

CHRISTOPHER PRENER, PH.D.
SPRING, 2018

WEEK 10
LECTURE 09

INTRO TO GIS_c

SPATIAL JOINS

AGENDA

1. Front Matter
2. GISc & Public Policy
3. Table Joins in R
4. Spatial Joins in ArcMap
5. Back Matter

1

FRONT MATTER

1. FRONT MATTER

ANNOUNCEMENTS



Midterm feedback was disseminated last week - please let us know asap if there are questions or issues to be addressed.



Lab-07 and LP-08 are due by 5pm **on Wednesday.**



Lab-08, PS-05, and LP-09 are due on **next Monday.**



Extra credit - "bounty" on typos & suggestions - see Slack for details!



New lecture repo template being used!

2 GIS_c & PUBLIC POLICY

3 TABLE JOINS IN R

KEY TERM

Table joins are a set of techniques for combining the variables of two sets of tabular data into a single data set. They are completed based on *shared attributes*.



KEY TERM

A **left join** combines the variables of the first specified table with the variables of the second specified table based on a common ID variable.



CONCEPTUALIZING JOINS

id	a

CONCEPTUALIZING JOINS

id	a		id	b

CONCEPTUALIZING JOINS

id	a
1	
2	
3	
4	
5	

= =

id	b
1	
2	
3	
4	
5	



CONCEPTUALIZING JOINS

id	a
1	
2	
3	
4	
5	

!=

id	b
a	
b	
c	
d	
e	



CONCEPTUALIZING JOINS

id	a
1	
2	
3	
4	
5	

!=

id	b
1	
2	
3	
4	
4	



CONCEPTUALIZING JOINS

id	a		id	b
1			1	
2			2	
3			3	
4			4	
5			4	

CONCEPTUALIZING JOINS

id	a
1	
2	
3	
4	
5	

x

!=

id	b
1	
2	
3	
4	
4	

y

```
x <- mutate(x,  
             as.character(id))
```

CONCEPTUALIZING JOINS

id	a
1	
2	
3	
4	
5	

==

id	b
1	
2	
3	
4	
4	

```
x <- mutate(x,  
  as.character(id))
```



CONCEPTUALIZING JOINS

id	a
1	
2	
3	
4	
5	

x

!=

id	b
1	
2	
3	
4	
4	

y

```
y <- mutate(y,  
             as.numeric(id))
```

CONCEPTUALIZING JOINS

id	a
1	
2	
3	
4	
5	

==

id	b
1	
2	
3	
4	
4	

```
y <- mutate(y,  
  as.character(id))
```



CONCEPTUALIZING JOINS

id a		+	id b		=	id a b		
1			1			1		
2			2			2		
3			3			3		
4			4			4		
5			5			5		

CONCEPTUALIZING JOINS

id a		+	id a		=	id a.x a.y		
1			1			1		
2			2			2		
3			3			3		
4			4			4		
5			5			5		

CONCEPTUALIZING JOINS

id a		+	id geo		=	id a geo		
1			1			1		
2			2			2		
3			3			3		
4			4			4		
5			5			5		

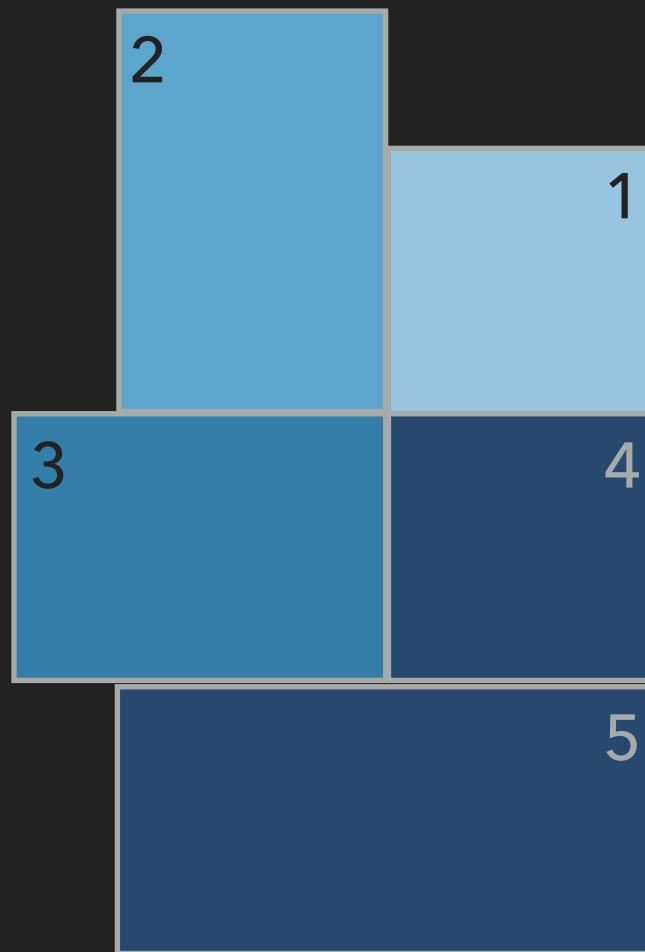
CONCEPTUALIZING JOINS

id	a	geo
1		
2		
3		
4		
5		

3. TABLE JOINS IN R

CONCEPTUALIZING JOINS

Use to combine tabular data with geometric data to create choropleth maps.



id	a	geo
1		
2		
3		
4		
5		

LEFT JOINS

f(x) `left_join(x = xtable, y = ytable, by = "id")`

Parameters:

- ▶ *xtable*
- ▶ *ytable*
- ▶ *id* is the shared identification variable



Available in `dplyr`
Download via CRAN

LEFT JOINS

f(x) `left_join(x = xtable, y = ytable, by = “id”)`

Parameters:

- ▶ *xtable* is the table whose columns will appear first
- ▶ *ytable* is the table whose columns will appear second
- ▶ *id* is the shared identification variable

3. TABLE JOINS IN R

LEFT JOINS

```
> # example set-up  
>  
> library(stlData)  
>  
> tracts <- stl_sf_tracts  
> asthma <- stl_tbl_asthma
```

3. TABLE JOINS IN R

LEFT JOINS

```
> tracts
Simple feature collection with 106 features and 12 fields
geometry type:  POLYGON
dimension:      XY
bbox:           xmin: -90.32052 ymin: 38.53185 xmax: -90.16657 ymax: 38.77443
epsg (SRID):    4269
proj4string:    +proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs
```

First 10 features:

	STATEFP	COUNTYFP	TRACTCE	GEOID	NAME	NAMELSAD	MTFCC	FUNCSTAT	ALAND	AWATER
313	29	510	112100	29510112100	1121 Census Tract	1121	G5020	S	6936664	0
314	29	510	116500	29510116500	1165 Census Tract	1165	G5020	S	904024	0
324	29	510	110300	29510110300	1103 Census Tract	1103	G5020	S	938287	0
536	29	510	103700	29510103700	1037 Census Tract	1037	G5020	S	910953	0
537	29	510	103800	29510103800	1038 Census Tract	1038	G5020	S	1640334	35369
538	29	510	104500	29510104500	1045 Census Tract	1045	G5020	S	1939712	0
539	29	510	106100	29510106100	1061 Census Tract	1061	G5020	S	962321	0

3. TABLE JOINS IN R

LEFT JOINS

```
> tracts
Simple feature collection with 106 features and 12 fields
geometry type: POLYGON
dimension: XY
bbox: xmin: -90.32052 ymin: 38.53185 xmax: -90.16657 ymax: 38.77443
epsg (SRID): 4269
proj4string: +proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs
```

First 10 features:

	STATEFP	COUNTYFP	TRACTCE	GEOID	NAME	NAMELSAD	MTFCC	FUNCSTAT	ALAND	AWATER
313	29	510	112100	29510112100	1121 Census Tract	1121	G5020	S	6936664	0
314	29	510	116500	29510116500	1165 Census Tract	1165	G5020	S	904024	0
324	29	510	110300	29510110300	1103 Census Tract	1103	G5020	S	938287	0
536	29	510	103700	29510103700	1037 Census Tract	1037	G5020	S	910953	0
537	29	510	103800	29510103800	1038 Census Tract	1038	G5020	S	1640334	35369
538	29	510	104500	29510104500	1045 Census Tract	1045	G5020	S	1939712	0
539	29	510	106100	29510106100	1061 Census Tract	1061	G5020	S	962321	0

3. TABLE JOINS IN R

LEFT JOINS

```
> asthma
# A tibble: 106 x 6
  geoID      tractCE nameLSAD      pctAsthma pctAsthma_Low pctAsthma_High
  <chr>      <int> <chr>      <dbl>      <dbl>      <dbl>
1 29510118100 118100 Census Tract 1181      11.9        11.1        12.7
2 29510117400 117400 Census Tract 1174       9.60        9.30       10.0
3 29510126700 126700 Census Tract 1267      14.5        13.5       15.5
4 29510119102 119102 Census Tract 1191.02     9.00        8.50        9.70
5 29510126800 126800 Census Tract 1268       9.30        8.80        9.80
6 29510126900 126900 Census Tract 1269      13.6        12.6       14.6
7 29510108100 108100 Census Tract 1081      12.7        11.8       13.8
8 29510127000 127000 Census Tract 1270      12.8        11.7       14.2
9 29510127400 127400 Census Tract 1274      12.7        11.9       13.8
10 29510103700 103700 Census Tract 1037       8.60        8.10        9.20
# ... with 96 more rows
```

3. TABLE JOINS IN R

LEFT JOINS

```
> asthma
# A tibble: 106 x 6
  geoID      tractCE nameLSAD      pctAsthma pctAsthma_Low pctAsthma_High
  <chr>      <int> <chr>      <dbl>      <dbl>      <dbl>
1 29510118100 118100 Census Tract 1181      11.9        11.1        12.7
2 29510117400 117400 Census Tract 1174       9.60        9.30       10.0
3 29510126700 126700 Census Tract 1267      14.5        13.5       15.5
4 29510119102 119102 Census Tract 1191.02     9.00        8.50        9.70
5 29510126800 126800 Census Tract 1268       9.30        8.80        9.80
6 29510126900 126900 Census Tract 1269      13.6        12.6       14.6
7 29510108100 108100 Census Tract 1081      12.7        11.8       13.8
8 29510127000 127000 Census Tract 1270      12.8        11.7       14.2
9 29510127400 127400 Census Tract 1274      12.7        11.9       13.8
10 29510103700 103700 Census Tract 1037       8.60        8.10        9.20
# ... with 96 more rows
```

LEFT JOINS

```
> class(tracts$GEOID)
[1] "character"
>
> is.character(tracts$GEOID)
[1] TRUE
>
> is.numeric(tracts$GEOID)
[1] FALSE
>
> class(asthma$geoID)
[1] "character"
```

- ▶ We can use the base R functions `class()`, `is.character()`, and `is.numeric()` to identify if the two identification variables are the same type.
- ▶ We can also do this visually, by inspecting objects in the global environment tab or using dplyr's `glimpse()` function.

