In [1]:
```
import os
from os import listdir
from os.path import isfile, join
import struct
import random
import operator
import gzip
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
import matplotlib.pyplot as plt
```

In [3]:
```
mypath= r'C:\Users\dwijj\Downloads\chsi_dataset'
os.chdir(mypath)
onlyfiles = [f for f in listdir(mypath) if isfile(join(mypath, f))]
onlyfiles
```

Out[3]:
```
['CHSI DataSet.xls',
 'CSV File Index.txt',
 'DATAELEMENTDESCRIPTION.csv',
 'DEFINEDDATAVALUE.csv',
 'DEMOGRAPHICS.csv',
 'HEALTHYPEOPLE2010.csv',
 'LEADINGCAUSESOFDEATH.csv',
 'MEASURESOFBIRTHANDDEATH.csv',
 'PREVENTIVESERVICESUSE.csv',
 'project.ipynb',
 'RELATIVEHEALTHIMPORTANCE.csv',
 'RISKFACTORSANDACCESSTOCARE.csv',
 'SUMMARYMEASURESOFHEALTH.csv',
 'VUNERABLEPOPSANDENVHEALTH.csv']
```

In [40]:
```
df_mbd = pd.read_csv('MEASURESOFBIRTHANDDEATH.csv')
df_mbd = df_mbd[['State_FIPS_Code','County_FIPS_Code','CHSI_County_Name','CHSI_St
ListofNans = [-9999,-2222,-2222.2,-2,-1111,-1,-9998.9]
df_mbd=df_mbd.replace([i for i in ListofNans], np.NAN)
df_mbd.head()
```

Out[40]:

	State_FIPS_Code	County_FIPS_Code	CHSI_County_Name	CHSI_State_Abbr	CHSI_State_Name
0	1	1	Autauga	AL	Alabama
1	1	3	Baldwin	AL	Alabama
2	1	5	Barbour	AL	Alabama
3	1	7	Bibb	AL	Alabama
4	1	9	Blount	AL	Alabama

5 rows × 28 columns

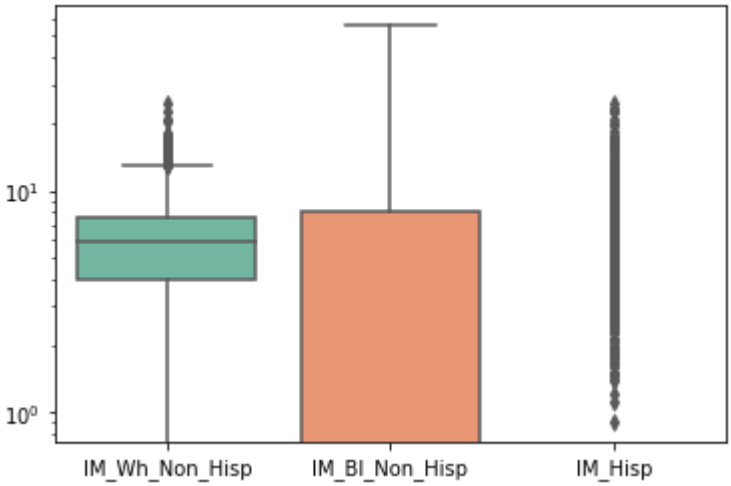
In [21]:
```
BirthStats = df_mbd['Total_Births'].describe()
DeathStats = df_mbd['Total_Deaths'].describe()
print("Births Across Counties Stats\n", BirthStats,"\n\n")
print("Deaths Across Counties Stats\n", DeathStats)
```

```
Births Across Counties Stats
 count      3140.000000
mean       4838.878344
std       13754.598791
min           2.000000
25%        1319.750000
50%        2283.000000
75%        3936.000000
max      457033.000000
Name: Total_Births, dtype: float64


Deaths Across Counties Stats
 count      3140.000000
mean       3107.701592
std        6432.756342
min           5.000000
25%        1164.000000
50%        1887.000000
75%        2858.250000
max      181018.000000
Name: Total_Deaths, dtype: float64
```

