Python: Beyond the Basics

Brittney White, Esri

Jordan Douthitt, Pennsylvania State University

2018 **Esri Federal GIS Conference** | Washington, DC

Topics Covered

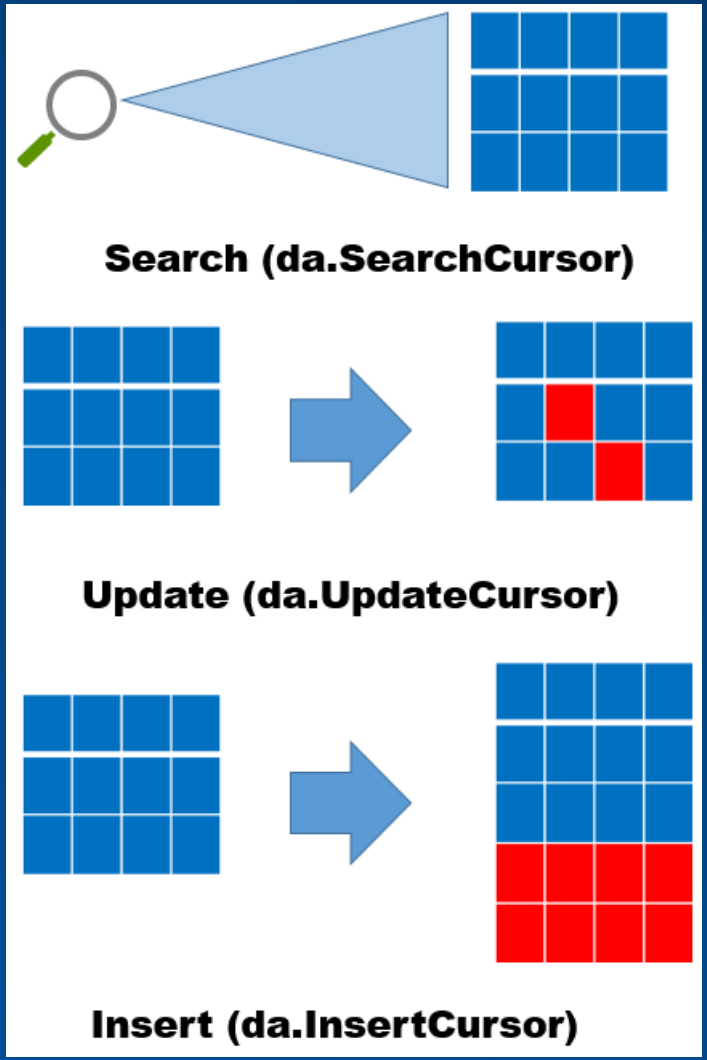
- Cursors
- Geometry objects
- Leveraging other Python modules
- *User presentation: Location-Based Analysis for Recruitment of United States Border Patrol Agents*

Cursors

- Used to:
  - Iterate over the set of rows in a table
  - Insert new rows into a table
- Two varieties:
  - arcpy.da cursors (10.1 onwards; significantly faster performance)
  - “Classic” cursors (provided only for continuing backward compatibility)

Legacy: The **arcpy.da** cursors (**arcpy.da.SearchCursor**, **arcpy.da.UpdateCursor**, and **arcpy.da.InsertCursor**) were introduced with ArcGIS 10.1 to provide significantly faster performance over the previously existing set of cursor functions (**arcpy.SearchCursor**, **arcpy.UpdateCursor**, and **arcpy.InsertCursor**). The original cursors are provided only for continuing backward compatibility.

Data Access Module Cursors



Required Arguments

- Table
  - The feature class, layer, table, or table view

- Fields
  - Single field or list of field names
  - Index position in fields parameter defines value access

```
#              0          1          2
fields = [“Name”, “Year”, “Count”]
```

Tokens

Used as shortcuts in place of field names

- OID@ —The value of the ObjectID field.
- SHAPE@ —A geometry object for the feature.
- SHAPE@XY —A tuple of the feature's centroid x,y coordinates.
- SHAPE@TRUECENTROID —A tuple of the feature's true centroid x,y coordinates.
- SHAPE@X —A double of the feature's x-coordinate.
- SHAPE@Y —A double of the feature's y-coordinate.
- SHAPE@Z —A double of the feature's z-coordinate.
- SHAPE@M —A double of the feature's m-value.
- SHAPE@JSON — The esri JSON string representing the geometry.
- SHAPE@WKB —The well-known binary (WKB) representation for OGC geometry. It provides a portable representation of a geometry value as a contiguous stream of bytes.
- SHAPE@WKT —The well-known text (WKT) representation for OGC geometry. It provides a portable representation of a geometry value as a text string.
- SHAPE@AREA —A double of the feature's area.
- SHAPE@LENGTH —A double of the feature's length.

arcpy.da.SearchCursor

```
arcpy.da.SearchCursor(in_table, field_names,
{where_clause}, {spatial_reference},
{explode_to_points}, {sql_clause})
```

arcpy.da.SearchCursor – optional where clause parameter

Crime_Incidents_2016

Field: Add Delete Calculate Selection: Zoom To Switch Clear Delete

1	Shape	OBJECTID	CCN	REPORTDATE	SHIFT	OFFENSE	METHOD	LASTMODIFI	BLOCKSITEA	BLOCKXCOOR	BLOCKYCOOR	WARD	ANC
	Point	15914	16063221	2016-04-22T21:27:41....	EVENING	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	700 - 799 BLOCK OF...	398098	136808	2	2C
	Point	15915	16063234	2016-04-22T21:30:25....	EVENING	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	2800 - 2899 BLOCK...	395007	137489	2	2E
	Point	15916	16063250	2016-04-22T22:09:09....	EVENING	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	1100 - 1127 BLOCK...	397430	137381	2	2F
	Point	15917	16063374	2016-04-23T03:06:29....	MIDNIGHT	THEFT F/AUTO	OTHERS	2017-01-10T07:18:37....	2400 - 2499 BLOCK...	402216	139352	5	5C
	Point	15918	16063392	2016-04-23T02:10:11....	MIDNIGHT	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	2000 - 2099 BLOCK...	398101	138793	1	1B
	Point	15919	16063419	2016-04-23T03:18:48....	MIDNIGHT	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	1200 - 1299 BLOCK...	400936	134825	6	6B
	Point	15920	16063431	2016-04-23T04:47:13....	MIDNIGHT	ROBBERY	GUN	2017-01-10T07:18:37....	1200 - 1299 BLOCK...	400986	142561	5	5A
	Point	15921	16063439	2016-04-23T04:25:02....	MIDNIGHT	ASSAULT W/DANGE...	OTHERS	2017-01-10T07:18:37....	1127 - 1199 BLOCK...	398944	137478	6	6E
	Point	15922	16063447	2016-04-23T05:02:31....	MIDNIGHT	ROBBERY	OTHERS	2017-01-10T07:18:37....	900 - 931 BLOCK OF...	397863	138792	1	1B
	Point	15923	16063460	2016-04-23T05:48:32....	MIDNIGHT	THEFT F/AUTO	OTHERS	2017-01-10T07:18:37....	1300 - 1399 BLOCK...	397302	139967	1	1A
	Point	15924	16063484	2016-04-23T08:51:12....	DAY	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	1240 - 1399 BLOCK...	401165	138586	5	5C
	Point	15925	16063495	2016-04-23T11:00:50....	DAY	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	1800 - 2299 BLOCK...	402158	138824	5	5C
	Point	15926	16063511	2016-04-23T10:23:43....	DAY	THEFT F/AUTO	OTHERS	2017-01-10T07:18:37....	1400 - 1499 BLOCK...	397116	138634	2	2B
	Point	15927	16063520	2016-04-23T10:53:05....	DAY	THEFT F/AUTO	OTHERS	2017-01-10T07:18:37....	600 - 699 BLOCK OF...	400251	140497	5	5E
	Point	15928	16063527	2016-04-23T11:35:51....	DAY	THEFT F/AUTO	OTHERS	2017-01-10T07:18:37....	400 - 499 BLOCK OF...	398392	142453	4	4D

0 of 35119 selected

100 %

```python
# Open a Search Cursor, print results
table = "Crime_Incidents_2016"
fields = ["Shift", "Offense", "Method", "Ward"]
qry = " Ward = '6' "
with arcpy.da.SearchCursor(table, fields, qry) as cursor:
    for row in cursor:
        print("Shift: {} Offense:{} Method: {} Ward: {}".format(row[0], row[1], row[2], row[3]))
```

