实验一

# 2 引用当前地图文档

# Notes:需要在ArcGIS中的Python窗口运行该代码

import arcpy.mapping as mapping

mxd = mapping.MapDocument("CURRENT")

mxd.title = "NEW"

layers = mapping.ListLayers(mxd)

for lyr in layers:

print(lyr)

# 3 引用磁盘上的地图文档

import arcpy.mapping as mapping

path = r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2.mxd"

mxd = mapping.MapDocument(path)

mxd.title = "Crime\_Ch2\_new"

print(mxd.title)

mxd.saveACopy(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

mxd\_new = mapping.MapDocument(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

print(mxd\_new.title)

layers = mapping.ListLayers(mxd)

for lyr in layers:

print(lyr)

# 4 获取地图文档中的图层列表

# (1)当前地图文档

# Notes:需要在ArcGIS中的Python窗口运行该代码

import arcpy.mapping as mapping

mxd = mapping.MapDocument("CURRENT")

mxd.title = "NEW"

print(mxd\_new.title)

layers = mapping.ListLayers(mxd)

for lyr in layers:

print(lyr)

# (2)磁盘地图文档

import arcpy.mapping as mapping

path = r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2.mxd"

mxd = mapping.MapDocument(path)

layers = mapping.ListLayers(mxd)

for lyr in layers:

print(lyr)

# (3)遍历地图文档所有数据框的所有图层并将所有图层依次添加到每个数据框

# 谨慎操作，循环次数过多，ArcGIS容易崩溃

import arcpy

import arcpy.mapping as mapping

mxd = mapping.MapDocument(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

ldfs = mapping.ListDataFrames(mxd)

for ldf in ldfs:

print(ldf.name)

layers = mapping.ListLayers(mxd)

for lyr in layers:

mapping.AddLayer(ldf,lyr,"AUTO\_ARRANGE")

arcpy.RefreshActiveView()

arcpy.RefreshTOC()

print(lyr.name)

# (4)将第二个数据框的图层添加到第一个数据框

import arcpy.mapping as mapping

mxd = mapping.MapDocument(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

ldf1 = mapping.ListDataFrames(mxd)[0]

ldf2 = mapping.ListDataFrames(mxd)[1]

for lyr in ldf2:

mapping.AddLayer(ldf1,lyr,"AUTO\_ARRANGE")

mxd.save(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new\_AddLayer.mxd") # 语句有问题

print("添加完毕")

# 5 限制图层列表

import arcpy.mapping as mapping

mxd = mapping.MapDocument(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

ldfs = mapping.ListDataFrames(mxd)

# 输出Crime数据框以B开头的图层名

for ldf in ldfs:

if(ldf.name == 'Crime'):

layers = mapping.ListLayers(mxd,"B\*",ldf)

for lyr in layers:

print(lyr.name)

# 输出第一个数据框的所有图层名

for lyr in ldfs[0]:

print(lyr.name)

# 6 缩放所选要素

import arcpy

import arcpy.mapping as mapping

mxd = mapping.MapDocument(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

ldf = mapping.ListDataFrames(mxd)[0]

lyr = mapping.ListLayers(mxd,"Burg\*",ldf)[0]

ldf.extent = lyr.getSelectedExtent()

# ldf.zoomToSelectedFeatures()

arcpy.RefreshActiveView()

# 7 改变地图范围

import arcpy.mapping as mapping

mxd = mapping.MapDocument(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

for df in mapping.ListDataFrames(mxd):

if(df.name == 'Crime'):

layers = mapping.ListLayers(mxd,'Crime Density by School District',df)

for lyr in layers:

query = '"NAME"=\'Lackland ISD\''

lyr.definitionQuery = query

df.extent = lyr.getExtent()

# 8 添加、插入图层到地图文档

# (1)添加图层

import arcpy.mapping as mapping

mxd = mapping.MapDocument(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

df = mapping.ListDataFrames(mxd)[0]

layer = mapping.Layer(r"E:\ArcPyStudy\Data\ArcpyBook\data\School\_Districts.lyr")

mapping.AddLayer(df,layer,"AUTO\_ARRANGE")

# (2)插入图层

import arcpy.mapping as mapping

mxd = mapping.MapDocument(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

df = mapping.ListDataFrames(mxd,"Crime")[0]

refLayer = mapping.ListLayers(mxd,"Burg\*",df)[0] # 指定图层

insertLayer = mapping.Layer(r"E:\ArcPyStudy\Data\ArcpyBook\data\CityOfSanAntonio.gdb\Crimes2009")

mapping.InsertLayer(df,refLayer,insertLayer,"BEFORE")

# 9 更新图层的符号系统

import arcpy.mapping as mapping

mxd = mapping.MapDocument(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

df = mapping.ListDataFrames(mxd,"Crime")[0]

updateLayer = mapping.ListLayers(mxd,"Crime Density by School District",df)[0]

sourceLayer = mapping.Layer(r"E:\ArcPyStudy\Data\ArcpyBook\data\CrimeDensityGradSym.lyr")

mapping.UpdateLayer(df,updateLayer,sourceLayer,True)

实验二

# 1 创建布局元素的列表

import arcpy.mapping as mapping

mxd = mapping.MapDocument(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

for el in mapping.ListLayoutElements(mxd):

if el.name != "":

print(el.name)

# 3 限制返回的布局元素

import arcpy.mapping as mapping

mxd = mapping.MapDocument(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

for el in mapping.ListLayoutElements(mxd,"LEGEND\_ELEMENT","\*Crime\*"):

print(el.name)

# 4 更新布局元素的属性

import arcpy.mapping as mapping

mxd = mapping.MapDocument(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

elLeg = mapping.ListLayoutElements(mxd,"LEGEND\_ELEMENT","\*Crime\*")[0]

elLeg.title = "Crime by School District"

for item in elLeg.listLegendItemLayers():

print(item.name)

# 5 导出地图为PDF文件

import arcpy.mapping as mapping

mxd = mapping.MapDocument(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

mapping.ExportToPDF(mxd,r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.pdf")

print("Export Done")

# 6 导出地图为图像文件

import arcpy.mapping as mapping

mxd = mapping.MapDocument(r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.mxd")

for df in mapping.ListDataFrames(mxd):

if df.name == "Crime":

mapping.ExportToJPEG(mxd,r"E:\ArcPyStudy\Data\ArcpyBook\Ch2\Crime\_Ch2\_new.jpg",df)

print("Export Done")

实验三

# 1 使用ListTools()函数查看特定工具箱（例如分析工具箱）包含的工具

import arcpy

tools = arcpy.ListTools('\*\_analysis')

for tool in tools:

print(tool)

# 获取工具箱别名

import arcpy

boxlist = arcpy.ListToolboxes()

for box in boxlist:

print(box)

# 2 使用Clip\_analysis()函数进行裁切分析

import arcpy

in\_feature = r'E:\ArcPyStudy\Data\ArcpyBook\data\CityOfSanAntonio.gdb\Burglary'

clip\_feature = r'E:\ArcPyStudy\Data\ArcpyBook\Ch5\EdgewoodSD.shp'

out\_feature\_class = r'E:\ArcPyStudy\Data\ArcpyBook\Ch5\ClpBurglary2.shp'

arcpy.Clip\_analysis(in\_feature,clip\_feature,out\_feature\_class)

print("Output Done")

# ☆3 使用脚本实现一个工具的输出作为另一个工具的输入（工具链）

import arcpy

arcpy.env.workspace = r'E:\ArcPyStudy\Data\ArcpyBook\data\TravisCounty'

streams = r'E:\ArcPyStudy\Data\ArcpyBook\data\TravisCounty\Streams.shp'

streamsBuffer = r'StreamsBuffer.shp'

distance = '2640 Feet'

schools2mile = 'Schools.shp'

schoolsLyrFile = 'Schools2Mile\_lyr'

arcpy.Buffer\_analysis(streams,streamsBuffer,distance,'FULL','ROUND','ALL')

arcpy.MakeFeatureLayer\_management(schools2mile,schoolsLyrFile)

arcpy.SelectLayerByLocation\_management(schoolsLyrFile,'INTERSECT',streamsBuffer)

print("Done")

# ☆4 创建自定义地理处理工具实现从逗号分割的文本文件中读取数据，并将数据写入到已定义好的要素类中。

import arcpy,os

try:

# The Output Feature Class Name

outputFC = arcpy.GetParameterAsText(0)

# Template Feature Class

fClassTemplate = arcpy.GetParameterAsText(1)

# Get the file to open Path

f\_path = arcpy.GetParameterAsText(2)

arcpy.CreateFeatureclass\_management(

os.path.split(outputFC)[0],

os.path.split(outputFC)[1],

'point',

fClassTemplate)

# open file to read

with open(f\_path,'r') as f:

lstFires = f.readlines()

cur = arcpy.InsertCursor(outputFC)

cntr = 0

for fire in lstFires[1:]:

vals = fire.split(',')

latitude = float(vals[0])

longitude = float(vals[1])

confid = int(vals[2])

# create points

point = arcpy.Point(longitude,latitude)

feat = cur.newRow()

feat.shape = point

feat.setValue('CONFIDENCEVALUE',confid)

cur.insertRow(feat)

arcpy.AddMessage("Record number " + str(cntr) + "Written to Feature Class")

cntr += 1

finally:

del cur

f.close()

实验四

# 2 创建要素图层和表视图

# 要素图层

import arcpy

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\data\CityOfSanAntonio.gdb"

try:

flayer = arcpy.MakeFeatureLayer\_management("Burglary","Burglary\_Layer")

print("Create Done")

except:

print("Create Error")

# 表视图

import arcpy

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\data\CityOfSanAntonio.gdb"

try:

tView = arcpy.MakeTableView\_management("Crime2009Table","Crime2009TView")

print("Create Done")

except:

print("Create Error")

# 3 使用Select Layer by Attribute选择要素

import arcpy

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\data\CityOfSanAntonio.gdb"

try:

qry = '"SVCAREA" = \'North\''

flayer = arcpy.MakeFeatureLayer\_management("Burglary","Burglary\_Layer")

arcpy.SelectLayerByAttribute\_management(flayer)

cnt = arcpy.GetCount\_management(flayer)

print("The number of selected records is:" + str(cnt))

except:

print("Error")

# 4 使用Select Layer by Location选择要素

import arcpy

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\data\CityOfSanAntonio.gdb"

try:

flayer = arcpy.MakeFeatureLayer\_management("Burglary","Burglary\_Layer")

arcpy.SelectLayerByLocation\_management(flayer,"COMPLETELY\_WITHIN",r"E:\ArcPyStudy\Data\ArcpyBook\Ch7\EdgewoodSD.shp")

cnt = arcpy.GetCount\_management(flayer)

print("The number of selected records is:" + str(cnt))

except:

print("Error")

