% 设置时间点

time\_points = [2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020

];

% 初始化一个矩阵用于存储核密度估计结果

density\_matrix = zeros(length(time\_points), 100); % 100是用于绘制曲线的点的数量

% 用ksdensity进行一维核密度估计并存储结果

for i = 1:length(time\_points)

time\_data = eval(['data\_t' num2str(i)]);

[f, xi] = ksdensity(time\_data);

density\_matrix(i, :) = f;

end

%i = 1:length(time\_points)：这行代码是一个循环的开始，它初始化一个循环变量 i，该变量从1遍历到 length(time\_points)，即从1到时间点的总数。

%time\_data = eval(['data\_t' num2str(i)])：这行代码构建一个字符串，通过 eval 函数将字符串转换为相应的变量。例如，如果 i 为1，这将生成字符串 'data\_t1'，然后 eval 函数会将其转换为变量 data\_t1，这个变量就是第一个时间点对应的数据集。

%[f, xi] = ksdensity(time\_data)：这行代码对当前时间点的数据集 time\_data 进行一维核密度估计，得到核密度估计结果 f 和对应的数据点 xi。

%density\_matrix(i, :) = f：这行代码将得到的核密度估计结果 f 存储在 density\_matrix 矩阵的第 i 行上，即每一行对应一个时间点的核密度估计结果。

% 生成时间网格

[T, XI] = meshgrid(time\_points, xi);

%meshgrid 函数用于生成网格矩阵，通常用于在二维平面上创建坐标网格。

%time\_points 是你提供的时间点，这是一个向量，例如 [1, 2, 3]，代表三个时间点。

%xi 是核密度估计的数据点，也是一个向量，代表一维核密度估计的数据点。

%meshgrid(time\_points, xi) 会生成两个矩阵：

%T 矩阵包含了 time\_points 中的每个时间点，重复了 xi 中的长度次，形成一个矩阵。

%XI 矩阵包含了 xi 中的每个数据点，重复了 time\_points 中时间点的长度次，形成一个矩阵。

% 绘制核密度估计曲面

figure;

surf(XI, T, density\_matrix');

title('1D Kernel Density Estimation over Time');

xlabel('X');

ylabel('Time');

zlabel('Density Estimate');