Crime_Incidents_2016

Field: Add Delete Calculate Selection: Zoom To Switch Clear Delete

1	Shape	OBJECTID	CCN	REPORTDATE	SHIFT	OFFENSE	METHOD	LASTMODIFI	BLOCKSITEA	BLOCKXCOOR	BLOCKYCOOR	WARD	ANC
	Point	15914	16063221	2016-04-22T21:27:41....	EVENING	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	700 - 799 BLOCK OF...	398098	136808	2	2C
	Point	15915	16063234	2016-04-22T21:30:25....	EVENING	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	2800 - 2899 BLOCK...	395007	137489	2	2E
	Point	15916	16063250	2016-04-22T22:09:09....	EVENING	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	1100 - 1127 BLOCK...	397430	137381	2	2F
	Point	15917	16063374	2016-04-23T03:06:29....	MIDNIGHT	THEFT F/AUTO	OTHERS	2017-01-10T07:18:37....	2400 - 2499 BLOCK...	402216	139352	5	5C
	Point	15918	16063392	2016-04-23T02:10:11....	MIDNIGHT	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	2000 - 2099 BLOCK...	398101	138793	1	1B
	Point	15919	16063419	2016-04-23T03:18:48....	MIDNIGHT	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	1200 - 1299 BLOCK...	400936	134825	6	6B
	Point	15920	16063431	2016-04-23T04:47:13....	MIDNIGHT	ROBBERY	GUN	2017-01-10T07:18:37....	1200 - 1299 BLOCK...	400986	142561	5	5A
	Point	15921	16063439	2016-04-23T04:25:02....	MIDNIGHT	ASSAULT W/DANGE...	OTHERS	2017-01-10T07:18:37....	1127 - 1199 BLOCK...	398944	137478	6	6E
	Point	15922	16063447	2016-04-23T05:02:31....	MIDNIGHT	ROBBERY	OTHERS	2017-01-10T07:18:37....	900 - 931 BLOCK OF...	397863	138792	1	1B
	Point	15923	16063460	2016-04-23T05:48:32....	MIDNIGHT	THEFT F/AUTO	OTHERS	2017-01-10T07:18:37....	1300 - 1399 BLOCK...	397302	139967	1	1A
	Point	15924	16063484	2016-04-23T08:51:12....	DAY	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	1240 - 1399 BLOCK...	401165	138586	5	5C
	Point	15925	16063495	2016-04-23T11:00:50....	DAY	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	1800 - 2299 BLOCK...	402158	138824	5	5C
	Point	15926	16063511	2016-04-23T10:23:43....	DAY	THEFT F/AUTO	OTHERS	2017-01-10T07:18:37....	1400 - 1499 BLOCK...	397116	138634	2	2B
	Point	15927	16063520	2016-04-23T10:53:05....	DAY	THEFT F/AUTO	OTHERS	2017-01-10T07:18:37....	600 - 699 BLOCK OF...	400251	140497	5	5E
	Point	15928	16063527	2016-04-23T11:35:51....	DAY	THEFT F/AUTO	OTHERS	2017-01-10T07:18:37....	400 - 499 BLOCK OF...	398392	142453	4	4D

0 of 35119 selected

100 %

```python
# Open a Search Cursor, print results
table = "Crime_Incidents_2016"
fields = ["Shift", "Offense", "Method", "Ward"]
qry = " Ward = '6' "
with arcpy.da.SearchCursor(table, fields, qry) as cursor:
    for row in cursor:
        print("Shift: {} Offense:{} Method: {} Ward: {}".format(row[0], row[1], row[2], row[3]))
```

```
Shift: MIDNIGHT Offense:THEFT/OTHER Method: OTHERS Ward: 6
Shift: MIDNIGHT Offense:ASSAULT W/DANGEROUS WEAPON Method: OTHERS Ward: 6
Shift: EVENING Offense:THEFT/OTHER Method: OTHERS Ward: 6
Shift: EVENING Offense:BURGLARY Method: OTHERS Ward: 6
Shift: MIDNIGHT Offense:THEFT/OTHER Method: OTHERS Ward: 6
Shift: EVENING Offense:ROBBERY Method: OTHERS Ward: 6
Shift: MIDNIGHT Offense:ROBBERY Method: OTHERS Ward: 6
Shift: EVENING Offense:THEFT F/AUTO Method: OTHERS Ward: 6
Shift: EVENING Offense:THEFT/OTHER Method: OTHERS Ward: 6
Shift: DAY Offense:THEFT/OTHER Method: OTHERS Ward: 6
Shift: DAY Offense:THEFT F/AUTO Method: OTHERS Ward: 6
Shift: EVENING Offense:BURGLARY Method: OTHERS Ward: 6
Shift: DAY Offense:BURGLARY Method: OTHERS Ward: 6
Shift: DAY Offense:THEFT/OTHER Method: OTHERS Ward: 6
Shift: MIDNIGHT Offense:THEFT F/AUTO Method: OTHERS Ward: 6
Shift: EVENING Offense:THEFT F/AUTO Method: OTHERS Ward: 6
Shift: DAY Offense:THEFT/OTHER Method: OTHERS Ward: 6
Shift: DAY Offense:THEFT/OTHER Method: OTHERS Ward: 6
Shift: DAY Offense:THEFT/OTHER Method: OTHERS Ward: 6
Shift: DAY Offense:THEFT F/AUTO Method: OTHERS Ward: 6
Shift: EVENING Offense:THEFT/OTHER Method: OTHERS Ward: 6
```

Crime_Incidents_2016

Field: Add Delete Calculate Selection:

1	Shape	OBJECTID	CCN	REPORTD/	RD	ANC
	Point	15914	16063221	2016-04-22		2C
	Point	15915	16063234	2016-04-22		2E
	Point	15916	16063250	2016-04-22		2F
	Point	15917	16063374	2016-04-23		5C
	Point	15918	16063392	2016-04-23		1B
	Point	15919	16063419	2016-04-23		6B
	Point	15920	16063431	2016-04-23		5A
	Point	15921	16063439	2016-04-23		6E
	Point	15922	16063447	2016-04-23		1B
	Point	15923	16063460	2016-04-23		1A
	Point	15924	16063484	2016-04-23		5C
	Point	15925	16063495	2016-04-23		5C
	Point	15926	16063511	2016-04-23		2B
	Point	15927	16063520	2016-04-23		5E
	Point	15928	16063527	2016-04-23		4D

0 of 35119 selected

100 %

arcpy.da.SearchCursor – optional spatial reference parameter

Spatial Reference

Geographic coordinate system	GCS WGS 1984
WKID	4326
Authority	EPSG
Angular Unit	Degree (0.0174532925199433)
Prime Meridian	Greenwich (0.0)
Datum	D WGS 1984
Spheroid	WGS 1984
Semimajor Axis	6378137.0
Semiminor Axis	6356752.314245179
Inverse Flattening	298.257223563

Crime_Incidents_2016

Field: Add Delete Calculate Selection: Zoom To Swit

1	Shape	OBJECTID	CCN	REPORTDATE	SHI						OR	BLOCKYCOOR	WARD	ANC
	Point	15914	16063221	2016-04-22T21:27:41....	EVEI						098	136808	2	2C
	Point	15915	16063234	2016-04-22T21:30:25....	EVEI						007	137489	2	2E
	Point	15916	16063250	2016-04-22T22:09:09....	EVEI						430	137381	2	2F
	Point	15917	16063374	2016-04-23T03:06:29....	MID						216	139352	5	5C
	Point	15918	16063392	2016-04-23T02:10:11....	MID						101	138793	1	1B
	Point	15919	16063419	2016-04-23T03:18:48....	MID						936	134825	6	6B
	Point	15920	16063431	2016-04-23T04:47:13....	MID						986	142561	5	5A
	Point	15921	16063439	2016-04-23T04:25:02....	MID						944	137478	6	6E
	Point	15922	16063447	2016-04-23T05:02:31....	MID						863	138792	1	1B
	Point	15923	16063460	2016-04-23T05:48:32....	MID						302	139967	1	1A
	Point	15924	16063484	2016-04-23T08:51:12....	DAY	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	1240 - 1399 BLOCK...	401165		138586	5	5C
	Point	15925	16063495	2016-04-23T11:00:50....	DAY	THEFT/OTHER	OTHERS	2017-01-10T07:18:37....	1800 - 2299 BLOCK...	402158		138824	5	5C
	Point	15926	16063511	2016-04-23T10:23:43....	DAY	THEFT F/AUTO	OTHERS	2017-01-10T07:18:37....	1400 - 1499 BLOCK...	397116		138634	2	2B
	Point	15927	16063520	2016-04-23T10:53:05....	DAY	THEFT F/AUTO	OTHERS	2017-01-10T07:18:37....	600 - 699 BLOCK OF...	400251		140497	5	5E
	Point	15928	16063527	2016-04-23T11:35:51....	DAY	THEFT F/AUTO	OTHERS	2017-01-10T07:18:37....	400 - 499 BLOCK OF...	398392		142453	4	4D