# 5 结合空间查询和属性查询选择要素

import arcpy

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\data\CityOfSanAntonio.gdb"

try:

qry = '"DOW" = \'Mon\''

flayer = arcpy.MakeFeatureLayer\_management("Burglary","Burglary\_Layer")

arcpy.SelectLayerByLocation\_management(flayer,"COMPLETELY\_WITHIN",r"E:\ArcPyStudy\Data\ArcpyBook\Ch7\EdgewoodSD.shp")

arcpy.SelectLayerByAttribute\_management(flayer,"SUBSET\_SELECTION",qry)

cnt = arcpy.GetCount\_management(flayer)

print("The total number of selected records is:" + str(cnt))

except Exception as e:

print(e.message)

实验五

# 1 使用SearchCursor检索要素

import arcpy,arcpy.da as da

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\Ch8"

with da.SearchCursor('Schools.shp',('Facility','Name')) as cursor:

for row in sorted(cursor):

print('School name:' + row[1])

# 2 使用where子句筛选记录

import arcpy,arcpy.da as da

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\Ch8"

with arcpy.da.SearchCursor('Schools.shp',("Facility","Name"),'"FACILITY" = \'HIGH SCHOOL\'')as cursor:

for row in sorted(cursor):

print("School name:" + row[1])

# 3 使用几何令牌返回几何的部分信息

import arcpy,arcpy.da as da

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\Ch8"

with arcpy.da.SearchCursor("coa\_parcels.shp",("PY\_FULL\_OW","SHAPE@XY")) as cursor:

for row in cursor:

print("Parcel owner: {0} has a location of:{1}").format(row[0],row[1])

# 4 使用InsertCursor插入行

import arcpy,os

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\Ch8\WildfireData\WildlandFires.mdb"

f = open(r"E:\ArcPyStudy\Data\ArcpyBook\Ch8\WildfireData\NorthAmericaWildfires\_2007275.txt",'r')

lstFires = f.readlines()

try:

with arcpy.da.InsertCursor("FireIncidents",("SHAPE@XY","CONFIDENCEVALUE")) as cur:

cntr = 1

for fire in lstFires:

if 'Latitude' in fire:

continue

vals = fire.split(",")

latitude = float(vals[0])

longitude = float(vals[1])

confidence = int(vals[2])

rowValue = [(longitude,latitude),confidence]

cur.insertRow(rowValue)

print("Record number " + str(cntr) + " written to feature class")

cntr += 1

except Exception as e:

print(e.message)

finally:

f.close()

# 5 使用UpdateCursor更新行

import arcpy

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\Ch8\WildfireData\WildlandFires.mdb"

try:

# create a new field to hold the values

arcpy.AddField\_management("FireIncidents","CONFID\_RATING","TEXT","10")

print("CONFID\_RATING field added to FireIncidents")

with arcpy.da.UpdateCursor("FireIncidents",("CONFIDENCEVALUE","CONFID\_RATING")) as cursor:

cntr = 1

for row in cursor:

if row[0] <=40:

row[1] = 'POOR'

elif row[0] > 40 and row[0] <= 60:

row[1] = 'FAIR'

elif row[0] > 60 and row[0] <= 85:

row[1] = 'GODD'

else:

row[1] = 'EXCELLENT'

cursor.updateRow(row)

print("Record number "+ str(cntr) + " updated")

cntr += 1

except Exception as e:

print(e.message)

# 6 使用UpdateCursor删除行

import arcpy,os

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\Ch8\WildfireData\WildlandFires.mdb"

try:

with arcpy.da.UpdateCursor("FireIncidents",("CONFID\_RATING"),'[CONFID\_RATING] = \'POOR\'') as cursor:

cntr = 1

for row in cursor:

cursor.deleteRow()

print("Record number " + str(cntr) + ' deleted')

cntr += 1

except Exception as e:

print(e.message)

# 7 在编辑会话中插入和更新行

import arcpy,os

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\Ch8\WildfireData\WildlandFires.mdb"

try:

edit = arcpy.da.Editor(r"E:\ArcPyStudy\Data\ArcpyBook\Ch8\WildfireData\WildlandFires.mdb")

edit.startEditing(True)

with arcpy.da.UpdateCursor("FireIncidents",("CONFIDENCEVALUE","CONFID\_RATING")) as cursor:

cntr = 1

for row in cursor:

if row[0] > 40 and row[0] <= 60:

row[1] = "GOOD"

elif row[0] > 60 and row[0] <= 85:

row[1] = "BETTER"

else:

row[1] = "BEST"

cursor.updateRow(row)

print("Record number " + str(cntr) + " updated")

cntr += 1

edit.stopEditing(True)

except Exception as e:

print(e.message)

# 8 读取要素类中的几何信息

import arcpy

infc = r"E:\ArcPyStudy\Data\ArcpyBook\data\CityOfSanAntonio.gdb\SchoolDistricts"

for row in arcpy.da.SearchCursor(infc,["OID@","SHAPE@"],'"NAME" = \'Lackland ISD\''):

# Print the current multipoint's ID

print("Feature {}:".format(row[0]))

partnum = 0

# Step through each part of the feature

for part in row[1]:

print("Part {}:".format(partnum))

for pnt in part:

if pnt:

print("{},{}".format(pnt.X,pnt.Y))

else:

print("Interior Ring:")

partnum += 1

# 9 使用Walk()遍历目录

import arcpy.da as da,os

os.chdir(r"E:\ArcPyStudy\Data\ArcpyBook\data\CityOfSanAntonio.gdb")

print("os.walk:")

for dirpath,dirnames,filenames in os.walk(os.getcwd()):

for filename in filenames:

print(filename)

print("arcpy.da.Walk:")

for dirpath,dirnames,filenames in da.Walk(os.getcwd(),datatype="FeatureClass"):

for filename in filenames:

print(os.path.join(dirpath,filename))

实验六

# 1 使用ListFeatureClasses()函数返回要素类的列表

import arcpy

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\data\CityOfSanAntonio.gdb"

fcList = arcpy.ListFeatureClasses('C\*','Polygon') # C开头且数据类型为Polygon的要素类

for fc in fcList:

print(fc)

# 2 使用ListFields()函数返回要素类的字段列表

import arcpy

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\data\CityOfSanAntonio.gdb"

fieldList = arcpy.ListFields('Burglary')

for fld in fieldList:

print('%s is a type of %s with a length of %i' % (fld.name,fld.type,fld.length))

# 3 使用Describe()函数返回要素类的形状类型、字段、范围等描述性信息；

import arcpy

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\New\_Data\CityOfSanAntonio\_Personal.mdb"

try:

descFC = arcpy.Describe("SchoolDistricts")

print("The shape type is: " + descFC.ShapeType)

flds = descFC.fields

for fld in flds:

print("Field: " + fld.name)

print("Type: " + fld.type)

print("Length: " + str(fld.length))

ext = descFC.extent

print("XMin: %f" % ext.XMin)

print("YMin: %f" % ext.YMin)

print("XMax: %f" % ext.XMax)

print("YMax: %f" % ext.YMax)

except:

print(arcpy.GetMessages)

# 4 使用Describe()函数返回栅格数据的范围、空间参考等描述性信息。

import arcpy

arcpy.env.workspace = r"E:\ArcPyStudy\Data\ArcpyBook\data"

try:

descRaster = arcpy.Describe("AUSTIN\_EAST\_NW.sid")

ext = descRaster.extent

print("XMin: %f" % ext.XMin)

print("YMin: %f" % ext.YMin)

print("XMax: %f" % ext.XMax)

print("YMax: %f" % ext.YMax)

sr = descRaster.SpatialReference

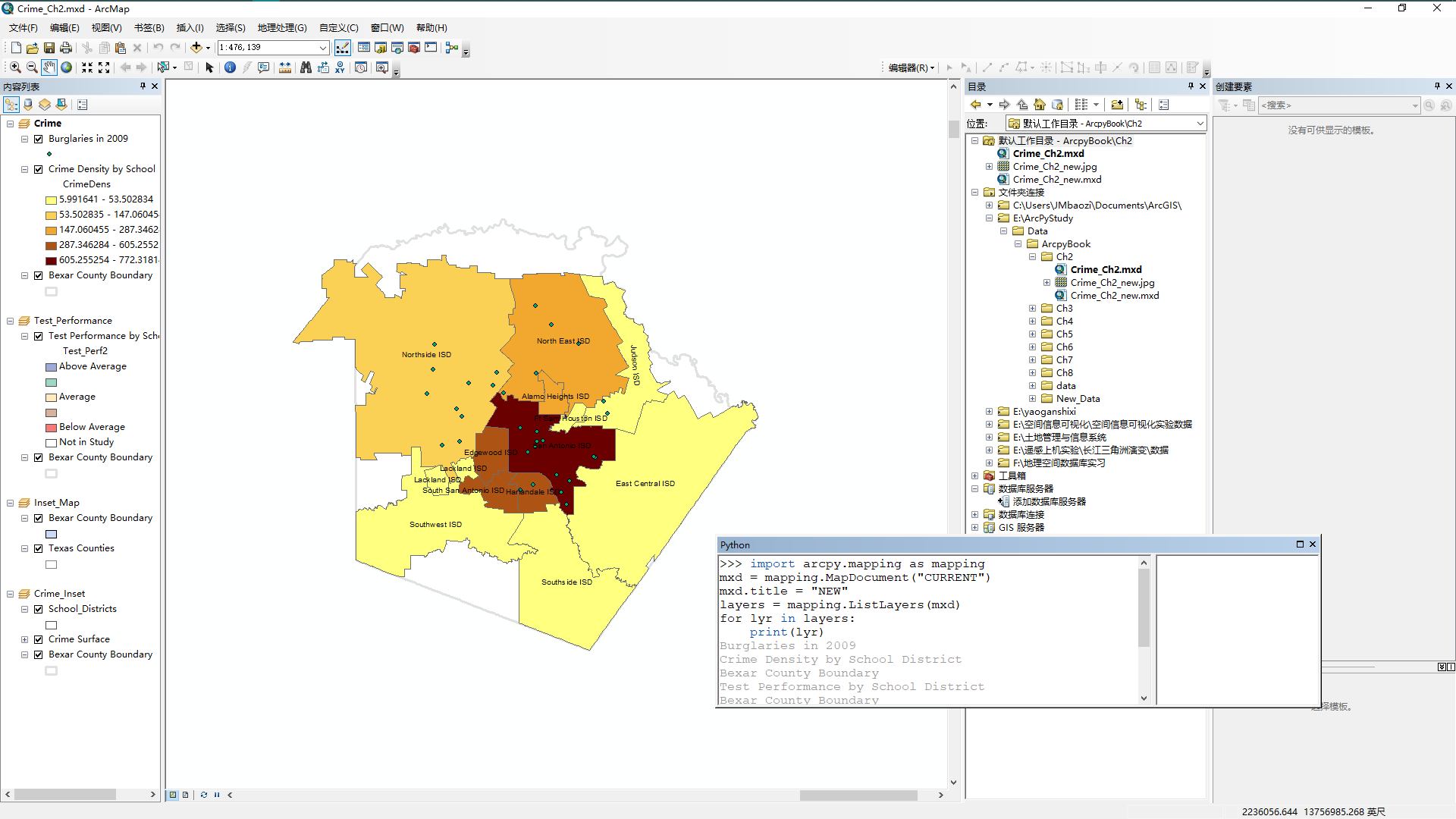
print("name: " + sr.name)

