##Join county-level resident population data in 2018 and the administrative boundary

import arcpy

arcpy.env.workspace = "C:/data"

arcpy.env.qualifiedFieldNames = False

arcpy.AddJoin\_management(“boundaries.shp”, joinField, “statistic\_population.xls”, joinField)

arcpy.CopyFeatures\_management(“boundaries.shp”, “statistical\_pop”)

##Calculate the total Tencent positioning frequency of each county *Tencenti*

from arcpy import env

from arcpy.sa import \*

env.workspace = "D:/data "

outZonalStats = ZonalStatistics("boundaries.shp", "PAC", "tencent", "SUM","NODATA")

outZonalStats.save("D:/data/county")

AddJoin\_management (county.shp, in\_field, “county”, join\_field, {KEEP\_COMMON})

##Use field calculator to get the log of population and Tencent positioning data

##Extract the Tencent positioning data to point with 0.01° resolution

import arcpy

from arcpy import env

from arcpy.sa import \*

env.workspace = "C:/data"

arcpy.env.cellSize = "0.01"

outPoint = " C:/data / Tencent.shp"

RasterToPoint\_conversion “tencent”, outPoint, {value})

## GWR, get the corresponding slope and intercept of each district and county (by GWR4)

##Join the result of GWR and county.shp

AddJoin\_management (county.shp, in\_field, “popGWR.shp”, join\_field, {KEEP\_COMMON})

##Join the point and county.shp based on location

target\_features = "C:/data/county.shp"

join\_features = "C:/data/Tencent.shp"

out\_feature\_class = "C:/data/join\_Output.shp "

arcpy.SpatialJoin\_analysis(target\_features, join\_features, out\_feature\_class)

##point to raster

import arcpy

from arcpy import env

env.workspace = "C:/data"

arcpy.PointToRaster\_conversion("join\_Output.shp", "lntencent", "C:/data/lntn", "MOST\_FREQUENT", "", 0.01)

arcpy.PointToRaster\_conversion("join\_Output.shp", slope\_field, "C:/data/a", "MOST\_FREQUENT", "", 0.01)

arcpy.PointToRaster\_conversion("join\_Output.shp", intercept\_field, "C:/data/b", "MOST\_FREQUENT", "", 0.01)

arcpy.PointToRaster\_conversion("join\_Output.shp", "Statistic\_population", "C:/data/county", "MOST\_FREQUENT", "", 0.01)

##calculate weight in each grid POPij

import arcpy

from arcpy import env

from arcpy.sa import \*

env.workspace = "C:/ data"

out1 = Ln("value")

out2 = Raster(“a”) \* Raster(“out1”) + Raster(“b”)

outExp = Exp("out2")

outExp.save("C:/data/outexp.tif")

outZonalStats = ZonalStatistics("statistical\_pop.shp", "PAC", " outexp.tif", "SUM","NODATA")

outZonalStats.save("C:/data/sumout")

##population distribution

pop2018 = Raster(“outexp”) \* Raster(“county”) / Raster(“sumout”)

out.save(“C:/data/pop2018”)