1.3 Pearson's Correlation Coefficient Comparison

This is to test whether the synthetic data has captured dependencies between variables of the original data.
We calculate Pearson's correlation coefficients – *r*, between categorical columns within the original and synthetic data.

To compare the how similar those *r*'s are,

- We calculate the **MSE** between every pair of *r*'s (for the same pair of columns but one from synthetic one from real data)
- We calculate the **SRA** (Synthetic Ranking Agreement,see explanation below) of *r*'s for each column.

In [6]:

```
def r_corr_test(df,PTable = False, CoefficientandPtabl
    '''Returns a table of Pearson's r correlation coef

    Args:
    df: The input dataframe
    PTable: False (default) or True, if True, then the
    CoefficientandPtable: False(default) or True, if t
    lower: True(default) or False. If True, the lower

    Returns:
    The requested table as specified in the args. If P

    '''
    from scipy.stats import pearsonr
    import pandas as pd
    import numpy as np

    df_index = (df.keys()).tolist()
    n = len(df_index)
    ini = [ [ None for y in range( n ) ]
                  for x in range( n ) ]

    #pearsonr returns two values: the correlation coef
    #so we create two empty dataframes to store them
    coefficient_table = pd.DataFrame(ini,index = df_in
    p_table = coefficient_table.copy()
    coe_and_p_table = coefficient_table.copy()

    for i in range(n):
        for j in range(i+1,n):
            name1 = df_index[i]
            name2 = df_index[j]
            obs_1 = df[name1].dropna()
            obs_2 = df[name2].dropna()
```

```
            dataframe = pd.DataFrame({name1: obs_1, na
            values = dataframe.dropna().values
            (coe,p) = pearsonr(values[:,0],values[:,1]
            coefficient_table.loc[name1,name2]=coe
            p_table.loc[name1,name2]=p
            coe_and_p_table.loc[name1,name2]=(coe,p)

    if lower:
        #A function that can fill the lower part of th
        #But for comparison reasons you may want them
        def fill_lower(df):
            n = df.values.shape[0]
            for j in range(n):
                for i in range(j+1,n):
                    df.iloc[i,j]=df.iloc[j,i]
            return df

        coefficient_table = fill_lower(coefficient_tab
        p_table = fill_lower(p_table)
        coe_and_p_table = fill_lower(coe_and_p_table)


    if PTable:
        return p_table
    elif CoefficientandPtable:
        return coe_and_p_table
    else:
        return coefficient_table
```

What is SRA?

SRA is used when we want to test whether a synthetic dataset respects a certain ranking. Suppose we have a list of metrics $R_1, R_2, R_3 \ldots R_n$ calculated from the real data and a list of same metrics $S_1, S_2, S_3 \ldots S_n$ calculated from the synthetic data. Then we define **SRA** as

$$SRA(R,S) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} Id((S_i - S_j) \times (R_i - R_j) > 0)$$

where Id is the identity function. $SRA \in [0,1]$, The closer the SRA to 1, the better the ranking agreement.

In the case of correlation comparison, suppose we have columns *A, B, C, D, E* , for column *A*, we calculate correlation coefficients $r_{AB}, r_{AC}, r_{AD}, r_{AE}$ for the real data, and $r'_{AB}, r'_{AC}, r'_{AD}, r'_{AE}$ for the synthetic data. We hope the ranking of r and r' agrees, e.g. if $r_{AB} > r_{AC}$ then $r'_{AB} > r'_{AC}$ as well. As a result, our R is $r_{AB}, r_{AC}, r_{AD}, r_{AE}$, S is $r'_{AB}, r'_{AC}, r'_{AD}, r'_{AE}$, and we can calulate the SRA for each column.