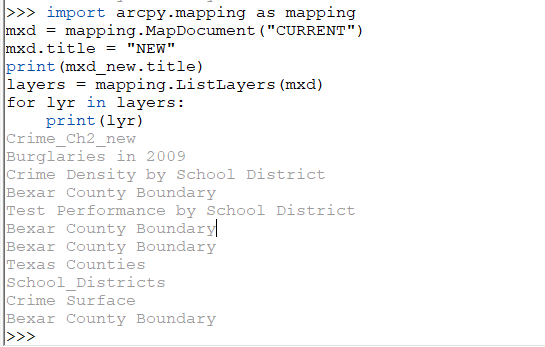
print("type: " + sr.type)

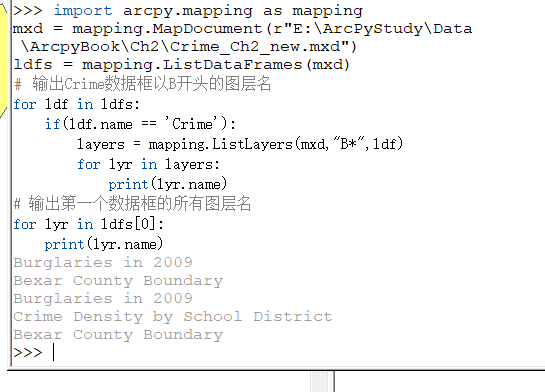
except Exception as e:

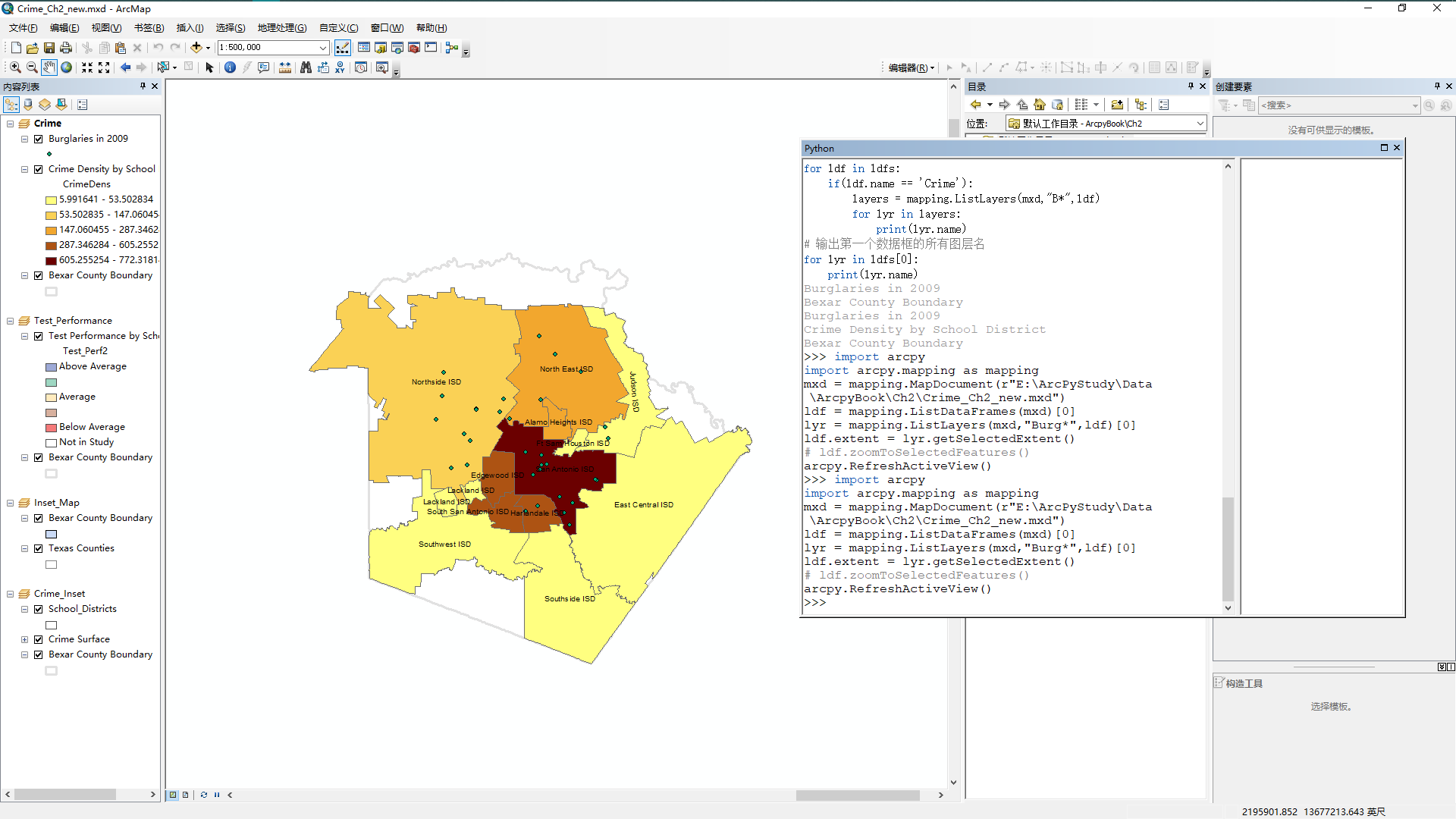
print(e.message)

