Geospatial data in R

11.26.2019

1. Assignment 6 will be released today before the midnight.

2. If you cannot bring a laptop to the Final Exam, please go to the link below and write your name there.
   https://forms.gle/oxJTfXAekDd9ct5J6

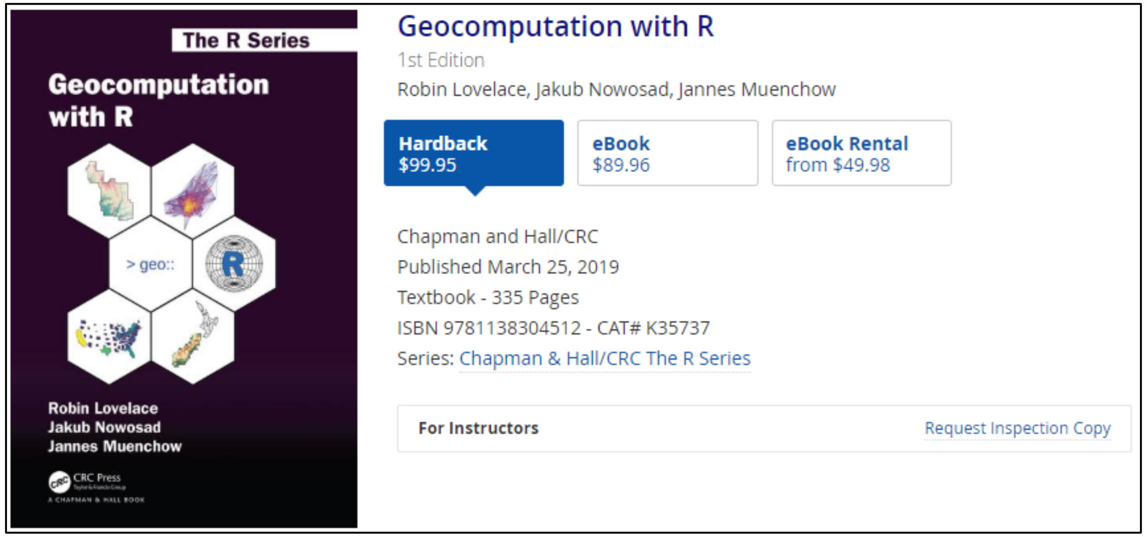
Purpose of this lecture:

- To introduce you to things that R can do with geographic data
- So that you will know where to start when you want to delve deeper into it

This lecture is NOT aimed at:

- Making you to abandon ArcMap and completely migrate to R (they are good at different things, so they are complementary)
- Making you a master in handling geographic data in R; it is an introductory session.

*Raster data is not covered in this lecture



Geocomputation with R

1st Edition

Robin Lovelace, Jakub Nowosad, Jannes Muenchow

Chapman and Hall/CRC
Published March 25, 2019
Textbook - 335 Pages
ISBN 9781138304512 - CAT# K35737
Series: Chapman & Hall/CRC The R Series

Online version of the textbook is free!

https://geocompr.robinlovelace.net/

```
install.packages(‘sf’, dependencies = T)
install.packages(‘tmap’, dependencies = T)
```

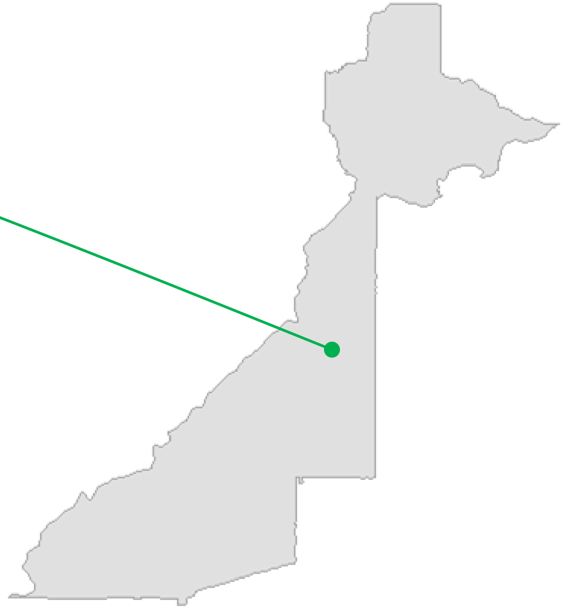
If you are asked to type in y/n,
type in y

data.frame & shapefile

data.frame

Name	GDP	Population
Fulton County	000,000,000	000,000,000

Geometry

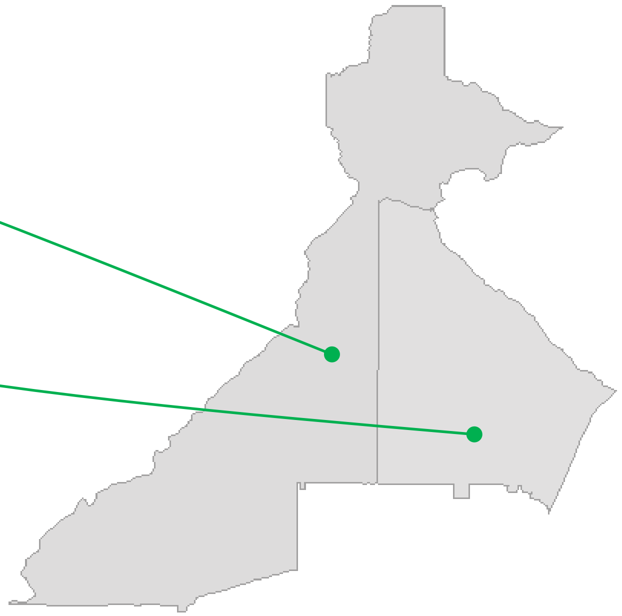


data.frame & shapefile

data.frame

Name	GDP	Population
Fulton County	000,000,000	000,000,000
DeKalb County	00,000,000	00,000,000

