In [ ]:

In [1]:
```
%matplotlib widget
import pandas as pd
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt
import analysis as analysis
from analysis import read_pickle, asnp
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.stats.anova import anova_lm
```

In [2]:
```
filename = 'base' #'mod'
#df = analysis.read_pickle('df_sayama_change_all')
#df = read_pickle('df_group_size10-40_changeall')
df_full = analysis.read_pickle('df_' + filename)
print(df_full.shape)
df_full.columns
df = analysis.data_cleanup(df_full)
```

```
(21600, 18)
```

In [3]:
```
#load SPL on undirected conversion of graph
filename = 'base' + 'mod'
dfmod = analysis.read_pickle('df_' + filename)
dfmod.columns
#load SPL on scc analysis of graph
filename = 'base' + 'modsc'
dfsc = analysis.read_pickle('df_' + filename)
```

In [9]:
```
df = df.loc[df['giantComponent'] >= 0.99]
dfsc = dfsc.loc[df_full['giantComponent'] >= 0.99]
```

In [10]:
```
#mplot3d doesnt handle sympy floats
stdd = np.asarray(df['std_d'].values, dtype = "float")
stdrs = np.asarray(df['std_rs'].values, dtype = "float")
stdrw = np.asarray(df['std_rw'].values, dtype = "float")
spl = np.asarray(df['SPL'].values, dtype = "float")
cd = np.asarray(df['CD'].values, dtype = "float")
cc = np.asarray(df['giantComponent'].values, dtype = "float")
sc = np.asarray(dfsc['sc'].values, dtype = "float")
```

In [11]:
```
analysis.plotfig(stdd, sc, spl, '$d$', 'stdd-sc-spl', fsize=(7,7) )
#analysis.plotfig(stdrs, cd, spl, '$r_s$', 'base-rs')
#analysis.plotfig(stdrw, cd, spl, '$r_w$', 'base-rw')
```

discrepancy in SPL

In [119]:
```
fig = plt.figure()
fig.set_size_inches(11,13)
ax = fig.gca(projection='3d')
alp = 0.2
#ax.scatter(asnp(dfsc['std_d']), asnp(dfsc['CD']), asnp(dfsc['SPL']),
          #label='Largest directed strongly connected component', c='g', marker=
'.', alpha=alp)
ax.scatter(stdd, cd, spl, label = 'Sayama and Yamanoi', c='r', marker='.',alpha=al
p)
ax.scatter(asnp(dfmod['std_d']), asnp(dfmod['CD']), asnp(dfmod['SPL']),
          #label='Largest undirected weakly connected component', c='b', marker='x
', alpha=alp)

ax.legend()
ax.set_xlabel("s.d. of $d$")
ax.set_ylabel("<CD>")
ax.set_zlabel("<SPL>")
ax.view_init(elev=3, azim=305)
plt.draw()
```

analysis of strongly/weakly connected components

In [107]:
```
def ddgroupby(df, labels):
    sdf = pd.DataFrame(df.groupby(labels)[['sc', 'giantComponent']].mean())
    sdf.reset_index(inplace=True)
    return sdf
def plt_surf(label, df, sdf):
    plt.close('all')
    fig = plt.figure()
    fig.set_size_inches(9, 9)
    ax = fig.gca(projection='3d')

    #ax.scatter(asnp(df[label1[0]]), asnp(df[label1[1]]), asnp(df['sc']), marker=
'.')
    ax.plot_trisurf(asnp(sdf[label[0]]), asnp(sdf[label[1]]), asnp(sdf['sc']), lab
el='Strongly Connected',
                    cmap=cm.coolwarm, linewidth=0, edgecolor='none')
    ax.plot_trisurf(asnp(sdf[label[0]]), asnp(sdf[label[1]]), asnp(sdf['giantCompo
nent']),
                    cmap=cm.viridis, linewidth=0, edgecolor='none')
    ax.set_xlabel('s.d. of $r_w$')
    ax.set_ylabel('s.d. of $r_s$')
    ax.set_zlabel('Component Size')
    plt.draw()
```

