GAUSSian FIT

```
Assuming data was sampled from a Gaussian distribution, returns the most
likely parameters for the underlying distribution.
Input:
X - A D-by-N matrix with observation locations in each column (thus the
        observations are in D-dimensions and there are N of them).
Output:
mu - D-by-1 vector indicating the center of the Gaussian distribution.
sigma - Scalar indicating the standard deviation of the Gaussian distribution.
```

```matlab
function [mu, sigma] = gaussfit(X)
    %extract (D, N) dimensions of matrix X
    [D,N] = size(X);

    %extract MLE mean for every row and stores into Dx1 vector mu
    mu = mean(X,2);

    %extract MLE variance
    %repmat(mu,[1 N]) is a matrix with N copies of mu along columns
    %so what sum_sq effectively computes is sum_i(l2norm(x_i-mu)^2)
    sum_sq = sum((X-repmat(mu,[1 N])).^2, 'all');

    %normalize then compute square root
    sigma = sqrt(sum_sq/(D*N));
end
```

Published with MATLAB® R2020b

Contents

1.2 load employees data

```
load employees.mat
```

1.3 learn mean, std.dev for each department (assuming gaussian)

```
num_dept = max(dept(:));

%initialize row vectors of mean & std dev of salary per department
%mu is also a row vector since salary is only 1 parameter
mu = zeros(1,num_dept);
sigma = zeros(1,num_dept);

%for each department, compute mu, sigma using gauss_fit function
for i = 1:num_dept
    % for each i, sal(dept==i) indicates only to consider the sliced
    % vector of salary which correspond to index of ith department
    [mu(i),sigma(i)] = gaussfit(sal(dept==i));
end
```

1.4 extract dept with highest and lowest mean salary

```
[max_mean_val,max_mean_IDX] = max(mu);
[min_mean_val,min_mean_IDX] = min(mu);

fprintf('Dept %d has max mean salary of %d\n', max_mean_IDX, max_mean_val);
fprintf('Dept %d has min mean salary of %d\n', min_mean_IDX, min_mean_val);
```

```
Dept 29 has max mean salary of 9.408353e+04
Dept 8 has min mean salary of 3.863530e+04
```

1.5.1 extract dept with highest and lowest variance in salary

```
[max_sig_val,max_sig_IDX] = max(sigma);
[min_sig_val,min_sig_IDX] = min(sigma);
fprintf('Dept %d has max salary std.dev of %d\n', max_sig_IDX, max_sig_val);
fprintf('Dept %d has min salary std.dev of %d\n', min_sig_IDX, min_sig_val);
```

```
Dept 14 has max salary std.dev of 4.291664e+04
Dept 35 has min salary std.dev of 0
```

1.5.2 observation for standard deviation

```
%we noticed the std.dev of dept 35 is 0, and so it is very likely that the
%sample for dept 35 contains only 1 person. We verify this below
fprintf('Dept 35 has sample size of %d\n', sum(dept == 35));
```

```
Dept 35 has sample size of 1
```

Published with MATLAB® R2020b

```
% Kernel Density Estimation
%   Samples the kernel density estimate of a probability distribution using the
%  data in X with Gaussian kernel of standard deviation h. Samples are calculated
%  for each location in C.
% Input:
%   X - A D-by-N matrix with observation locations in each column (thus the
%       observations are in D-dimensions and there are N of them).
%   h - A number indicating the standard deviation of the Gaussian kernel used.
%   C - Locations to evaluate the estimated distribution. Hence D-by-M, where if
%      M = 1 this function calculates the KDE at one location.
% Output:
%   E - Evaluation of the estimated distribution at each of M locations given by
%      the input C. Should be returned as a column vector.


function [E] = kde(X, h, C)
    [D,N] = size(X);
    [~,M] = size(C);

    E = zeros(M,1);

    %adjustment multiplier
    K = 1/(N*((sqrt(2*pi)*h)^D));

    for i = 1:M
        %first get row vector w/ l2 norm squared of xk-C(:,i) in each component k
        %then normalize by the gaussian coefficient
        xSUM = -sum((X - repmat(C(:,i),[1 N])).^2)/(2*h^2);

        %exponentiate row vector then accumulate into a scalar value
        E(i) = sum(exp(xSUM));
    end

    %adjust length M row vector by multiplier elementwise
    E = K*E;
end
```