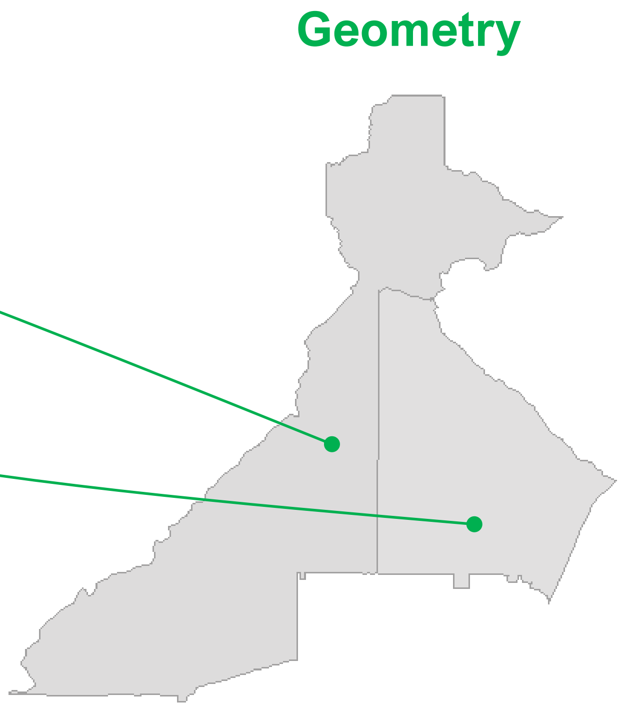
Geometry



data.frame & shapefile

data.frame

Name	GDP	Population
Fulton County	000,000,000	000,000,000
DeKalb County	00,000,000	00,000,000



1. How do we represent geometry in computers?

2. How do we combine data.frame and geometry?

Representing Geometry in Computers

Simple Features (SF)

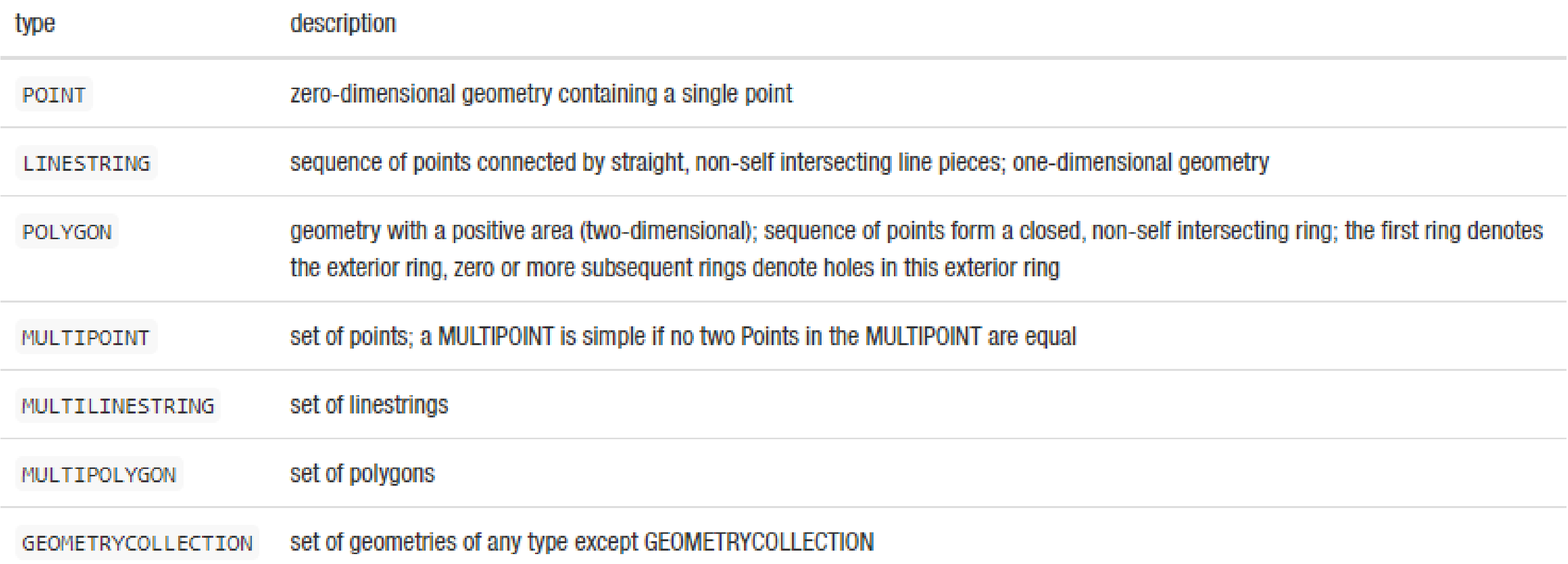
“Simple features or *simple feature access* refers to a formal standard (ISO 19125-1:2004) that describes **how objects in the real world can be represented in computers**, with emphasis on the *spatial* geometry of these objects” (sf vignette, https://r-spatial.github.io/sf/articles/sf1.html).

In other words, people made a standardized way to represent geometries in computers.

Simple Feature Geometry

Examples of geometry information

Point: X & Y coordinate
Line: 2 pairs of XY coordinate
Polygon: At least 3 pairs of XY coordinate + the order



type	description
`POINT`	zero-dimensional geometry containing a single point
`LINESTRING`	sequence of points connected by straight, non-self intersecting line pieces; one-dimensional geometry
`POLYGON`	geometry with a positive area (two-dimensional); sequence of points form a closed, non-self intersecting ring; the first ring denotes the exterior ring, zero or more subsequent rings denote holes in this exterior ring
`MULTIPOINT`	set of points; a MULTIPOINT is simple if no two Points in the MULTIPOINT are equal
`MULTILINESTRING`	set of linestrings
`MULTIPOLYGON`	set of polygons
`GEOMETRYCOLLECTION`	set of geometries of any type except GEOMETRYCOLLECTION

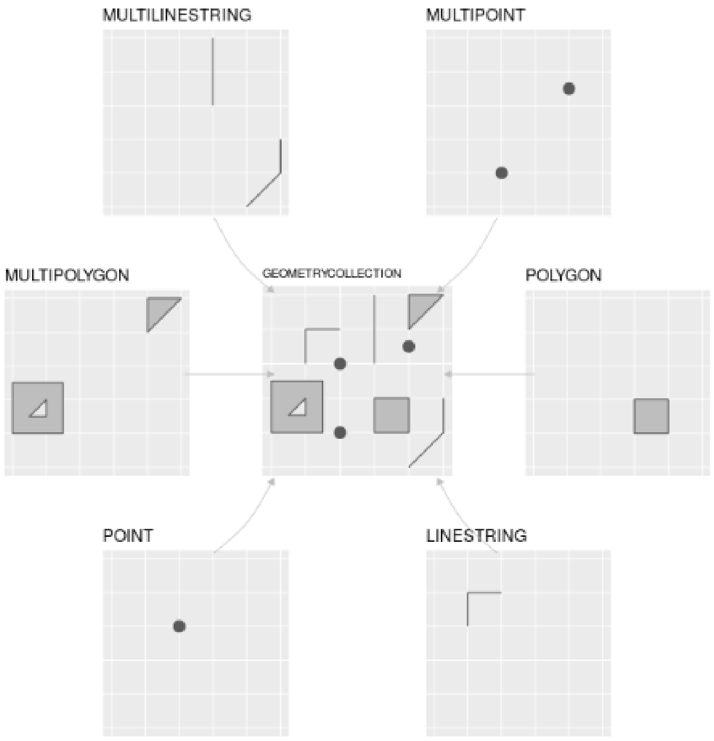


FIGURE 2.2: Simple feature types fully supported by sf.