LEFT JOINS

```
> class(tracts$GEOID)
[1] "character"
>
> is.character(tracts$GEOID)
[1] TRUE
>
> is.numeric(tracts$GEOID)
[1] FALSE
>
> class(asthma$geoID)
[1] "character"
```

- ▶ Finally, we want to note if our identification variables are identically named.
- ▶ If they **are**, we can proceed with the join.
- ▶ If they are **not**, we need to modify the join syntax **or** rename one of the variables.

LEFT JOINS

f(x)

```
left_join(x = xtable, y = ytable, by = "id")
```



Join tract and asthma data if id variables are identically named:

```
> left_join(x = tracts, y = asthma, by = "GEOID")
```



This will not work because the assumption of identically named identification variables does not hold!

LEFT JOINS

f(x)

```
left_join(x = xtable, y = ytable, by = "id")
```



Join tract and asthma data if id variables are identically named:

```
> left_join(x = tracts, y = asthma,  
            by = c("GEOID" = "geoID"))
```

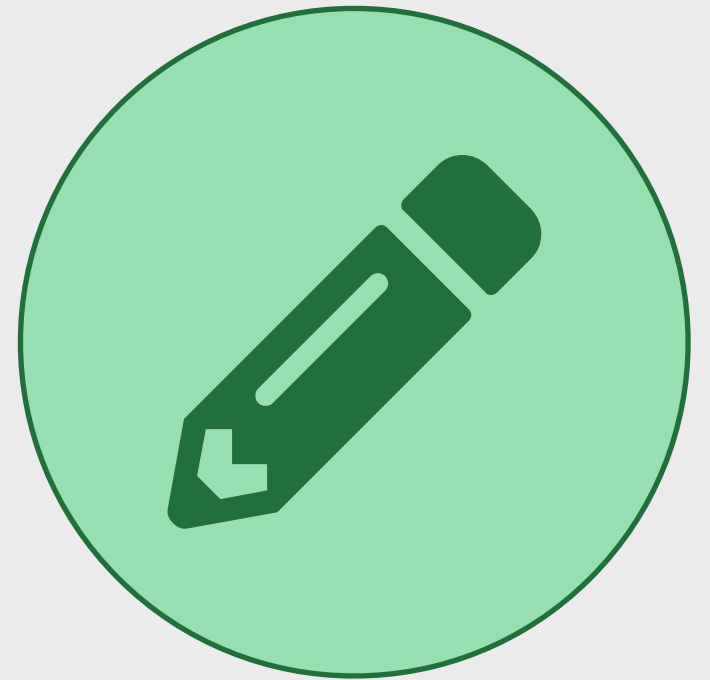


This will not work because the assumption of identically named identification variables does not hold!

4 SPATIAL JOINS IN ARCMAP

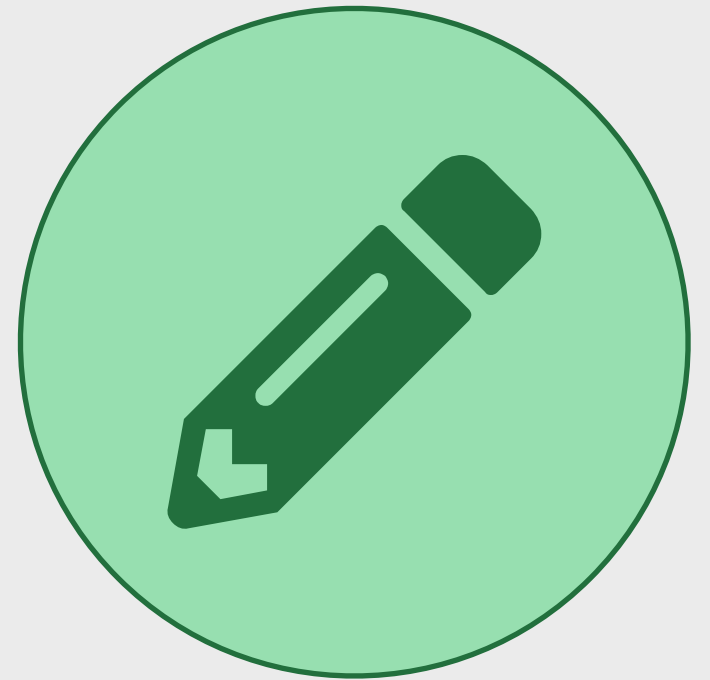
KEY TERM

Spatial joins are a set of techniques for combining or summarizing the variables from two geometric data sets into a single data set. They are completed based on *spatial proximity*.