Published with MATLAB® R2020b

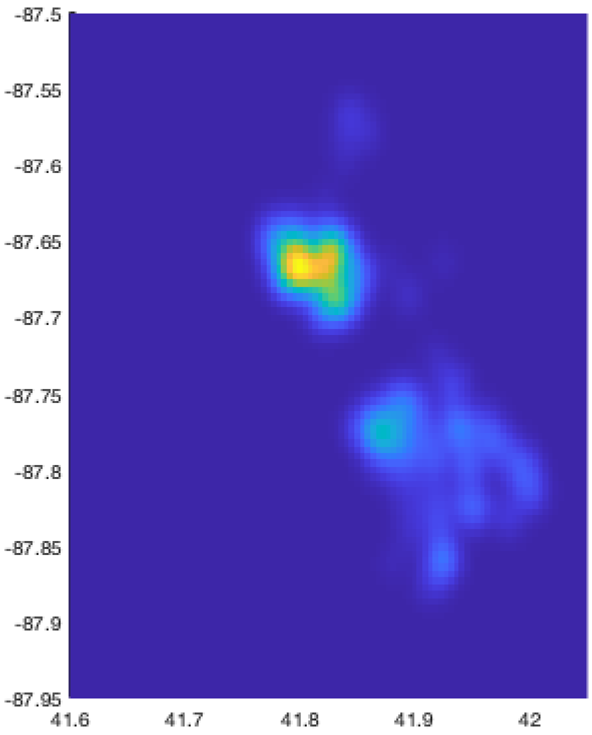
Contents

2.2 Load file

```
load crimes.mat
```

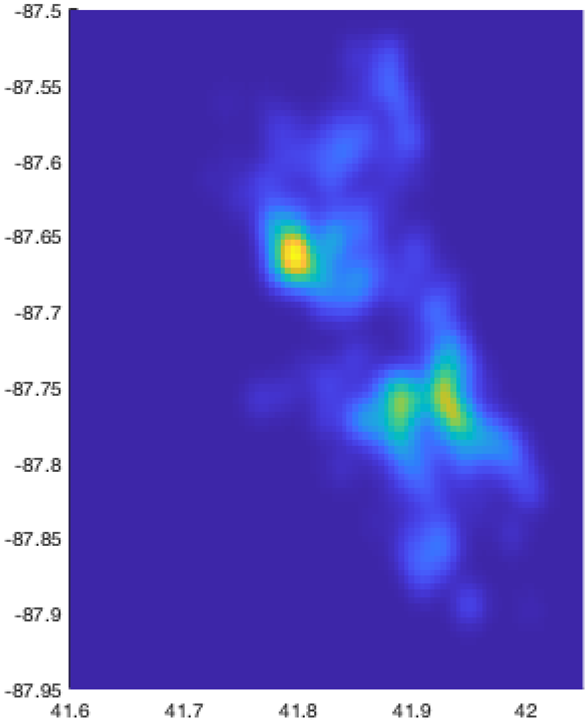
2.3 heat map of gambling crimes in 2014

```
h = 0.01;
N = 100;
map = kdemap(lat(type == 15 & year == 2014), lon(type == 15 & year == 2014), h, N);

%map boundary corresponding to those in kdemap.m
x = [41.6, 42.05];
y = [-87.95, -87.5];

%referenced code from TA, this declares the specification and plots
figure; hold on; set(gca, 'XLim', x, 'YLim', y);
imagesc(x, y, flipud(map));
daspect([1 cos(41/180*pi) 1]);
```