In [45]:
```
def cum_components(dfsc):
    x = []
    sc_ = []
    gc_ = []
    for i in np.linspace(0, 1.0, 21):
        cum = 1 - i
        con = dfsc['sc'] >= cum
        con2 = dfsc['giantComponent'] >= cum
        x.append(cum)
        sc_.append(dfsc.loc[con].shape[0])
        gc_.append(dfsc.loc[con2].shape[0])
    for i in np.linspace(0,0.15,15):
        cum = 1 - i
        con = dfsc['sc'] >= cum
        con2 = dfsc['giantComponent'] >= cum
        x.append(cum)
        sc_.append(dfsc.loc[con].shape[0])
        gc_.append(dfsc.loc[con2].shape[0])
    return x, sc_, gc_
```

In [12]:
```
dfca = read_pickle('df_sayama_change_allmodsc')
#x, sc_, gc_ = cum_components(dfca)
```

In [48]:
```
plt.close('all')
fig = plt.figure()
ax = fig.add_subplot(111)
ax.scatter(x,sc_, label = 'Strongly Connected',color='g')
ax.scatter(x,gc_, label='Weakly Connected', color='b', marker ='x')
ax.legend()
ax.set_ylabel('Number of Experiments')
ax.set_xlabel('Relative size of Largest Component')
plt.draw()
```

In [ ]:

Start with 0.8 as minimum gc size. Filter df:

In [108]:
```
#df's are df, dfmod (undir SPL), dfsc, dfca(changeall)
label1 = ['std_rw', 'std_rs']
label2 = ['std_d', 'std_rw']
label3 = ['std_d', 'std_rs']
labels = label1

sdf = ddgroupby(dfca, labels)
plt_surf(labels, df, sdf)
```

linreg

In [112]:
```
res = smf.ols(formula='sc ~ (std_d + std_rs + std_rw)**2', data=dfsc).fit()
print(str(res.summary()))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                     sc   R-squared:                       0.700
Model:                            OLS   Adj. R-squared:                  0.700
Method:                 Least Squares   F-statistic:                     8388.
Date:                Mon, 20 Jul 2020   Prob (F-statistic):               0.00
Time:                        21:53:39   Log-Likelihood:                 37033.
No. Observations:               21600   AIC:                        -7.405e+04
Df Residuals:                   21593   BIC:                        -7.400e+04
Df Model:                           6
Covariance Type:            nonrobust
================================================================================
=
                   coef    std err          t      P>|t|      [0.025      0.97
5]
--------------------------------------------------------------------------------
-
Intercept        1.0365      0.001    759.174      0.000       1.034       1.03
9
std_d           -0.3997      0.004   -100.137      0.000      -0.408      -0.39
2
std_rs          -0.0061      0.004     -1.530      0.126      -0.014       0.00
2
std_rw          -0.0172      0.004     -4.298      0.000      -0.025      -0.00
9
std_d:std_rs    -0.1080      0.010    -10.623      0.000      -0.128      -0.08
8
std_d:std_rw     0.1630      0.010     16.038      0.000       0.143       0.18
3
std_rs:std_rw    0.0068      0.010      0.670      0.503      -0.013       0.02
7
==============================================================================
Omnibus:                     5927.231   Durbin-Watson:                   1.438
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            38082.159
Skew:                          -1.163   Prob(JB):                         0.00
Kurtosis:                       9.075   Cond. No.                         42.4
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

In [110]:
```
print(str(anova_lm(res)))
```