KEY TERM

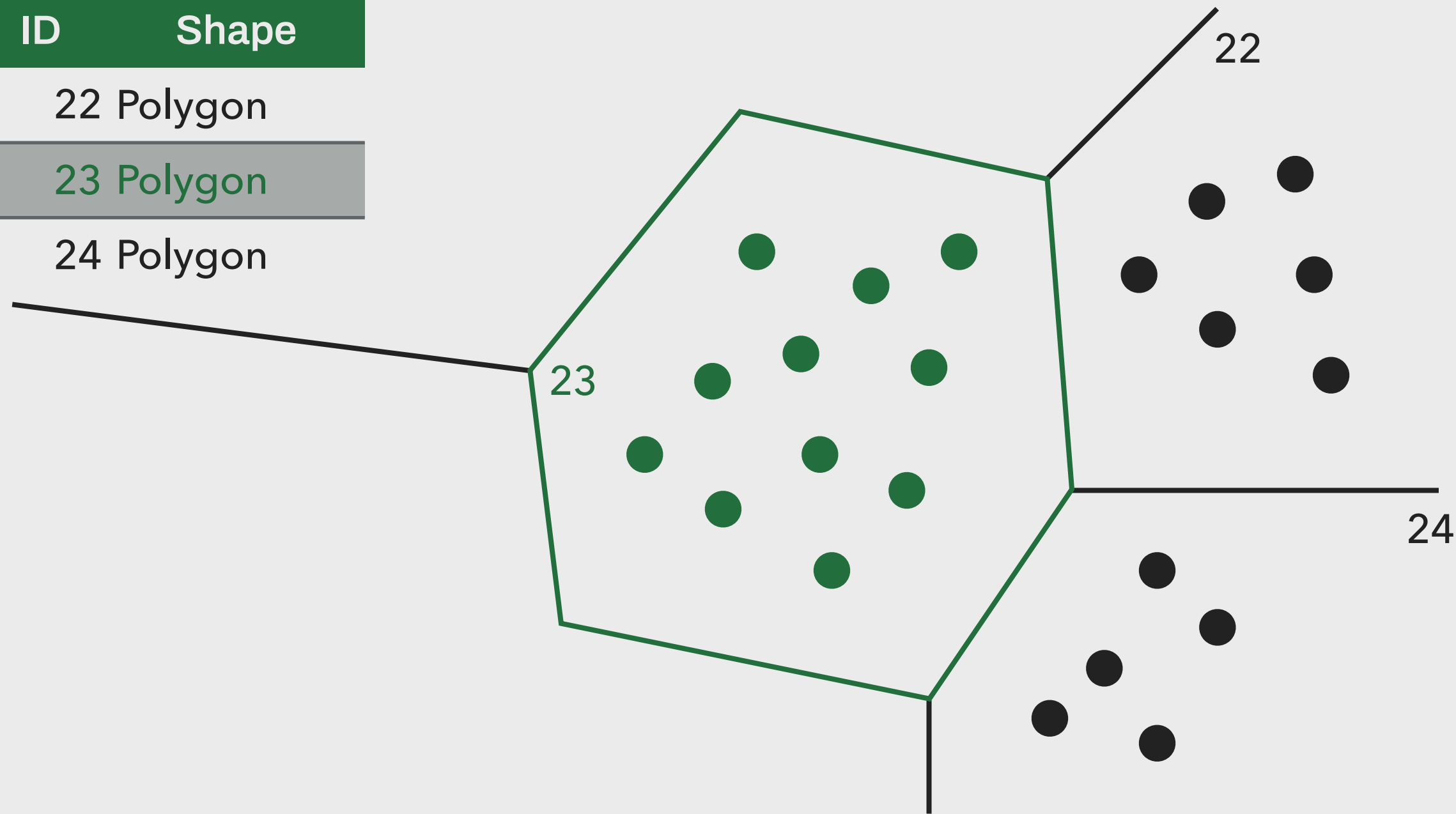
A **point to polygon** join can provide a count or mean of point values that is proximate to it.



CONCEPTUALIZING POINT TO POLYGON

Target Layer

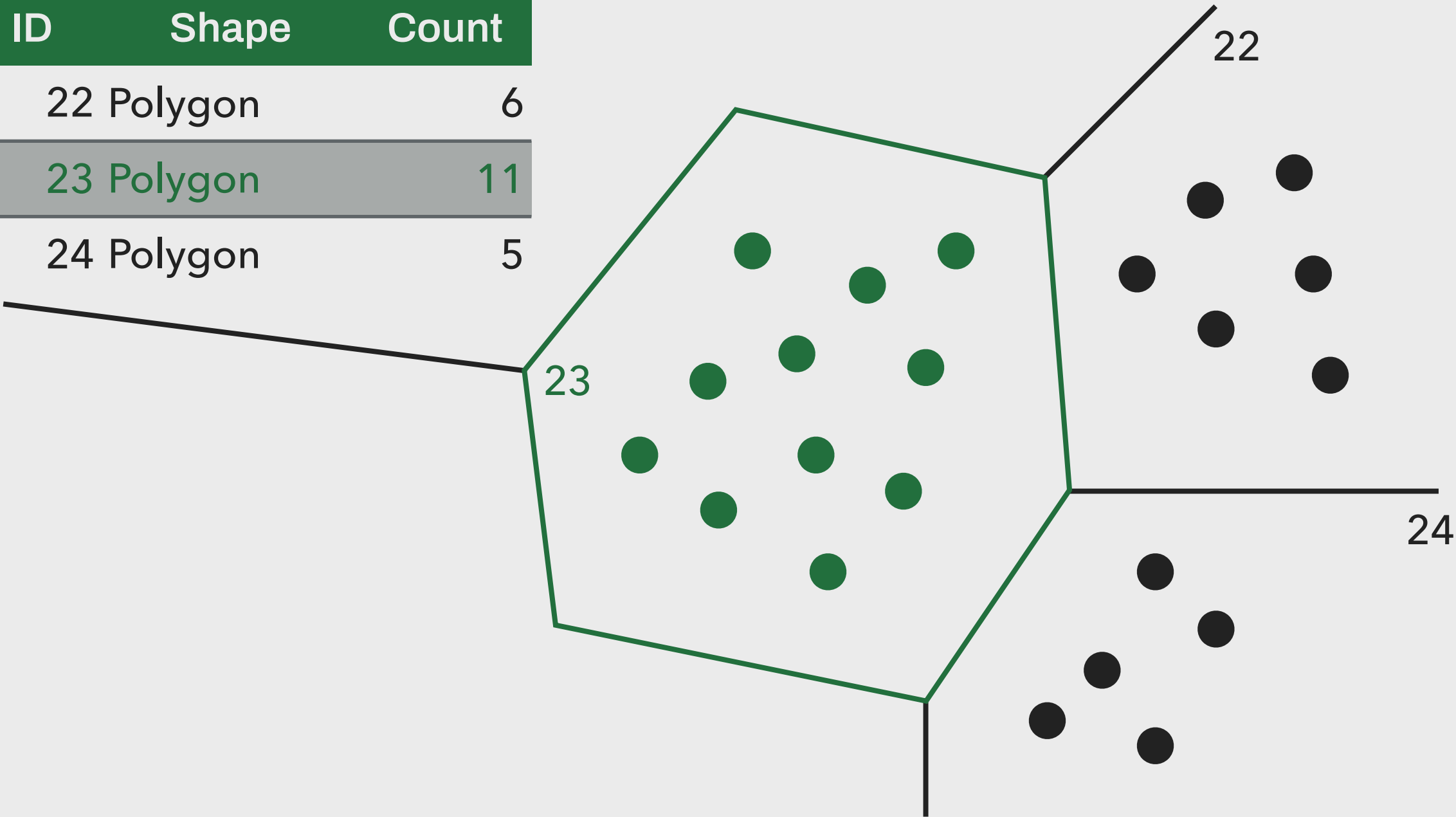
ID	Shape
22	Polygon
23	Polygon
24	Polygon