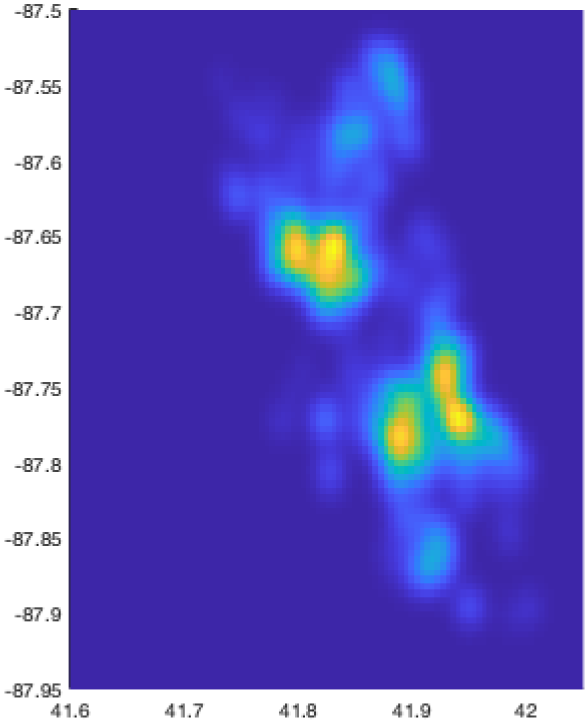


2.4, 2.5 gambling crimes in years 2001 to 2014, 14 heat maps

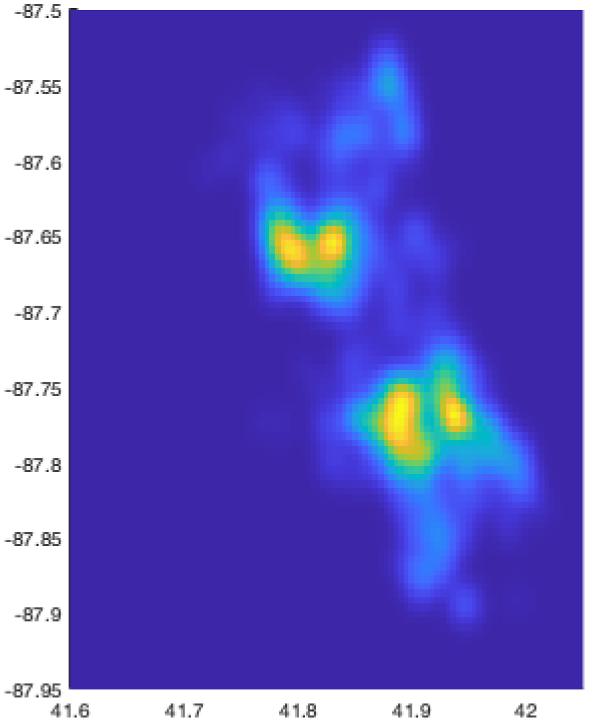
```
h = 0.01;
N = 100;

%map boundary corresponding to those in kdemap.m
x = [41.6, 42.05];
y = [-87.95, -87.5];

%2.5 explanation/observation:
%As we can see from the heat map generated for the 14 years, one of the central
%location near (41.8, -87.66) remains a place where gambling crime is heavy
%throughout the years, but gambling crime fell in surrounding central location,
%as seen by the fading from yellow/lime/light blue to dark blue indicating
%dissapearance of crimes in those areas as year approach 2014.
for yr = 2001:2014
    map = kdemap(lat(type == 15 & year == yr), lon(type == 15 & year == yr), h, N);
    figure; hold on; set(gca, 'XLim', x, 'YLim', y);
    imagesc(x, y, flipud(map));
    daspect([1 cos(41/180*pi) 1]);
    title('Gambling crimes in year', yr);
end
```