Source: sf vignette, https://r-spatial.github.io/sf/articles/sf1.html).

Source: Geocomputation with R (https://geocompr.robinlovelace.net/)

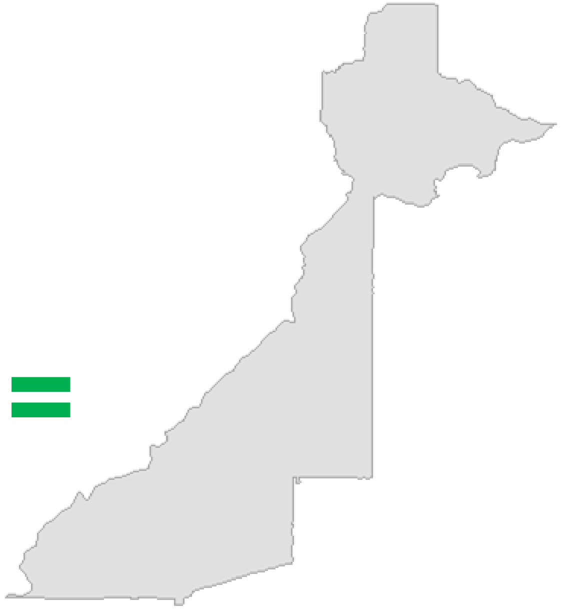
Simple Feature Geometry

Simple feature

MULTIPOLYGON (((-82.31535 31.94637, -82.31262 31.94746, -82.31068 31.9473, -82.30861 31.94555, -82.30654 31.94161, -82.30602 31.93869, -82.30452 31.93792, -82.3028 31.9339, -82.30111 31.93327, -82.29809 31.93399, -82.29533 31.93886, -82.29334 31.94084, -82.29137 31.94248, -82.28642 31.94244, -82.28327 31.94076, -82.27955 31.93798, -82.27762 31.93707, -82.27033 31.93587, -82.2682 31.93279, -82.26716 31.92862, -82.26316 31.92795, -82.26139 31.92899, -82.2597 31.93241, -82.25604 31.93209, -82.25534 31.93142, -82.25459 31.92747, -82.25119 31.92351, -82.24903 31.92186, -82.24801 31.9173, -82.24681 31.9151, -82.24531 31.9146, -82.24172 31.916, -82.23822 31.9201, -82.23692 31.9205, -82.23313 31.9192, -82.23183 31.9171, -82.23164 31.9134, -82.22964 31.9118, -82.22504 31.91307, -82.22114 31.9128, -82.21934 31.9097, -82.2127 31.9095, -82.21044 31.9095, -82.20664 31.9103, -82.20484 31.9093, -82.20134 31.9063, -82.19914 31.9061, -82.19524 31.9068, -82.19314 31.9068, -82.19134 31.9059, -82.19074 31.9035, -82.19224 31.9018, -82.19134 31.9005, -82.18704 31.9012, -82.18204 31.9012, -82.17724 31.9018, -82.16584 31.9036, -82.15924 31.9039, -82.15544 31.9031, -82.15154 31.901, -82.14964 31.8995, -82.14704 31.8983, -82.14514 31.8981, -82.14284 31.8993, -82.14044 31.9052, -82.13804 31.9071, -82.13574 31.9081, -82.13354 31.9077, -82.13174 31.9058, -82.13114 31.9038, -82.13154 31.9017, -82.13415 31.89741, -82.13649 31.89455, -82.13554 31.8917, -82.13344 31.8904, -82.12944 31.8904, -82.12554 31.8897, -82.12104 31.8863, -82.11954 31.886, -82.11744 31.8866, -82.11544 31.8888, -82.11624 31.8919, -82.11604 31.8952, -82.11684 31.8974, -82.11664 31.8997, -82.11491 31.90098, -82.11386 31.90119, -82.11097 31.89963, -82.11024 31.8979, -82.11054 31.8955, -82.11204 31.8913, -82.10854 31.8857, -82.10784 31.8837, -82.10944 31.88, -82.10984 31.878, -82.10962 31.87031, -82.11066 31.86918, -82.11641 31.86635, -82.11663 31.86428, -82.11548 31.86285, -82.10724 31.8638, -82.1044 31.86178, -82.09832 31.85838, -82.09424 31.85365, -82.08451 31.84794, -82.08131 31.84281, -82.07238 31.84044, -82.06607 31.83821, -82.05636 31.83095, -82.04858 31.82707, -82.05284 31.8278, -82.05626 31.82784, -82.05875 31.82744, -82.06131 31.82547, -82.06498 31.82559, -82.0681 31.82744, -82.07008 31.82596, -82.07172 31.82573, -82.07205 31.82738, -82.07425 31.8293, -82.07669 31.82919, -82.0799 31.82762, -82.08012 31.82597, -82.08204 31.82494, -82.08385 31.82282, -82.0861 31.82311, -82.0874 31.82162, -82.08724 31.821, -82.08725 31.79951, -82.10654 31.7998, -82.10663 31.79748, -82.11898 31.80395, -82.12666 31.80436, -82.1326 31.80713, -82.14055 31.80792 -82.14785 31.81062, -82.17108 31.81073, -82.17174 31.81086, -82.17849 31.8146, -82.18475 31.81117, -82.19264 31.81104, -82.19946 31.81098, -82.19708 31.81965, -82.19729 31.8249, -82.19895 31.82995, -82.19917 31.83477, -82.20159 31.83474, -82.20176 31.83591, -82.2058 31.83687, -82.20722 31.83655, -82.20912 31.83741, -82.21166 31.83744, -82.21241 31.83577, -82.21518 31.83436, -82.21944 31.83408, -82.221 31.83547, -82.22501 31.83445, -82.22628 31.83516, -82.22916 31.835, -82.23092 31.83714, -82.23204 31.83701, -82.23611 31.84104, -82.24259 31.83886, -82.2436 31.83957, -82.24399 31.84186, -82.24962 31.8451, -82.25498 31.8467, -82.2589 31.84749, -82.25996 31.85066, -82.2608 31.85147, -82.26036 31.85312, -82.26147 31.85478, -82.2672 31.85501, -82.28049 31.84936, -82.28213 31.84844, -82.28215 31.85023, -82.2853 31.85049, -82.28944 31.85227, -82.29088 31.85371, -82.29135 31.85343, -82.29321 31.85043, -82.31184 31.85038, -82.31653 31.8505, -82.32621 31.85061, -82.33523 31.85219, -82.3366 31.84597, -82.34001 31.8423, -82.35006 31.84186, -82.35209 31.84147, -82.35302 31.84397, -82.35972 31.84373, -82.36311 31.84391, -82.36906 31.84486, -82.37132 31.84421, -82.37609 31.84406, -82.37653 31.8394, -82.38011 31.8301, -82.38267 31.83002, -82.38914 31.83275, -82.39419 31.83161, -82.39642 31.82701, -82.40066 31.82461, -82.40478 31.83239, -82.40884 31.83355, -82.40948 31.8344, -82.41326 31.83303, -82.41982 31.83596, -82.42552 31.83894, -82.42819 31.83973, -82.42954 31.84095, -82.42957 31.84408, -82.43145 31.84437, -82.43146 31.8704, -82.43153 31.96618, -82.43015 31.96599, -82.42475 31.96419, -82.42315 31.96339, -82.41785 31.95959, -82.41215 31.95719, -82.40975 31.95679, -82.40805 31.95569, -82.40755 31.95329, -82.40845 31.95129, -82.40725 31.94949, -82.40554 31.94919, -82.40354 31.94979, -82.40204 31.95169, -82.40194 31.95369, -82.40414 31.95649, -82.40424 31.95849, -82.40354 31.95949, -82.40154 31.96029, -82.39874 31.95949, -82.39324 31.95558, -82.39204 31.95108, -82.39114 31.94958, -82.38884 31.94838, -82.38714 31.94818, -82.38584 31.94928, -82.38574 31.95248, -82.38914 31.95708, -82.38764 31.95968, -82.38446 31.96002, -82.38315 31.95946, -82.38092 31.95601, -82.37953 31.95498, -82.3721 31.95398, -82.37155 31.95349, -82.37255 31.95079, -82.37484 31.94969, -82.37785 31.94689, -82.37875 31.94509, -82.37805 31.94209, -82.37715 31.94169, -82.37242 31.94146, -82.36503 31.94364, -82.36233 31.94369, -82.35931 31.94174, -82.35801 31.93969, -82.35489 31.93956, -82.35306 31.93902, -82.3476 31.93856, -82.33732 31.93717, -82.32967 31.93478, -82.32328 31.93449, -82.31448 31.93078, -82.31314 31.93091, -82.30887 31.9327, -82.30768 31.93727, -82.31216 31.93939, -82.31533 31.9428, -82.31609 31.94415, -82.31535 31.94637)))