```
                    df     sum_sq    mean_sq             F         PR(>F)
std_d              1.0  93.831200  93.831200  49415.864717   0.000000e+00
std_rs             1.0   0.621218   0.621218    327.162022   1.370843e-72
std_rw             1.0   0.403358   0.403358    212.426995   6.853623e-48
std_d:std_rs       1.0   0.214294   0.214294    112.856957   2.693272e-26
std_d:std_rw       1.0   0.488377   0.488377    257.201753   1.503337e-57
std_rs:std_rw      1.0   0.000852   0.000852      0.448904   5.028630e-01
Residual       21593.0  41.000944   0.001899           NaN            NaN
```

how much does sc explain SPL?

In [113]:
```
df['sc'] = dfsc['sc']
df.columns
```

Out[113]:
```
Index(['degrees', 'clusterCoeff', 'reciprocity', 'center_1', 'center_2',
       'mean_centers', 'overall_mean_culture', 'giantComponent', 'diam', 'SPL',
       'CD', 'c1_init', 'c2_init', 'c_avg_init', 'std_d', 'std_rs', 'std_rw',
       'tags', 'comps', 'sc'],
      dtype='object')
```

In [122]:
```
res = smf.ols(formula='SPL ~ (sc + CD) ** 2', data=dfsc).fit()
print(str(res.summary()))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    SPL   R-squared:                       0.388
Model:                            OLS   Adj. R-squared:                  0.388
Method:                 Least Squares   F-statistic:                     4561.
Date:                Mon, 20 Jul 2020   Prob (F-statistic):               0.00
Time:                        22:09:41   Log-Likelihood:                 16000.
No. Observations:               21600   AIC:                        -3.199e+04
Df Residuals:                   21596   BIC:                        -3.196e+04
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      2.1451      0.117     18.387      0.000       1.916       2.374
sc             0.0675      0.118      0.574      0.566      -0.163       0.298
CD             0.3514      0.040      8.776      0.000       0.273       0.430
sc:CD         -0.2545      0.040     -6.290      0.000      -0.334      -0.175
==============================================================================
Omnibus:                     3506.301   Durbin-Watson:                   1.899
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            19684.744
Skew:                           0.666   Prob(JB):                         0.00
Kurtosis:                       7.483   Cond. No.                         881.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

In [71]:
```
plt.close('all')
fig = plt.figure()
fig.set_size_inches(9, 9)
ax = fig.gca(projection='3d')

ax.scatter(asnp(dfsc['SPL']), asnp(dfsc['std_d']), asnp(dfsc['sc']), marker='.')
ax.scatter(asnp(dfsc['SPL']), asnp(dfsc['std_d']), asnp(dfsc['giantComponent']), ma
rker='.', color='r')

ax.set_ylabel('s.d. of d')
ax.set_xlabel('<SPL>')
ax.set_zlabel('Size of largest Strongly Connected Component')

plt.draw()
```

In [ ]:
```
#
```

linear relation between splsc and splgc?

In [94]:
```
dtemp = dfmod['SPL']
dtemp.name = "SPLweak"
dtemp2 = dfsc['SPL']
dtemp2.name = "SPLstrong"
dfspl = pd.concat([dtemp, dtemp2], axis=1)
dfspl.head()
```

Out[94]:

	SPLweak	SPLstrong
0	1.974694	2.463265
1	1.951837	2.430204
2	2.040000	2.560816
3	1.937143	2.337143
4	1.932245	2.351429