In [7]:

```python
def SRA(R,S):
    '''Calculate the SRA of lists R and S

    Args:
    - R: A list of performance metrics of different pr
    - S: A list of performance metrics of different pr

    Returns:
    - SRA: SRA value

    '''
    def identity_function(statement):
        v = 0
        if statement:
            v = 1
        return v

    k = len(R)
    sum_ = 0
    for i in range(k):
        for j in range(k):
            if i != j:
                if (R[i]-R[j])==0:
                    if (S[i]-S[j])==0:
                        agree = True
                    else:
                        agree = False
                else:
                    agree = (R[i]-R[j])*(S[i]-S[j])>0
                sum_ += identity_function(agree)
    SRA = sum_ / (k*(k-1))
    return SRA
```

In [8]:

```
def CorrelationSRA(ori_correlation_df,gen_correlation_
    '''Returns the value of SRA for the absolute Pears
    all other columns. SRA is between 0 and 1, the clo
    the more similar the synthetic data and the real d

    Args:
    ori_correlation_df: the correlation coefficient da
                        r_corr_test.
    gen_correlation_df: the correlation coefficient da
                        r_corr_test.
    ColumnWise: False(default) or True. If True, the r
                Otherwise, the return is the average o

    Returns:
    s: It is either a column-wise SRA series or the av

    '''
    import numpy as np
    import pandas as pd

    columns = (ori_correlation_df.keys()).tolist()
    n = len(columns)
    ini = np.ones(n)

    for i in range(n):
        ori_values = ori_correlation_df.iloc[i,:].drop
        gen_values = gen_correlation_df.iloc[i,:].drop
        ini[i] = SRA(abs(ori_values), abs(gen_values))

    if ColumnWise:
        s = pd.Series(ini,index = columns)
        s['average'] = sum(ini)/n
    else:
        s = sum(ini)/n
```

```
    return s
```

```
def MSE(r_table_ori,r_table_gen):
    '''
    Returns the MSE for each position between two data
    '''
    import pandas as pd
    import numpy as np
    ori = r_table_ori.fillna(0).values
    gen = r_table_gen.fillna(0).values
    columns = (r_table_gen.keys()).tolist()
    matrix = (ori-gen)**2
    df = pd.DataFrame(matrix, index = columns, columns
    score = np.sum(matrix)/(len(ori)*(len(ori)-1)) #Th
    return df, score
```

```
# Data Loading
```

In [32]:

```
import numpy as np
import pandas as pd
dp_ori_df = pd.read_csv('synthetic data/doppelGANger/d
dp_gen_df = pd.read_csv('synthetic data/doppelGANger/d
tgan_ori_df = pd.read_csv('synthetic data/TGAN/tgan_or
tgan_gen_df = pd.read_csv('synthetic data/TGAN/tgan_ge
ori_df = pd.read_csv('synthetic data/2_no_id/ori_df.cs
gen_1_df = pd.read_csv('synthetic data/2_no_id/gen_1_d
gen_2_df = pd.read_csv('synthetic data/2_no_id/gen_2_d
gen_3_df = pd.read_csv('synthetic data/2_no_id/gen_3_d
gen_4_df = pd.read_csv('synthetic data/2_no_id/gen_4_d

synthetic_data_dic = {'DoppelGANger ini':[dp_ori_df, d
                      'gen 1':[tgan_ori_df,gen_1_df],'
                     'gen 4':[tgan_ori_df,gen_4_df]}
syn_keys = list(synthetic_data_dic.keys())
```

In [37]:

```
n = len(syn_keys)
MSE_array = np.zeros(n)
for i in range(n):
    key = syn_keys[i]
    df_ori = synthetic_data_dic[key][0]
    df_gen = synthetic_data_dic[key][1]
    r_table_ori = r_corr_test(df_ori)
    r_table_gen = r_corr_test(df_gen)

    #Highlight all r values > 0.5 as yellow, indicatin
    def color_threshold_yellow(val):
        threshold = 0.5
        if ((val != None) and (abs(val) > threshold)):
            color = 'yellow'
        else:
            color = 'black'
        return 'color: %s' % color

    display(key+' '+'generated r table',r_table_gen.st
    display(key+' '+'real r table',r_table_ori.style.a
    sra = CorrelationSRA(r_table_ori,r_table_gen,Colum
    if i==0:
        sra_df = pd.DataFrame(sra,columns = [key])
    else:
        sra_df = pd.concat([sra_df,pd.DataFrame(sra,co
    display(key+' '+'SRA',sra)
    MSE_df, MSE_score = MSE(r_table_gen,r_table_ori)
    display(key+' '+'MSE table', MSE_df)
    MSE_array[i] = MSE_score
MSE_series = pd.Series(MSE_array,index = syn_keys)
```

```
'DoppelGANger ini generated r table'
```

