3$ 25 to 75th percentiles",
              linewidth=2, alpha=.2, color='gold')
plt.plot(f_primary_marginal_stats_df.index[0:43],
        f_O3_marginal_stats_df['Median'][0:43],
        label="O$_3$ Median",
        linewidth=4, alpha=0.9, color='gold')

plt.plot(f_primary_marginal_stats_df.index[0:43],
        f_OH_marginal_stats_df['Median'][0:43],
        label="OH Median",
        linewidth=4, alpha=0.9, color='darkorange')

plt.plot(f_primary_marginal_stats_df.index[0:43],
        f_H2O2_marginal_stats_df['Median'][0:43],
        label="H$_2$O$_2$ median",
```

```
        linewidth=4, alpha=0.9, color='mediumblue')
plt.fill_between(f_primary_marginal_stats_df.index[0:43],
                y1=f_H2O2_marginal_stats_df['2.5th percentile'][0:43],
                y2=f_H2O2_marginal_stats_df['97.5th percentile'][0:43],
                #label="H$_2$O$_2$ 25 to 75th percentiles",
                linewidth=2, alpha=.05, color='mediumblue')

plt.plot(f_primary_marginal_stats_df.index[0:43],
        f_TMI_marginal_stats_df['Median'][0:43],
        label="TMI-O$_2$ median",
        linewidth=4, alpha=0.9, color='mediumvioletred')
plt.fill_between(f_primary_marginal_stats_df.index[0:43],
                y1=f_TMI_marginal_stats_df['2.5th percentile'][0:43],
                y2=f_TMI_marginal_stats_df['97.5th percentile'][0:43],
               #label="TMI-O$_2$ 25 to 75th percentiles",
                linewidth=2, alpha=.1, color='mediumvioletred')

plt.plot(f_primary_marginal_stats_df.index[0:43],
        f_NO2_marginal_stats_df['Median'][0:43],
        label="NO$_2$ median",
        linewidth=4, alpha=0.9, color='gray')
plt.fill_between(f_primary_marginal_stats_df.index[0:43],
                y1=f_NO2_marginal_stats_df['2.5th percentile'][0:43],
                y2=f_NO2_marginal_stats_df['97.5th percentile'][0:43],
               #label="TMI-O$_2$ 25 to 75th percentiles",
                linewidth=2, alpha=.15, color='gray')


ax.set_xticks(f_primary_marginal_stats_df.index[0:43])
plt.ylim(0,1)
#ax.xaxis.grid()
#ax.set_xlabel('Sample ID',fontsize=20)
ax.set_ylabel('Fraction of PM0.7 Sulfate',fontsize=24)
ax.set_title('Sources and Formation of Fairbanks Sulfate',fontsize=24)

ax.legend(prop={'size': 14},loc='upper center', bbox_to_anchor=(0.5, -0.3),
         fancybox=True, shadow=True, ncol=5)
ax.tick_params(axis = 'y', labelsize=20, rotation=0)
plt.tick_params(rotation=90, axis='x',labelsize=10)
plt.show()
```