In [24]:
```
Races_df = df_mbd[['IM_Wh_Non_Hisp','IM_Bl_Non_Hisp','IM_Hisp']]
Races_df.rename(columns = {'IM_Wh_Non_Hisp':'White', 'IM_Bl_Non_Hisp':'Black','IM
print("Races \n\n", Races_df.describe())
```

```
Races

              White        Black    Hispanic
count  3140.000000  3140.000000  3140.00000
mean   -145.765732  -803.439904  -891.16328
std     383.559354   501.836686   444.37318
min   -1111.100000 -1111.100000 -1111.10000
25%       4.000000 -1111.100000 -1111.10000
50%       5.900000 -1111.100000 -1111.10000
75%       7.600000     8.025000 -1111.10000
max      24.600000    55.600000    24.60000
```

In [25]:
```
ax = sns.boxplot(data=df_mbd[['IM_Wh_Non_Hisp','IM_Bl_Non_Hisp','IM_Hisp']], pale
ax.set_yscale('log')
```



In [28]:
```
AgeBorn = df_mbd[['Under_18','Over_40']]
print("Age Groups \n",AgeBorn.describe())
```

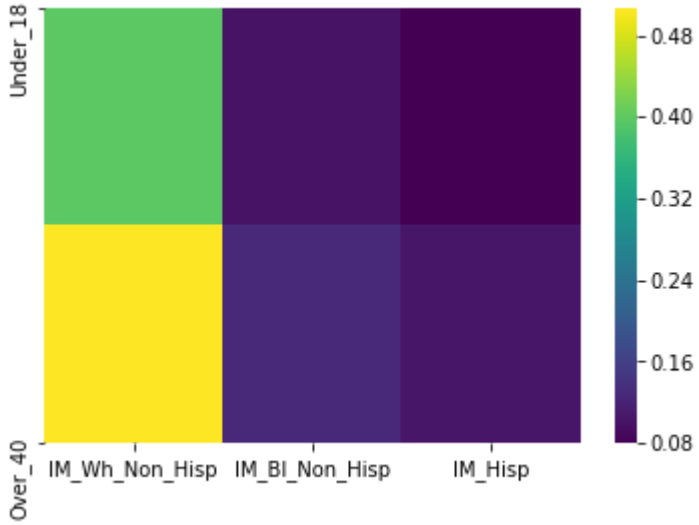
```
Age Groups
          Under_18      Over_40
count  3140.000000  3140.000000
mean    -22.968439   -43.975732
std     173.712929   220.916017
min   -1111.100000 -1111.100000
25%       2.900000     1.100000
50%       4.300000     1.500000
75%       6.000000     2.100000
max      14.500000     9.100000
```

In [29]:
```
Race_Age = df_mbd[['IM_Wh_Non_Hisp','IM_Bl_Non_Hisp','IM_Hisp','Under_18','Over_4
Race_AgeCorr = pd.DataFrame(Race_Age.corr())
Race_AgeCorr = Race_AgeCorr[Race_AgeCorr.index.isin(['IM_Wh_Non_Hisp','IM_Bl_Non_
Race_AgeCorr = Race_AgeCorr[['Under_18','Over_40']]
print(Race_AgeCorr)
```

```
                Under_18   Over_40
IM_Wh_Non_Hisp  0.399284  0.506627
IM_Bl_Non_Hisp  0.100756  0.126865
IM_Hisp         0.079410  0.103268
```