	dday	weight	height	age	temp
dday	None	0.460590	0.579901	0.385757	0.572924
weight	0.460590	None	0.852360	0.765801	0.725225
height	0.579901	0.852360	None	0.700063	0.951644
age	0.385757	0.765801	0.700063	None	0.561933
temp	0.572924	0.725225	0.951644	0.561933	None

```
'DoppelGANger ini real r table'
```

	dday	weight	height	age	temp
dday	None	0.547442	0.625742	0.431480	0.624604
weight	0.547442	None	0.904009	0.888397	0.787127
height	0.625742	0.904009	None	0.739106	0.964485
age	0.431480	0.888397	0.739106	None	0.589636
temp	0.624604	0.787127	0.964485	0.589636	None

```
'DoppelGANger ini SRA'
```

```
dday       1.0
weight     1.0
height     1.0
age        1.0
temp       1.0
average    1.0
dtype: float64
```

```
'DoppelGANger ini MSE table'
```

	dday	weight	height	age	temp
dday	0.000000	0.007543	0.002101	0.002091	0.002671
weight	0.007543	0.000000	0.002668	0.015030	0.003832
height	0.002101	0.002668	0.000000	0.001524	0.000165
age	0.002091	0.015030	0.001524	0.000000	0.000767
temp	0.002671	0.003832	0.000165	0.000767	0.000000

```
'tGAN generated r table'
```

	dday	height	weight	temp	vomit_dur
dday	None	0.630282	0.674201	−0.518343	0.509896
height	0.630282	None	0.967306	0.271986	0.840133
weight	0.674201	0.967306	None	0.264717	0.934518
temp	−0.518343	0.271986	0.264717	None	0.417741
vomit_dur	0.509896	0.840133	0.934518	0.417741	None
cough_dur	0.106260	0.607276	0.704238	0.737285	0.848396
diar_No	−0.508896	−0.476903	−0.642798	−0.165282	−0.777692
diar_Yes	0.505435	0.444277	0.610614	0.136410	0.744161
head_No	0.013767	−0.627081	−0.679343	−0.845591	−0.802912
head_Yes	−0.013388	0.627144	0.679569	0.845392	0.803213

```
'tGAN real r table'
```

	dday	height	weight	temp	vomit_dur
dday	None	0.176702	0.216054	−0.090951	−0.039075
height	0.176702	None	0.873474	−0.143156	−0.026711
weight	0.216054	0.873474	None	−0.122154	−0.018737
temp	−0.090951	−0.143156	−0.122154	None	0.125559
vomit_dur	−0.039075	−0.026711	−0.018737	0.125559	None
cough_dur	−0.091015	−0.059380	−0.041915	0.112293	0.020258
diar_No	0.017397	0.053020	0.015691	−0.048428	−0.147209
diar_Yes	−0.017397	−0.053020	−0.015691	0.048428	0.147209
head_No	0.069809	−0.108343	−0.093078	−0.281744	−0.086867
head_Yes	−0.069809	0.108343	0.093078	0.281744	0.086867

```
'tGAN SRA'
```

```
dday          0.694444
height        0.638889
weight        0.555556
temp          0.750000
vomit_dur     0.166667
cough_dur     0.583333
diar_No       0.500000
diar_Yes      0.500000
head_No       0.805556
head_Yes      0.805556
average       0.600000
dtype: float64
```