0 of 35119 selected

100 %

```python
# Open a Search Cursor with a where clause, print results
table = "Crime_Incidents_2016"
fields = ["Offense", "SHAPE@XY"]
qry = " Ward = '6' "
# WKID: 2248 for NAD_1983_StatePlane_Maryland_FIPS_1900_Feet
sro = arcpy.SpatialReference(2248)
with arcpy.da.SearchCursor(table, fields, qry, sro) as cursor:
    for row in cursor:
        print("Offense:{} Coordinates: {}".format(row[0], row[1]))
```

```
Offense:ROBBERY Coordinates: (1308231.6409734625, 451799.8333681458)
Offense:THEFT F/AUTO Coordinates: (1312467.197828326, 440789.35427620757)
Offense:THEFT F/AUTO Coordinates: (1315108.2692477861, 443617.4334204988)
Offense:THEFT F/AUTO Coordinates: (1313418.6396913962, 443062.97236885206)
Offense:THEFT/OTHER Coordinates: (1316751.9669016926, 449053.7753106581)
Offense:SEX ABUSE Coordinates: (1313602.366157454, 450996.0289629688)
Offense:THEFT/OTHER Coordinates: (1312772.315310747, 447859.55179770023)
Offense:THEFT/OTHER Coordinates: (1316781.4944870411, 441826.09797414625)
Offense:THEFT F/AUTO Coordinates: (1305656.1861242952, 453459.935325456)
Offense:BURGLARY Coordinates: (1310990.8224155093, 443309.03488714236)
Offense:THEFT F/AUTO Coordinates: (1311322.1865406574, 449207.97437929467)
Offense:THEFT F/AUTO Coordinates: (1306384.5313509994, 450343.14282557205)
```

arcpy.da.UpdateCursor

```
arcpy.da.UpdateCursor(in_table, field_names,
{where_clause}, {spatial_reference},
{explode_to_points}, {sql_clause})
```

Method	Explanation
deleteRow ()	Deletes the current row
updateRow (row)	Updates the current row in the table.

arcpy.da.UpdateCursor

Parcel

Field: Add Delete Calculate Selection: Zoom To Switch Clear Delete

_S	CLASS	PLAT	LOT	BLOCK	BENTON_ZON	DEED_NUMBE	CREATION_D	BC_DATE	X_COORD	Y_COORD	Shape_Length	Shape_Area
02	400					287674-00	90	09-JUL-01	1270706.28095	331311.75629	887.174211	43627.16464
26								09-JUL-01	1271497.4375	332581.78125	12006.791035	4540105.783864
11	401					91989-87	27-AUG-87	09-JUL-01	1269824.54145	354034.79116	2276.933054	218262.976049
09	400				UR5/MHA	162707-93		09-JUL-01	1269464.76864	353662.88804	2081.500782	226910.95821
12	401				UR5/MHA	254560-98		09-JUL-01	1269904.5625	353556.90088	2427.249501	288014.722954
23	101	PARKSIDE	5		RS-5	134151-91		09-JUL-01	1278525.36595	346032.29688	559.342887	9704.661188
46	101				RS-3.5	244512-98		09-JUL-01	1273656.875	345696.4375	507.393847	15068.496938
25	101				RS-3.5	242080-98		09-JUL-01	1273630.625	345588.35937	417.976086	10903.048499
23	101				RS-5	176849-92		09-JUL-01	1273799.1875	343453.5625	400.093019	10004.6495
37	101				RS-5	131822-91		09-JUL-01	1273796.75	343341.07813	450.046661	12502.917
73	581				RR-5	70774-85	02-JUN-88	09-JUL-01	1251701.25491	338904.66116	8413.791687	1754300.600793

0 of 15041 selected

100 %

arcpy.da.UpdateCursor

Parcel

Field: Add Delete Calculate Selection: Zoom To Switch Clear Delete

_S	CLASS	PLAT	LOT	BLOCK	BENTON_ZON	DEED_NUMBE	CREATION_D	BC_DATE	X_COORD	Y_COORD	Shape_Length	Shape_Area
02	400					287674-00	90	09-JUL-01	1270706.28095	331311.75629	887.174211	43627.16464
26								09-JUL-01	1271497.4375	332581.78125	12006.791035	4540105.783864
11	401					91989-87	27-AUG-87	09-JUL-01	1269824.54145	354034.79116	2276.933054	218262.976049
09	400				UR5/MHA	162707-93		09-JUL-01	1269464.76864	353662.88804	2081.500782	226910.95821
12	401				UR5/MHA	254560-98		09-JUL-01	1269904.5625	353556.90088	2427.249501	288014.722954
23	101	PARKSIDE	5		RS-5	134151-91		09-JUL-01	1278525.36595	346032.29688	559.342887	9704.661188
46	101				RS-3.5	244512-98		09-JUL-01	1273656.875	345696.4375	507.393847	15068.496938
25	101				RS-3.5	242080-98		09-JUL-01	1273630.625	345588.35937	417.976086	10903.048499
23	101				RS-5	176849-92		09-JUL-01	1273799.1875	343453.5625	400.093019	10004.6495
37	101				RS-5	131822-91		09-JUL-01	1273796.75	343341.07813	450.046661	12502.917
73	581				RR-5	70774-85	02-JUN-88	09-JUL-01	1251701.25491	338904.66116	8413.791687	1754300.600793

0 of 15041 selected 100 %

```python
# Add a new field named ACRES to the Parcel feature class
arcpy.AddField_management("Parcel", "ACRES", "Double")

# Update ACRES field using the SHAPE@AREA token
with arcpy.da.UpdateCursor("Parcel", ["SHAPE@AREA", "ACRES"]) as cursor:
    for row in cursor:
        row[1] = row[0] / 43560
        cursor.updateRow(row)

print ("Script completed")
```

arcpy.da.UpdateCursor

Parcel

Field: Add Delete Calculate Selection: Zoom To Switch Clear Delete

PLAT	LOT	BLOCK	BENTON_ZON	DEED_NUMBE	CREATION_D	BC_DATE	X_COORD	Y_COORD	Shape_Length	Shape_Area	ACRES
				287674-00	90	09-JUL-01	1270706.28095	331311.75629	887.174211	43627.16464	1.001542
						09-JUL-01	1271497.4375	332581.78125	12006.791035	4540105.783864	104.226487
				91989-87	27-AUG-87	09-JUL-01	1269824.54145	354034.79116	2276.933054	218262.976049	5.010628
			UR5/MHA	162707-93		09-JUL-01	1269464.76864	353662.88804	2081.500782	226910.95821	5.209159
			UR5/MHA	254560-98		09-JUL-01	1269904.5625	353556.90088	2427.249501	288014.722954	6.611908
PARKSIDE	5		RS-5	134151-91		09-JUL-01	1278525.36595	346032.29688	559.342887	9704.661188	0.222788
			RS-3.5	244512-98		09-JUL-01	1273656.875	345696.4375	507.393847	15068.496938	0.345925
			RS-3.5	242080-98		09-JUL-01	1273630.625	345588.35937	417.976086	10903.048499	0.2503
			RS-5	176849-92		09-JUL-01	1273799.1875	343453.5625	400.093019	10004.6495	0.229675
			RS-5	131822-91		09-JUL-01	1273796.75	343341.07813	450.046661	12502.917	0.287027
			RR-5	70774-85	02-JUN-88	09-JUL-01	1251701.25491	338904.66116	8413.791687	1754300.600793	40.2732

0 of 15041 selected 100 %

```python
arcpy.AddField_management("Parcel", "ACRES", "Double")

# Update ACRES field using the SHAPE@AREA token
with arcpy.da.UpdateCursor("Parcel", ["SHAPE@AREA", "ACRES"]) as cursor:
    for row in cursor:
        row[1] = row[0] / 43560
        cursor.updateRow(row)

print ("Script completed")
```

arcpy.da.InsertCursor

```
arcpy.da.InsertCursor(in_table, field_names)
```

Method	Explanation
insertRow (row)	Inserts a row into a table.