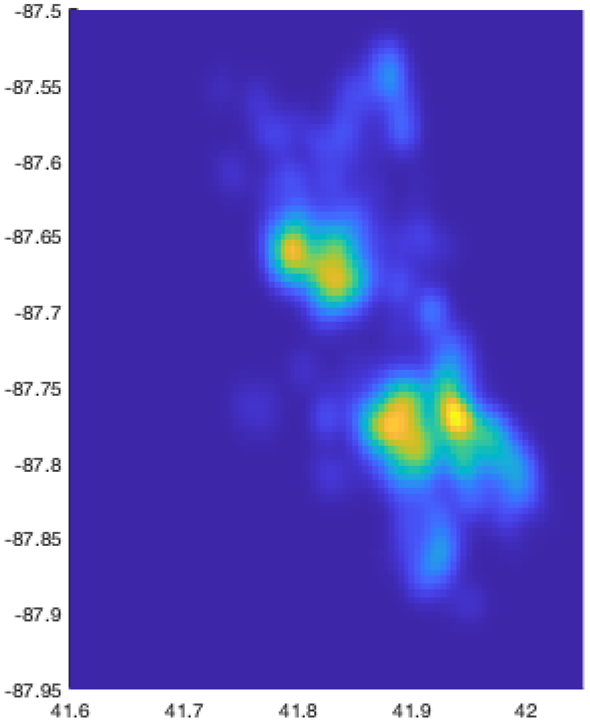
Gambling crimes in year
2001



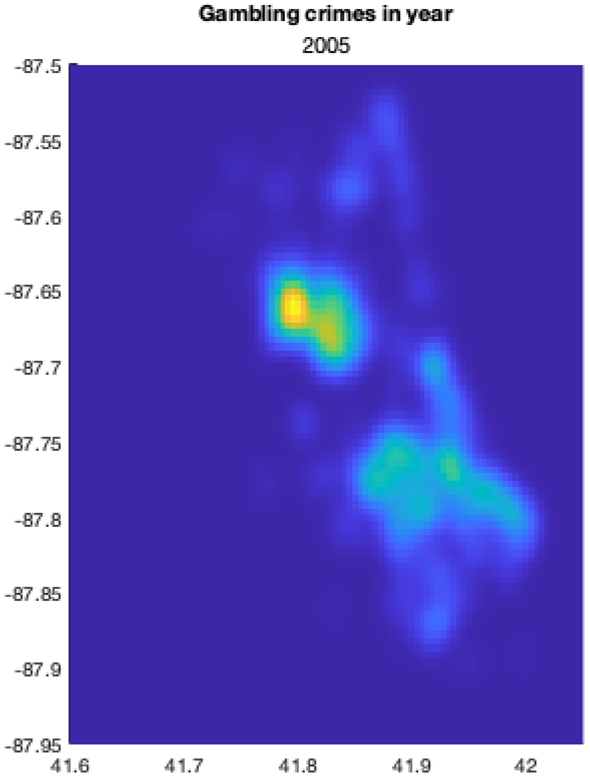
Gambling crimes in year
2002

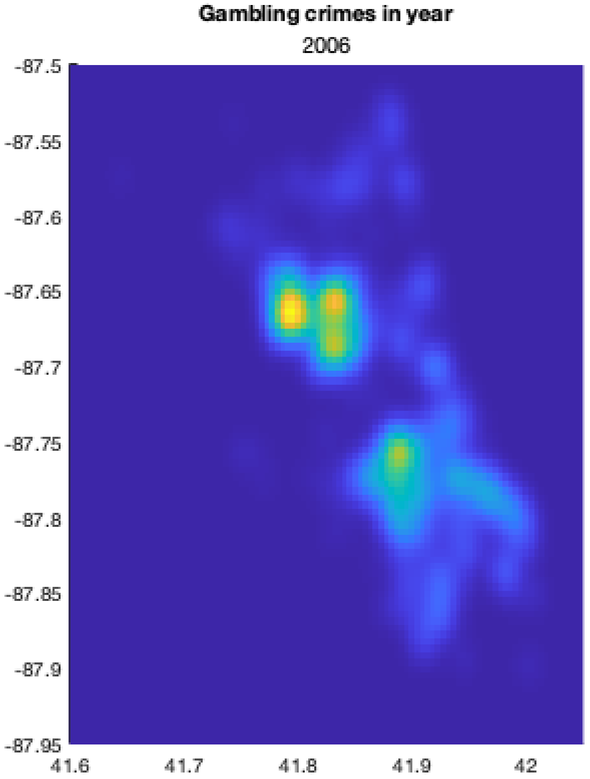


Gambling crimes in year
2003