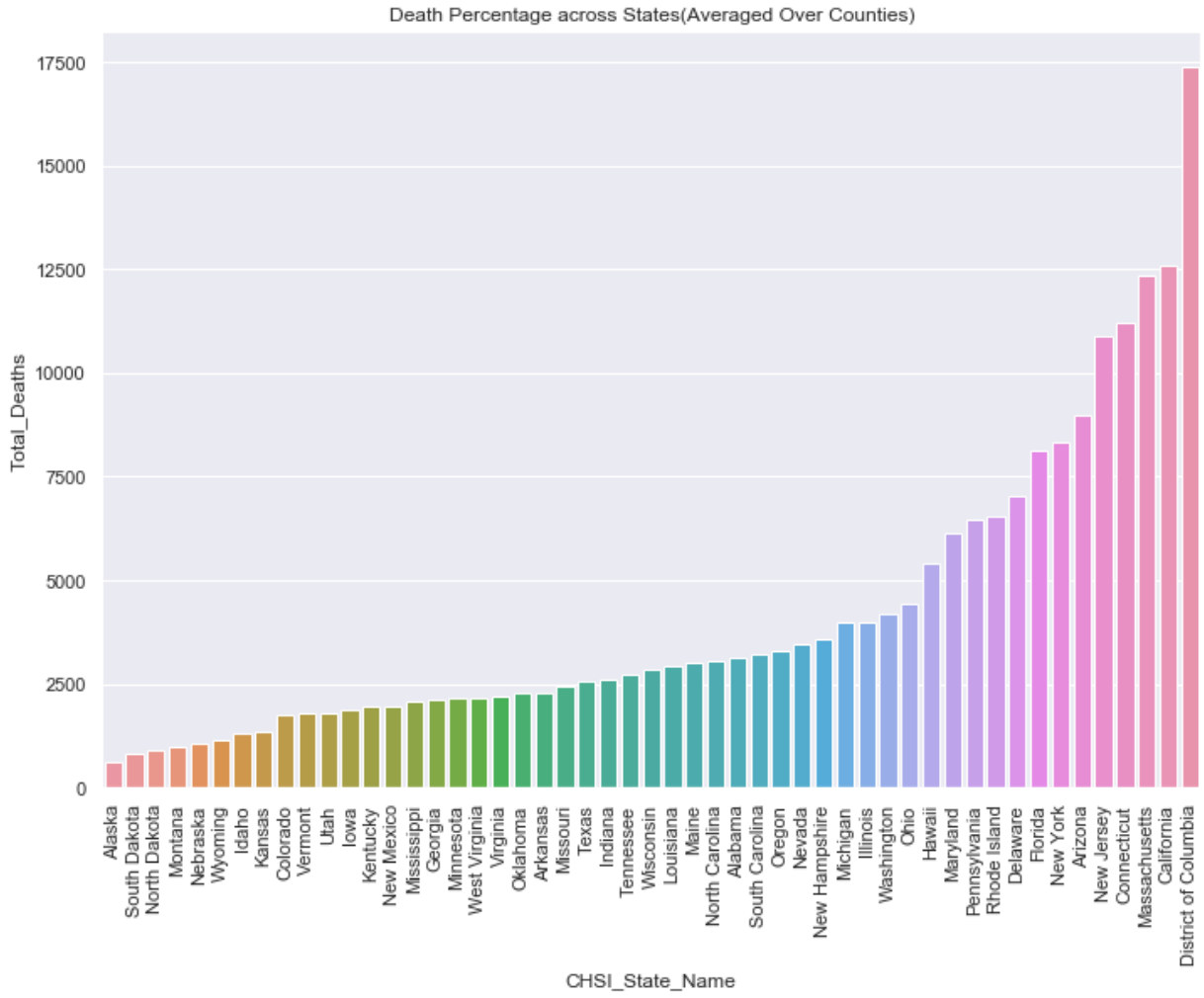
The table represents the correlation values among the features.