'tGAN MSE table'

	dday	height	weight	temp	vomit_dur	cou
dday	0.000000	0.205735	0.209899	0.182664	0.301369	0.0
height	0.205735	0.000000	0.008805	0.172343	0.751419	0.4
weight	0.209899	0.008805	0.000000	0.149669	0.908696	0.5
temp	0.182664	0.172343	0.149669	0.000000	0.085370	0.3
vomit_dur	0.301369	0.751419	0.908696	0.085370	0.000000	0.6
cough_dur	0.038918	0.444430	0.556745	0.390614	0.685812	0.0
diar_No	0.276985	0.280819	0.433608	0.013655	0.397508	0.5
diar_Yes	0.273354	0.247304	0.392258	0.007741	0.356351	0.4
head_No	0.003141	0.269089	0.343707	0.317924	0.512721	0.7
head_Yes	0.003183	0.269154	0.343971	0.317700	0.513152	0.7

'gen 1 generated r table'

	dday	height	weight	temp	vomit_du
dday	None	0.162797	0.097409	−0.339070	0.110246
height	0.162797	None	0.937405	−0.563485	−0.07500
weight	0.097409	0.937405	None	−0.514464	−0.04737
temp	−0.339070	−0.563485	−0.514464	None	0.589742
vomit_dur	0.110246	−0.075009	−0.047376	0.589742	None

	dday	height	weight	temp	vomit_du
cough_dur	0.021433	−0.798786	−0.693646	0.729940	0.560289
diar_No	0.225388	0.606384	0.488053	−0.108582	0.235987
diar_Yes	−0.242572	−0.623032	−0.500043	0.125227	−0.25027
head_No	0.366513	−0.024406	−0.007414	−0.748058	−0.64679
head_Yes	−0.368334	0.024948	0.008275	0.745608	0.647633

```
'gen 1 real r table'
```

	dday	height	weight	temp	vomit_dur
dday	None	0.176702	0.216054	−0.090951	−0.039075
height	0.176702	None	0.873474	−0.143156	−0.026711
weight	0.216054	0.873474	None	−0.122154	−0.018737
temp	−0.090951	−0.143156	−0.122154	None	0.125559
vomit_dur	−0.039075	−0.026711	−0.018737	0.125559	None
cough_dur	−0.091015	−0.059380	−0.041915	0.112293	0.020258
diar_No	0.017397	0.053020	0.015691	−0.048428	−0.147209
diar_Yes	−0.017397	−0.053020	−0.015691	0.048428	0.147209
head_No	0.069809	−0.108343	−0.093078	−0.281744	−0.086867
head_Yes	−0.069809	0.108343	0.093078	0.281744	0.086867

```
'gen 1 SRA'
```

```
dday          0.333333
height        0.527778
weight        0.527778
temp          0.861111
vomit_dur     0.611111
cough_dur     0.500000
diar_No       0.527778
diar_Yes      0.555556
head_No       0.722222
head_Yes      0.722222
average       0.588889
dtype: float64
```

'gen 1 MSE table'

	dday	height	weight	temp	vomit_dur	cou
dday	0.000000	0.000193	0.014077	0.061563	0.022297	0.0
height	0.000193	0.000000	0.004087	0.176676	0.002333	0.5
weight	0.014077	0.004087	0.000000	0.153907	0.000820	0.4
temp	0.061563	0.176676	0.153907	0.000000	0.215465	0.3
vomit_dur	0.022297	0.002333	0.000820	0.215465	0.000000	0.2
cough_dur	0.012645	0.546722	0.424753	0.381487	0.291634	0.0
diar_No	0.043260	0.306211	0.223126	0.003619	0.146840	0.1
diar_Yes	0.050704	0.324913	0.234597	0.005898	0.157996	0.1
head_No	0.088033	0.007046	0.007338	0.217449	0.313523	0.0
head_Yes	0.089117	0.006955	0.007192	0.215170	0.314458	0.0