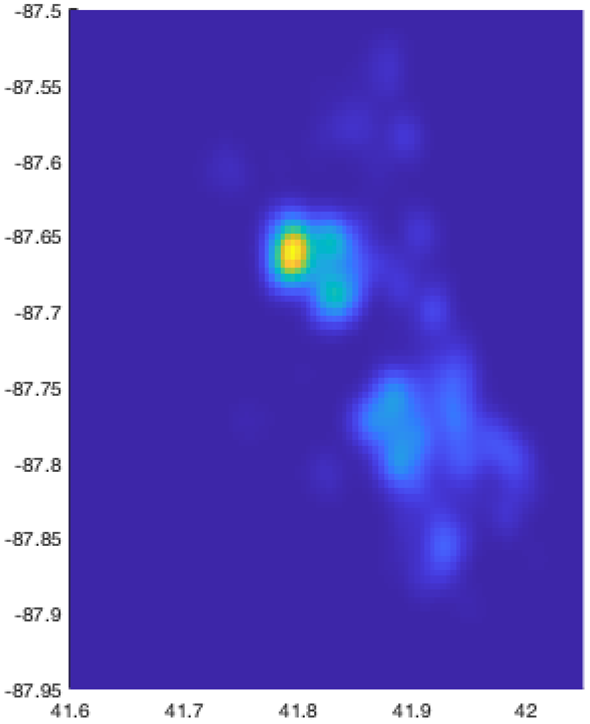


Gambling crimes in year
2004

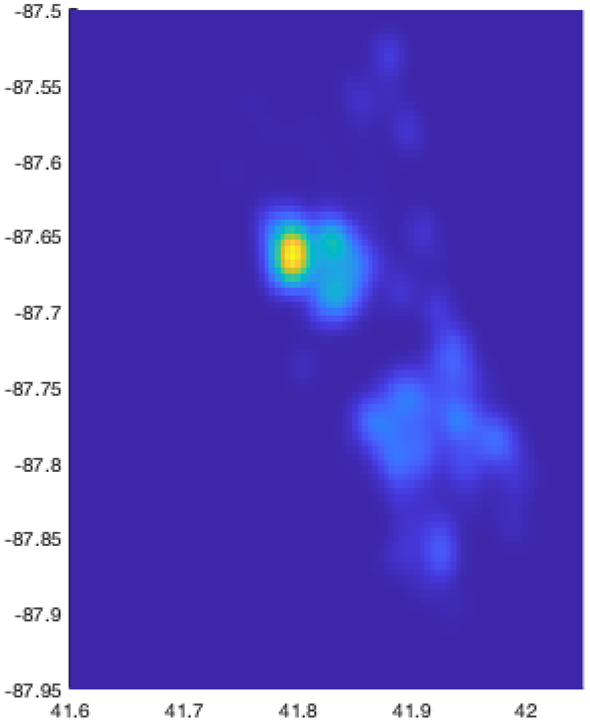

Gambling crimes in year
2005
-87.5
-87.55
-87.6
-87.65
-87.7
-87.75
-87.8
-87.85
-87.9
-87.95
41.6
41.7
41.8
41.9
42

