

```
% C: an nx5 table of probabilities. Each row is five non-negative numbers
% that add up to 1, representing the proportions of the five ethnicities in
% one census tract. There are as 415 rows (we removed rows containing NA
% values and rows where the race of the victim was unknown).
```

```
% E: same as C, except that the proportions come from the population of
% victims. Rows in E correspond to the rows in C, i.e. they come
% from the same census tract.
```

```
% Define a statistic 'S', which is the L1 distance between E and C
```

```
% Build a null distribution for S by creating "surrogate" versions of E.
% For example, let SE be an nx5 table with rows that correspond to the rows
% of C and E, except that the entries are random and come from random
% samples from the distributions represented in C. Each row of SE is
% determined from 'ESums' selections from the C distribution, where ESums
% is the number of victims recorded in the corresponding tract.
```

```
% Get the data (Tract data and victim data)
```

```
C=csvread('RacesOfCounties.csv',1);
E=csvread('RacesOfVictims.csv',1);
```

```
NumSamples=1000;
[NumTracts,NTypes]=size(C);
```

```
% Clean C, meaning force total in each row to be 1
```

```
CSums=sum(C,2);
for col=1:NTypes
    C(:,col)=C(:,col)./CSums;
end
```

```
% Get number of attacks in each row, and then normalize rows of E
```

```
ESums=sum(E,2);
for col=1:NTypes
    E(:,col)=E(:,col)./ESums;
end
```

```
S=sum(sum(abs(E-C))); % The observed value of the statistic
```

```
% Make surrogate NumSamples surrogate E's and compute, for each, a
% surrogate S ('SS')
```

```
SS=zeros(NumSamples,1);
```

```
for samp=1:NumSamples
```

```
    ES=zeros(NumTracts,NTypes); % ES will hold the current surrogate E
```

```
    for row=1:NumTracts
        r = mnrnd(ESums(row),C(row,:)); % Multinomial selection of
        % number of victims, but from the distribution in C
```

```
    % load the victim numbers into the surrogate, ES
```

```
    for col=1:NTypes
        ES(row,col)=r(col);
```

```
        end
    end

    % Normalize the rows of ES (make them probability distributions
    ESSums=sum(ES,2);
    for col=1:NTypes
        ES(:,col)=ES(:,col)./ESSums;
    end

    % Compute the L1 distance between C and the surrogate ES
    SS(samp)=sum(sum(abs(ES-C)));

end

% Display Results

figure(1)
close(1)
figure(1)
hist(SS);
hold on
scatter(S,0,100,'filled','r')
hold off
pvalue=(sum(SS>=S)+1)/(NumSamples+1);
disp(['p-value: ',num2str(pvalue)])
```