'gen 2 generated r table'

	dday	height	weight	temp	vomit_dur
dday	None	0.669932	0.600616	−0.421997	−0.319321
height	0.669932	None	0.936421	−0.027013	0.160809
weight	0.600616	0.936421	None	−0.117218	0.049449
temp	−0.421997	−0.027013	−0.117218	None	0.861116
vomit_dur	−0.319321	0.160809	0.049449	0.861116	None
cough_dur	−0.365681	−0.505931	−0.566673	0.197196	0.291194
diar_No	0.303687	−0.073868	0.027920	−0.615479	−0.684354
diar_Yes	−0.324059	0.053488	−0.042256	0.611958	0.660524
head_No	0.355109	−0.323211	−0.260020	−0.551272	−0.723152
head_Yes	−0.355177	0.323054	0.259797	0.551744	0.723586

'gen 2 real r table'

	dday	height	weight	temp	vomit_dur
dday	None	0.176702	0.216054	−0.090951	−0.039075
height	0.176702	None	0.873474	−0.143156	−0.026711
weight	0.216054	0.873474	None	−0.122154	−0.018737
temp	−0.090951	−0.143156	−0.122154	None	0.125559
vomit_dur	−0.039075	−0.026711	−0.018737	0.125559	None
cough_dur	−0.091015	−0.059380	−0.041915	0.112293	0.020258
diar_No	0.017397	0.053020	0.015691	−0.048428	−0.147209
diar_Yes	−0.017397	−0.053020	−0.015691	0.048428	0.147209
head_No	0.069809	−0.108343	−0.093078	−0.281744	−0.086867
head_Yes	−0.069809	0.108343	0.093078	0.281744	0.086867

```
'gen 2 SRA'
```

```
dday          0.861111
height        0.694444
weight        0.805556
temp          0.361111
vomit_dur     0.750000
cough_dur     0.500000
diar_No       0.833333
diar_Yes      0.805556
head_No       0.527778
head_Yes      0.527778
average       0.666667
dtype: float64
```

```
'gen 2 MSE table'
```

	dday	height	weight	temp	vomit_dur	c
dday	0.000000	0.243275	0.147888	0.109591	0.078538	0.
height	0.243275	0.000000	0.003962	0.013489	0.035164	0.
weight	0.147888	0.003962	0.000000	0.000024	0.004649	0.
temp	0.109591	0.013489	0.000024	0.000000	0.541044	0.
vomit_dur	0.078538	0.035164	0.004649	0.541044	0.000000	0.

	dday	height	weight	temp	vomit_dur	c
cough_dur	0.075441	0.199408	0.275371	0.007208	0.073406	0.
diar_No	0.081962	0.016101	0.000150	0.321547	0.288525	0.
diar_Yes	0.094041	0.011344	0.000706	0.317566	0.263492	0.
head_No	0.081396	0.046168	0.027870	0.072646	0.404859	0.
head_Yes	0.081435	0.046101	0.027795	0.072900	0.405410	0.

```
'gen 3 generated r table'
```

	dday	height	weight	temp	vomit_dur
dday	None	−0.162625	−0.230949	0.090384	0.047434
height	−0.162625	None	0.767620	0.250086	0.004119
weight	−0.230949	0.767620	None	0.241961	−0.101985
temp	0.090384	0.250086	0.241961	None	0.074358
vomit_dur	0.047434	0.004119	−0.101985	0.074358	None
cough_dur	−0.064280	−0.098146	−0.036908	0.113351	−0.030081
diar_No	0.083909	0.314467	0.213542	0.351852	−0.051109
diar_Yes	−0.083909	−0.314467	−0.213542	−0.351852	0.051109
head_No	0.072363	0.170356	0.138130	0.293400	−0.047358
head_Yes	−0.072363	−0.170356	−0.138130	−0.293400	0.047358

```
'gen 3 real r table'
```