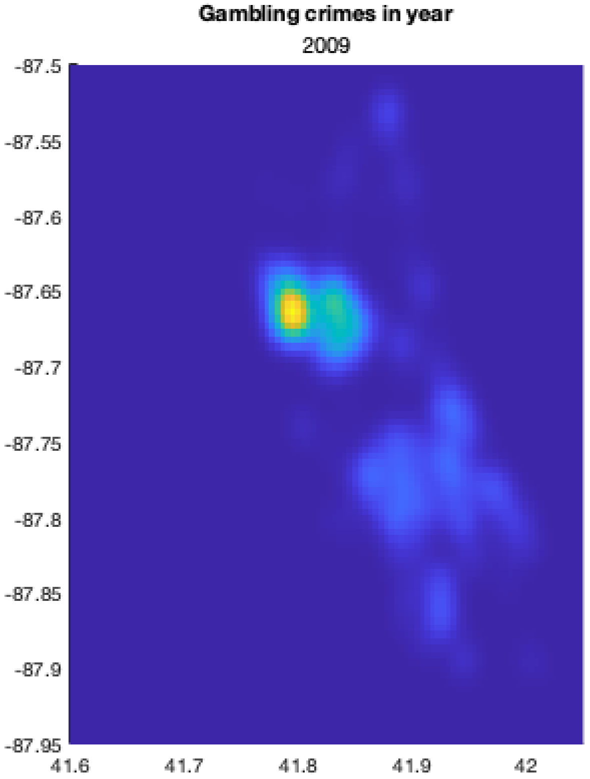

Gambling crimes in year
2006
-87.5
-87.55
-87.6
-87.65
-87.7
-87.75
-87.8
-87.85
-87.9
-87.95
41.6
41.7
41.8
41.9
42