arcpy.da.InsertCursor

	A	B	C	D
1	Latitude	Longitude	Description	
2	38.99583	-77.041	North Corner	
3	38.9855	-77.0277	Northeast Mile 1	
4	38.97533	-77.0145	Northeast Mile 2	
5	38.96517	-77.0017	Northeast Mile 3	
6	38.95467	-76.9882	Northeast Mile 4	
7	38.94433	-76.9748	Northeast Mile 5	
8	38.93367	-76.9613	Northeast Mile 6	
9	38.91357	-76.9355	Northeast Mile 8	
10	38.90322	-76.9223	Northeast Mile 9	
11	38.8929	-76.9092	East Corner	
12	38.90352	-77.1591	Northwest Mile 1	
13	38.91407	-77.1459	Northwest Mile 2	
14	38.9376	-77.1157	Northwest Mile 4	
15	38.94442	-77.1067	Northwest Mile 5	
16	38.95488	-77.0936	Northwest Mile 6	
17	38.96517	-77.0803	Northwest Mile 7	
18	38.97533	-77.0672	Northwest Mile 8	
19	38.98567	-77.054	Northwest Mile 9	

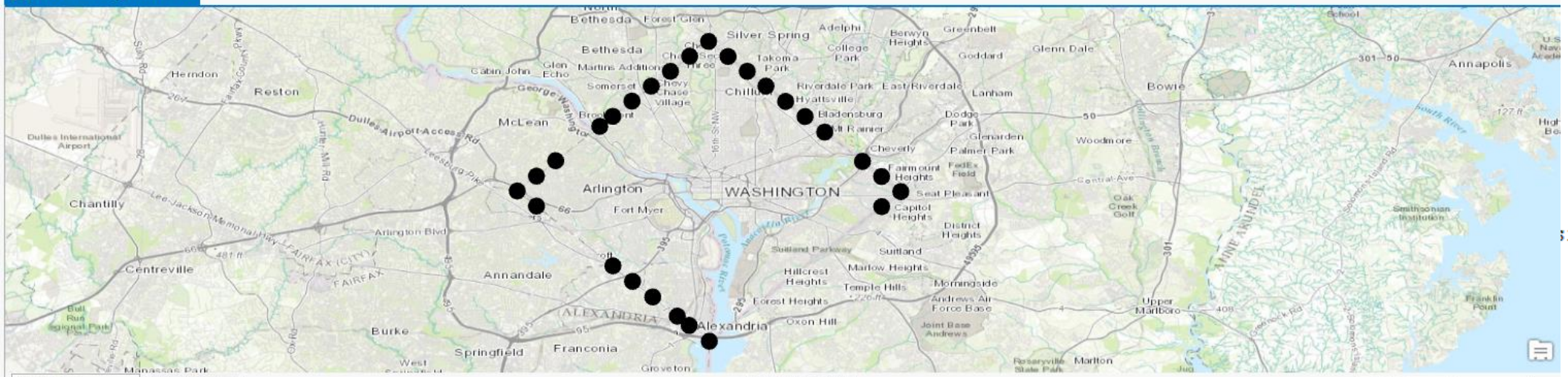
boundaryStones

```python
import arcpy
import csv

arcpy.env.workspace = r"C:\demo\DC.gdb"
file = r"C:\demo\boundaryStones.csv"

# Create empty feature class
sr = arcpy.SpatialReference(4326)

# Create an empty feature class
arcpy.CreateFeatureclass_management(r"C:\demo\DC.gdb", "BoundaryStones", "Point", spatial_reference=sr)

# Add fields to  feature class
arcpy.AddField_management("BoundaryStones", "Description", "Text")

# Create Insert Cursor
cursor = arcpy.da.InsertCursor("BoundaryStones", ["SHAPE@XY", "Description"])

with open(file, newline='') as csvfile:
    reader = csv.reader(csvfile)
    count = 0
    for line in reader:
        if count > 0:
            # SHAPE@XY (X,Y) passed as a tuple
            cursor.insertRow([(float(line[1]), float(line[0])),line[2]])
        count = +1
del cursor
```

boundaryStones

```python
import arcpy
import csv
```

DCBoundaryStones



BoundaryStones

Field: Add Delete Calculate Selection: Zoom To Switch Clear Delete

OBJECTID	Shape	Description
1	Point	North Corner
2	Point	Northeast Mile 1
3	Point	Northeast Mile 2
4	Point	Northeast Mile 3
5	Point	Northeast Mile 4
6	Point	Northeast Mile 5
7	Point	Northeast Mile 6
8	Point	Northeast Mile 8
9	Point	Northeast Mile 9
10	Point	East Corner

0 of 27 selected

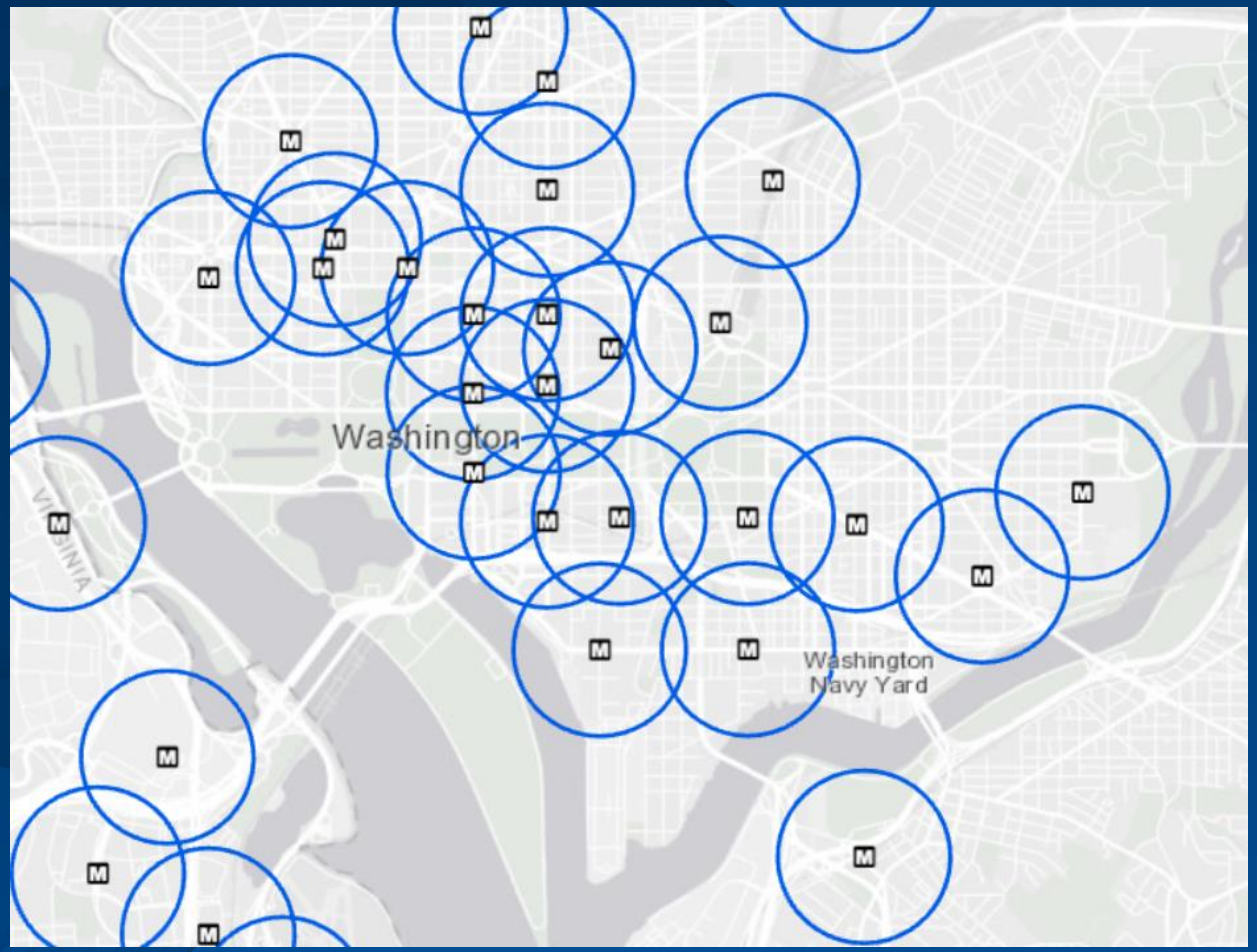
100 %

Working with Geometry Objects

- Reading geometries
  - arcpy.da.SearchCursor
  - arcpy.da.UpdateCursor
- Writing geometries
  - arcpy.da.UpdateCursor
  - arcpy.da.InsertCursor
- Work with geoprocessing tools

Using geometry as input with geoprocessing tools

```
import arcpy

# List of coordinates.
coordinates = [
    [2365000, 7355000],
    [2365000, 7455000],
    [2465000, 7455000],
    [2465000, 7355000]]

# Create an array with a point object for each coordinate pair
array = arcpy.Array([arcpy.Point(x, y) for x, y in coordinates])

# Create a polygon geometry object using the array object
boundary = arcpy.Polygon(array, arcpy.SpatialReference(2953))

# Use the geometry to clip an input feature class
arcpy.Clip_analysis('c:/data/rivers.shp',
                    boundary,
                    'c:/data/rivers_clipped.shp')
```



Demo: Which buffers overlap?