CONCEPTUALIZING POINT TO POLYGON

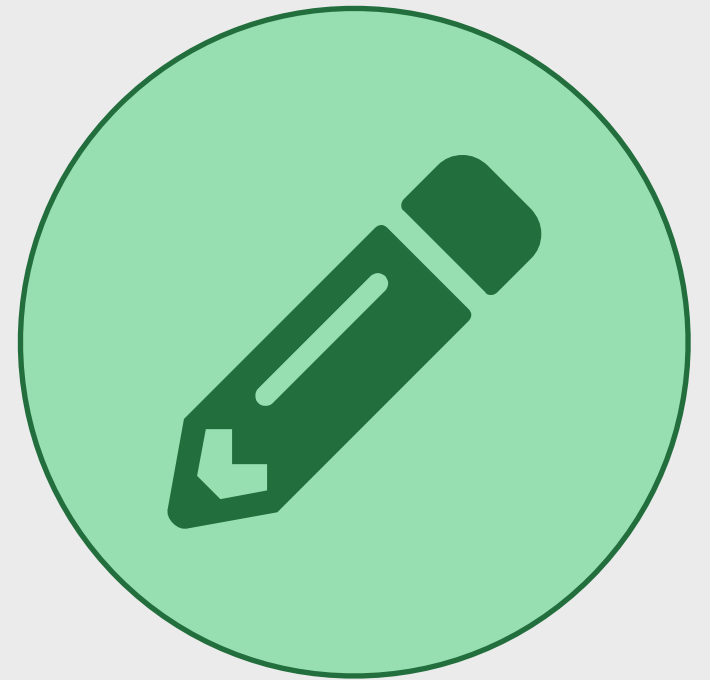
Target Layer

ID	Shape	Count
22	Polygon	6
23	Polygon	11
24	Polygon	5



KEY TERM

A **polygon to point** join will append the attributes of a polygon to each point that is proximate to it.