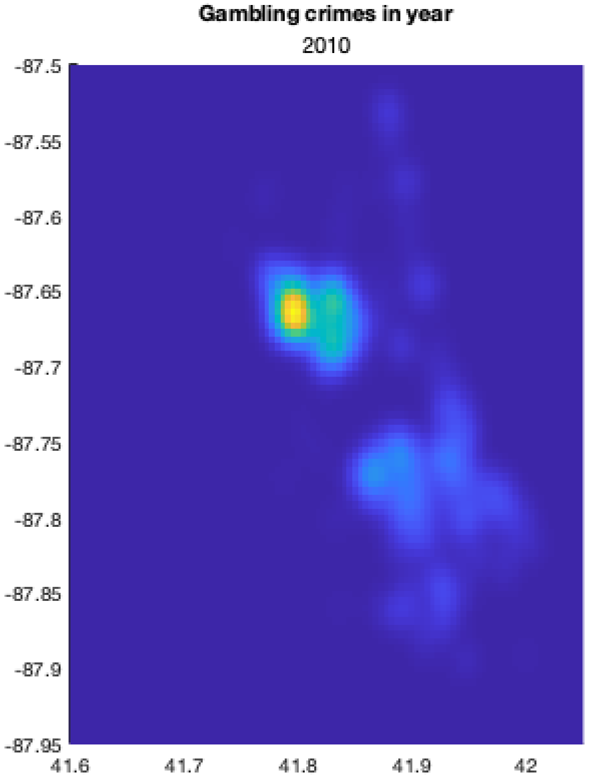
Gambling crimes in year
2007

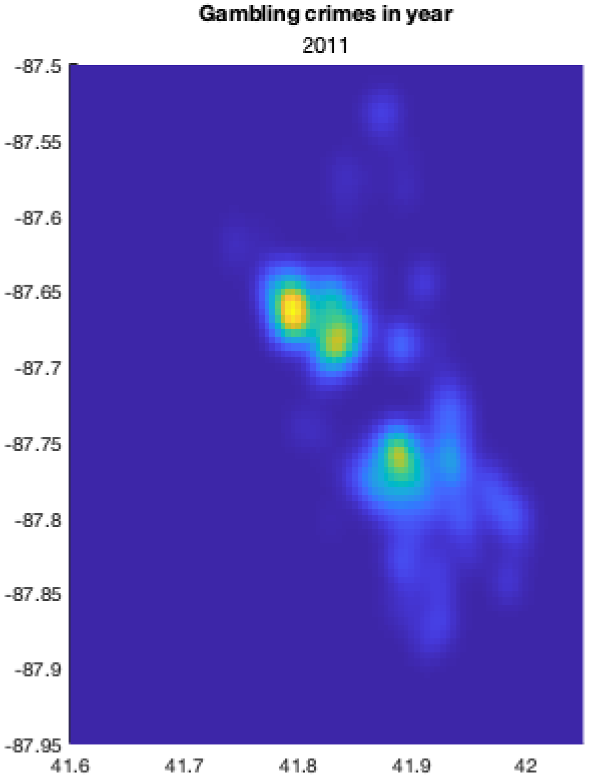


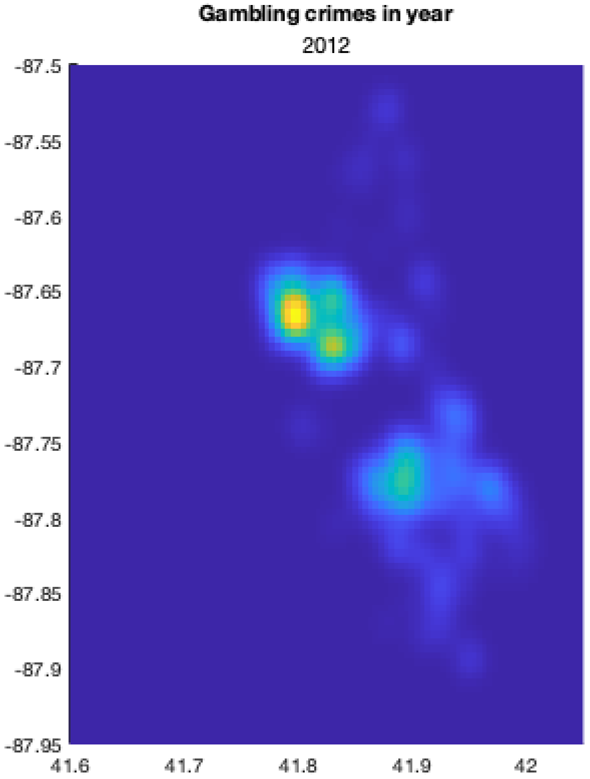
Gambling crimes in year
2008

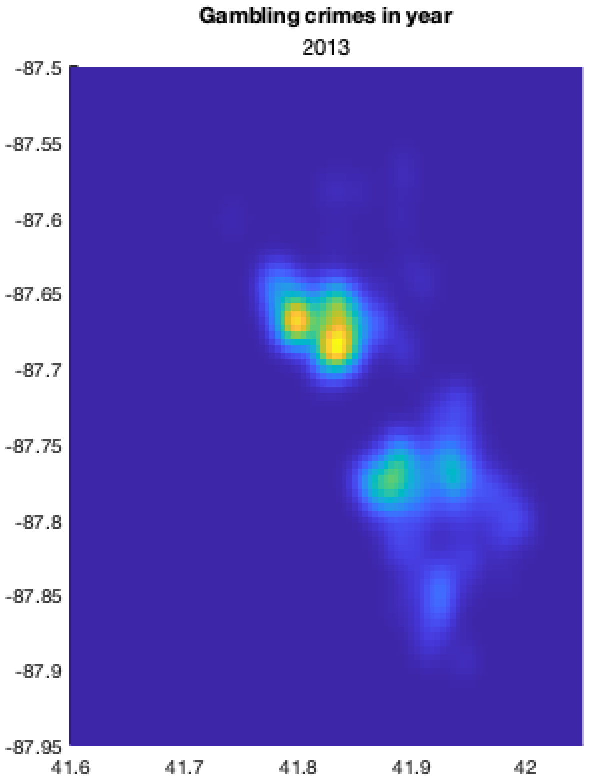


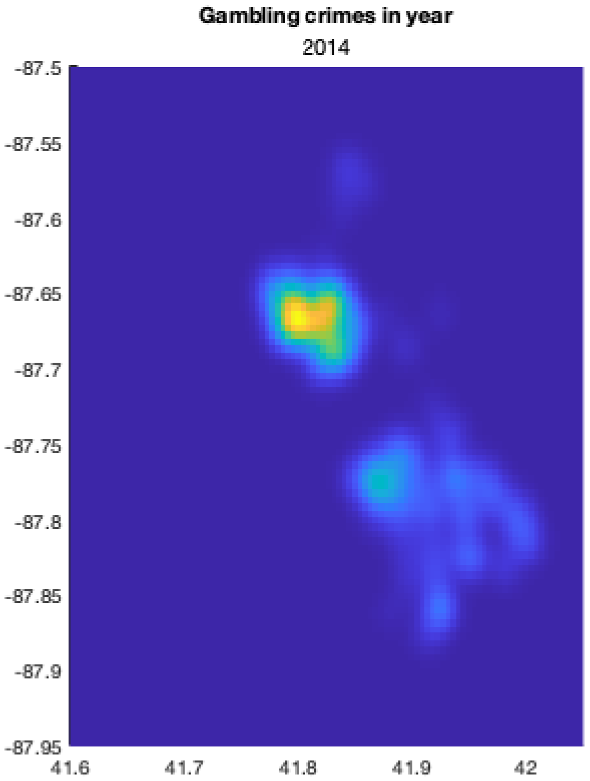

Gambling crimes in year
2009
-87.5
-87.55
-87.6
-87.65
-87.7
-87.75
-87.8
-87.85
-87.9
-87.95
41.6
41.7
41.8
41.9
42


Gambling crimes in year
2010
-87.5
-87.55
-87.6
-87.65
-87.7
-87.75
-87.8
-87.85
-87.9
-87.95
41.6
41.7
41.8
41.9
42


Gambling crimes in year
2011
-87.5
-87.55
-87.6
-87.65
-87.7
-87.75
-87.8
-87.85
-87.9
-87.95
41.6
41.7
41.8
41.9
42


Gambling crimes in year
2012
-87.5
-87.55
-87.6