Which buffers overlap?

```
for row in arcpy.da.SearchCursor('BuffHalfMi',['OID@','SHAPE@', 'NAME']):
    for row2 in arcpy.da.SearchCursor('BuffHalfMi',['OID@','SHAPE@','NAME']):
        if row2[1].overlaps(row[1]):
            print ('{0} (oid:{1}) overlaps {2} (oid:{3})'.format(row2[2], row2[0], row[2], row[0]))
```

```
Federal Center SW (oid:6) overlaps Archives-Navy Mem'l (oid:1)
Judiciary Sq (oid:7) overlaps Archives-Navy Mem'l (oid:1)
Metro Center (oid:9) overlaps Archives-Navy Mem'l (oid:1)
Gallery Pl-Chinatown (oid:10) overlaps Archives-Navy Mem'l (oid:1)
L'Enfant Plaza (oid:88) overlaps Archives-Navy Mem'l (oid:1)
```

Leveraging other Python modules

- Python Standard Library
  - *Read a csv with the csv module*
  - *Unzip a .zip file with the zipfile module*
  - *Download data from the internet with the urllib module*
  - *Automatically open an output with the os module*
  - *… many more*
- Can install other pre-existing Python packages with ArcGIS Pro Python Package Manager

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

Jordan Douthitt

Advisor: Stephen A. Matthews

March 20 & 21, 2018

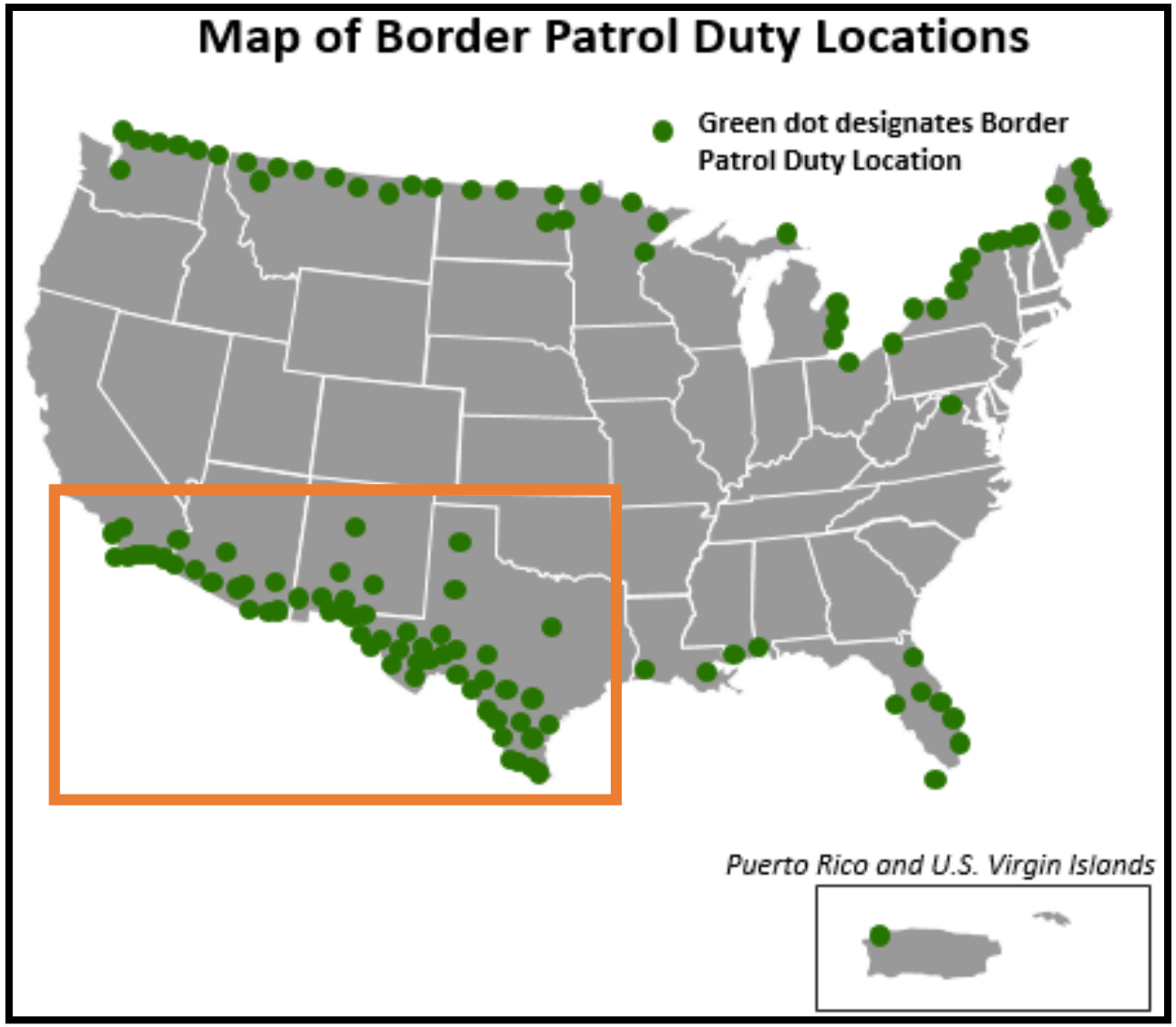
Background Information

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



Problem

Need to strategically recruit a large amount of skilled employees (United States Border Patrol Agents) to hard to fill positions in undesirable locations.

Goal

Analysis answering two questions

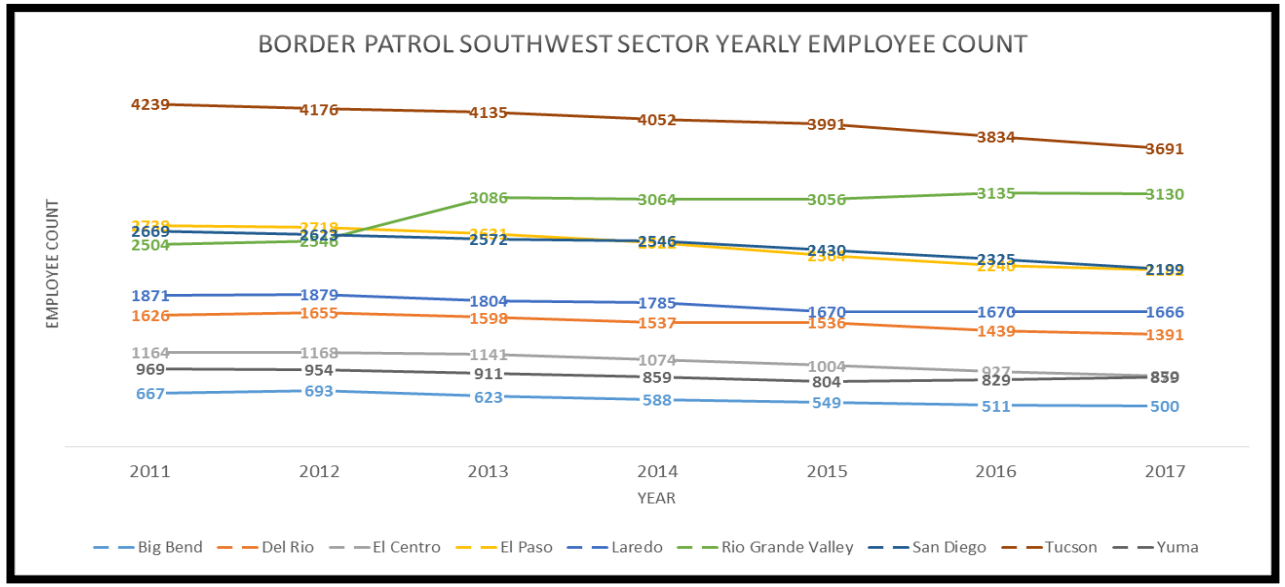
1. Which Border Patrol Sectors have the greatest recruitment support need?