CONCEPTUALIZING POLYGON TO POINT

Target Layer

ID	Shape
----	-------

104 Point

105	Point
-----	-------

106 Point

105	Point
-----	-------

106 Point

107	Point
-----	-------

108 Point

109	Point
-----	-------

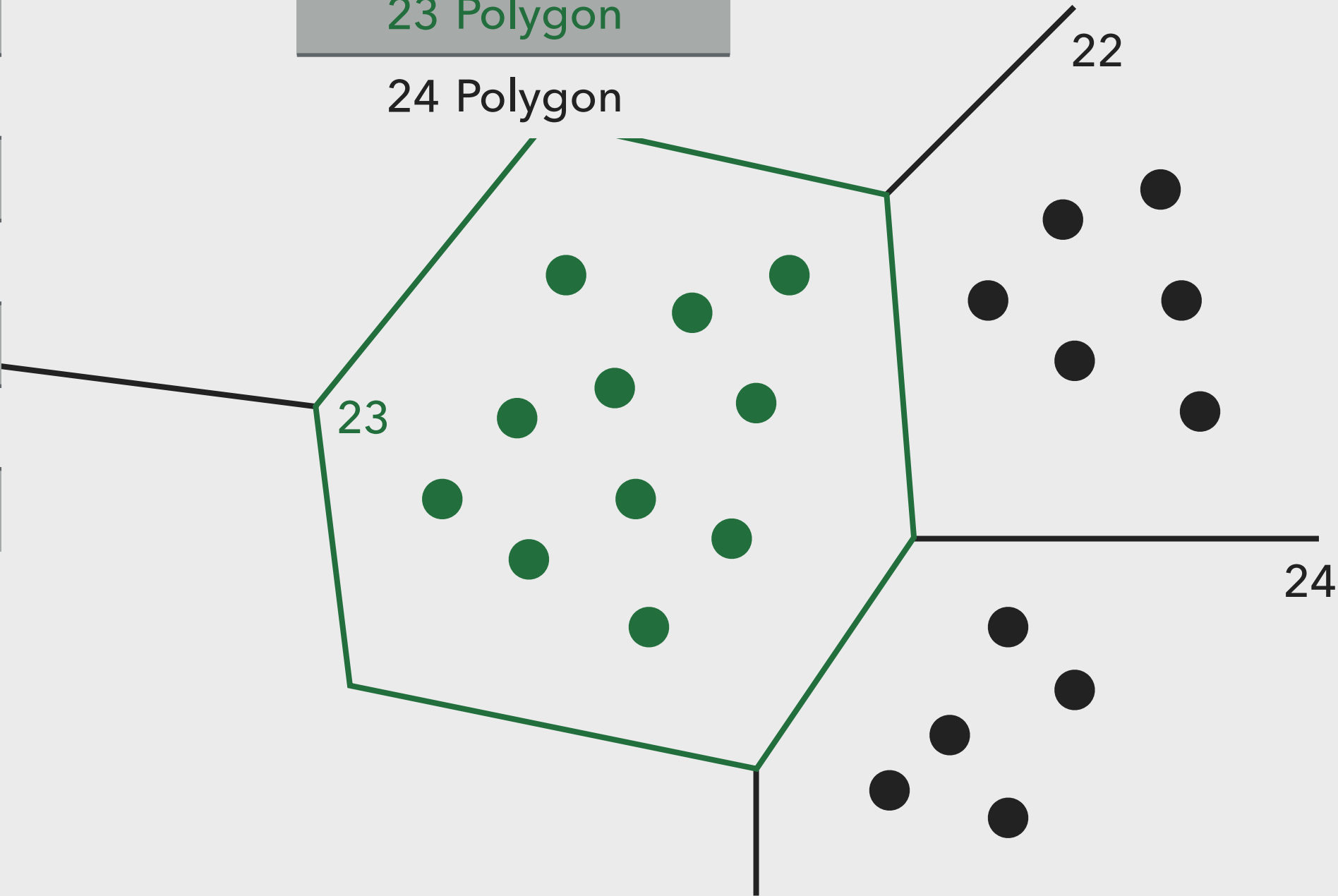
Reference Layer

ID	Shape
----	-------

22 Polygon

23	Polygon
----	---------

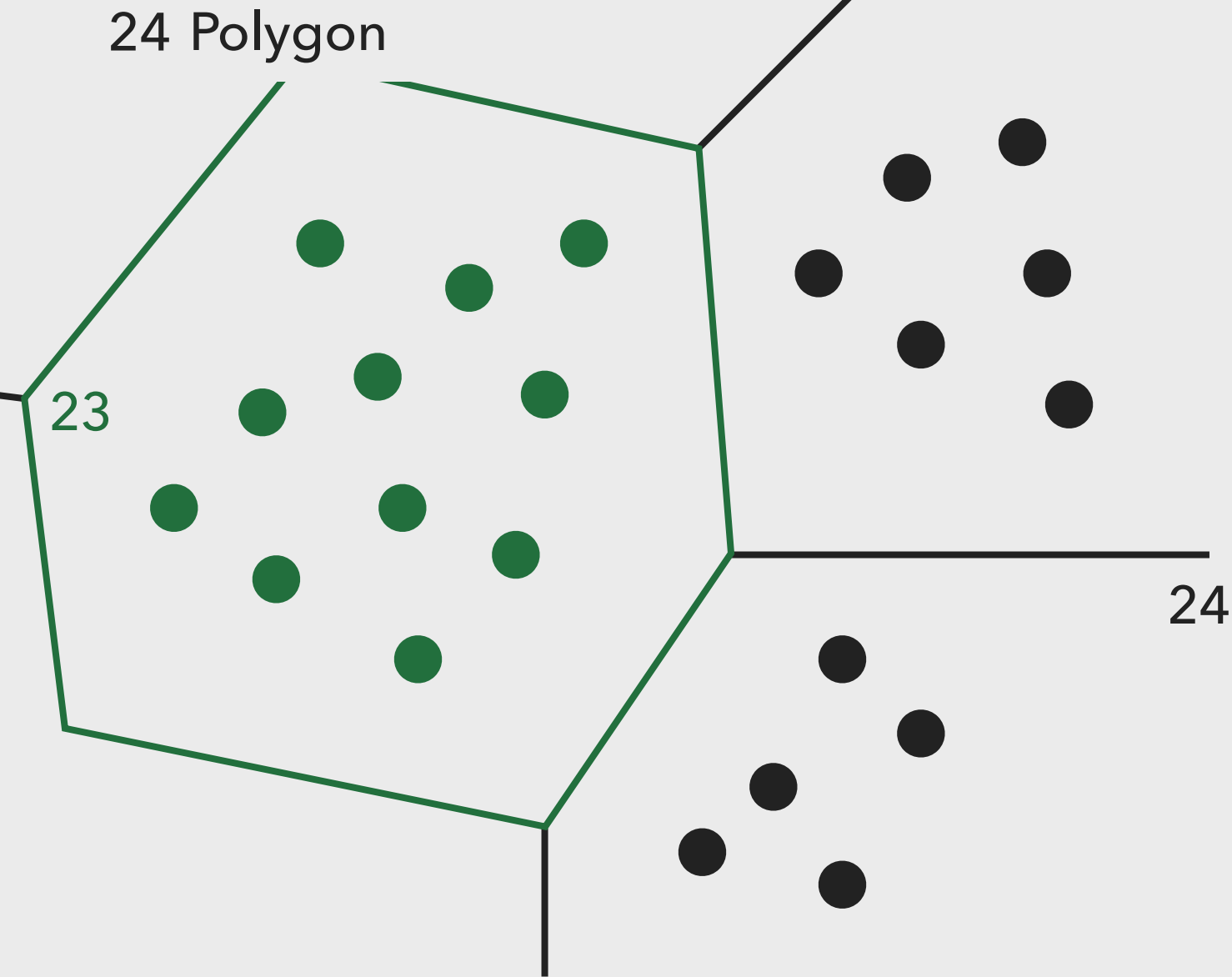
24 Polygon



