Location-Based Analysis for Recruitment of United States Border Patrol Agents

Jordan Douthitt
Advisor: Stephen A. Matthews

March 20 & 21, 2018
Executive Order 13767 on January 25th, 2017
  • Need to hire 5,000 additional U.S. Border Patrol Agents

Historical difficulty hiring Agents

>10% loss in Southwest Agents 2011-2017
Need to strategically recruit a large amount of skilled employees (United States Border Patrol Agents) to hard to fill positions in undesirable locations.
Analysis answering two questions

1. Which Border Patrol Sectors have the greatest recruitment support need?

2. Which areas outside of these Border Patrol Sectors can additional recruiting focus on?
Factors Affecting Recruitment

**Border Patrol Agent Factors**

- Starting age under 39
- Degree/Background in Law Enforcement
- Ability to Speak/Learn Spanish
- Workload & Location

**Labor Pool & Migration Factors**

- Total Population
- Age
- Unemployment Rate
- Salary
- Community Size & Demography
- Distance from Community of Origin
- Previous Migration/Social Ties
DECISION:
Which sectors are most in need of additional recruitment assistance
Methodology: Labor Pool Analysis

- **Sensitivity Testing**
  - $R^2$ correlation between factor values

- **Measure Factors**
  - 1-5 quantile rating
  - Final Score: sum of individual ratings

- **Results Testing**
  - $R^2$ correlation between factor scores & total score

- **Spatial Cluster Analysis**
  - Univariate Local Moran’s I
  - Getis-Ord Gi*

**DECISION:**
Which outside labor pools are most likely to provide more viable recruits
**Results: Sector Profiles**

<table>
<thead>
<tr>
<th>Sector Information</th>
<th>Workload</th>
<th>Labor Pool</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2017 Agent Count</strong>: 500</td>
<td><strong>Border Crossing Count</strong>: 2</td>
<td><strong>Percent of Stations in Nonmetropolitan Areas</strong>: 69.2%</td>
</tr>
<tr>
<td><strong>2011-2017 Agent Change</strong>: 25% loss</td>
<td><strong>Apprehensions per Agent</strong>: 12</td>
<td><strong>Population of Surrounding Labor Pools</strong>: 2,142,736</td>
</tr>
<tr>
<td><strong>Station Count</strong>: 13</td>
<td><strong>Border Entries per Agent</strong>: 2,990</td>
<td></td>
</tr>
</tbody>
</table>

**Background**: 

**Problem & Goal**: 

**Data**: 

**Methodology**: 

**ArcPy**: 

**Legend** 

- **Southwest Region** 
- **Big Bend Sector** 
  - Sector Stations 
  - U.S. States 
  - Total Factor Rating 
  - 5 Quantiles 
    - 14 - 24 
    - 25 - 27 
    - 28 - 30 
    - 31 - 33 
    - 34 - 41 

**Big Bend Sector Border Patrol Stations** 

- **Border Crossing Count**: 2 
- **Apprehensions per Agent**: 12 
- **Border Entries per Agent**: 2,990 
- **Percent of Stations in Nonmetropolitan Areas**: 69.2% 
- **Population of Surrounding Labor Pools**: 2,142,736
Results: Labor Pool Analysis

Total Rating of Potential Border Patrol Labor Pools (Metropolitan and Nonmetropolitan Areas)

Border Patrol Potential Labor Pool Hot and Cold Clusters Using Getis-Ord Gi* Statistic

Legend
Total Factor Rating
5 Quantiles
14 - 24
25 - 27
28 - 30
31 - 33
34 - 41

Legend
Hot and Cold Spots
Cold Spot - 99% Confidence
Cold Spot - 95% Confidence
Cold Spot - 90% Confidence
Not Significant
Hot Spot - 99% Confidence
Hot Spot - 95% Confidence
Hot Spot - 90% Confidence

Projection: USA Contiguous Albers Equal Area Conic USGS

Background  Problem & Goal  Data  Methodology  Results  ArcPy
Value to Human Resources Field:

• Previously minimal demographic or location-based research performed

• Model for focused recruitment of hard to fill positions that can be used in other similar scenarios
Need:
• Join a large amount of Census CSV files to a feature class to visualize factors within
• Tell which areas are within a desired distance of the border between the United States and Mexico
• Easily allow for changes needed as data is researched

Python:
• Automate Repetitive Tasks
• Changeable Parameters
• Ease of Use with ArcGIS Script Tool
• Sharable
# Import both arcpy and csv modules for use in script
import arcpy
import csv
import os

# Set overwriteOutput to true, so that files can be overwritten if an error happens
arcpy.env.overwriteOutput = True

# Set needed input and output parameters
inputTablesFolder = arcpy.GetParameterAsText(0)
inputJoinShapefile = arcpy.GetParameterAsText(1)
inputBufferShapefile = arcpy.GetParameterAsText(2)
inputBufferDistance = arcpy.GetParameterAsText(3)
outputFolder = arcpy.GetParameterAsText(4)

# Start try statement for bulk of script
try:
    # Check if this geodatabase already exists in the output folder
    # Create name and location to check for geodatabase
    outputGDBCheck = os.path.join(outputFolder,"Output_GDB.gdb")
    # Start if statement if geodatabase exists, set output geodatabase path
    if arcpy.Exists(outputGDBCheck):
        arcpy.AddMessage("FileGDB Already Exists")
        outputGDB = outputGDBCheck
    else:
        # Create file geodatabase to output new shapefiles into
        outputGDB = arcpy.CreateFileGDB_management(outputFolder, "Output_GDB")
ArcPy

```python
# Sets workspace for the arcpy.ListFiles() function
arcpy.env.workspace = inputTablesFolder
# Lists all files from input folder that end in ".csv"
csvList = arcpy.ListFiles("*.csv")

# Start for loop for csv files
for table in csvList:
    # Create output feature layer name from csv name
    fieldName = table.replace(".csv", "")

    # Make layer from input shapefile to allow for select layer by location later in script
    arcpy.MakeFeatureLayer_management(inputJoinShapefile, fieldName)

    # Create buffer distance name
    # This removes all spaces from the buffer distance string
    fieldNameString = inputBufferDistance.replace(" ", "")
    # Length of field name is kept to 10 characters with 'bf_' and the first 7 characters of the buffer distance
    fieldName = "bf_" + fieldNameString[:7]
    # Adds field
    arcpy.AddField_management(fcName, fieldName, "TEXT")

    # Select layer by location based on input buffer shapefile and distance
    arcpy.SelectLayerByLocation_management(fcName,"WITHIN A DISTANCE",inputBufferShapefile, inputBufferDistance)

    # Update cursor to add whether or not each record falls in the specified distance of the selected feature
    # If record falls within buffer distance (select by location), it prints "YES" in the field, otherwise a null value is left
    with arcpy.da.UpdateCursor(fcName, (fieldName,)) as cursor:
        for row in cursor:
            row[0] = "YES"
            cursor.updateRow(row)
```

Create New Field

Select by Buffer Distance

Update Field Values
# Clear selected features before join to csv, so that all features are joined, not just selected features
arcpy.SelectLayerByAttribute_management(fcName,"CLEAR_SELECTION")

# Join csv to new feature layer based on the GEOID2 field in each
arcpy.AddJoin_management(fcName, "GEOID2", table, "GEOID2")

# Create shapefile from layer and joined table
# Set name of output shapefile
lyrName = fcName
lyrName2 = lyrName + "_fc"
# Create path for output shapefile
outPath = str(outputGDB)
fcPath = os.path.join(outPath, lyrName2)
# Create shapefile
arcpy.CopyFeatures_management(lyrName, fcPath)
arcpy.AddMessage(arcpy.GetMessages())

# Get error messages if script doesn't work
except:
    # Report error message
    arcpy.AddMessage("Could Not Complete Process")
    # Report any error messages that tools in this script might have generated
    arcpy.AddMessage(arcpy.GetMessages(2))

# Finally, delete the built layer, even if the rest of the script does not run
finally:
    # Delete layer
    arcpy.Delete_management(lyrName)
Python Script Tool

- Use script in other projects
- Rework process
- Change input files
- Change buffer distance
Thank You


References