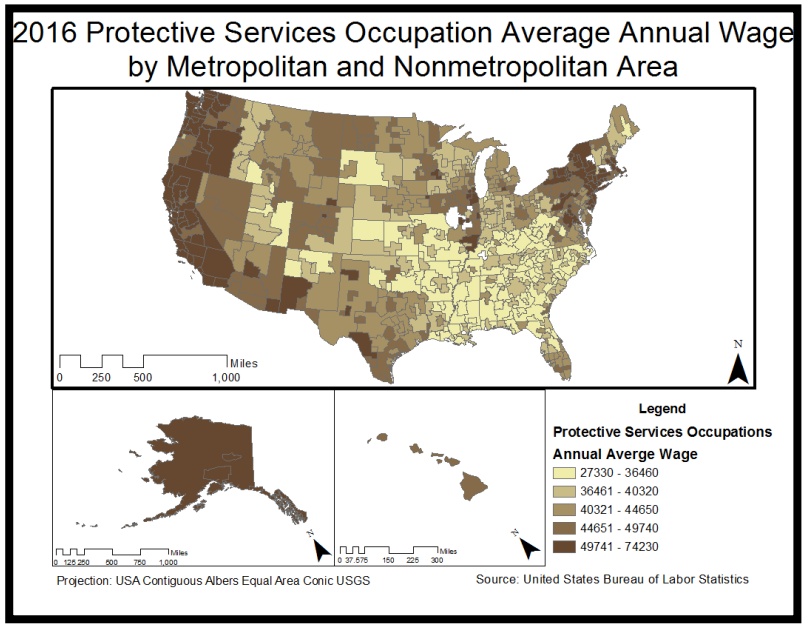
2. Which areas outside of these Border Patrol Sectors can additional recruiting focus on?

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

Methodology: Sector Profiles

Sector Information
- Agent Count
- Agent Turnover
- Station Count

Workload
- Border crossing locations
- Apprehensions per agent
- Border entries per agent

Labor Pool
- Percent of stations in nonmetropolitan areas
- Population of labor pools

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

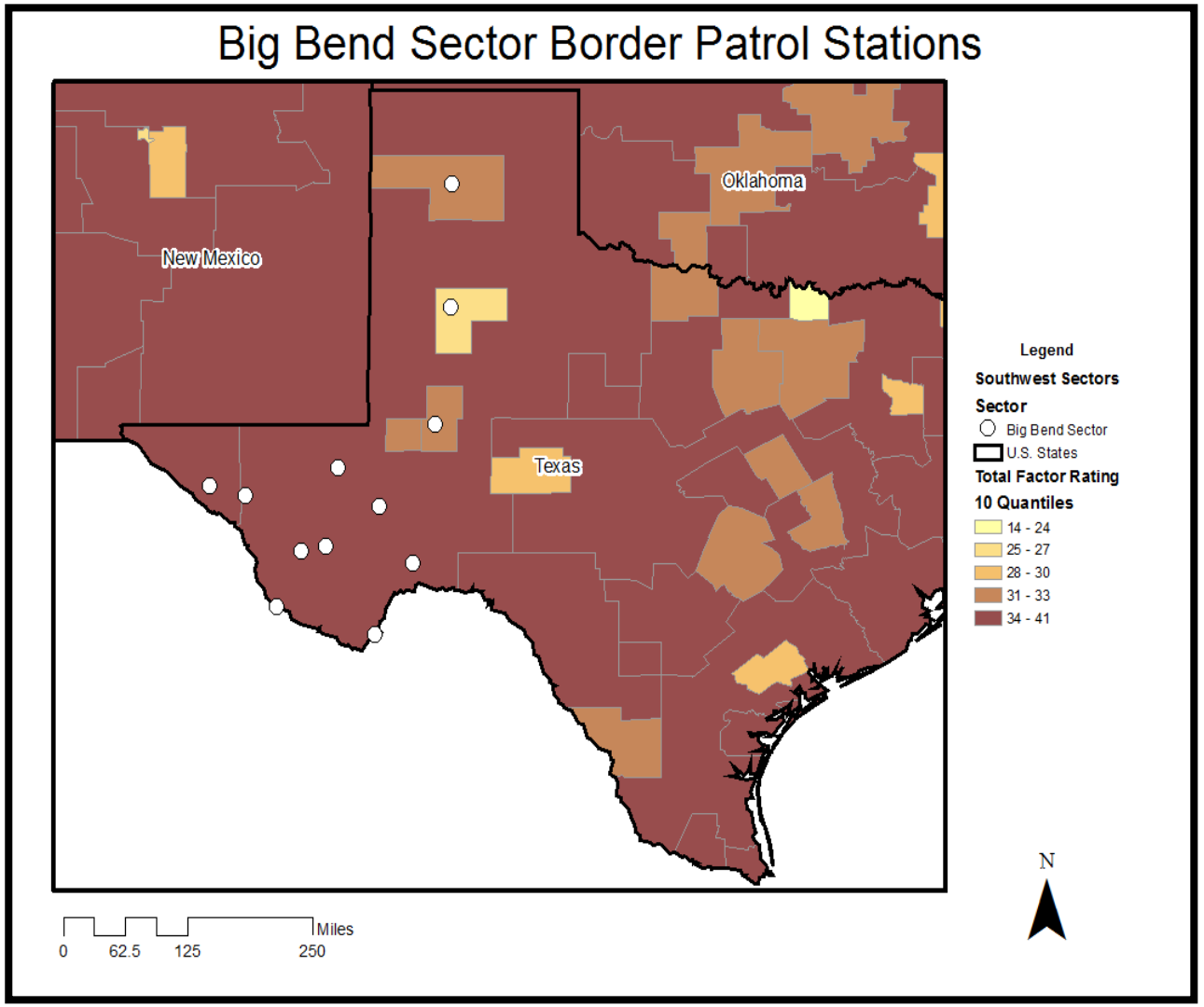
Results Testing
- R^2 correlation between factor scores
- & total score

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



Sector Information

- 2017 Agent Count: 500
- 2011-2017 Agent Change: 25% loss
- Station Count: 13

Workload

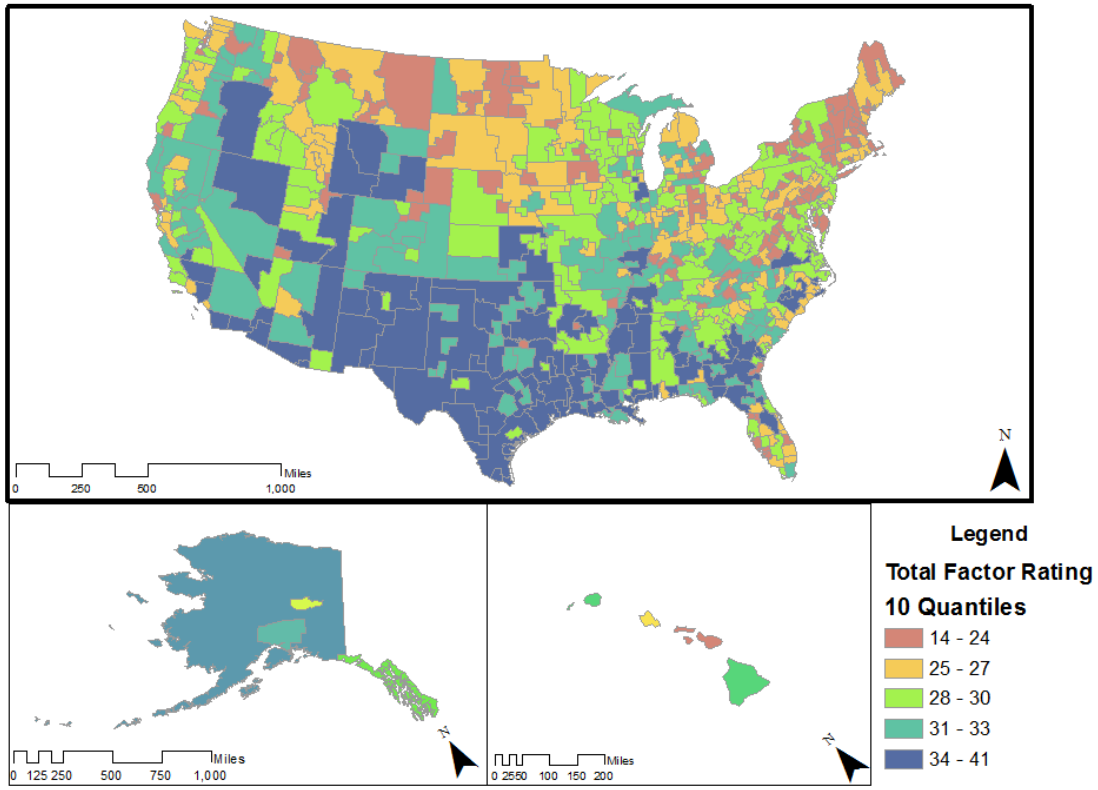
- Border Crossing Count: 2
- Apprehensions per Agent: 12
- Border Entries per Agent: 2,990