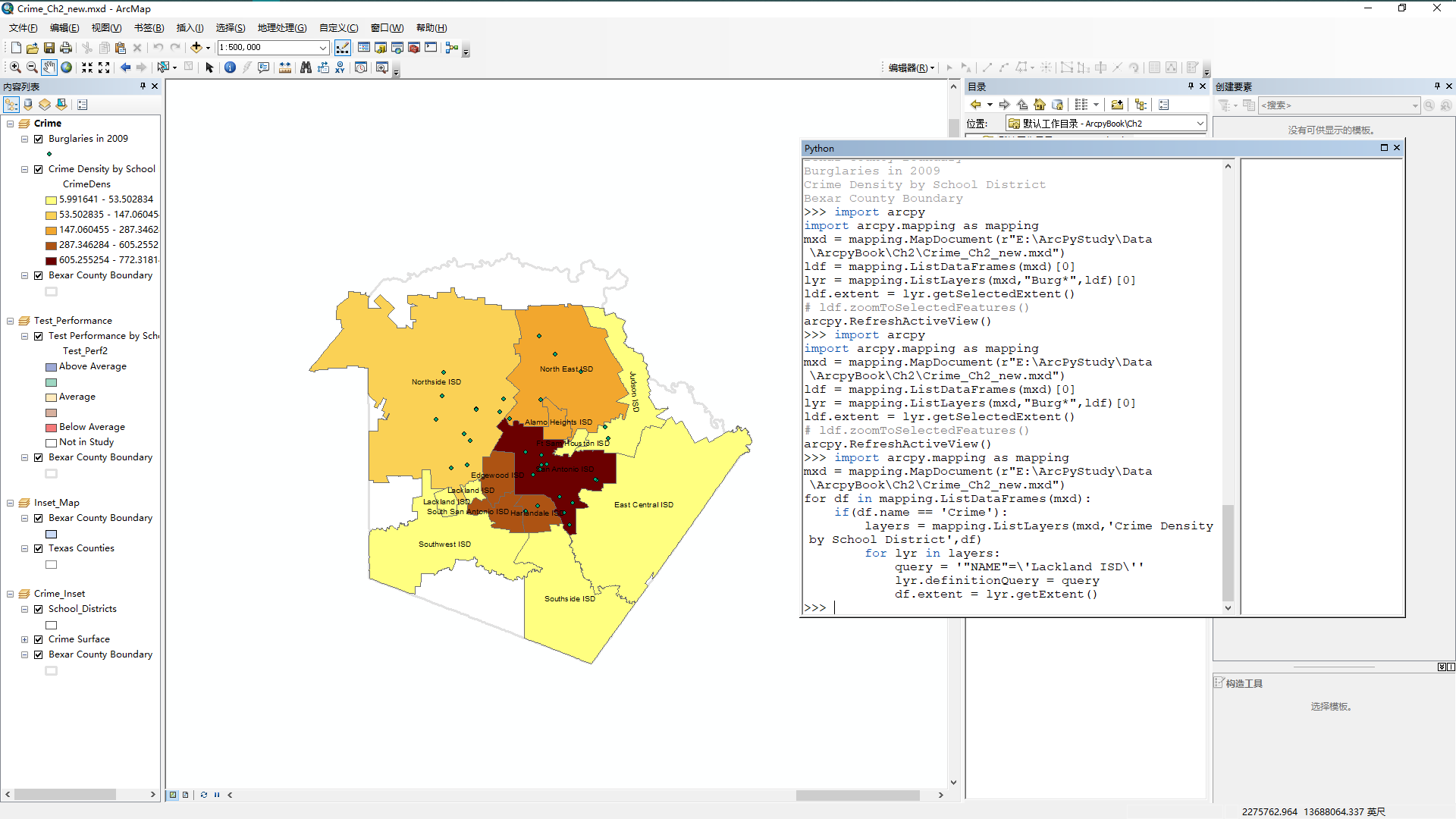
实验一

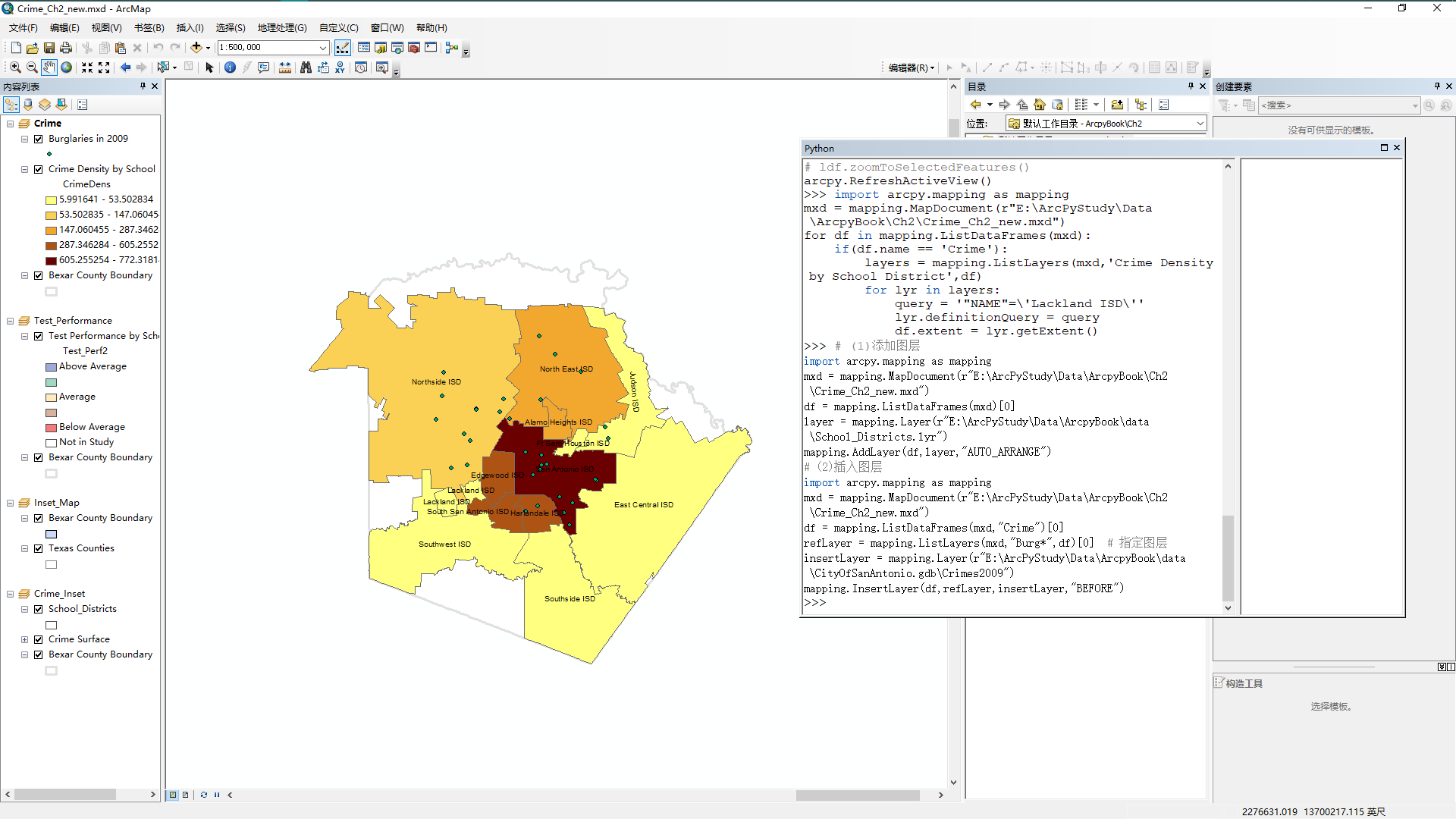


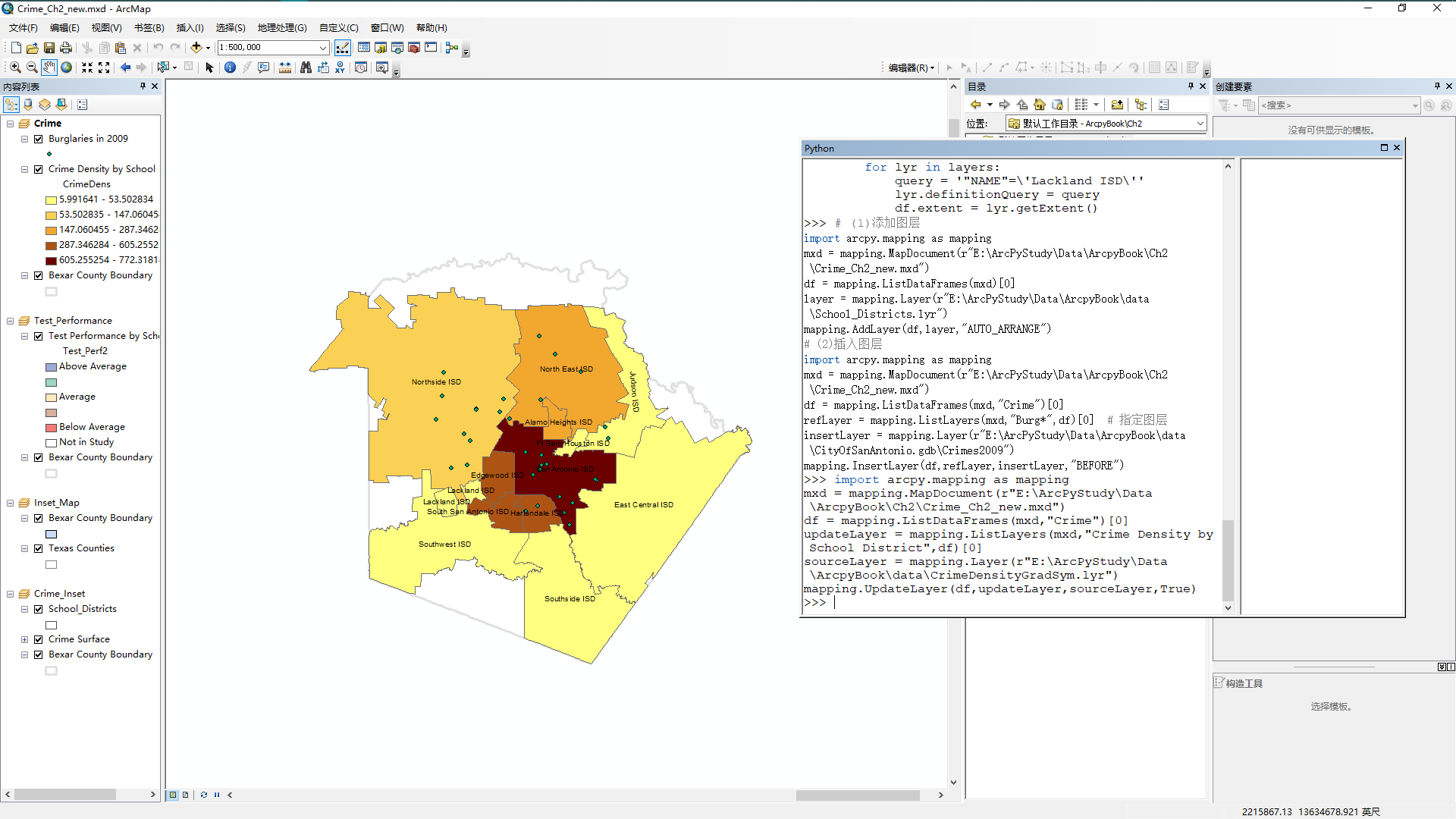




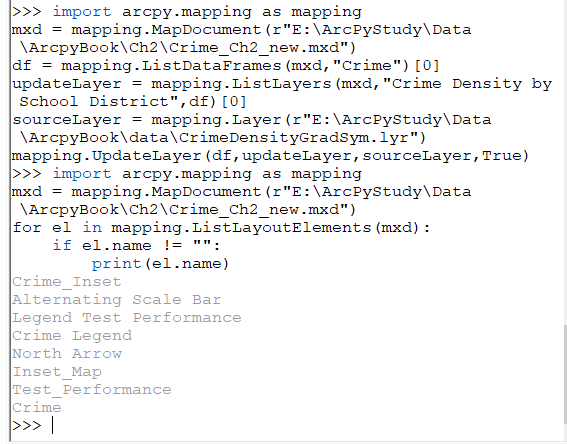


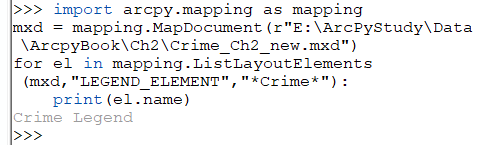