-87.65
-87.7
-87.75
-87.8
-87.85
-87.9
-87.95
41.6
41.7
41.8
41.9
42



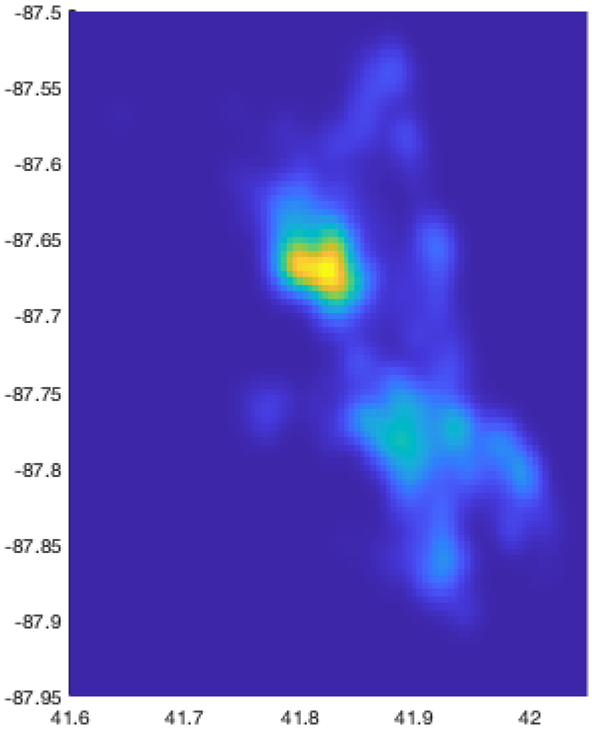


page padding(this is for pdf formatting help, ignore)

```
%hello
```

2.6 KDE visualization (heat map) for interference w/ officer (type 1) crime in 2014

```
h = 0.01;
N = 100;
map = kdemap(lat(type == 1 & year == 2014), lon(type == 1 & year == 2014), h, N);

%map boundary corresponding to those in kdemap.m
x = [41.6, 42.05];
y = [-87.95, -87.5];

%referenced code from TA, this declares the specification and plots
figure; hold on; set(gca, 'XLim', x, 'YLim', y);
imagesc(x, y, flipud(map));
daspect([1 cos(41/180*pi) 1]);
```



Published with MATLAB® R2020b