Labor Pool

- Percent of Stations in Nonmetropolitan Areas: 69.2%
- Population of Surrounding Labor Pools: 2,142,736

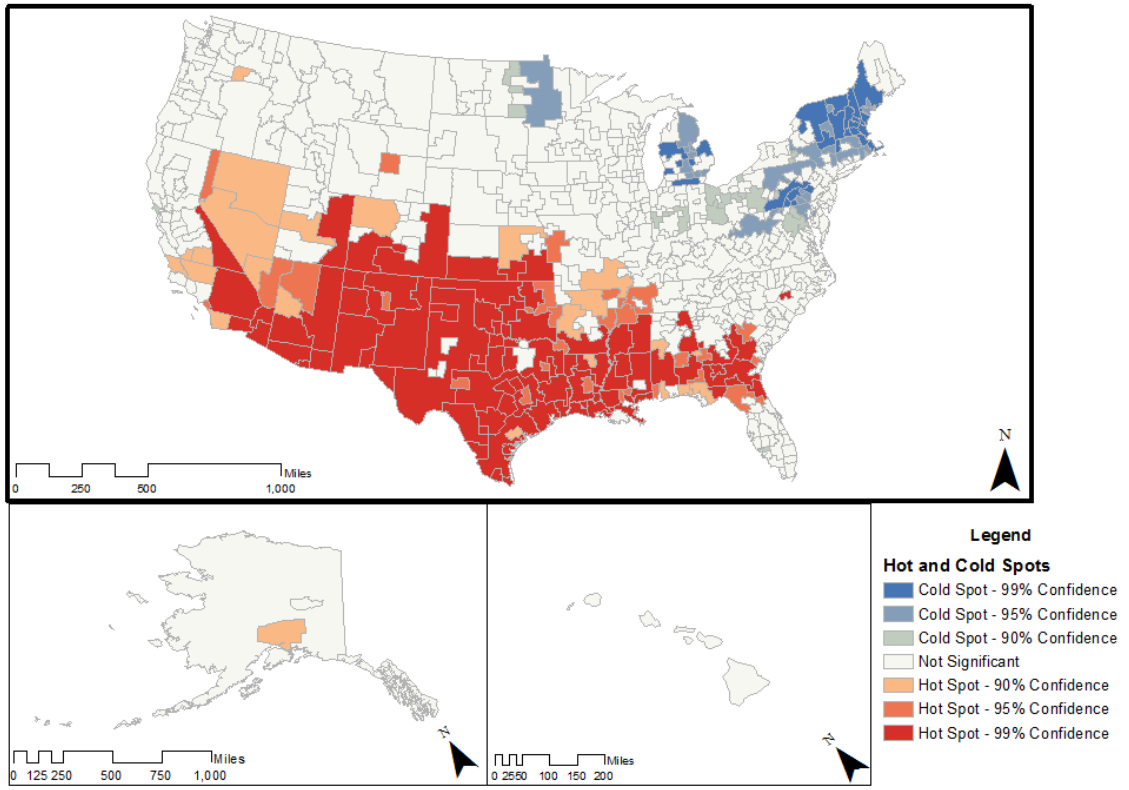
Results: Labor Pool Analysis

Total Rating by Metropolitan and Nonmetropolitan Area



Projection: USA Contiguous Albers Equal Area Conic USGS

Hot and Cold Spot Clusters by Metropolitan and Nonmetropolitan Area Using Getis-Ord Gi* Statistic



Projection: USA Contiguous Albers Equal Area Conic USGS

Conclusions

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

ArcPy

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

ArcPy

```
#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 ouput geodatabase path
    if arcpy.Exists(outputGDBCheck):
        arcpy.AddMessage("FileGDB Already Exists")
        outputGDB = outputGDBCheck
    #If the output file geodatabase does not already exist in the output folder, create it
    else:
        #Create file geodatabase to output new shapefiles into
        outputGDB = arcpy.CreateFileGDB_management(outputFolder, "Output_GDB")
```

Import Modules

Input Parameters (script tool)

Check for / Create Geodatabase

ArcPy

```
#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
    fcName = table.replace(".csv", "")

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

    #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[:6]
    #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

ArcPy

```
        #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))