CONCEPTUALIZING POLYGON TO POINT

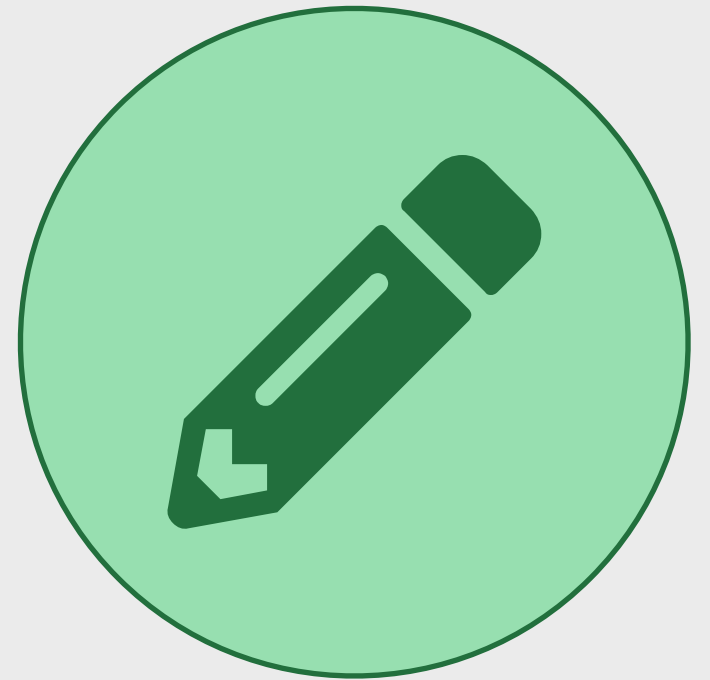
Target Layer		
ID	Shape	polyID
104	Point	22
105	Point	23
106	Point	23
105	Point	23
106	Point	24
107	Point	23
108	Point	23
109	Point	24

Reference Layer	
ID	Shape
22	Polygon
23	Polygon



KEY TERM

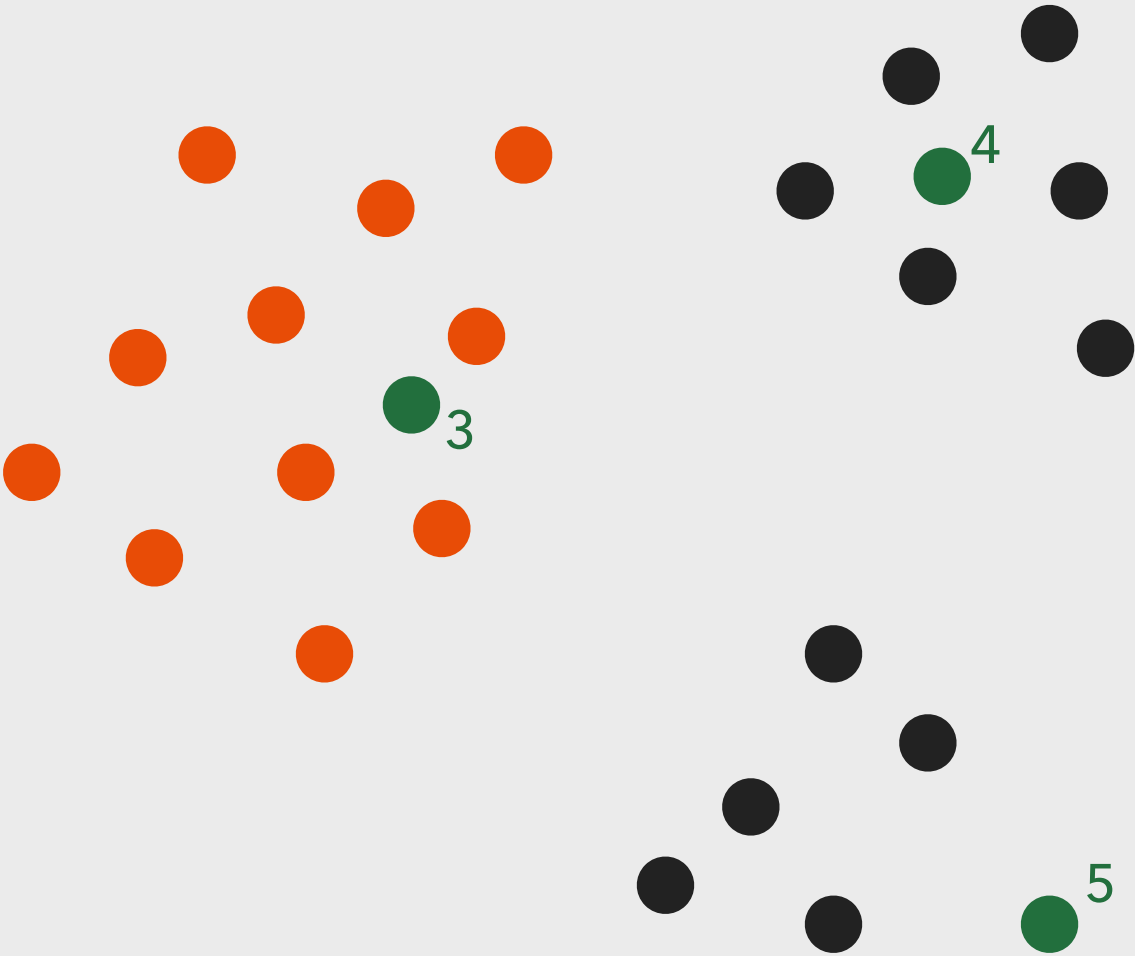
A **point to point** join will append the attributes of a point to each point that is proximate to it.