	dday	height	weight	temp	vomit_du
dday	None	−0.036634	0.000878	−0.059368	−0.02994
height	−0.036634	None	0.881265	−0.175099	−0.05927
weight	0.000878	0.881265	None	−0.160768	−0.04065
temp	−0.059368	−0.175099	−0.160768	None	0.178328
vomit_dur	−0.029941	−0.059278	−0.040652	0.178328	None
cough_dur	−0.051150	−0.101346	−0.063729	0.133677	0.045722

	dday	height	weight	temp	vomit_du
diar_No	0.021721	0.105971	0.055076	−0.051018	−0.172186
diar_Yes	−0.021721	−0.105971	−0.055076	0.051018	0.172186
head_No	0.048467	−0.149711	−0.139772	−0.200484	−0.05612
head_Yes	−0.048467	0.149711	0.139772	0.200484	0.056125

```
'gen 3 SRA'


dday          0.361111
height        0.750000
weight        0.638889
temp          0.444444
vomit_dur     0.555556
cough_dur     0.833333
diar_No       0.583333
diar_Yes      0.611111
head_No       0.638889
head_Yes      0.611111
average       0.602778
dtype: float64


'gen 3 MSE table'
```

	dday	height	weight	temp	vomit_dur	cou
dday	0.000000	0.015874	0.053744	0.022426	0.005987	0.0
height	0.015874	0.000000	0.012915	0.180783	0.004019	0.0
weight	0.053744	0.012915	0.000000	0.162191	0.003762	0.0
temp	0.022426	0.180783	0.162191	0.000000	0.010810	0.0
vomit_dur	0.005987	0.004019	0.003762	0.010810	0.000000	0.0
cough_dur	0.000172	0.000010	0.000719	0.000413	0.005746	0.0
diar_No	0.003867	0.043471	0.025112	0.162304	0.014660	0.0
diar_Yes	0.003867	0.043471	0.025112	0.162304	0.014660	0.0
head_No	0.000571	0.102443	0.077229	0.243921	0.000077	0.0
head_Yes	0.000571	0.102443	0.077229	0.243921	0.000077	0.0

```
'gen 4 generated r table'
```

	dday	height	weight	temp	vomit_dur
dday	None	0.725814	0.780287	−0.703732	−0.398647
height	0.725814	None	0.903226	−0.237532	−0.087795
weight	0.780287	0.903226	None	−0.289294	−0.045537
temp	−0.703732	−0.237532	−0.289294	None	0.486734
vomit_dur	−0.398647	−0.087795	−0.045537	0.486734	None
cough_dur	−0.284812	−0.554406	−0.315996	0.058875	0.440218
diar_No	0.345047	0.303831	0.183392	−0.363811	−0.142562
diar_Yes	−0.367659	−0.318263	−0.198528	0.400465	0.150752
head_No	0.300212	−0.093038	−0.116846	−0.476977	−0.921556
head_Yes	−0.299526	0.093150	0.117114	0.476242	0.921134

```
'gen 4 real r table'
```

	dday	height	weight	temp	vomit_dur
dday	None	0.176702	0.216054	−0.090951	−0.039075
height	0.176702	None	0.873474	−0.143156	−0.026711
weight	0.216054	0.873474	None	−0.122154	−0.018737
temp	−0.090951	−0.143156	−0.122154	None	0.125559
vomit_dur	−0.039075	−0.026711	−0.018737	0.125559	None
cough_dur	−0.091015	−0.059380	−0.041915	0.112293	0.020258
diar_No	0.017397	0.053020	0.015691	−0.048428	−0.147209
diar_Yes	−0.017397	−0.053020	−0.015691	0.048428	0.147209
head_No	0.069809	−0.108343	−0.093078	−0.281744	−0.086867
head_Yes	−0.069809	0.108343	0.093078	0.281744	0.086867

```
'gen 4 SRA'
```

```
dday          0.611111
height        0.694444
weight        0.694444
```

```
temp          0.527778
vomit_dur     0.555556
cough_dur     0.444444
diar_No       0.583333
diar_Yes      0.583333
head_No       0.694444
head_Yes      0.694444
average       0.608333
dtype: float64


'gen 4 MSE table'
```

	dday	height	weight	temp	vomit_dur	cou
dday	0.000000	0.301524	0.318359	0.375500	0.129292	0.0
height	0.301524	0.000000	0.000885	0.008907	0.003731	0.2
weight	0.318359	0.000885	0.000000	0.027936	0.000718	0.0
temp	0.375500	0.008907	0.027936	0.000000	0.130447	0.0
vomit_dur	0.129292	0.003731	0.000718	0.130447	0.000000	0.1
cough_dur	0.037557	0.245051	0.075120	0.002854	0.176366	0.0
diar_No	0.107354	0.062906	0.028124	0.099466	0.000022	0.0
diar_Yes	0.122683	0.070354	0.033429	0.123930	0.000013	0.0
head_No	0.053085	0.000234	0.000565	0.038116	0.696706	0.0
head_Yes	0.052770	0.000231	0.000578	0.037830	0.696002	0.0

In [34]:

```
display('MSE values of r for each synthetic data',MSE_
```

```
'MSE values of r for each synthetic data'
```

```
DoppelGANger ini    0.003839
tGAN                0.315190
gen 1               0.129868
gen 2               0.111499
gen 3               0.048782
gen 4               0.098235
dtype: float64
```

Conclusion

- The lower the value of MSE, the smaller the average difference between the correlation coefficients between the real and synthetic data, hence the better the result.
- DoppelGANger ini is good but it contains only 5 columns.
- tGAN, gen 1 and gen 2 have large MSE values, by comparing the correlation tables we find that these generated datas have wronly too strong dependencies (r>0.5) between some columns.
- gen 3 agrees with its original data in all r>0.5.

In [35]:

```
1 display('SRA for each column and synthetic data',sra_d
```

'SRA for each column and synthetic data'

	DoppelGANger ini	tGAN	gen 1	gen 2	gen 3
dday	1.0	0.694444	0.333333	0.861111	0.361111
weight	1.0	0.555556	0.527778	0.805556	0.638889
height	1.0	0.638889	0.527778	0.694444	0.750000
age	1.0	NaN	NaN	NaN	NaN
temp	1.0	0.750000	0.861111	0.361111	0.444444
average	1.0	0.600000	0.588889	0.666667	0.602778
vomit_dur	NaN	0.166667	0.611111	0.750000	0.555556
cough_dur	NaN	0.583333	0.500000	0.500000	0.833333
diar_No	NaN	0.500000	0.527778	0.833333	0.583333
diar_Yes	NaN	0.500000	0.555556	0.805556	0.611111
head_No	NaN	0.805556	0.722222	0.527778	0.638889
head_Yes	NaN	0.805556	0.722222	0.527778	0.611111

Conclusion

- we can conclude that DoppleGANger ini preserves the dependency ranking between columns very well.
- By comparing the 'average', gen 2 is best at preserve the ranking. Though in the previous section we find that gen 2 tends to have a large r value in average.

Possible Improvements in this method

Note in the TGAN data, it contains categorical columns e.g. diar_No and diar_Yes between which the r is –1. This corresponding to the logic fact that if diar_No = 1, then diar_Yes = 0; if diar_No = 0, then diar_Yes = 1. A reasonable synthetic data has to respect this kind of 'logic' relationship.

As a result, it's reasonable to say that that larger the absolute value of r is, the more important the relationship is, that's why we choose to use MSE rather than MAE in quantitative evaluation, a possible improvement is adding weight according to r rather than averaging the MSE.

Reference:

- James Jordon, Jinsung Yoon, Mihaela van der Schaar. PATE-GAN: GENERATING SYNTHETIC DATA WITH DIFFERENTIAL PRIVACY GUARANTEES (https://openreview.net/pdf?id=S1zk9iRqF7)