#Delete the built layer, even if the rest of the script does not run
finally:
    #Delete layer
    arcpy.Delete_management(lyrName)
```

Join CSV to feature class

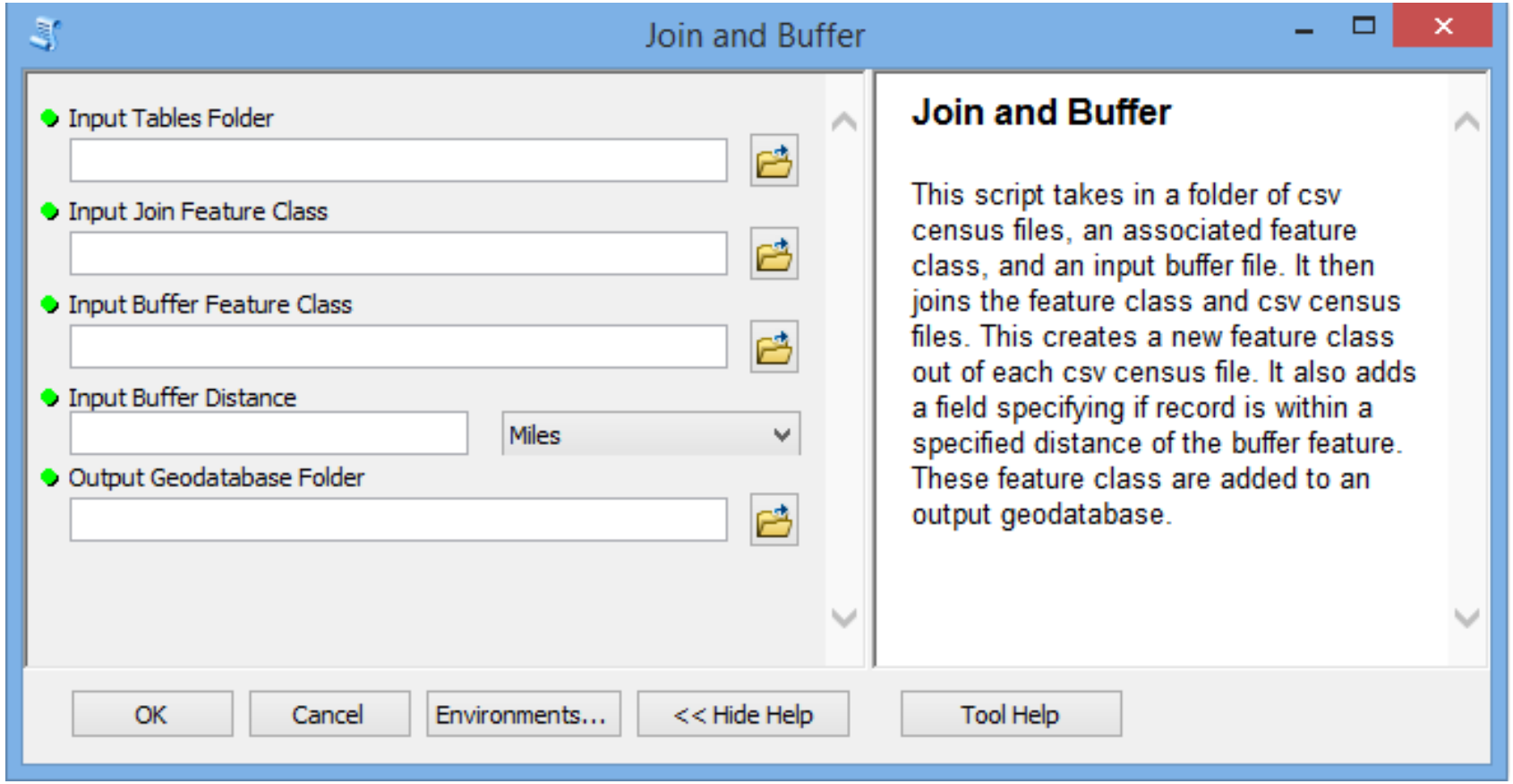
Create feature class

Error Handling

ArcPy

Python Script Tool

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



Thank You

Additional Python Sessions

- Python: Building Geoprocessing Tools (145 B)
  - *Wednesday 4:00 pm*
- ArcGIS API for Python: Scripting Your Web GIS (145 B)
  - *Wednesday 8:15 am*

Print Your Certificate of Attendance

Print stations located in the 140 Concourse

Tuesday

12:30 pm – 6:30 pm
GIS Solutions Expo
Hall B

5:00 pm – 6:30 pm
GIS Solutions Expo Social
Hall B

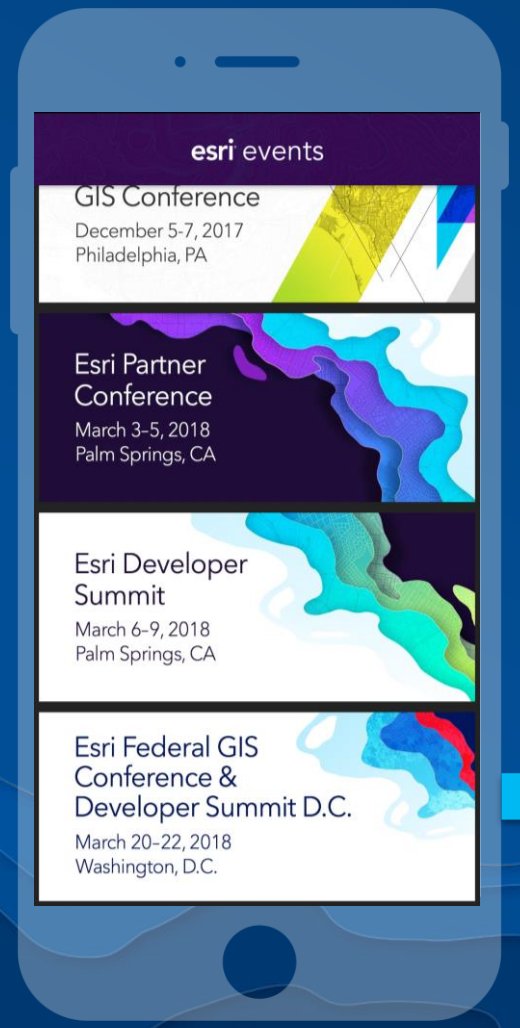
Wednesday

10:30 am – 5:15 pm
GIS Solutions Expo
Hall B

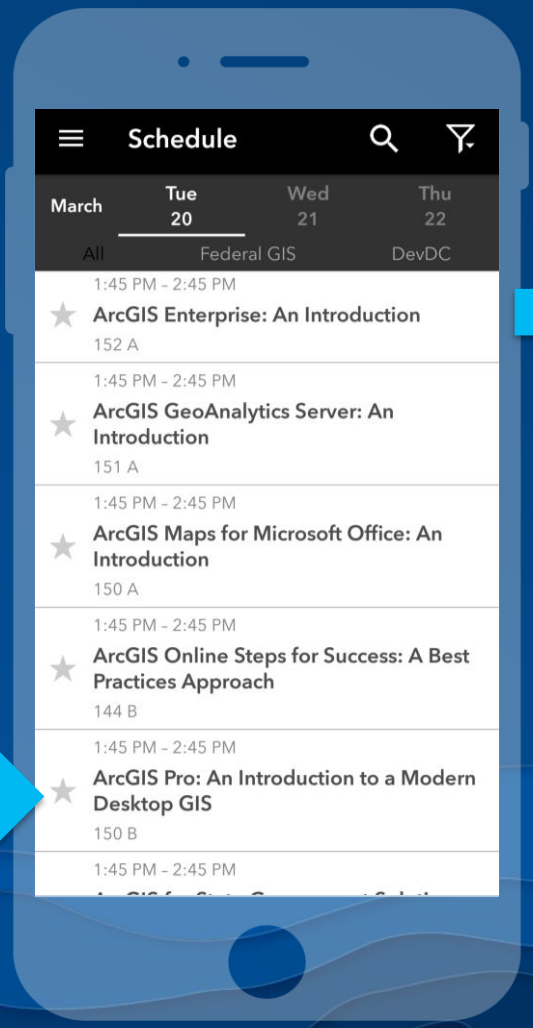
6:30 pm – 9:00 pm
Networking Reception
Smithsonian National Portrait Gallery

Please Take Our Survey in the Esri Events App

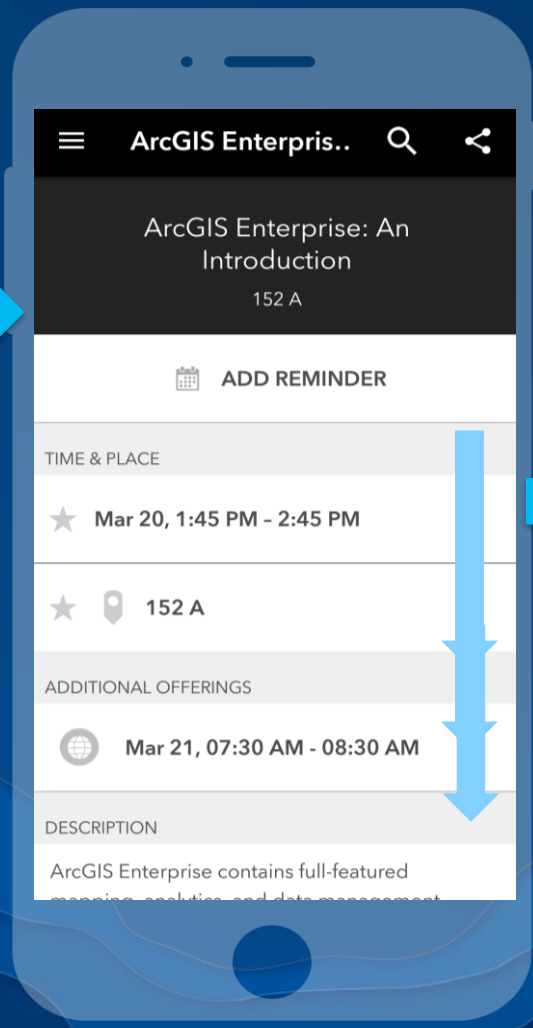
Download the Esri Events app and find your event



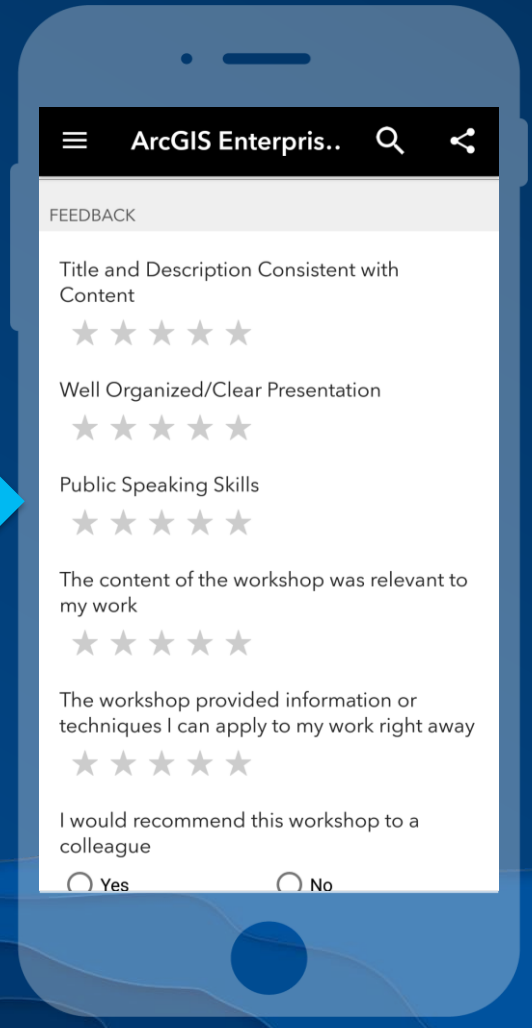
Select the session you attended



Scroll down to find the feedback section



Complete answers and select "Submit"



esri

THE SCIENCE OF WHERE