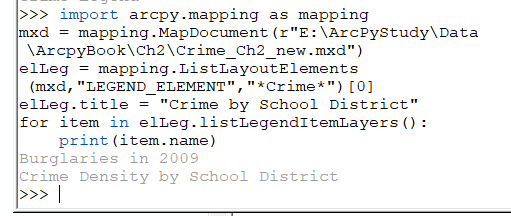


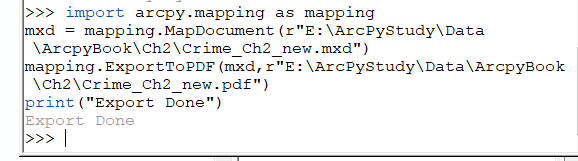


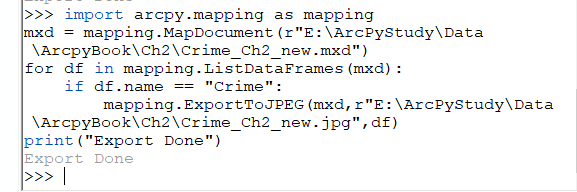
实验二



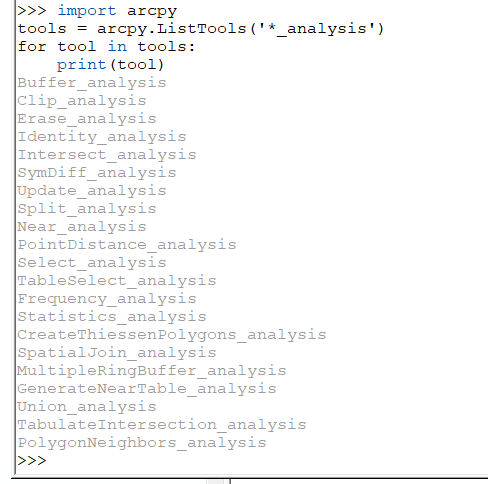


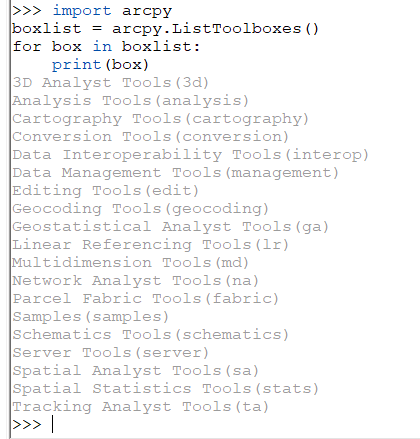


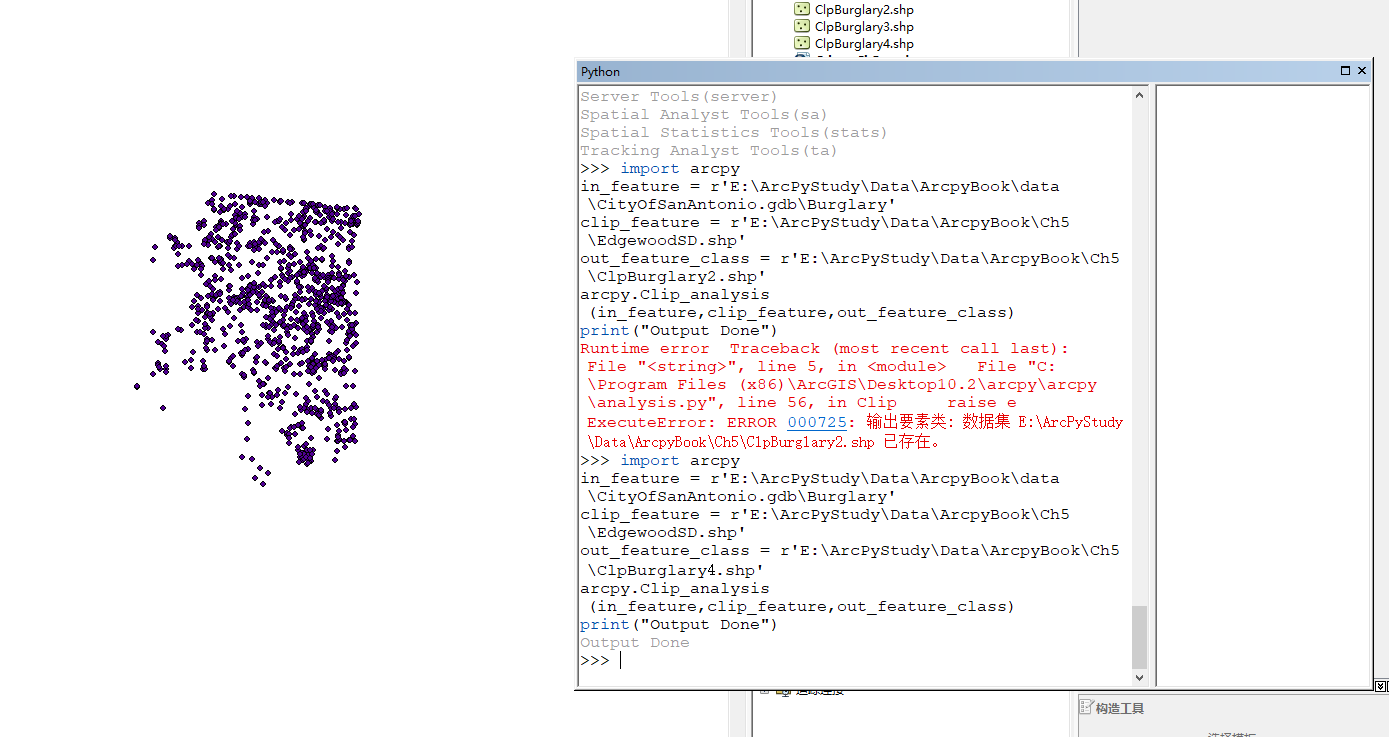