In [95]:
```
res = smf.ols(formula='SPLstrong ~ SPLweak', data=dfspl).fit()
print(str(res.summary()))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              SPLstrong   R-squared:                       0.798
Model:                            OLS   Adj. R-squared:                  0.798
Method:                 Least Squares   F-statistic:                 8.514e+04
Date:                Mon, 20 Jul 2020   Prob (F-statistic):               0.00
Time:                        21:32:56   Log-Likelihood:                 27956.
No. Observations:               21600   AIC:                        -5.591e+04
Df Residuals:                   21598   BIC:                        -5.589e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.2541      0.009    -26.756      0.000      -0.273      -0.235
SPLweak        1.3652      0.005    291.795      0.000       1.356       1.374
==============================================================================
Omnibus:                     5398.807   Durbin-Watson:                   1.887
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           255132.610
Skew:                          -0.393   Prob(JB):                         0.00
Kurtosis:                      19.819   Cond. No.                         53.1
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

analysis with 0.8 of graph as scc

In [128]:
```
dfred = dfsc.loc[dfsc['sc'] >= 0.8]
dfred.shape
```

Out[128]: (20259, 20)

In [134]:
```
analysis.plotfig(asnp(dfred['std_rw']), asnp(dfred['CD']), asnp(dfred['SPL']), '$r
_w$')
```

In [130]:
```
import linReg
linReg.fit_model(dfred, 'reduced-base-linreg')
```

In [131]:
```
linReg.fit_model(df, 'sayama-base')
```

analyze experiment where we change all parameters.

- 0.8 cutoff for scc

In [6]:
```
import linReg
```

In [15]:
```
dfcaR = dfca.loc[dfca['sc'] >= 0.8]
dfcaR.shape
analysis.plotfig(asnp(dfcaR['std_rw']), asnp(dfcaR['CD']), asnp(dfcaR['SPL']), '$r_
s$', fsize=(7,7))
```

In [136]:
```
dfcaR.shape
```

Out[136]:
```
(15687, 20)
```

In [7]:
```
linReg.fit_model(dfcaR, 'linreg changeall reduced')
```

In [137]:
```
res = smf.ols(formula='sc ~ (std_d + std_rs + std_rw)**2', data=dfcaR).fit()
print(str(res.summary()))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                     sc   R-squared:                       0.622
Model:                            OLS   Adj. R-squared:                  0.622
Method:                 Least Squares   F-statistic:                     4305.
Date:                Mon, 20 Jul 2020   Prob (F-statistic):               0.00
Time:                        22:29:17   Log-Likelihood:                 29496.
No. Observations:               15687   AIC:                        -5.898e+04
Df Residuals:                   15680   BIC:                        -5.892e+04
Df Model:                           6
Covariance Type:            nonrobust
================================================================================
=
                    coef    std err          t      P>|t|      [0.025      0.97
5]
--------------------------------------------------------------------------------
-
Intercept         1.0237      0.001    805.053      0.000       1.021       1.02
6
std_d            -0.3050      0.005    -63.511      0.000      -0.314      -0.29
6
std_rs           -0.0087      0.004     -2.299      0.022      -0.016      -0.00
1
std_rw           -0.0046      0.004     -1.207      0.227      -0.012       0.00
3
std_d:std_rs     -0.1346      0.013    -10.762      0.000      -0.159      -0.11
0
std_d:std_rw      0.0034      0.012      0.277      0.782      -0.021       0.02
8
std_rs:std_rw    -0.0314      0.010     -3.086      0.002      -0.051      -0.01
1
==============================================================================
Omnibus:                      829.822   Durbin-Watson:                   1.467
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1020.324
Skew:                          -0.544   Prob(JB):                    2.75e-222
Kurtosis:                       3.613   Cond. No.                         49.4
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

how much does spl weak correlate with spl strong?

In [138]:
```
#load SPL on undirected conversion of graph
filename = 'sayama_change_all' + 'mod'
camod = analysis.read_pickle('df_' + filename)
camod.columns
```

Out[138]:
```
Index(['degrees', 'clusterCoeff', 'reciprocity', 'center_1', 'center_2',
       'mean_centers', 'overall_mean_culture', 'giantComponent', 'diam', 'SPL',
       'CD', 'c1_init', 'c2_init', 'c_avg_init', 'std_d', 'std_rs', 'std_rw',
       'tags', 'comps'],
      dtype='object')
```

In [140]:
```
camodR = camod.loc[dfca['sc'] >= 0.8]
```

In [141]:
```
dtemp = camodR['SPL']
dtemp.name = "SPLweak"
dtemp2 = dfcaR['SPL']
dtemp2.name = "SPLstrong"
dfspl = pd.concat([dtemp, dtemp2], axis=1)
dfspl.head()
```

Out[141]:

	SPLweak	SPLstrong
0	1.951020	2.432653
1	1.906122	2.347755
2	1.906939	2.315918
3	1.931429	2.431837
4	1.989388	2.409796

In [142]:
```
res = smf.ols(formula='SPLstrong ~ SPLweak', data=dfspl).fit()
print(str(res.summary()))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              SPLstrong   R-squared:                       0.848
Model:                            OLS   Adj. R-squared:                  0.848
Method:                 Least Squares   F-statistic:                 8.733e+04
Date:                Mon, 20 Jul 2020   Prob (F-statistic):               0.00
Time:                        22:34:29   Log-Likelihood:                 13139.
No. Observations:               15687   AIC:                        -2.627e+04
Df Residuals:                   15685   BIC:                        -2.626e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.6226      0.011    -56.629      0.000      -0.644      -0.601
SPLweak        1.5513      0.005    295.524      0.000       1.541       1.562
==============================================================================
Omnibus:                     7291.102   Durbin-Watson:                   1.960
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           163281.140
Skew:                           1.720   Prob(JB):                         0.00
Kurtosis:                      18.426   Cond. No.                         33.8
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

In [ ]: