SPATIAL MAPS CONT.

STA465: Theory and Methods for Complex Spatial Data

Instructor: Dr. Vianey Leos Barajas

QUICKLY BACK TO MAP PROJECTIONS

- We saw that there are ways to ‘map’ the Earth in 2D
- We can only preserve one to two elements (), so we always introduce biases of some sort into the maps we make.

The **True Size** of **Africa**

A small contribution in the fight against rampant *Immappancy*, by **Kai Krause**

Graphic layout for visualization only (some countries are cut and rotated)
But the conclusions are very accurate: refer to table below for exact data

COUNTRY	AREA x 1000 km^2
China	9.597
USA	9.629
India	3.287
Mexico	1.964
Peru	1.285
France	633
Spain	506
Papua New Guinea	462
Sweden	441
Japan	378
Germany	357
Norway	324
Italy	301
New Zealand	270
United Kingdom	243
Nepal	147
Bangladesh	144
Greece	132
TOTAL	**30.102**
AFRICA	**30.221**

In addition to the well known social issues of *illiteracy* and *innumeracy*, there also should be such a concept as "*immappancy*", meaning *insufficient geographical knowledge*.

A survey with random American schoolkids let them guess the population and land area of their country. Not entirely unexpected, but still rather unsettling, the majority chose *"1-2 billion"* and *"largest in the world"*, respectively.

Even with Asian and European college students, geographical estimates were often off by factors of *2-3*. This is partly due to the highly distorted nature of the predominantly used mapping projections (such as *Mercator*).

A particularly extreme example is the worldwide misjudgement of the true size of *Africa*. This single image tries to embody the massive scale, which is larger than the *USA*, *China*, *India*, *Japan* and *all of Europe....... combined!*

Top 100 Countries

Area in square kilometers, Percentage of World Total
Sources: Britannica, Wikipedia, Almanac 2010

		AREA km^2	%
1	Russia	17.098.242	11,50
2	Canada	9.984.670	6,70
3	China	9.596.961	6,40
4	United States	9.629.091	6,40
5	Brazil	8.514.877	5,70
6	Australia	7.692.024	5,20
7	India	3.287.263	2,30
8	Argentina	2.780.400	2,00
9	Kazakhstan	2.724.900	1,80
10	Sudan	2.505.813	1,70
11	Algeria	2.381.741	1,60
12	Congo	2.344.858	1,60
13	Greenland	2.166.086	1,50
14	Saudi Arabia	2.149.690	1,40
15	Mexico	1.964.375	1,30
16	Indonesia	1.860.360	1,30
17	Libya	1.759.540	1,20
18	Iran	1.628.750	1,10
19	Mongolia	1.564.100	1,10
20	Peru	1.285.216	0,86
21	Chad	1.284.000	0,86
22	Niger	1.267.000	0,85
23	Angola	1.246.700	0,85
24	Mali	1.240.192	0,83
25	South Africa	1.221.037	0,82
26	Colombia	1.141.748	0,76
27	Ethiopia	1.104.300	0,74
28	Bolivia	1.098.581	0,74
29	Mauritania	1.025.520	0,69
30	Egypt	1.002.000	0,67
31	Tanzania	945.087	0,63
32	Nigeria	923.768	0,62
33	Venezuela	912.050	0,61
34	Namibia	824.116	0,55
35	Mozambique	801.590	0,54
36	Pakistan	796.095	0,53
37	Turkey	783.562	0,53
38	Chile	756.102	0,51
39	Zambia	752.612	0,51
40	Myanmar	676.578	0,45
41	Afghanistan	652.090	0,44
42	Somalia	637.657	0,43
43	France	632.834	0,43
44	C. African Rep	622.984	0,42
45	Ukraine	603.500	0,41
46	Madagascar	587.041	0,39
47	Botswana	582.000	0,39
48	Kenya	580.367	0,39
49	Yemen	527.968	0,35
50	Thailand	513.120	0,34
51	Spain	505.992	0,34
52	Turkmenistan	488.100	0,33
53	Cameroon	475.442	0,32
54	Papua New Guinea	462.840	0,31
55	Uzbekistan	447.400	0,30
56	Morocco	446.550	0,30
57	Sweden	441.370	0,30
58	Iraq	438.317	0,29
59	Paraguay	406.752	0,27
60	Zimbabwe	390.757	0,26
61	Japan	377.930	0,25
62	Germany	357.114	0,24
63	Rep o.t. Congo	342.000	0,23
64	Finland	338.419	0,23
65	Vietnam	331.212	0,22
66	Malaysia	330.803	0,22
67	Norway	323.802	0,22
68	Côte d'Ivoire	322.463	0,22
69	Poland	312.685	0,21
70	Oman	309.500	0,21
71	Italy	301.336	0,20
72	Philippines	300.000	0,20
73	Burkina Faso	274.222	0,18
74	New Zealand	270.467	0,18
75	Gabon	267.668	0,18
76	Western Sahara	266.000	0,18
77	Ecuador	256.369	0,20
78	Guinea	245.857	0,17
79	United Kingdom	242.900	0,16
80	Uganda	241.038	0,16
81	Ghana	238.539	0,16
82	Romania	238.391	0,16
83	Laos	236.800	0,16
84	Guyana	214.969	0,14
85	Belarus	207.600	0,14
86	Kyrgyzstan	199.951	0,13
87	Senegal	196.722	0,13
88	Syria	185.180	0,12
89	Cambodia	181.035	0,12
90	Uruguay	176.215	0,12
91	Suriname	163.820	0,11
92	Tunisia	163.610	0,11
93	Nepal	147.181	0,10
94	Bangladesh	143.998	0,10
95	Tajikistan	143.100	0,10
96	Greece	131.957	0,09
97	Nicaragua	130.373	0,09
98	North Korea	120.538	0,08
99	Malawi	118.484	0,08
100	Eritrea	117.600	0,08
	TOP 100 TOTAL	**132.632.524**	**89,34**


United States


Europe


India


Japan


China

A BIG PART OF BEING A STATISTICIAN: IDENTIFYING BIASES

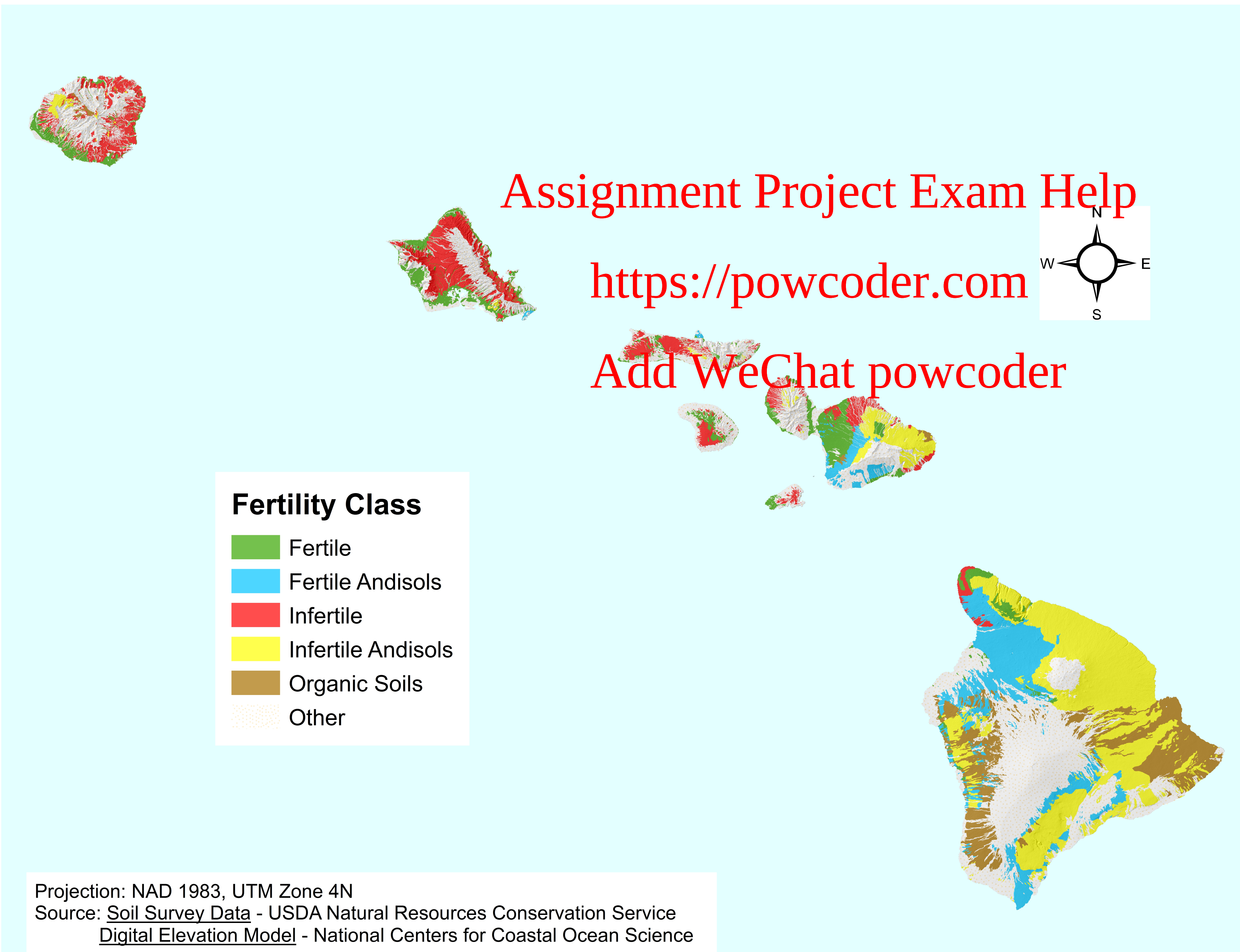
- ‘The True Size Of…”
- In mathematical statistics, you learn about biases of estimators, asymptotic behaviour, etc.

- From a non-mathematical perspective, individuals introduce biases into an analysis by the choices they make:
  - The models they choose
  - The visualizations that are made — like maps!
  - How results are presented

MANIPULATING SPATIAL DATA STRUCTURES

COMBINING DATA WITH DIFFERENT CRS/STRUCTURE — OAHU, HI, USA

Hawaii Soil Atlas: https://gis.ctahr.hawaii.edu/SoilAtlas



Downloads

- Download Info (README)
- All Properties
- Nutrient Holding Capacity
  - Image
  - Shapefile
- Fertility Class
  - Image
  - Shapefile
- Organic Matter
  - Image
  - Shapefile
- Acidity
  - Image
  - Shapefile
- Shrink Swell
  - Image
  - Shapefile
- Soil Order Series
  - Image
  - Shapefile
- Water Permeability
  - Image
  - Shapefile

SOIL FERTILITY CLASS DATA SET

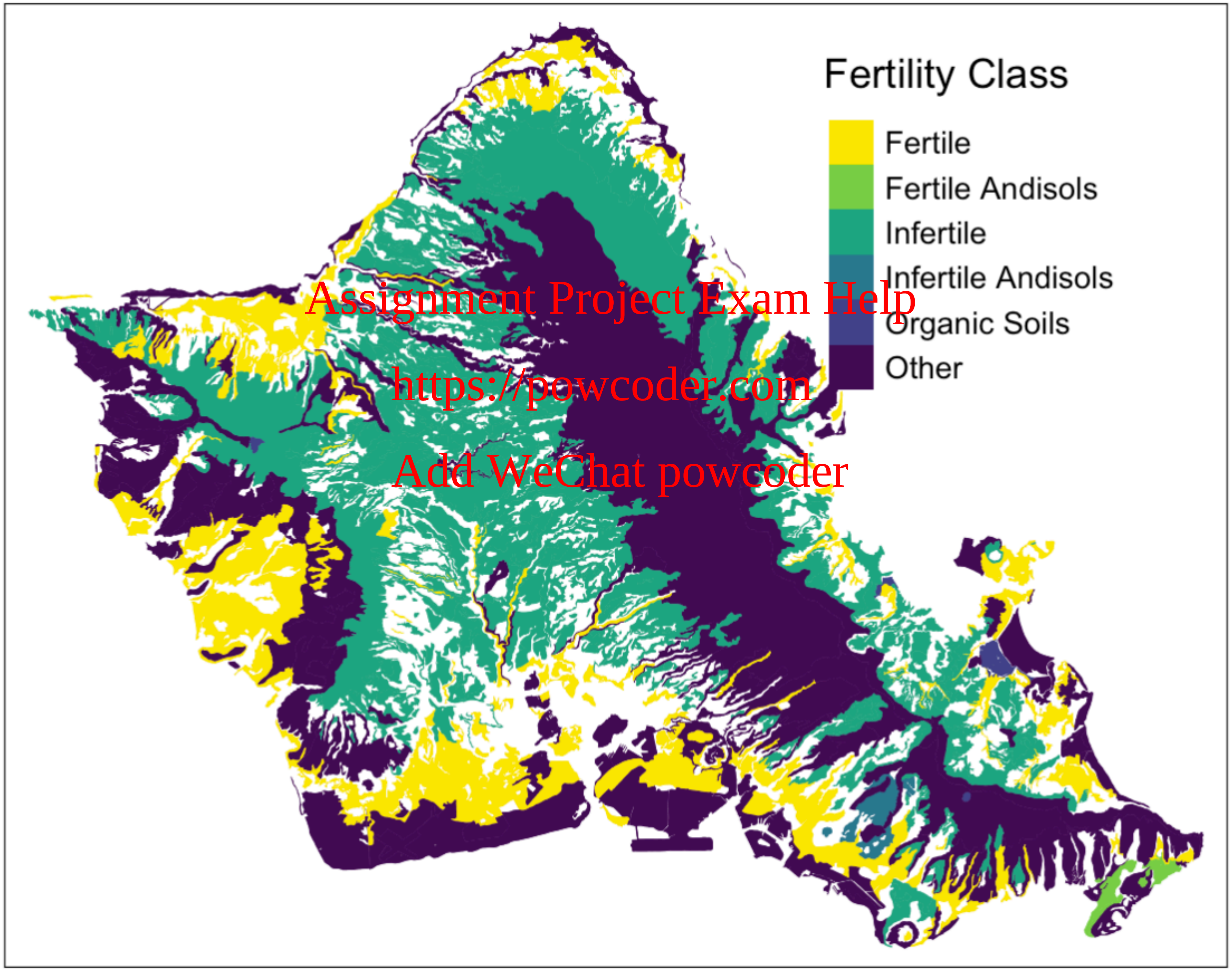
- What kind of file is it?
- Simple feature or raster?
- CRS?
- Geometry type?


```
> library(sf)
> fertility <- read_sf("FertilityClass/FertilityClass_State.shp")
```

WHAT IT LOOKS LIKE IN R

```
> fertility
Simple feature collection with 46456 features and 11 fields
geometry type:  POLYGON
dimension:      XY
bbox:           xmin: 418652.5 ymin: 2094250 xmax: 940270.3 ymax: 2458909
projected CRS:  NAD83 / UTM zone 4N
# A tibble: 46,456 x 12
   OBJECTID MUSYM MUKEY  CompKey  CompName TaxClass       SoilOrder Suborder GreatGroup Subgroup FertClass                       geometry
      <int> <chr> <chr>  <chr>    <chr>    <chr>          <chr>     <chr>    <chr>      <chr>    <chr>                      <POLYGON [m]>
 1        1 903   24242… 2424228… Hilo     Medial over h… Andisols  Udands   Hydrudands Acrudox… Infertil… ((904046.1 2199761, 904046.1 …
 2        2 346   23717… 2371729… Waikaloa Medial, amorp… Andisols  Torrands Haplotorr… Typic H… Fertile … ((828521.2 2198652, 828551.8 …
 3        3 156   23716… 2371641… Napuu    Medial-skelet… Andisols  Ustands  Haplustan… Lithic … Fertile … ((834699 2198774, 834717.6 21…
 4        4 903   24242… 2424228… Hilo     Medial over h… Andisols  Udands   Hydrudands Acrudox… Infertil… ((910356.9 2198352, 910340.4 …
 5        5 843   24215… 2421568… Puu Oo   Medial over h… Andisols  Udands   Hydrudands Acrudox… Infertil… ((883980.8 2198689, 883996.4 …
 6        6 907   24312… 2431212… Kaiwiki  Hydrous, ferr… Andisols  Udands   Hydrudands Acrudox… Infertil… ((902926.1 2199867, 902934.5 …
 7        7 156   23716… 2371641… Napuu    Medial-skelet… Andisols  Ustands  Haplustan… Lithic … Fertile … ((836739.2 2198581, 836730.1 …
 8        8 356   23717… 2371731… Akahipuu Medial-skelet… Andisols  Ustands  Haplustan… Pachic … Fertile … ((839337.9 2198406, 839407.6 …
 9        9 382   23718… 2371818… Waimea   Medial, amorp… Andisols  Ustands  Haplustan… Humic H… Fertile … ((848558.8 2198700, 848563.7 …
10       10 317   23717… 2371722… Kiholo   Clayey, isoti… Entisols  Orthents Torriorth… Lithic … Other     ((824167.1 2198050, 824140.7 …
# … with 46,446 more rows
```

MAPPING SOIL FERTILITY CLASS ACROSS OAHU

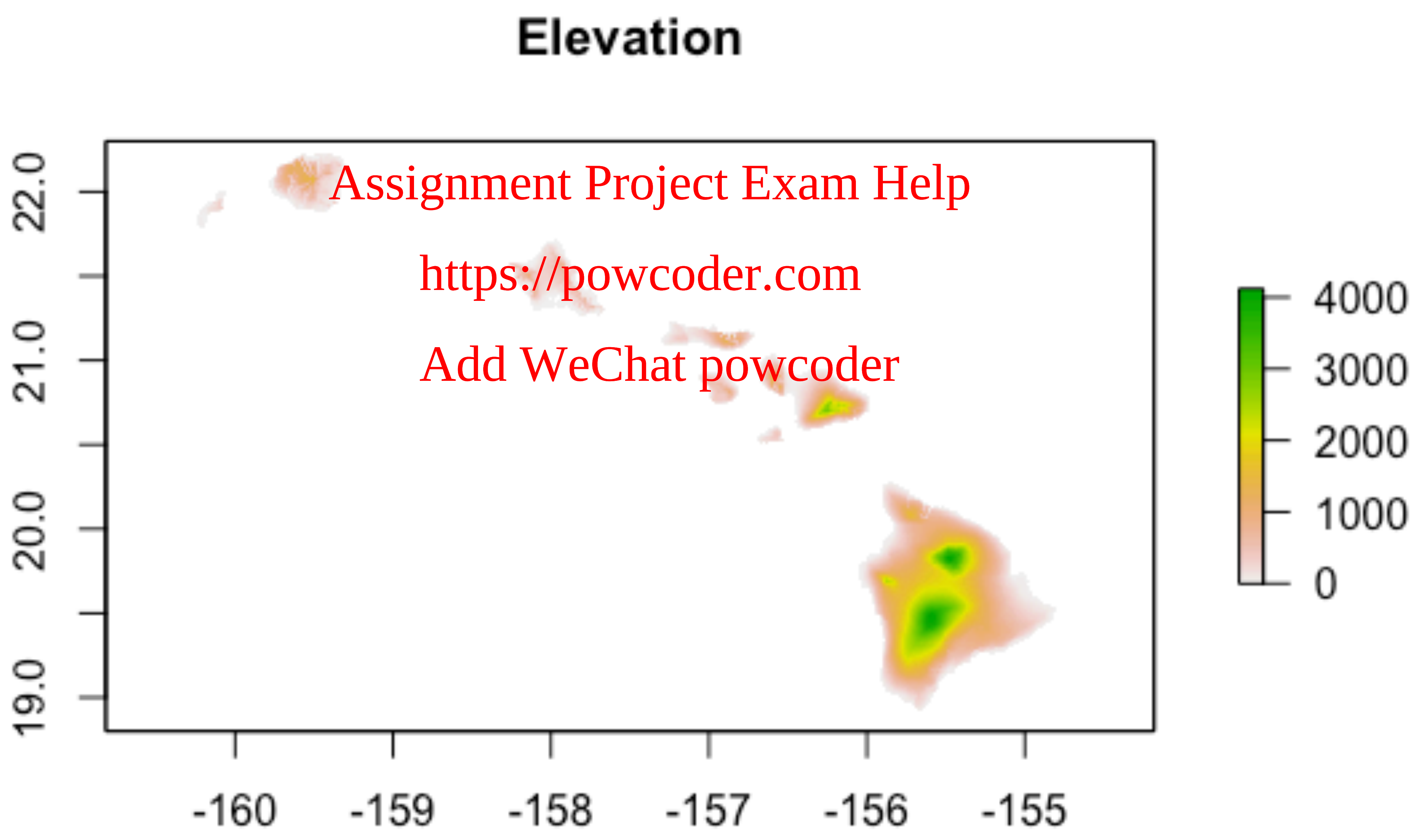


USA ELEVATION

usa_elev <- raster::getData('alt', country='USA', level=1)


```
> usa_elev[[4]]
class       : RasterLayer
dimensions  : 420, 672, 282240  (nrow, ncol, ncell)
resolution  : 0.008333333, 0.008333333  (x, y)
extent      : -160.3, -154.7, 18.8, 22.3  (xmin, xmax, ymin, ymax)
crs         : +proj=longlat +ellps=WGS84 +no_defs
source      : /Users/vianeylb/Dropbox/PostDoc/SpatialNTM/USA4_msk_alt.grd
names       : USA4_msk_alt
values      : -3, 4120  (min, max)
```

MAPPING ELEVATIONS ACROSS HAWAII



WHAT I NEEDED FOR MY WORK

- At specific points across, Oahu, I wanted to identify soil fertility class, elevation, etc.

- How do you put it all together?

- It takes a few steps that, for me, were not very intuitive!

TO START, WHAT POINTS DO I NEED INFORMATION FOR?

- I first started out with coordinates that I extracted from the USA elevation source:
- *rasterToPoints(usa_elev[[4]])*

```
              x        y USA4_msk_alt
 [1,] -157.9792 21.71250            3
 [2,] -157.9708 21.71250            3
 [3,] -158.0042 21.70417            2
 [4,] -157.9958 21.70417            7
 [5,] -157.9875 21.70417            6
 [6,] -157.9792 21.70417            9
 [7,] -157.9708 21.70417            5
 [8,] -157.9625 21.70417            4
 [9,] -158.0208 21.69583            2
[10,] -158.0125 21.69583            6
```


- This is a matrix, with two columns x and y for longitude and latitude — not a spatial object, yet.

THEN WHAT?

- dsp <- sf::st_as_sf(data.frame(oahu), coords=c("x", "y"), crs=4326)
- dsp <- sf::st_transform(dsp, crs=st_crs(fertility))
- oahufertility <- sf::st_join(x = dsp, y=fertility[dsp,c("FertClass")], join=st_intersects)[1+2*(0:1912),]

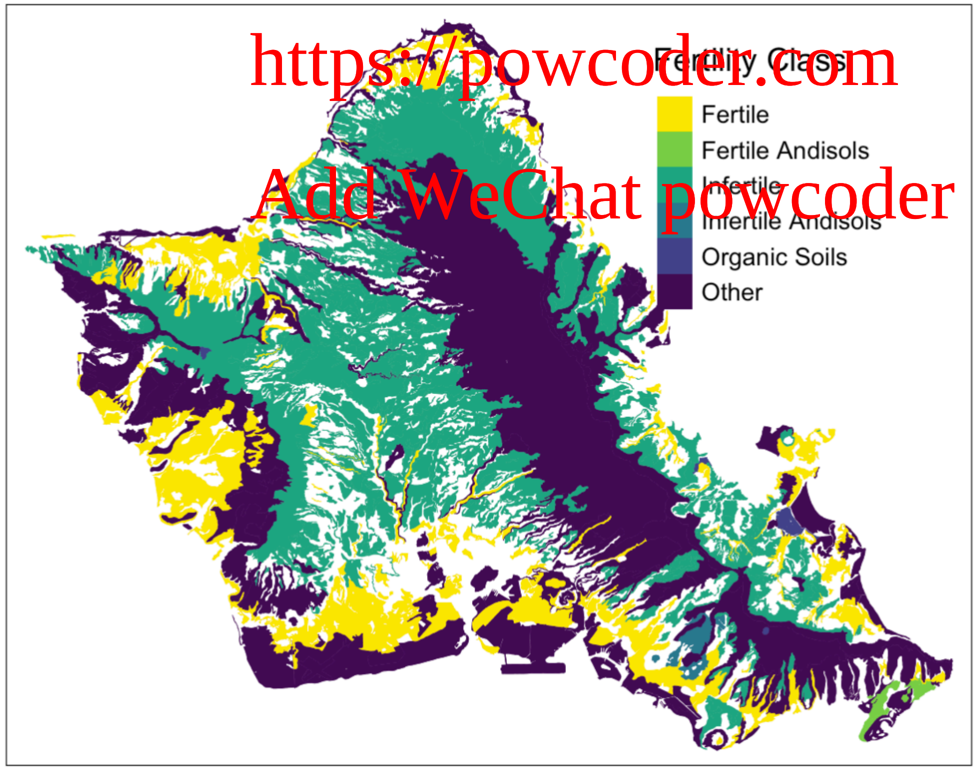
- st_transform(); st_as_sf(); st_intersects()

FINAL DATA SET

```
Simple feature collection with 1913 features and 2 fields
geometry type:  POINT
dimension:      XY
bbox:           xmin: 575487.3 ymin: 2354411 xmax: 638738.9 ymax: 2401354
projected CRS:  NAD83 / UTM zone 4N
First 10 features:
     USA4_msk_alt FertClass                 geometry
1               3      <NA>   POINT (605587 2401354)
3               2      <NA> POINT (603006.9 2400414)
4.1             7     Other   POINT (603869 2400420)
5.1             6     Other   POINT (604731 2400426)
6.1             9     Other POINT (605593.1 2400431)
7.1             5     Other POINT (606455.1 2400437)
9               2      <NA> POINT (601288.7 2399481)
10.1            6     Other POINT (602150.8 2399486)
11.1            9   Fertile POINT (603012.9 2399492)
12.1           10   Fertile   POINT (603875 2399498)
```

TO MAKE THE MAP OF FERTILITY CLASS:

```
tm_shape(fertility[oahufertility,]) +
        tm_fill("FertClass", pal="-viridis", title="Fertility Class")
```

CLIPPING, CROPPING AND MORE

WHAT DOES THE DATA LOOK LIKE WHEN COMBINED?

```
Simple feature collection with 1913 features and 2 fields
geometry type:  POINT
dimension:      XY
bbox:           xmin: 575487.3 ymin: 2354411 xmax: 638738.9 ymax: 2401354
projected CRS:  NAD83 / UTM zone 4N
First 10 features:
     USA4_msk_alt FertClass                 geometry
1               3      <NA>   POINT (605587 2401354)
3               2      <NA> POINT (603006.9 2400414)
4.1             7     Other   POINT (603869 2400420)
5.1             6     Other   POINT (604731 2400426)
6.1             9     Other POINT (605593.1 2400431)
7.1             5     Other POINT (606455.1 2400437)
9               2      <NA> POINT (601288.7 2399481)
10.1            6     Other POINT (602150.8 2399486)
11.1            9   Fertile POINT (603012.9 2399492)
12.1           10   Fertile   POINT (603875 2399498)
```

CLIPPING AND CROPPING (+ MASKING)

- **Spatial clipping:** spatial subsetting that changes the geometry columns of at least some of the affected features

- **Raster cropping + masking:**

  - Often a raster may be bigger than the area of study, cropping is one way to reduce the dimension.
  - Masking sets values outside of a boundary to “NA”

CLIPPING TYPES

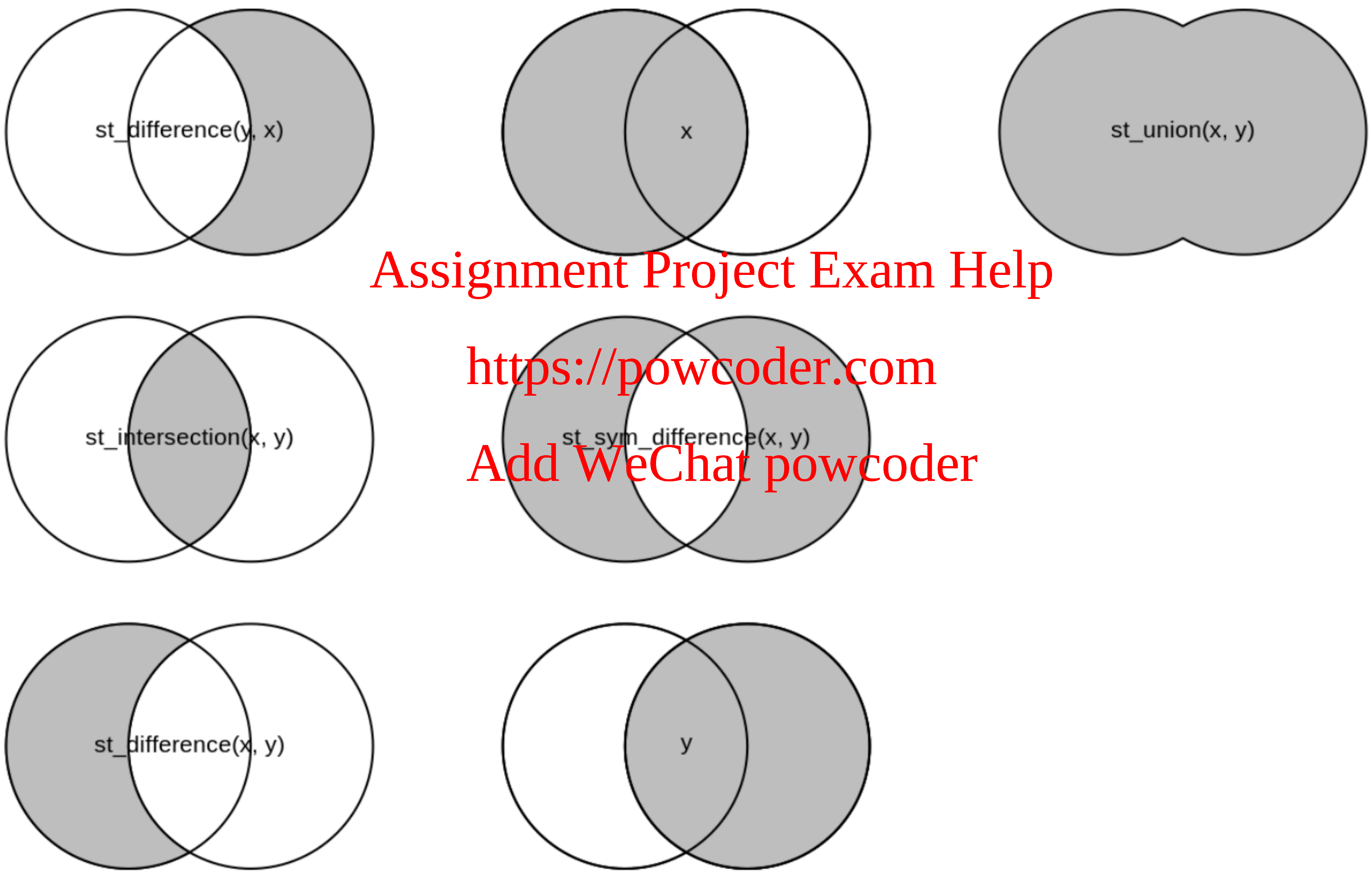


FIGURE 5.8: Spatial equivalents of logical operators.

CLIPPING EXAMPLE

```
b = st_sfc(st_point(c(0, 1)), st_point(c(1, 1))) # create 2 points
b = st_buffer(b, dist = 1) # convert points to circles
plot(b)
text(x = c(-0.5, 1.5), y = 1, labels = c("x", "y")) # add text
```

```
x = b[1]
y = b[2]
x_and_y = st_intersection(x, y)
plot(b)
plot(x_and_y, col = "lightgrey", add = TRUE) # color intersecting area
```

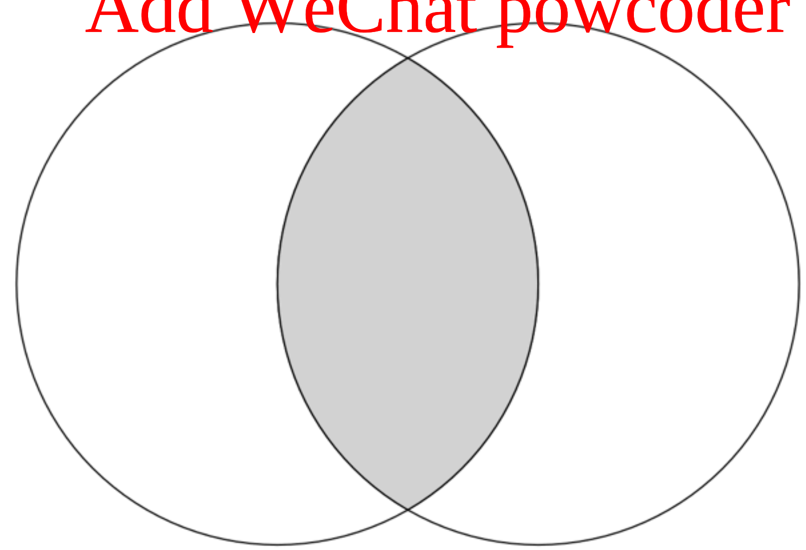



FIGURE 5.7: Overlapping circles with a gray color indicating intersection between them.

CROPPING

```
srtm = raster(system.file("raster/srtm.tif", package = "spDataLarge"))
zion = st_read(system.file("vector/zion.gpkg", package = "spDataLarge"))
zion = st_transform(zion, projection(srtm))
```

```
> zion = st_read(system.file("vector/zion.gpkg", package = "spDataLarge"))
Reading layer `zion' from data source `/Library/Frameworks/R.framework/Ver
sing driver `GPKG'
Simple feature collection with 1 feature and 11 fields
geometry type:  POLYGON
dimension:      XY
bbox:           xmin: 302903.1 ymin: 4112244 xmax: 334735.5 ymax: 4153087
projected CRS:  UTM Zone 12, Northern Hemisphere
```



```
srtm_cropped = crop(srtm, zion)
```

```
> zion
Simple feature collection with 1 feature and 11 fields
geometry type:  POLYGON
dimension:      XY
bbox:           xmin: -113.2283 ymin: 37.14135 xmax: -1
CRS:            +proj=longlat +datum=WGS84 +no_defs
  UNIT_CODE
1      ZION Lands - http://landsnet.nps.gov/tractsnet/d
  STATE REGION GNIS_ID     UNIT_TYPE CREATED_BY
1    UT     IM 1455157 National Park      Lands https:/
  PARKNAME                              geom
1     Zion POLYGON ((-113.0844 37.1713...
```

```
srtm_masked = mask(srtm, zion)
```

```
srtm_inv_masked = mask(srtm, zion, inverse = TRUE)
```

CROPPING

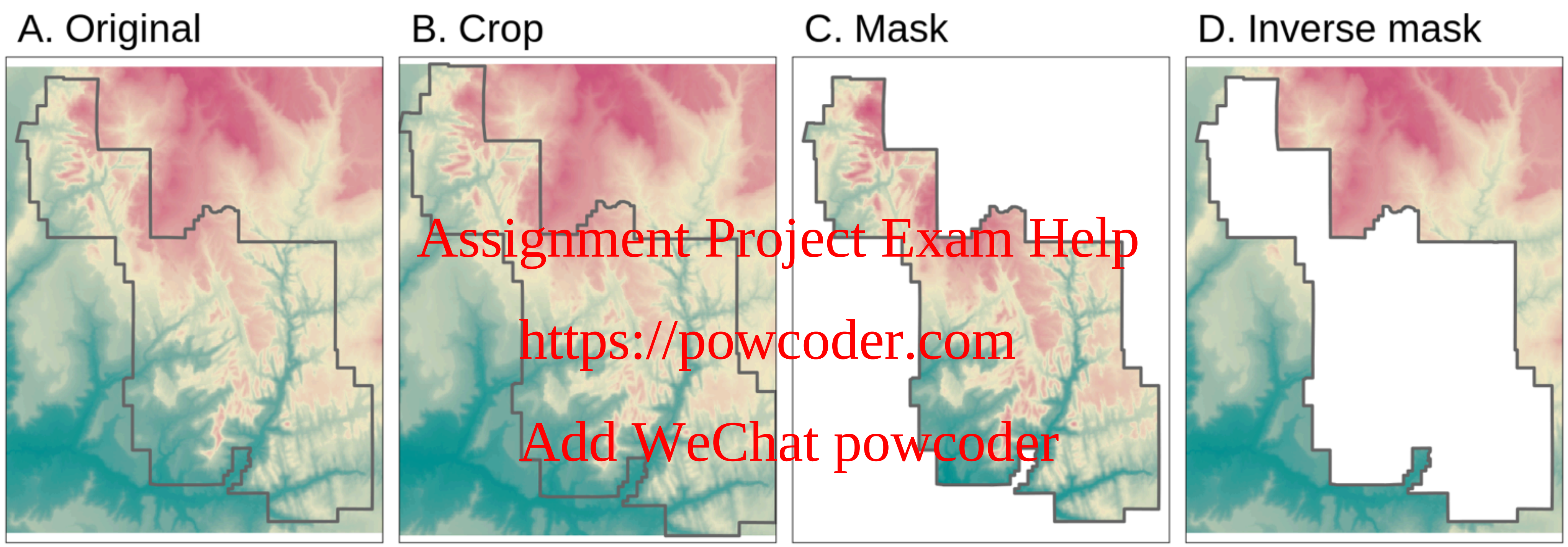


FIGURE 5.17: Illustration of raster cropping and raster masking.

GEOMETRY UNIONS: US_STATES DATA SET — SPDATA R PACKAGE

```
> us_states
Simple feature collection with 49 features and 6 fields
geometry type:  MULTIPOLYGON
dimension:      XY
bbox:           xmin: -124.7042 ymin: 24.55868 xmax: -66.9824 ymax: 49.38436
geographic CRS: NAD83
First 10 features:
   GEOID        NAME   REGION             AREA total_pop_10 total_pop_15                       geometry
1     01     Alabama    South 133709.27 [km^2]      4712651      4830620 MULTIPOLYGON (((-88.20006 3...
2     04     Arizona     West 295281.25 [km^2]      6246816      6641928 MULTIPOLYGON (((-114.7196 3...
3     08    Colorado     West 269573.06 [km^2]      4887061      5278906 MULTIPOLYGON (((-109.0501 4...
4     09 Connecticut Norteast  12976.59 [km^2]      3545837      3593222 MULTIPOLYGON (((-73.48731 4...
5     12     Florida    South 151052.01 [km^2]     18511620     19645772 MULTIPOLYGON (((-81.81169 2...
6     13     Georgia    South 152725.21 [km^2]      9468815     10006693 MULTIPOLYGON (((-85.60516 3...
7     16       Idaho     West 216512.66 [km^2]      1526797      1616547 MULTIPOLYGON (((-116.916 45...
8     18     Indiana  Midwest  93648.40 [km^2]      6417398      6568645 MULTIPOLYGON (((-87.52404 4...
9     20      Kansas  Midwest 213037.08 [km^2]      2809329      2892987 MULTIPOLYGON (((-102.0517 4...
10    22   Louisiana    South 122345.76 [km^2]      4429940      4625253 MULTIPOLYGON (((-92.01783 2...
```

GEOMETRY UNIONS

```
regions = aggregate(x = us_states[, "total_pop_15"], by = list(us_states$REGION),
                    FUN = sum, na.rm = TRUE)
regions2 = us_states %>% group_by(REGION) %>%
  summarize(pop = sum(total_pop_15, na.rm = TRUE))
```

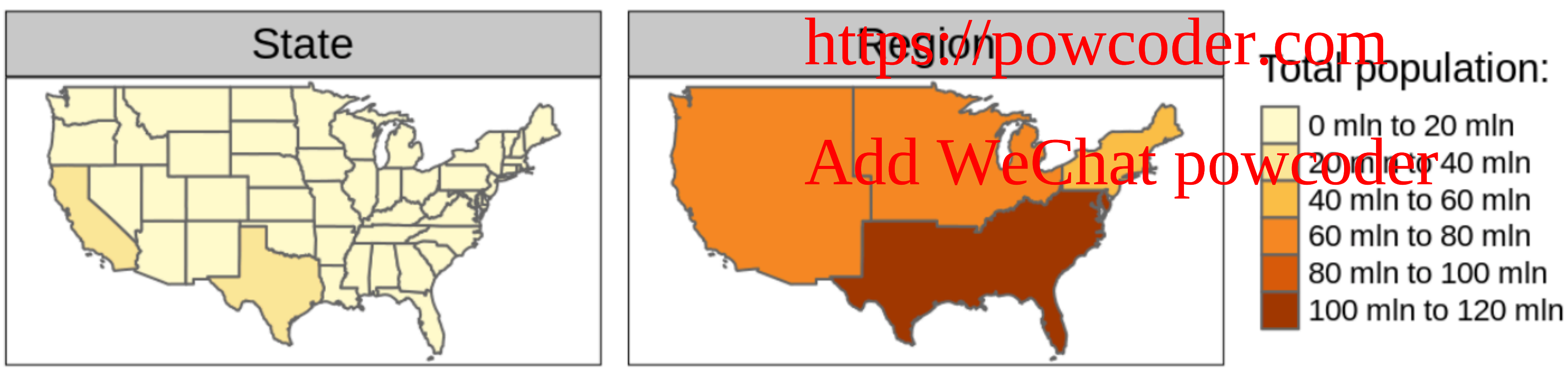



FIGURE 5.10: Spatial aggregation on contiguous polygons, illustrated by aggregating the population of US states into regions, with population represented by color. Note the operation automatically dissolves boundaries between states.

RASTERIZATION AND VECTORIZATION

- Sometimes you will want/need to switch between rasters and vector objects
- In R, it's possible to change between types to suit your needs

- **Rasterization:** conversion of vector objects into their representation in raster objects
- **Vectorization**: converts spatially continuous raster data into spatially discrete data such as points, lines or polygons

RASTERIZATION: 5.4.3 IN GEOCOMPUTATION WITH R

- In the ‘raster’ R package, we have the function: ‘rasterize()’

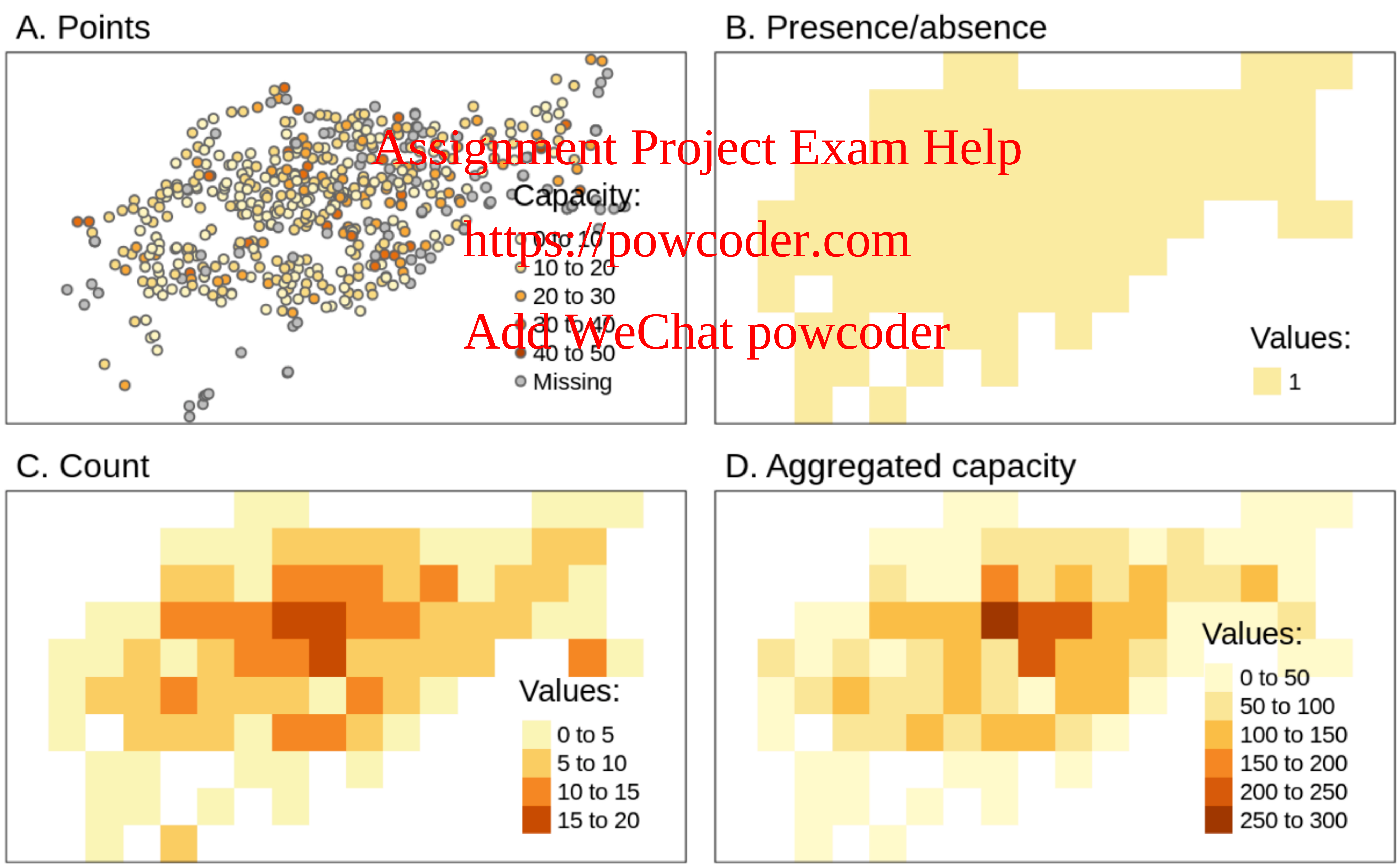


FIGURE 5.21: Examples of point rasterization.

IN THE PREVIOUS EXAMPLE, TO RASTERIZE….

- There were a few steps to making a raster.

- First, we need to make a 'raster template' - it contains no values


```
> raster_template
class       : RasterLayer
dimensions  : 10, 16, 160  (nrow, ncol, ncell)
resolution  : 1000, 1000  (x, y)
extent      : 523038.6, 539038.6, 174971.4, 184971.4  (xmin, xmax, ymin, ymax)
crs         : +proj=tmerc +lat_0=49 +lon_0=-2 +k=0.9996012717 +x_0=400000 +y_0=-100000 +ellps=airy +units=m +no_defs

> plot(raster_template)
Error in .plotraster2(x, col = col, maxpixels = maxpixels, add = add,  :
  no values associated with this RasterLayer
```

- Then, we can combine the points and raster template to create the new rasterized objects!

RASTERIZATION: 5.4.3 IN GEOCOMPUTATION WITH R

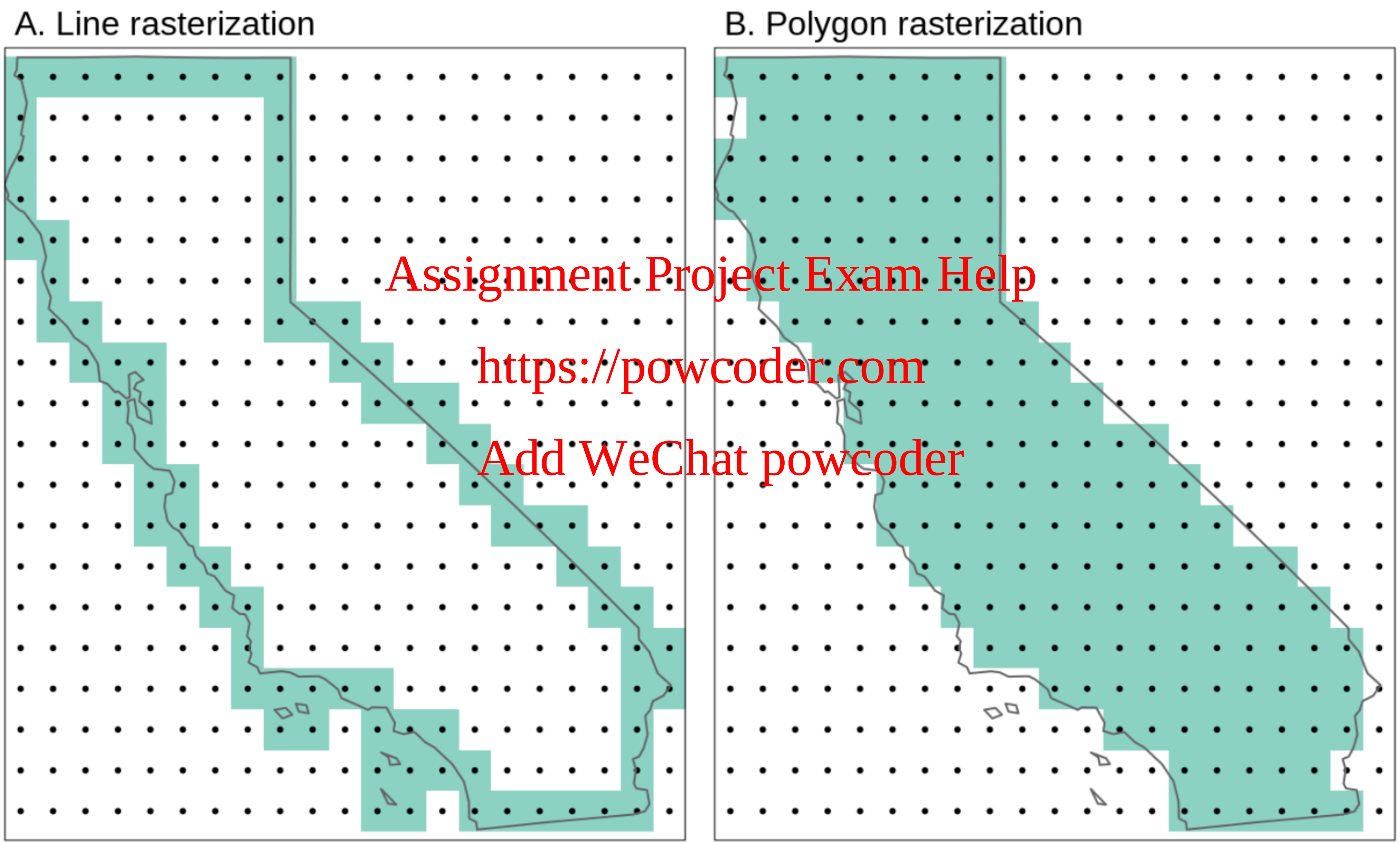


FIGURE 5.22: Examples of line and polygon rasterizations.

VECTORIZATION (OF RASTER OBJECTS)

```
elev_point = rasterToPoints(elev, spatial = TRUE) %>%
  st_as_sf()
```

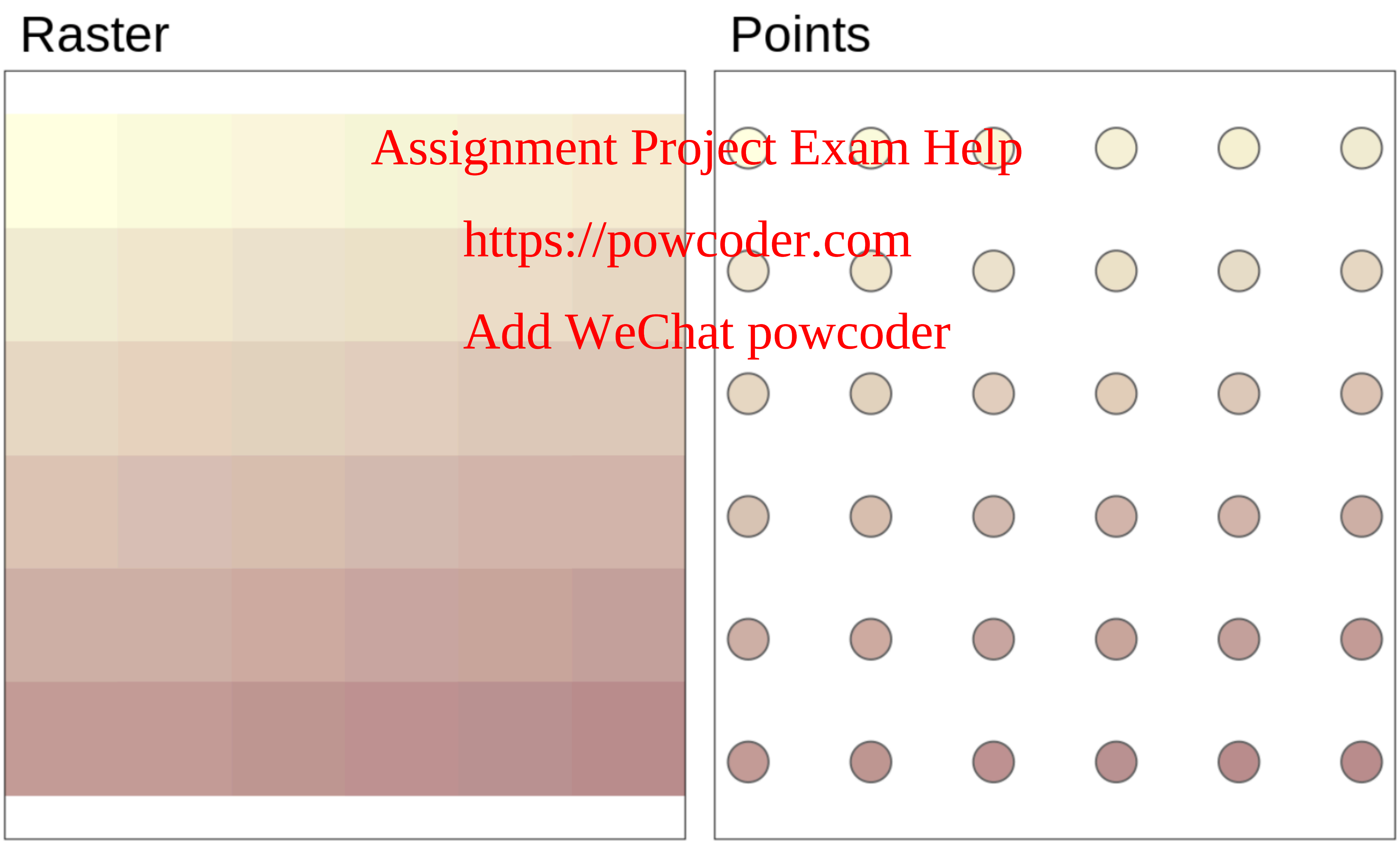


FIGURE 5.23: Raster and point representation of the elev object.

VECTORIZATION — TO POLYGONS!

```r
grain_poly = rasterToPolygons(grain) %>%
  st_as_sf()
grain_poly2 = grain_poly %>%
  group_by(layer) %>%
  summarize()
#> although coordinates are longitude/latitude, st_union assumes that they are planar
#> although coordinates are longitude/latitude, st_union assumes that they are planar
#> although coordinates are longitude/latitude, st_union assumes that they are planar
```

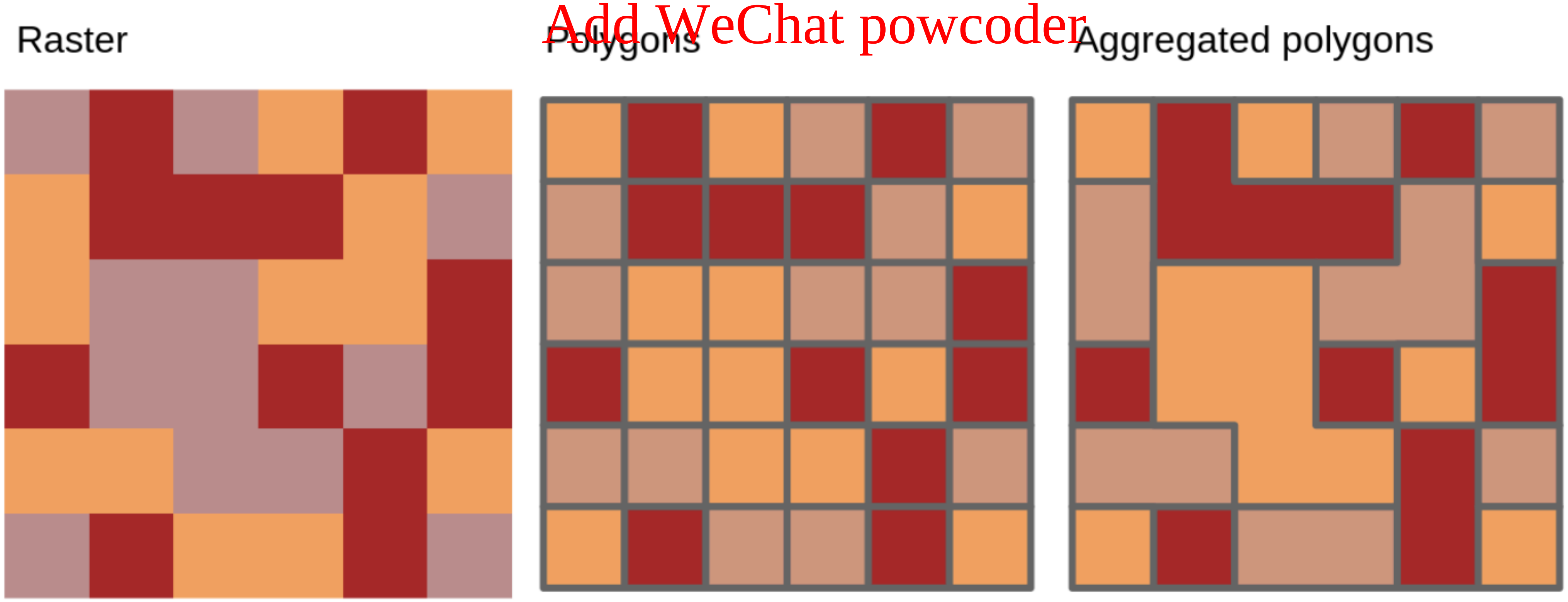



FIGURE 5.25: Illustration of vectorization of raster (left) into polygon (center) and polygon aggregation (right)

RASTER TO….POINTS, POLYGONS, AND CONTOURS

- In the `raster' R package, there exist a few more options:

  - rasterToPoints()
  - rasterToPolygons()
  - rasterToContour()

MAPS + INTERACTIVE VISUALIZATIONS

UP NEXT

INTERACTIVE MAPS

- There are lots of ways to make maps interactive in R
- Interactivity allows for greater exploration of the data
- We can work with different R packages to make maps interactive in different ways:

    - Shiny
    - leaflet
    - tmap
    - mapview