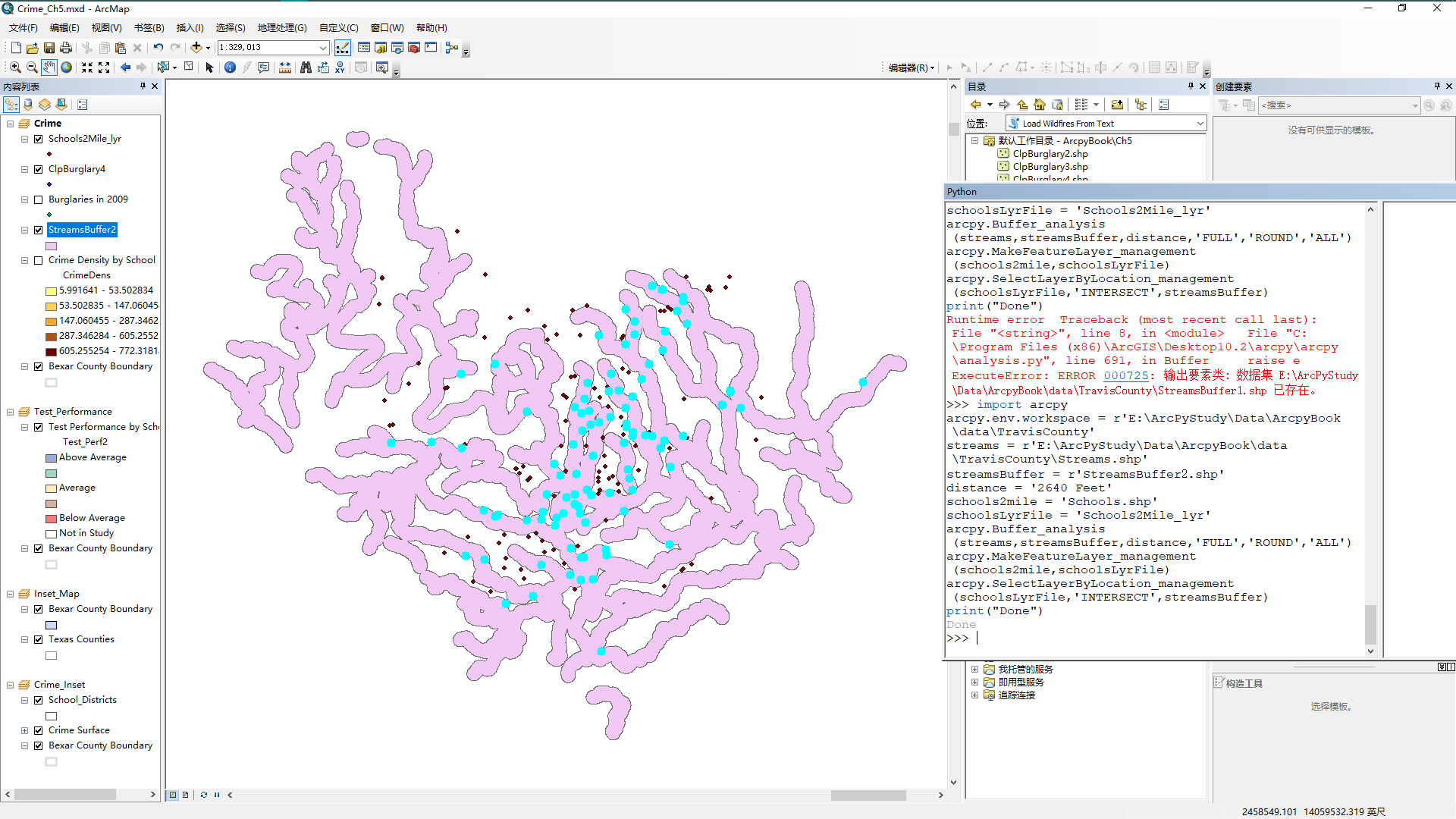


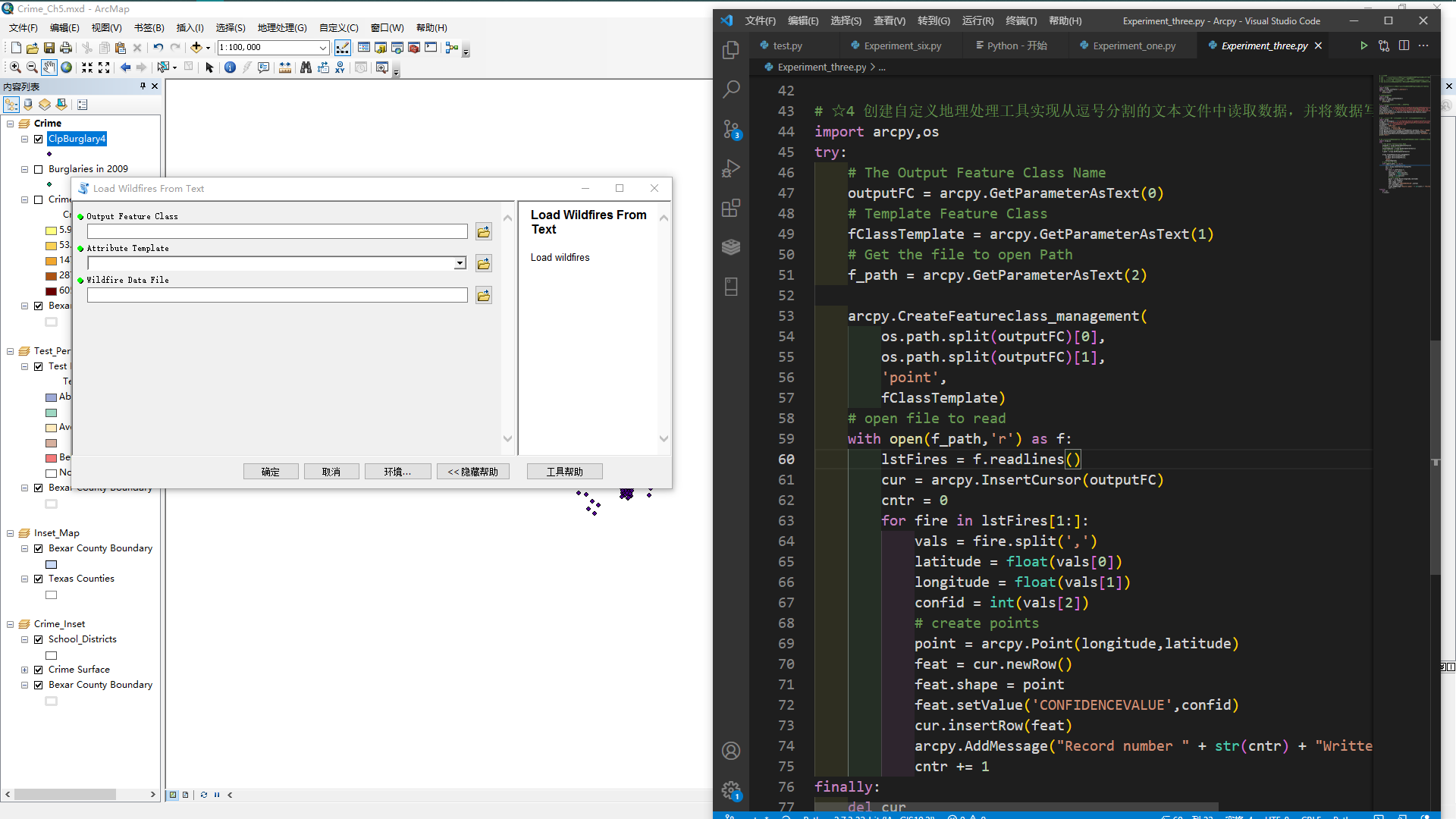
实验三



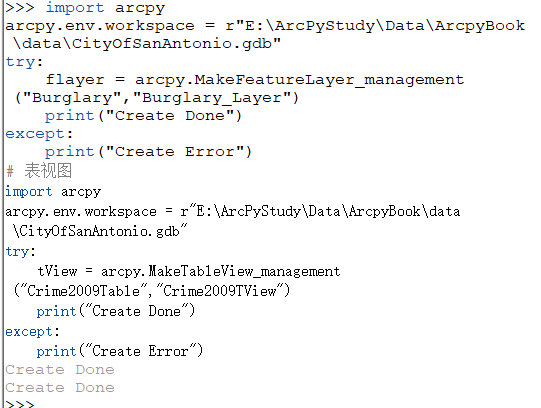


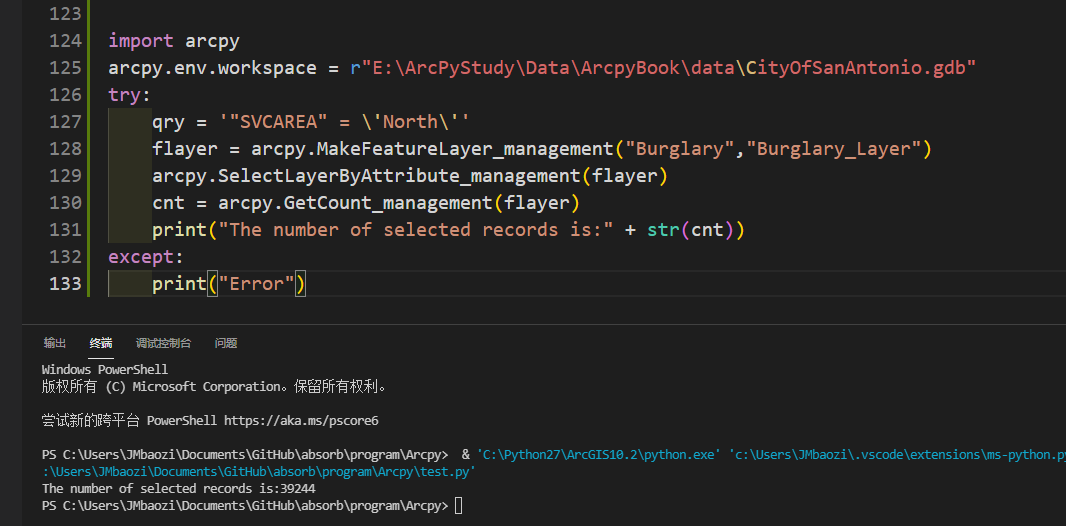


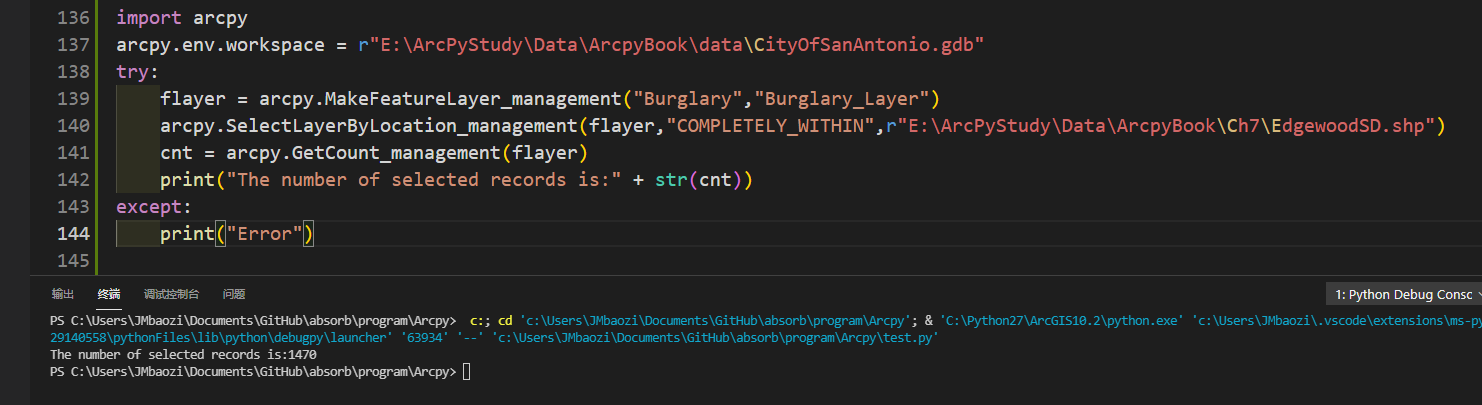


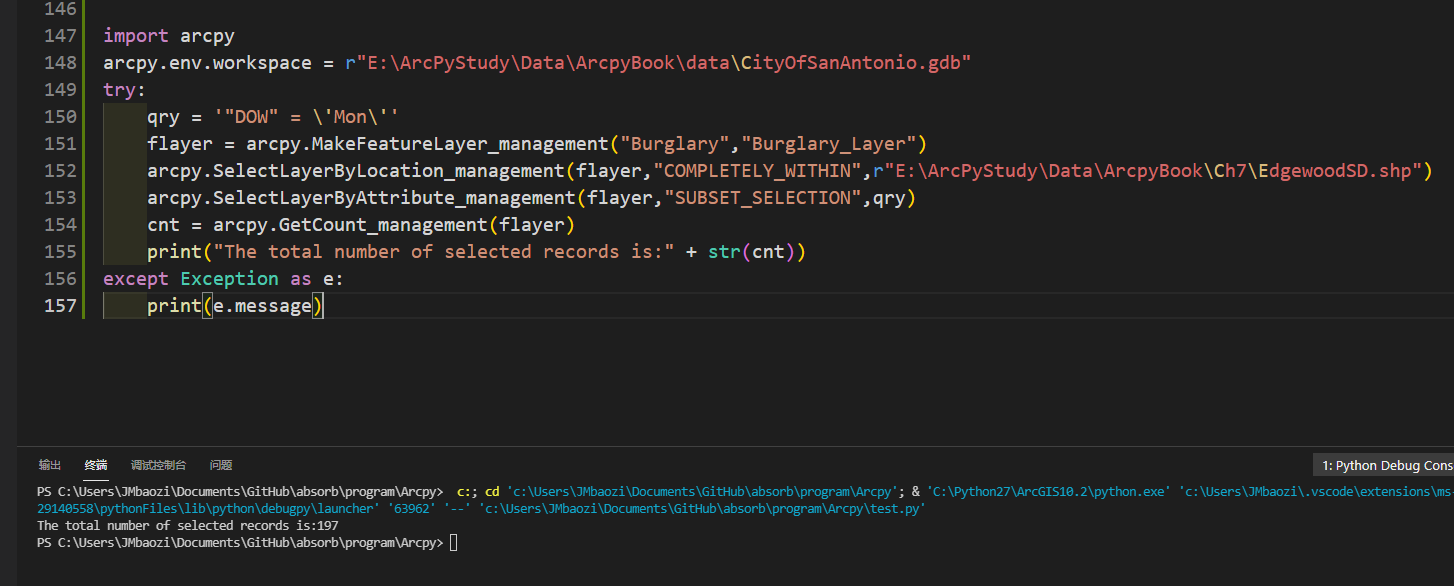


实验四

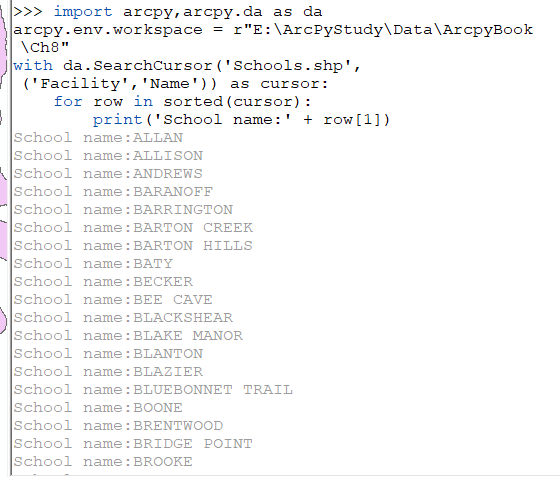


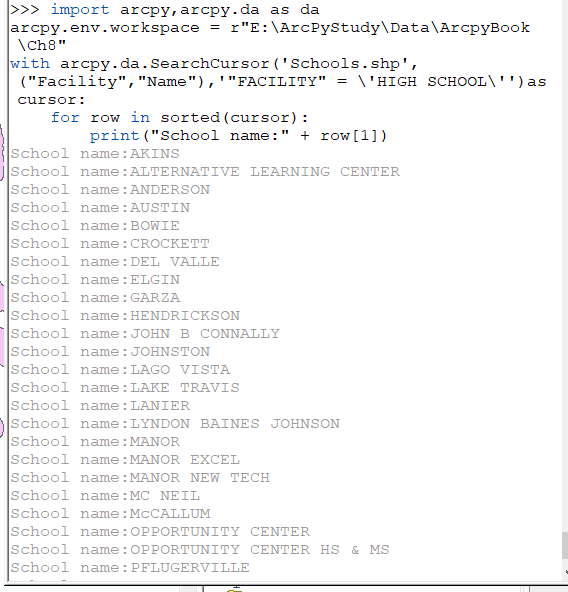


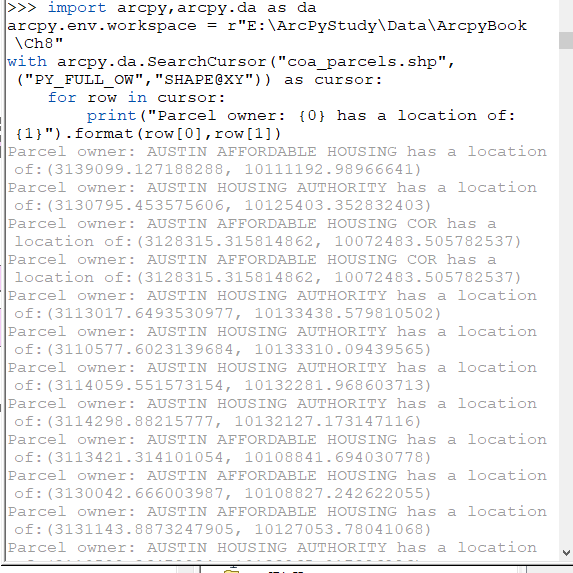


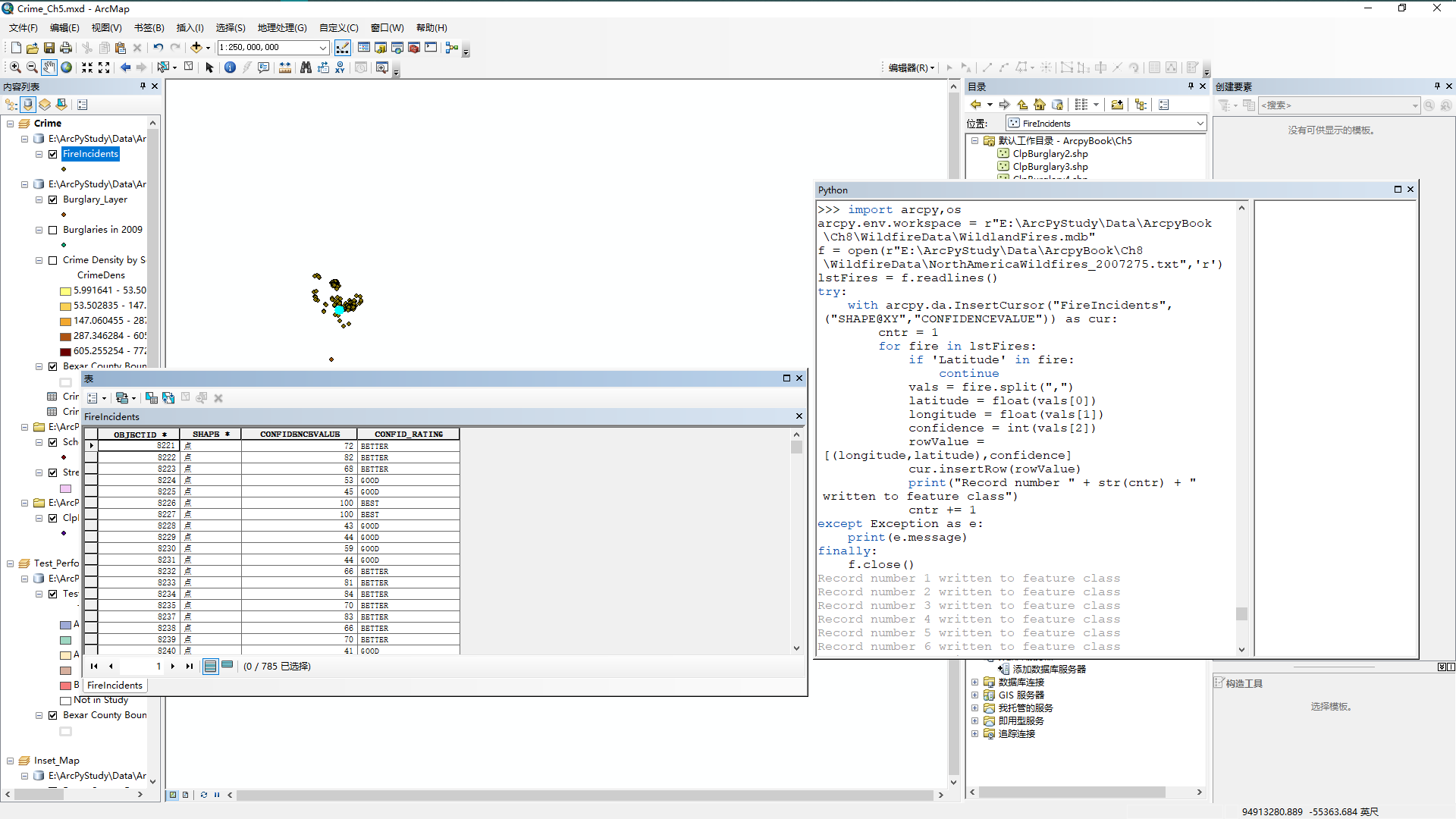


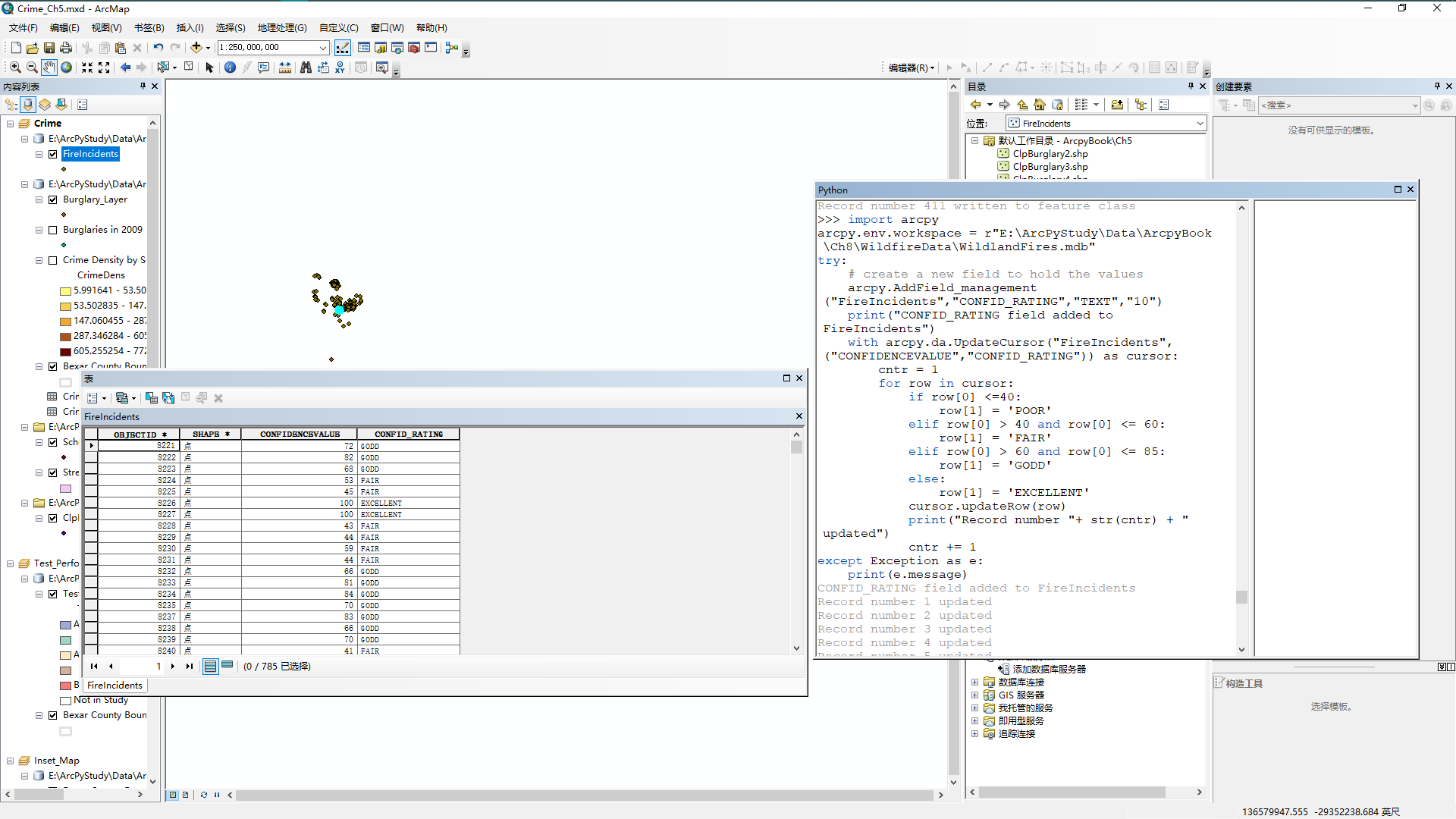
实验五

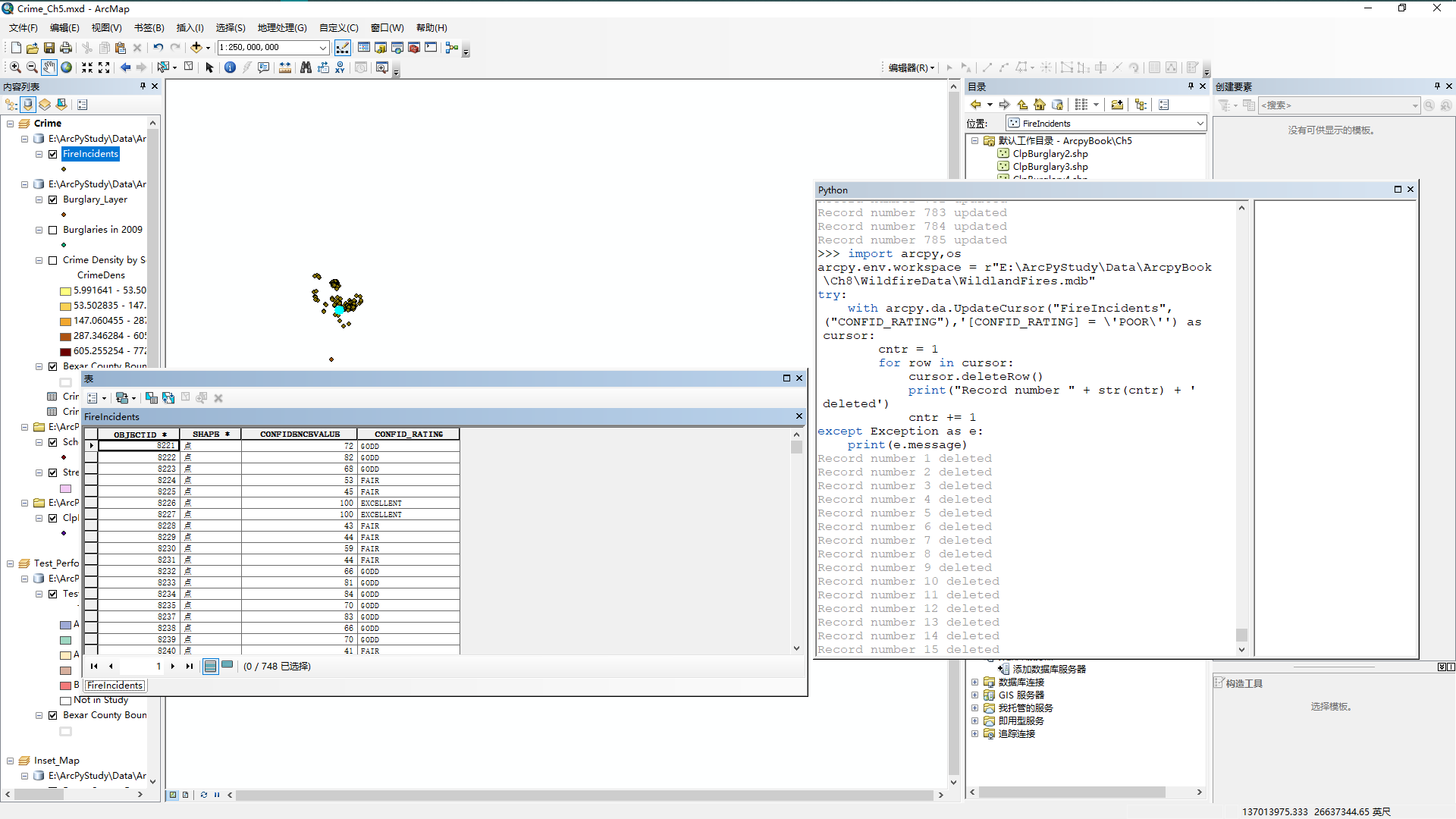


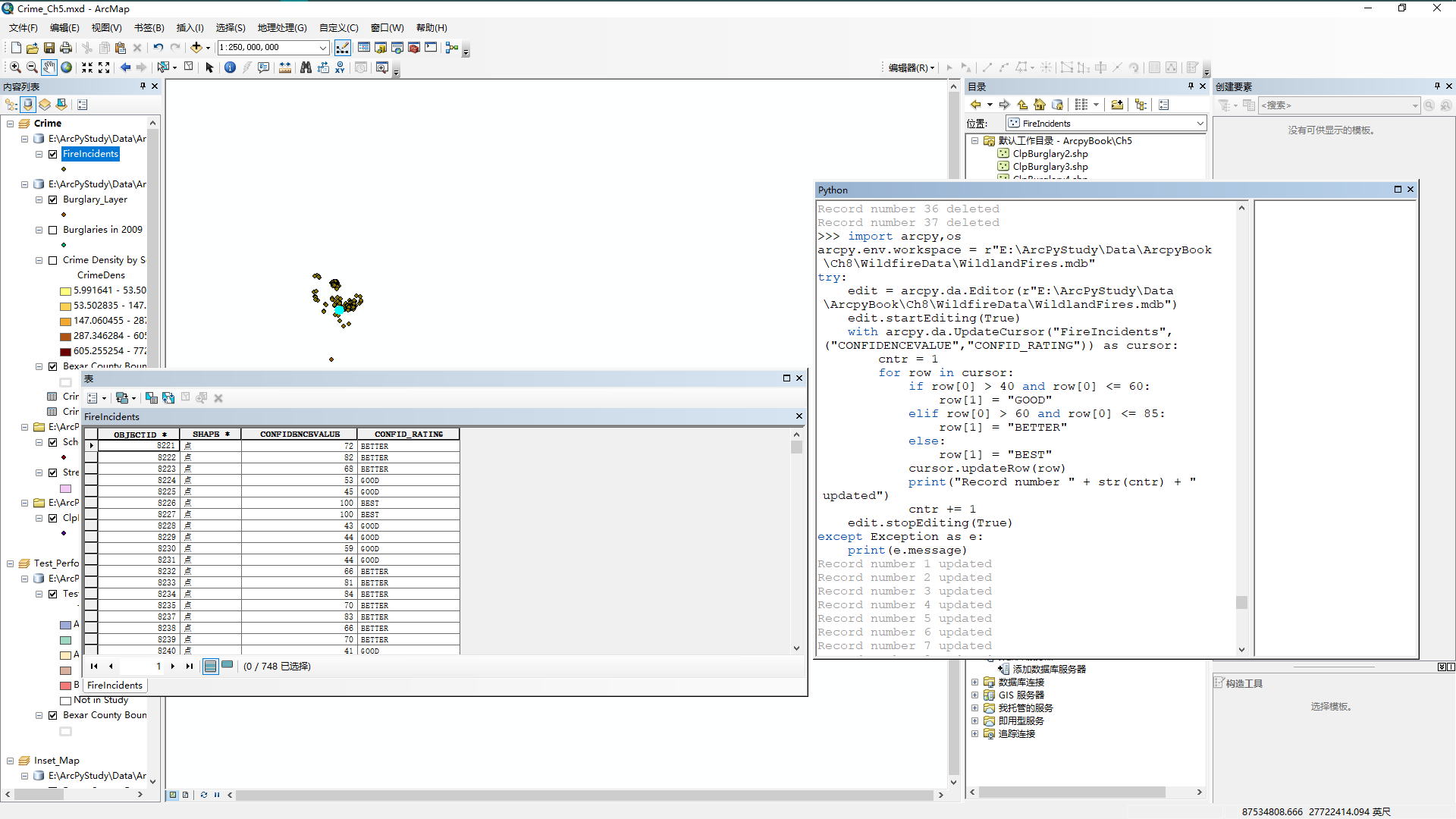


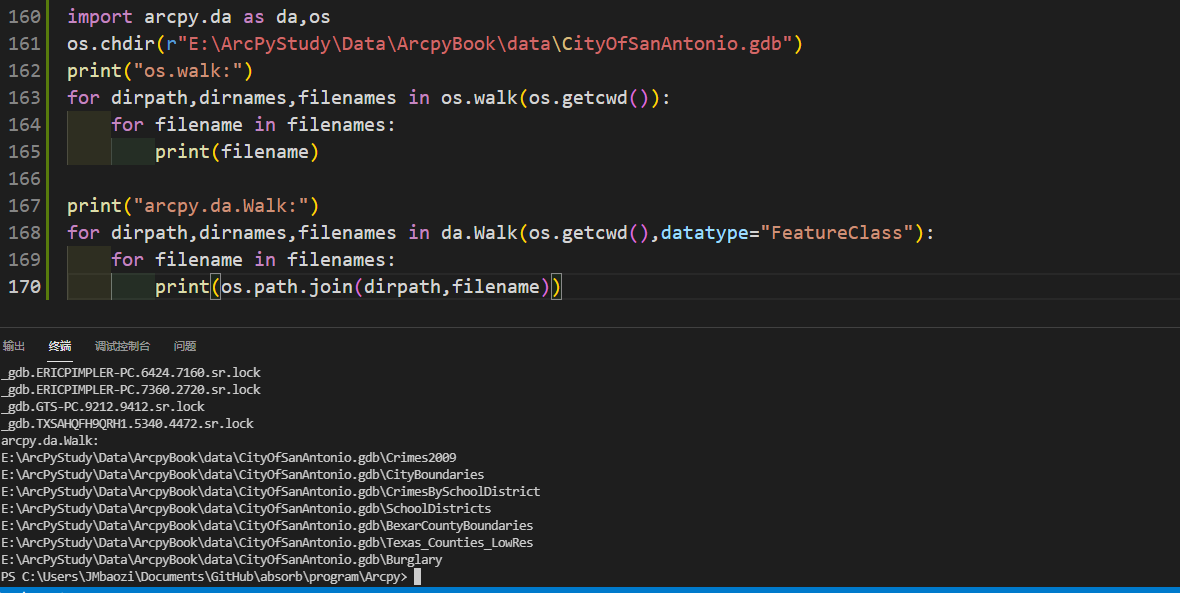
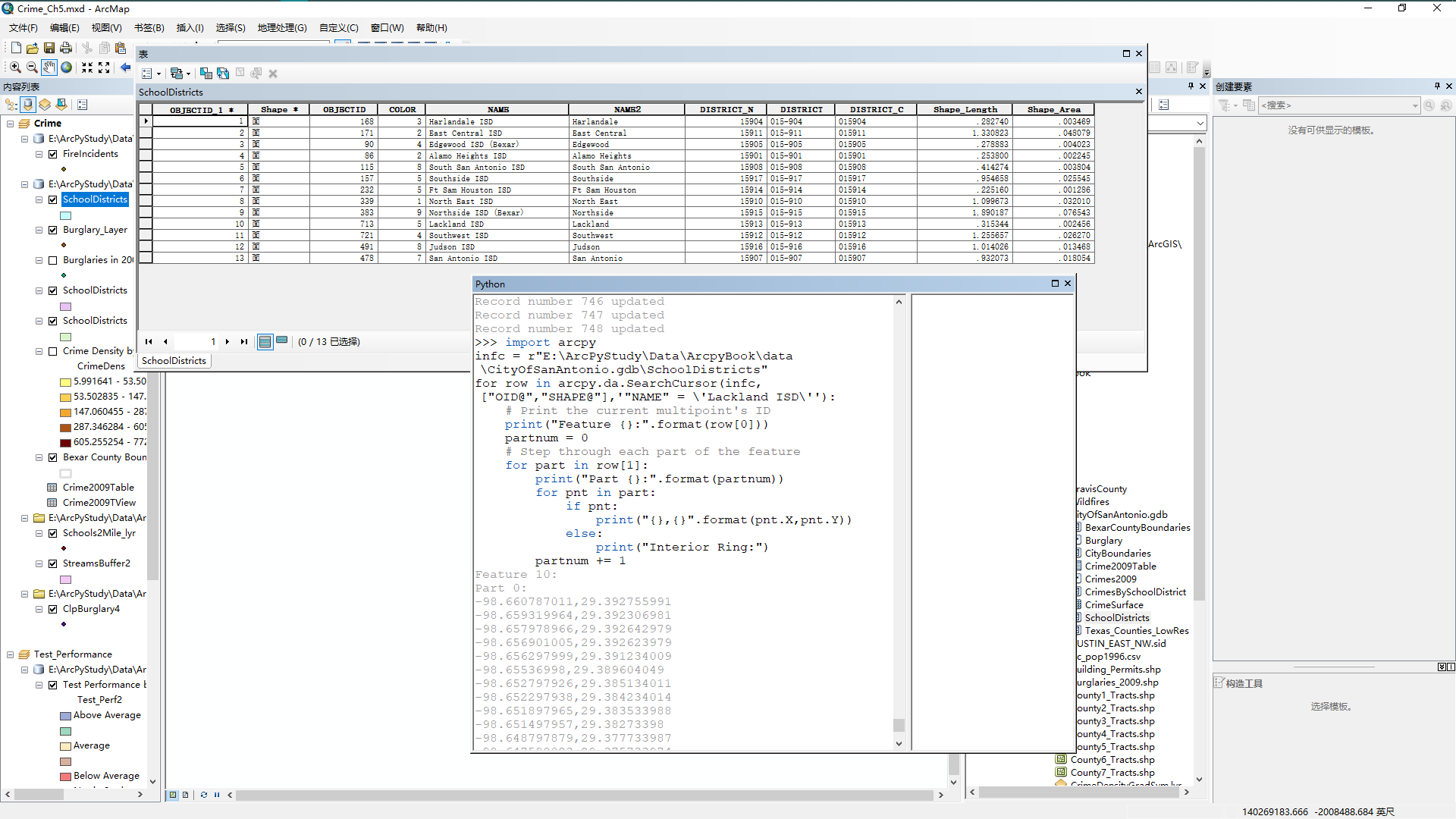












实验六