In [30]:
```
ax = sns.heatmap(Race_AgeCorr.transpose(),cmap='viridis')
```



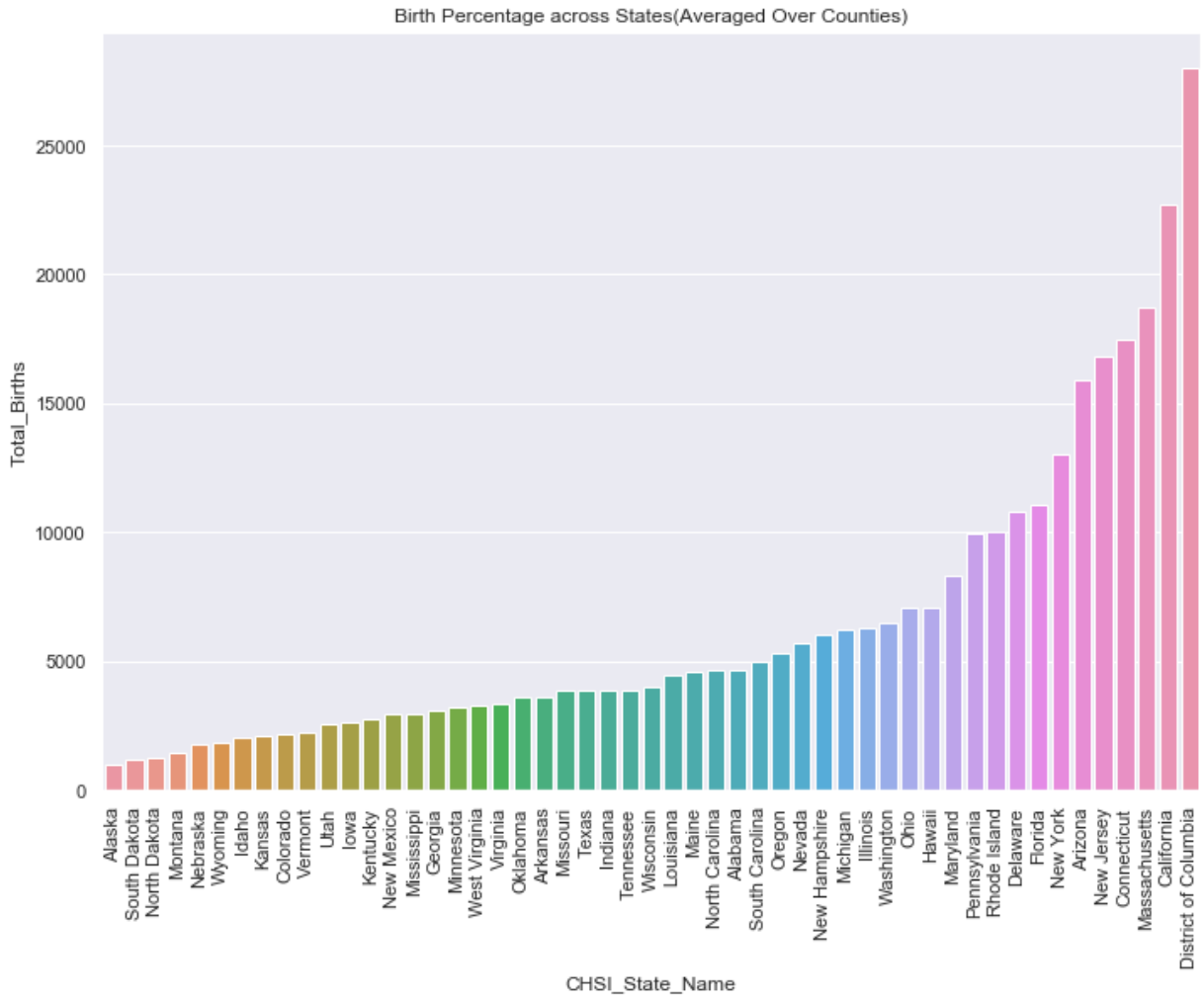
In [33]:
```
DeathDF = df_mbd[['Total_Deaths']].groupby(df_mbd['CHSI_State_Name']).mean().sort
sns.set(rc={'figure.figsize':(11.7,8.27)})
chart = sns.barplot(x=DeathDF.index, y='Total_Deaths', data=DeathDF)
plt.xticks(rotation=90)
plt.title('Death Percentage across States(Averaged Over Counties)')
```

Out[33]: Text(0.5, 1.0, 'Death Percentage across States(Averaged Over Counties)')



In [36]:
```
BirthDF = df_mbd[['Total_Births']].groupby(df_mbd['CHSI_State_Name']).mean().sort
sns.set(rc={'figure.figsize':(11.7,8.27)})
chart = sns.barplot(x=DeathDF.index, y='Total_Births', data=BirthDF)
plt.xticks(rotation=90)
plt.title('Birth Percentage across States(Averaged Over Counties)')
```

Out[36]: Text(0.5, 1.0, 'Birth Percentage across States(Averaged Over Counties)')