CONCEPTUALIZING POINT TO POINT

Target Layer	
ID	Shape
104	Point
105	Point
106	Point
105	Point
106	Point
107	Point
108	Point
109	Point

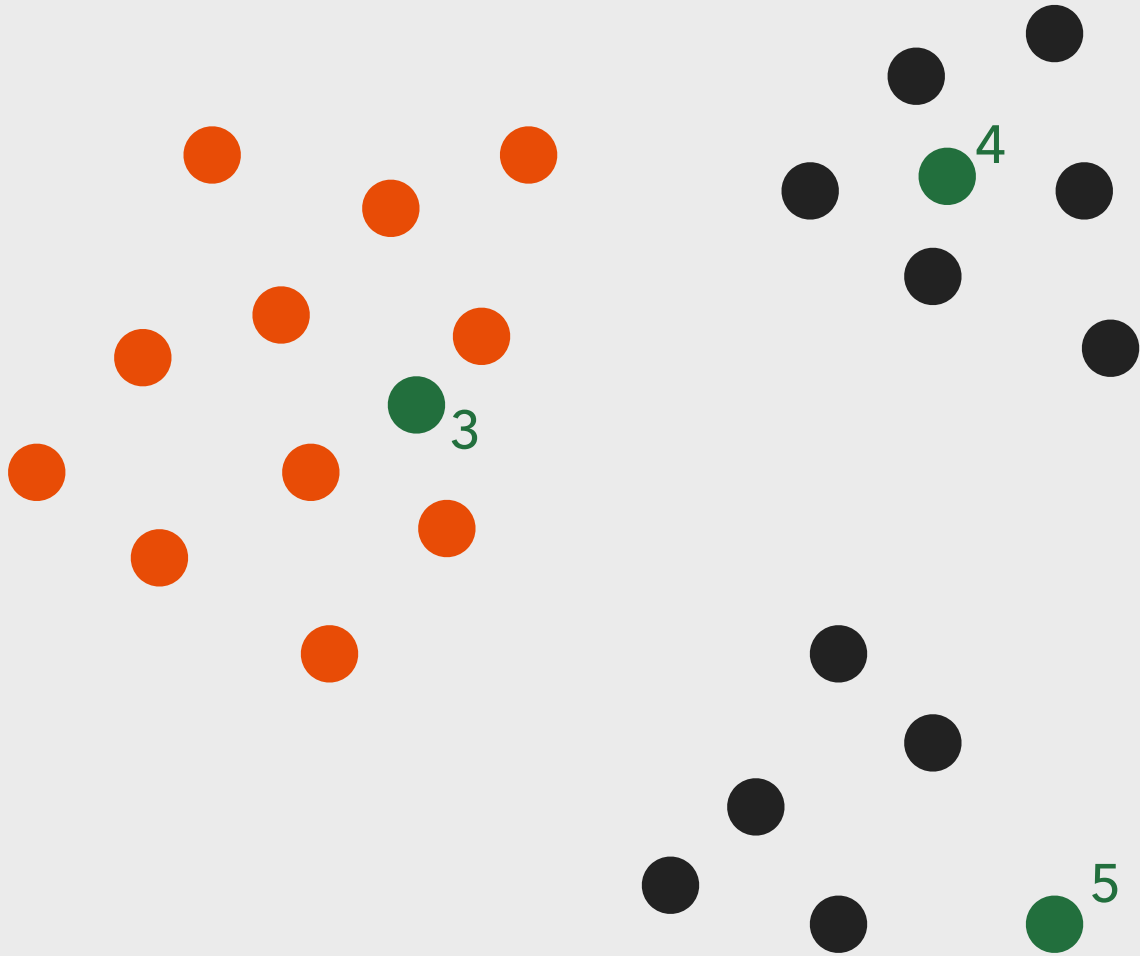
Reference Layer	
sID	Shape
3	Point
4	Point
5	Point



CONCEPTUALIZING POINT TO POINT

Target Layer		
ID	Shape	sID
104	Point	3
105	Point	3
106	Point	4
105	Point	3
106	Point	3
107	Point	3
108	Point	3
109	Point	5

Reference Layer	
sID	Shape
3	Point
4	Point
5	Point



5 BACK MATTER

AGENDA REVIEW

2. GISc & Public Policy

3. Table Joins in R

4. Spatial Joins in ArcMap

5. BACK MATTER

REMINDERS



Midterm feedback was disseminated last week - please let us know asap if there are questions or issues to be addressed.



Lab-07 and LP-08 are due by 5pm **on Wednesday.**



Lab-08, PS-05, and LP-09 are due on **next Monday.**



Extra credit - "bounty" on typos & suggestions - see Slack for details!



New lecture repo template being used!