Compute the rankability of an unweighted graph read from a file.

```
% Add the paths needed for this tutorial
addpath('exhaustive')
addpath('lp')
```

```
D = csvread('data/unweighted1.csv');
disp('The size of the D matrix is:')
```

```
The size of the D matrix is:
```

```
disp(size(D))
```

```
     8     8
```

```
D
```

```
D =
     0     0     0     0     0     0     0     0
     1     0     0     0     0     0     1     1
     1     0     0     1     1     0     0     0
     0     1     0     0     1     0     0     1
     1     0     0     0     0     0     0     0
     1     1     0     1     0     0     0     1
     1     0     1     0     0     0     0     1
     1     0     1     0     0     1     0     0
```

```
[k,p,P,stats] = rankability_exhaustive(D,'transform',true);
```

```
fprintf('k=%d, p=%d, r=%f, rtransformed=%f\n',k,p,stats.r,stats.rtransformed)
```

```
k=12, p=1, r=0.999989, rtransformed=0.571429
```

Compute the rankability of a weighted graph read from a file

```
D = csvread('data/weighted1.csv');
disp(['The size of the D matrix is:'])
```

```
The size of the D matrix is:
```

```
disp(size(D))
```

```
     8     8
```

```
D
```

```
D =
     0     0    79    82    74     0    61    79
     0     0    79    89    89    74     0     0
    21    21     0    68    63    68     0    66
    18    11    32     0    87    63    61    76
    26    11    37    13     0     0    55     0
     0    26    32    37     0     0    74    76
    39     0    34    39    45    26     0    84
    21    16    34     0    21    24    16     0
```

```
[k,p,P,stats] = rankability_exhaustive(D,'transform',true);
```

```
fprintf('k=%d, p=%d, r=%f, rtransformed=%f\n',k,p,stats.r,stats.rtransformed)
```

```
k=1523, p=2, r=0.999970, rtransformed=0.194499
```

For larger graphs and datasets, the parallel implementation can be called:

```
[k,p,P,stats] = rankability_exhaustive_parallel(D,10,'transform',true);
```

```
fprintf('k=%d, p=%d, r=%f, rtransformed=%f\n',k,p,stats.r,stats.rtransformed);
```

```
k=1523, p=2, r=0.999970, rtransformed=0.194499
```

Standard examples from the paper

Strong Dominance

```
D=[0 1 1 1 1 1 1 1; 0 0 1 1 1 1 1 1; 0 0 0 1 1 1 1 1; 0 0 0 0 1 1 1 1; 0 0 0 0 0 1 1 1; 0 0 0
[k,p,P,stats] = rankability_exhaustive(D,'transform',true);
fprintf('k=%d, p=%d, r=%f, rtransformed=%f\n',k,p,stats.r,stats.rtransformed);
```

```
k=0, p=1, r=1.000000, rtransformed=1.000000
```

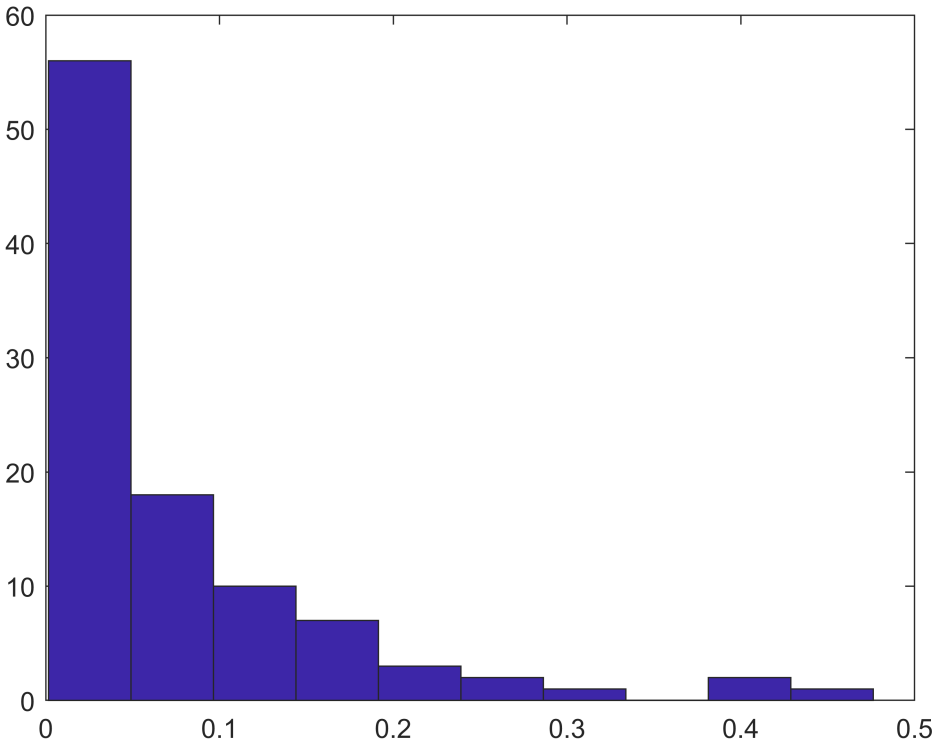
Fully Connected

```
D=ones(7,7);
for i = 1:size(D,1)
    D(i,i) = 0;
end
[k,p,P,stats] = rankability_exhaustive(D,'transform',true);
fprintf('k=%d, p=%d, r=%f, rtransformed=%f\n',k,p,stats.r,stats.rtransformed);
```

```
k=21, p=5040, r=0.000000, rtransformed=0.000000
```

Random graphs

```
ntimes = 100;
rtransformed = zeros(1,ntimes);
for j = 1:ntimes
    D=round(rand(7,7));
    for i = 1:size(D,1)
        D(i,i) = 0;
    end
    [k,p,P,stats] = rankability_exhaustive(D,'transform',true);
    rtransformed(j) = stats.rtransformed;
end
hist(rtransformed);
```



Weak Dominance

```
D=zeros(7,7);
for i = 1:size(D,1)-1
    D(i,i+1) = 1;
end
[k,p,P,stats] = rankability_exhaustive(D,'transform',true);
fprintf('k=%d, p=%d, r=%f, rtransformed=%f\n',k,p,stats.r,stats.rtransformed);
```

```
k=15, p=1, r=0.999858, rtransformed=0.285714
```