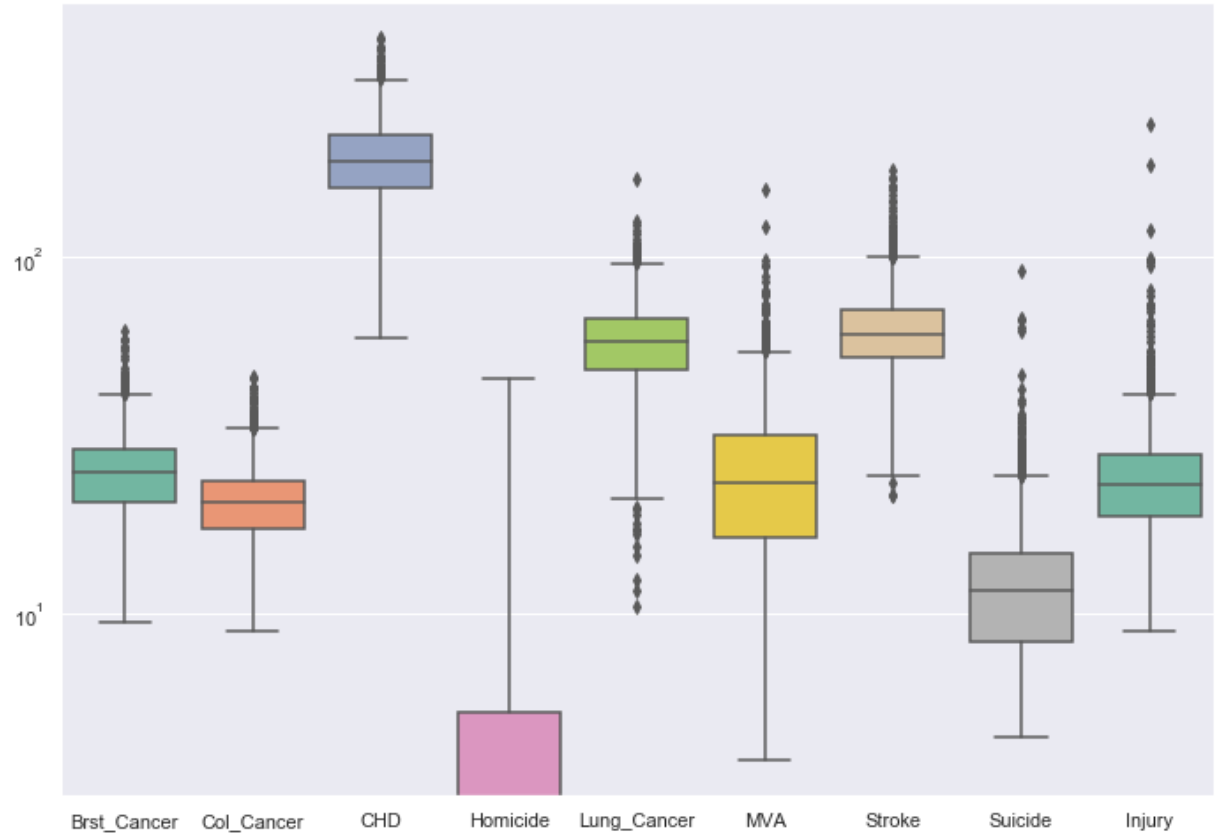
In [42]:
```
DReason_df = df_mbd[['Brst_Cancer','Col_Cancer','CHD','Homicide','Lung_Cancer','M
print("Death Types \n", DReason_df.describe())
```

```
Death Types
        Brst_Cancer   Col_Cancer          CHD     Homicide  Lung_Cancer  \
count  3140.000000  3140.000000  3140.000000  3140.000000  3140.000000
mean   -114.947325   -59.431720   183.601847  -680.712038    29.955127
std     375.238313   291.561394   109.473949   544.384791   181.508161
min   -1111.100000 -1111.100000 -1111.100000 -1111.100000 -1111.100000
25%      20.500000    17.400000   156.475000 -1111.100000    48.500000
50%      24.900000    20.500000   187.000000 -1111.100000    58.050000
75%      28.900000    23.700000   221.250000     5.300000    67.500000
max      62.300000    46.300000   412.900000    46.000000   166.400000

                MVA       Stroke      Suicide       Injury
count  3140.000000  3140.000000  3140.000000  3140.000000
mean    -39.297643    38.701783  -173.418408   -41.501815
std     264.597871   169.237718   418.797706   266.283995
min   -1111.100000 -1111.100000 -1111.100000 -1111.100000
25%      16.300000    52.300000     8.400000    18.775000
50%      23.300000    61.100000    11.700000    23.000000
75%      31.700000    71.400000    14.825000    27.900000
max     154.600000   175.800000    91.300000   236.200000
```

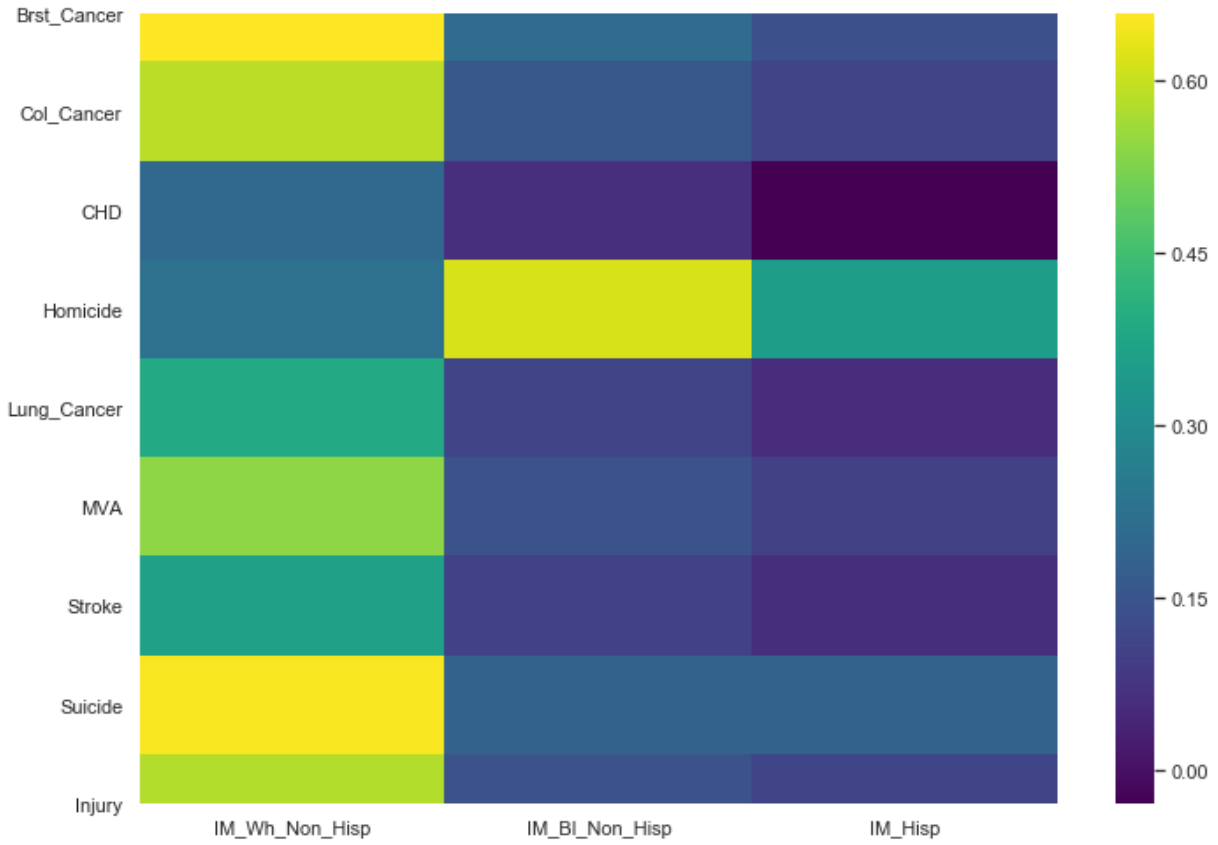
In [43]:
```
Death_Reason = df_mbd[['IM_Wh_Non_Hisp','IM_Bl_Non_Hisp','IM_Hisp','Brst_Cancer',
Death_ReasonCorr = pd.DataFrame(Death_Reason.corr())
Death_ReasonCorr = Death_ReasonCorr[Death_ReasonCorr.index.isin(['IM_Wh_Non_Hisp'
Death_ReasonCorr = Death_ReasonCorr[['Brst_Cancer','Col_Cancer','CHD','Homicide',
print(Death_ReasonCorr)
```

```
                Brst_Cancer  Col_Cancer       CHD  Homicide  Lung_Cancer  \
IM_Wh_Non_Hisp     0.658974    0.587406  0.201669  0.225717     0.385497
IM_Bl_Non_Hisp     0.210620    0.158220  0.064671  0.617122     0.111377
IM_Hisp            0.138884    0.108712 -0.028988  0.350097     0.059588

                     MVA    Stroke   Suicide    Injury
IM_Wh_Non_Hisp  0.543205  0.360929  0.653414  0.579592
IM_Bl_Non_Hisp  0.143649  0.101371  0.186623  0.144415
IM_Hisp         0.104681  0.060058  0.185411  0.113137
```

In [44]:
```
ax = sns.boxplot(data=df_mbd[['Brst_Cancer','Col_Cancer','CHD','Homicide','Lung_C
ax.set_yscale('log')
```



In [45]:
```
ax = sns.heatmap(Death_ReasonCorr.transpose(), cmap='viridis')
```



In [ ]:
```

```