=

Geometry



Simple Feature Geometry

data.frame

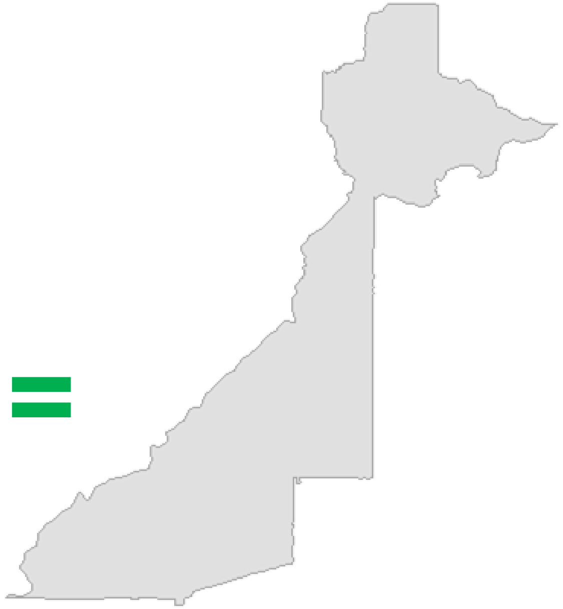
Name	GDP	Population	geometry
Fulton County	000,000,000	000,000,000	

Simple feature

MULTIPOLYGON (((-82.31535 31.94637, -82.31262 31.94746, -82.31068 31.9473, -82.30861 31.94555, -82.30654 31.94161, -82.30602 31.93869, -82.30452 31.93792, -82.3028 31.9339, -82.30111 31.93327, -82.29809 31.93399, -82.29533 31.93886, -82.29334 31.94084, -82.29137 31.94248, -82.28642 31.94244, -82.28327 31.94076, -82.27955 31.93798, -82.27762 31.93707, -82.27033 31.93587, -82.2682 31.93279, -82.26716 31.92862, -82.26316 31.92795, -82.26139 31.92899, -82.2597 31.93241, -82.25604 31.93209, -82.25534 31.93142, -82.25459 31.92747, -82.25119 31.92351, -82.24903 31.92186, -82.24801 31.9173, -82.24681 31.9151, -82.24531 31.9146, -82.24172 31.916, -82.23822 31.9201, -82.23692 31.9205, -82.23313 31.9192, -82.23183 31.9171, -82.23164 31.9134, -82.22964 31.9118, -82.22504 31.91307, -82.22114 31.9128, -82.21934 31.9097, -82.2127 31.9095, -82.21044 31.9095, -82.20664 31.9103, -82.20484 31.9093, -82.20134 31.9063, -82.19914 31.9061, -82.19524 31.9068, -82.19314 31.9068, -82.19134 31.9059, -82.19074 31.9035, -82.19224 31.9018, -82.19134 31.9005, -82.18704 31.9012, -82.18204 31.9012, -82.17724 31.9018, -82.16584 31.9036, -82.15924 31.9039, -82.15544 31.9031, -82.15154 31.901, -82.14964 31.8995, -82.14704 31.8983, -82.14514 31.8981, -82.14284 31.8993, -82.14044 31.9052, -82.13804 31.9071, -82.13574 31.9081, -82.13354 31.9077, -82.13174 31.9058, -82.13114 31.9038, -82.13154 31.9017, -82.13415 31.89741, -82.13649 31.89455, -82.13554 31.8917, -82.13344 31.8904, -82.12944 31.8904, -82.12554 31.8897, -82.12104 31.8863, -82.11954 31.886, -82.11744 31.8866, -82.11544 31.8888, -82.11624 31.8919, -82.11604 31.8952, -82.11684 31.8974, -82.11664 31.8997, -82.11491 31.90098, -82.11386 31.90119, -82.11097 31.89963, -82.11024 31.8979, -82.11054 31.8955, -82.11204 31.8913, -82.10854 31.8857, -82.10784 31.8837, -82.10944 31.88, -82.10984 31.878, -82.10962 31.87031, -82.11066 31.86918, -82.11641 31.86635, -82.11663 31.86428, -82.11548 31.86285, -82.10724 31.8638, -82.1044 31.86178, -82.09832 31.85838, -82.09424 31.85365, -82.08451 31.84794, -82.08131 31.84281, -82.07238 31.84044, -82.06607 31.83821, -82.05636 31.83095, -82.04858 31.82707, -82.05284 31.8278, -82.05626 31.82784, -82.05875 31.82744, -82.06131 31.82547, -82.06498 31.82559, -82.0681 31.82744, -82.07008 31.82596, -82.07172 31.82573, -82.07205 31.82738, -82.07425 31.8293, -82.07669 31.82919, -82.0799 31.82762, -82.08012 31.82597, -82.08204 31.82494, -82.08385 31.82282, -82.0861 31.82311, -82.0874 31.82162, -82.08724 31.821, -82.08725 31.79951, -82.10654 31.7998, -82.10663 31.79748, -82.11898 31.80395, -82.12666 31.80436, -82.1326 31.80713, -82.14055 31.80792, -82.14785 31.81062, -82.17108 31.81073, -82.17174 31.81086, -82.17849 31.8146, -82.18475 31.81117, -82.19264 31.81104, -82.19946 31.81098, -82.19708 31.81965, -82.19729 31.8249, -82.19895 31.82995, -82.19917 31.83477, -82.20159 31.83474, -82.20176 31.83591, -82.2058 31.83687, -82.20722 31.83655, -82.20912 31.83741, -82.21166 31.83744, -82.21241 31.83577, -82.21518 31.83436, -82.21944 31.83408, -82.221 31.83547, -82.22501 31.83445, -82.22628 31.83516, -82.22916 31.835, -82.23092 31.83714, -82.23204 31.83701, -82.23611 31.84104, -82.24259 31.83886, -82.2436 31.83957, -82.24399 31.84186, -82.24962 31.8451, -82.25498 31.8467, -82.2589 31.84749, -82.25996 31.85066, -82.2608 31.85147, -82.26036 31.85312, -82.26147 31.85478, -82.2672 31.85501, -82.28049 31.84936, -82.28213 31.84844, -82.28215 31.85023, -82.2853 31.85049, -82.28944 31.85227, -82.29088 31.85371, -82.29135 31.85343, -82.29321 31.85043, -82.31184 31.85038, -82.31653 31.8505, -82.32621 31.85061, -82.33523 31.85219, -82.3366 31.84597, -82.34001 31.8423, -82.35006 31.84186, -82.35209 31.84147, -82.35302 31.84397, -82.35972 31.84373, -82.36311 31.84391, -82.36906 31.84486, -82.37132 31.84421, -82.37609 31.84406, -82.37653 31.8394, -82.38011 31.8301, -82.38267 31.83002, -82.38914 31.83275, -82.39419 31.83161, -82.39642 31.82701, -82.40066 31.82461, -82.40478 31.83239, -82.40884 31.83355, -82.40948 31.8344, -82.41326 31.83303, -82.41982 31.83596, -82.42552 31.83894, -82.42819 31.83973, -82.42954 31.84095, -82.42957 31.84408, -82.43145 31.84437, -82.43146 31.8704, -82.43153 31.96618, -82.43015 31.96599, -82.42475 31.96419, -82.42315 31.96339, -82.41785 31.95959, -82.41215 31.95719, -82.40975 31.95679, -82.40805 31.95569, -82.40755 31.95329, -82.40845 31.95129, -82.40725 31.94949, -82.40554 31.94919, -82.40354 31.94979, -82.40204 31.95169, -82.40194 31.95369, -82.40414 31.95649, -82.40424 31.95849, -82.40354 31.95949, -82.40154 31.96029, -82.39874 31.95949, -82.39324 31.95558, -82.39204 31.95108, -82.39114 31.94958, -82.38884 31.94838, -82.38714 31.94818, -82.38584 31.94928, -82.38574 31.95248, -82.38914 31.95708, -82.38764 31.95968, -82.38446 31.96002, -82.38315 31.95946, -82.38092 31.95601, -82.37953 31.95498, -82.3721 31.95398, -82.37155 31.95349, -82.37255 31.95079, -82.37484 31.94969, -82.37785 31.94689, -82.37875 31.94509, -82.37805 31.94209, -82.37715 31.94169, -82.37242 31.94146, -82.36503 31.94364, -82.36233 31.94369, -82.35931 31.94174, -82.35801 31.93969, -82.35489 31.93956, -82.35306 31.93902, -82.3476 31.93856, -82.33732 31.93717, -82.32967 31.93478, -82.32328 31.93449, -82.31448 31.93078, -82.31314 31.93091, -82.30887 31.9327, -82.30768 31.93727, -82.31216 31.93939, -82.31533 31.9428, -82.31609 31.94415, -82.31535 31.94637)))

=

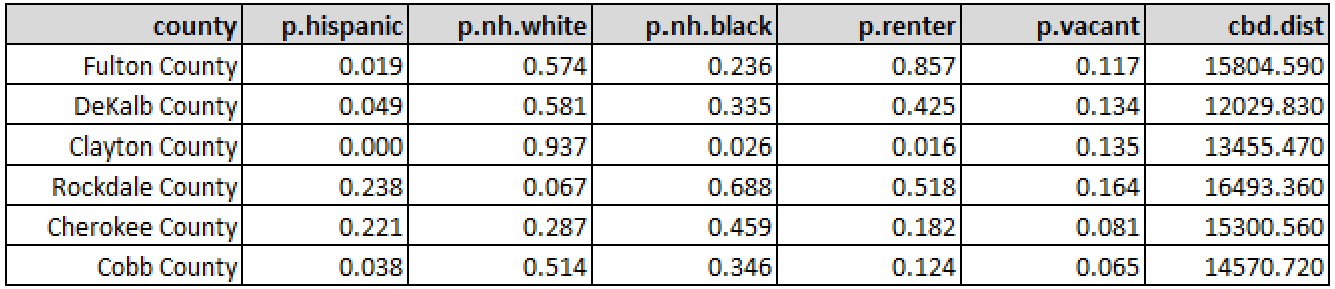
Geometry



sf Package in R

written in sf format

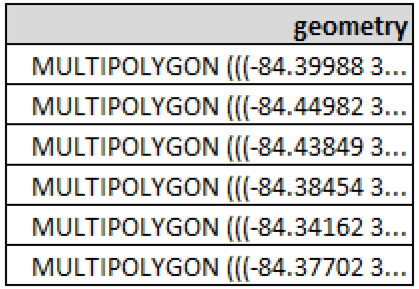
data.frame



county	p.hispanic	p.nh.white	p.nh.black	p.renter	p.vacant	cbd.dist
Fulton County	0.019	0.574	0.236	0.857	0.117	15804.590
DeKalb County	0.049	0.581	0.335	0.425	0.134	12029.830
Clayton County	0.000	0.937	0.026	0.016	0.135	13455.470
Rockdale County	0.238	0.067	0.688	0.518	0.164	16493.360
Cherokee County	0.221	0.287	0.459	0.182	0.081	15300.560
Cobb County	0.038	0.514	0.346	0.124	0.065	14570.720

+

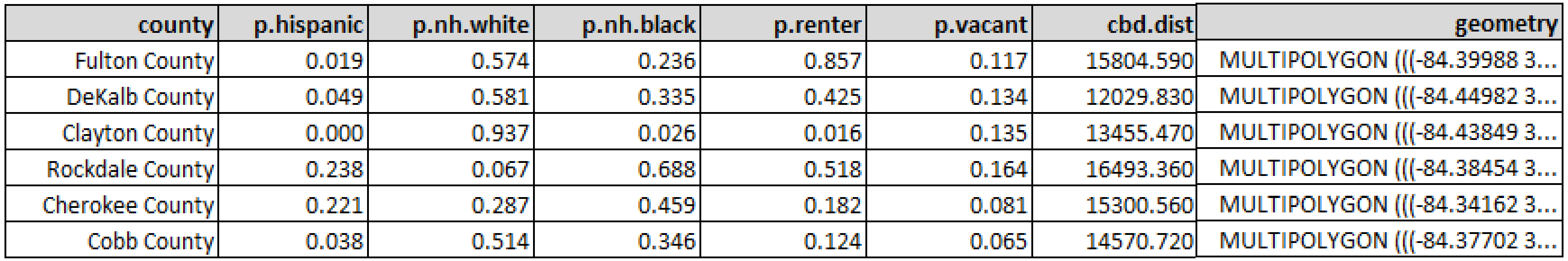
geometry



geometry
MULTIPOLYGON (((-84.39988 3...
MULTIPOLYGON (((-84.44982 3...
MULTIPOLYGON (((-84.43849 3...
MULTIPOLYGON (((-84.38454 3...
MULTIPOLYGON (((-84.34162 3...
MULTIPOLYGON (((-84.37702 3...

=

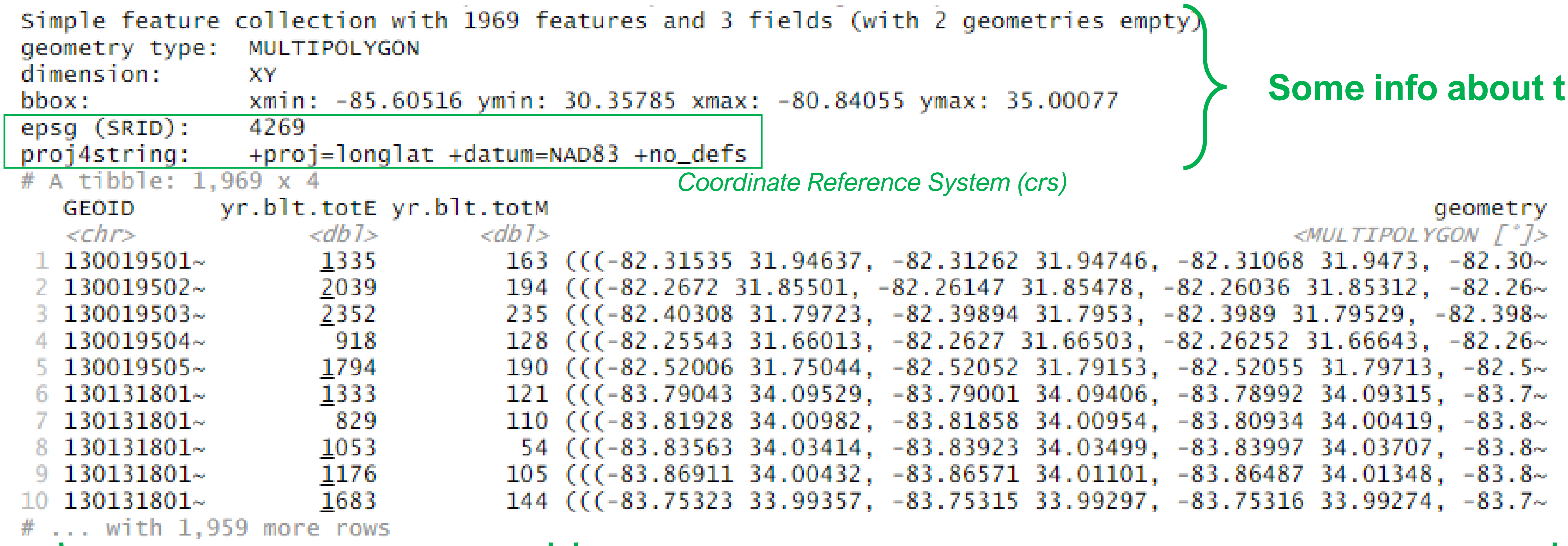
Simple feature collection



county	p.hispanic	p.nh.white	p.nh.black	p.renter	p.vacant	cbd.dist	geometry
Fulton County	0.019	0.574	0.236	0.857	0.117	15804.590	MULTIPOLYGON (((-84.39988 3...
DeKalb County	0.049	0.581	0.335	0.425	0.134	12029.830	MULTIPOLYGON (((-84.44982 3...
Clayton County	0.000	0.937	0.026	0.016	0.135	13455.470	MULTIPOLYGON (((-84.43849 3...
Rockdale County	0.238	0.067	0.688	0.518	0.164	16493.360	MULTIPOLYGON (((-84.38454 3...
Cherokee County	0.221	0.287	0.459	0.182	0.081	15300.560	MULTIPOLYGON (((-84.34162 3...
Cobb County	0.038	0.514	0.346	0.124	0.065	14570.720	MULTIPOLYGON (((-84.37702 3...

sf Package in R

Simple feature collection



```
Simple feature collection with 1969 features and 3 fields (with 2 geometries empty)
geometry type:  MULTIPOLYGON
dimension:      XY
bbox:           xmin: -85.60516 ymin: 30.35785 xmax: -80.84055 ymax: 35.00077
epsg (SRID):    4269
proj4string:    +proj=longlat +datum=NAD83 +no_defs
# A tibble: 1,969 x 4
   GEOID      yr.blt.totE yr.blt.totM                                                   geometry
   <chr>            <dbl>       <dbl>                                         <MULTIPOLYGON [°]>
 1 130019501~        1335         163 (((-82.31535 31.94637, -82.31262 31.94746, -82.31068 31.9473, -82.30~
 2 130019502~        2039         194 (((-82.2672 31.85501, -82.26147 31.85478, -82.26036 31.85312, -82.26~
 3 130019503~        2352         235 (((-82.40308 31.79723, -82.39894 31.7953, -82.3989 31.79529, -82.398~
 4 130019504~         918         128 (((-82.25543 31.66013, -82.2627 31.66503, -82.26252 31.66643, -82.26~
 5 130019505~        1794         190 (((-82.52006 31.75044, -82.52052 31.79153, -82.52055 31.79713, -82.5~
 6 130131801~        1333         121 (((-83.79043 34.09529, -83.79001 34.09406, -83.78992 34.09315, -83.7~
 7 130131801~         829         110 (((-83.81928 34.00982, -83.81858 34.00954, -83.80934 34.00419, -83.8~
 8 130131801~        1053          54 (((-83.83563 34.03414, -83.83923 34.03499, -83.83997 34.03707, -83.8~
 9 130131801~        1176         105 (((-83.86911 34.00432, -83.86571 34.01101, -83.86487 34.01348, -83.8~
10 130131801~        1683         144 (((-83.75323 33.99357, -83.75315 33.99297, -83.75316 33.99274, -83.7~
# ... with 1,959 more rows
```

Coordinate Reference System (crs)

Some info about the file

Attributes (normal data.frame part)

geometry information

sf Package in R

You can do (almost) everything you learned in *Intro to GIS* class in R

- Fast reading and writing of data.
- Enhanced plotting performance.
- sf objects can be treated as data frames in most operations.
- sf functions can be combined using %>% operator and works well with the tidyverse collection of R packages.
- sf function names are relatively consistent and intuitive (all begin with st_).

Source: Geocomputation with R (https://geocompr.robinlovelace.net/)

Exercise 1

- Reading
- Printing
- Plotting

Source: Geocomputation with R (https://geocompr.robinlovelace.net/)

Exercise 1

For some of other mapping functions..

http://zevross.com/blog/2018/10/02/creating-beautiful-demographic-maps-in-r-with-the-tidycensus-and-tmap-packages/

Source: Geocomputation with R (https://geocompr.robinlovelace.net/)

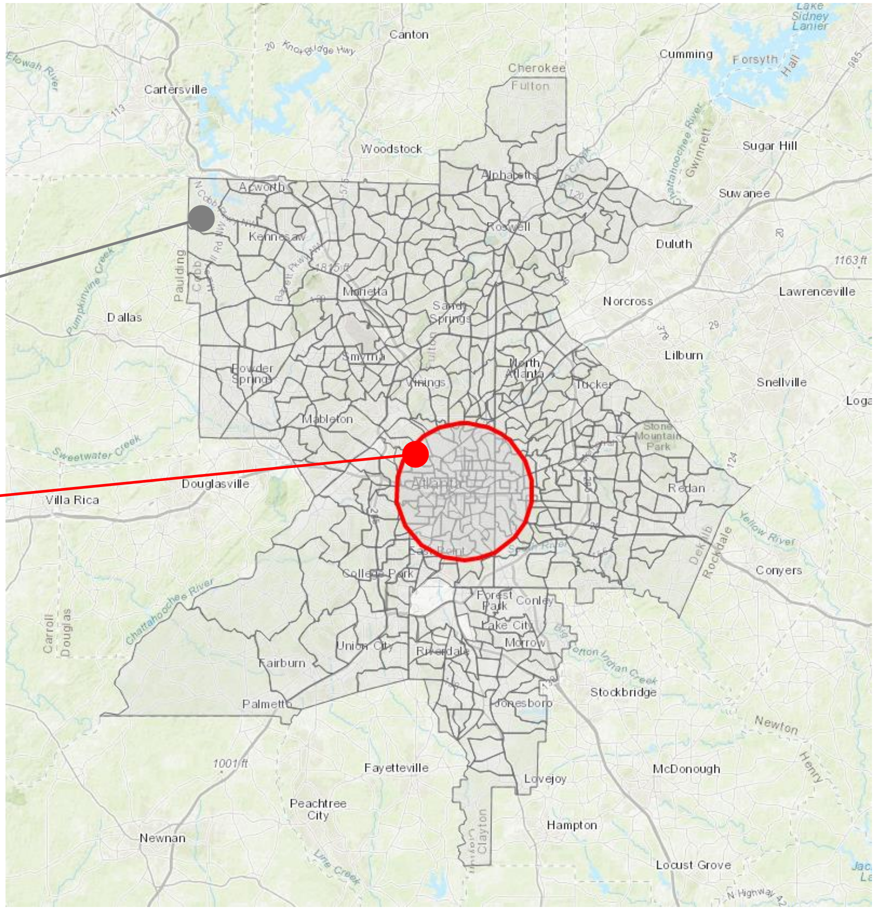
Subsetting

Select by attributes

- Nothing new to learn for this. All the techniques using ***data.frame[,]*** format works for sf objects too.

Select by location

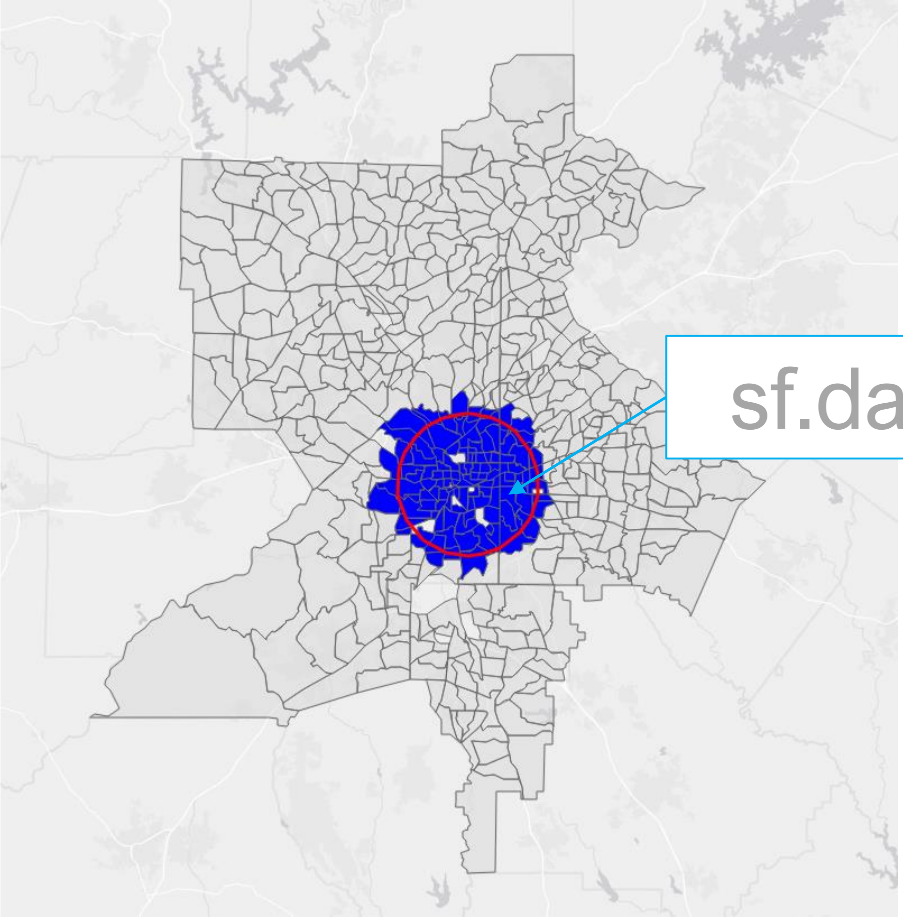
- You need two or more sf objects for this.
- In this example,
  - ✓ One is the census tract sf object
  - ✓ The other is the red circle



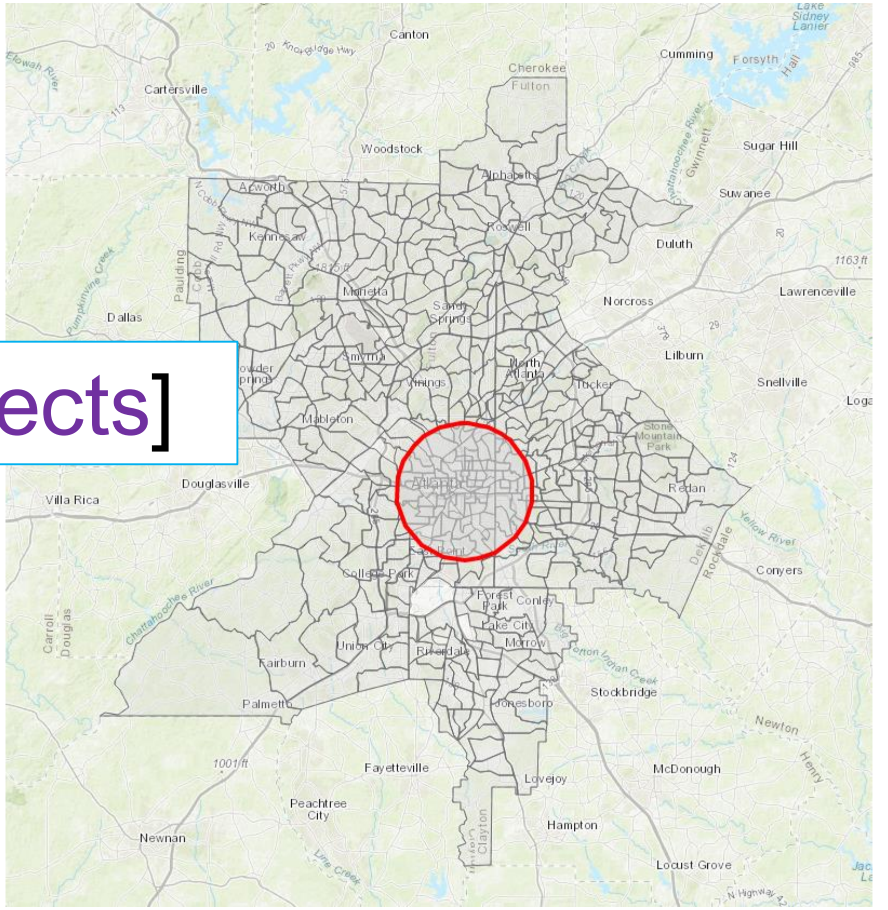
Let's first select Census Tracts that ***intersects*** the circle

Subsetting

sf.obj1[sf.obj2, , operation]



sf.data[cbd, , st_intersects]

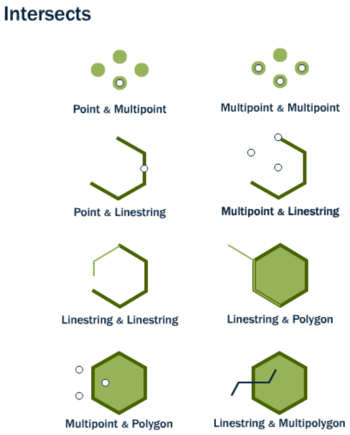


st_is_within_distance

Subsetting

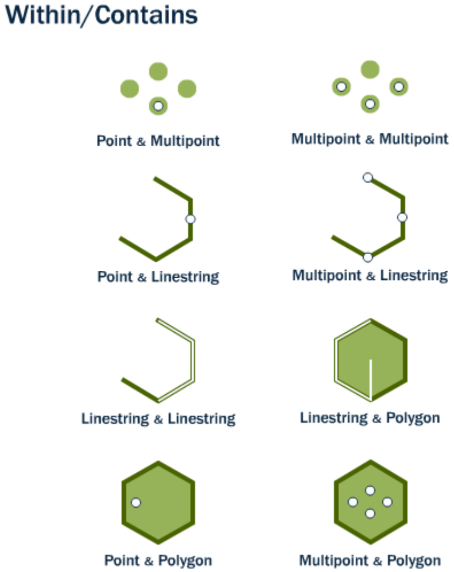
st_intersects

if the two shapes have any space in common, i.e., if their boundaries or interiors intersect.



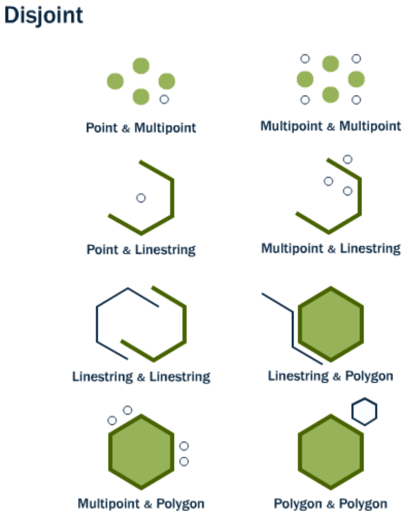
st_within

if the first geometry is completely within the second geometry. st_within tests for the exact opposite result of st_contains.

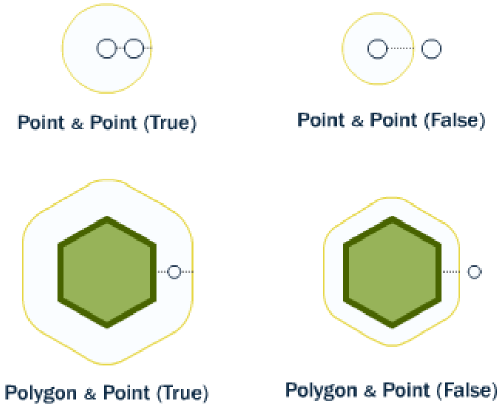


st_disjoint

If two geometries are disjoint, they do not intersect

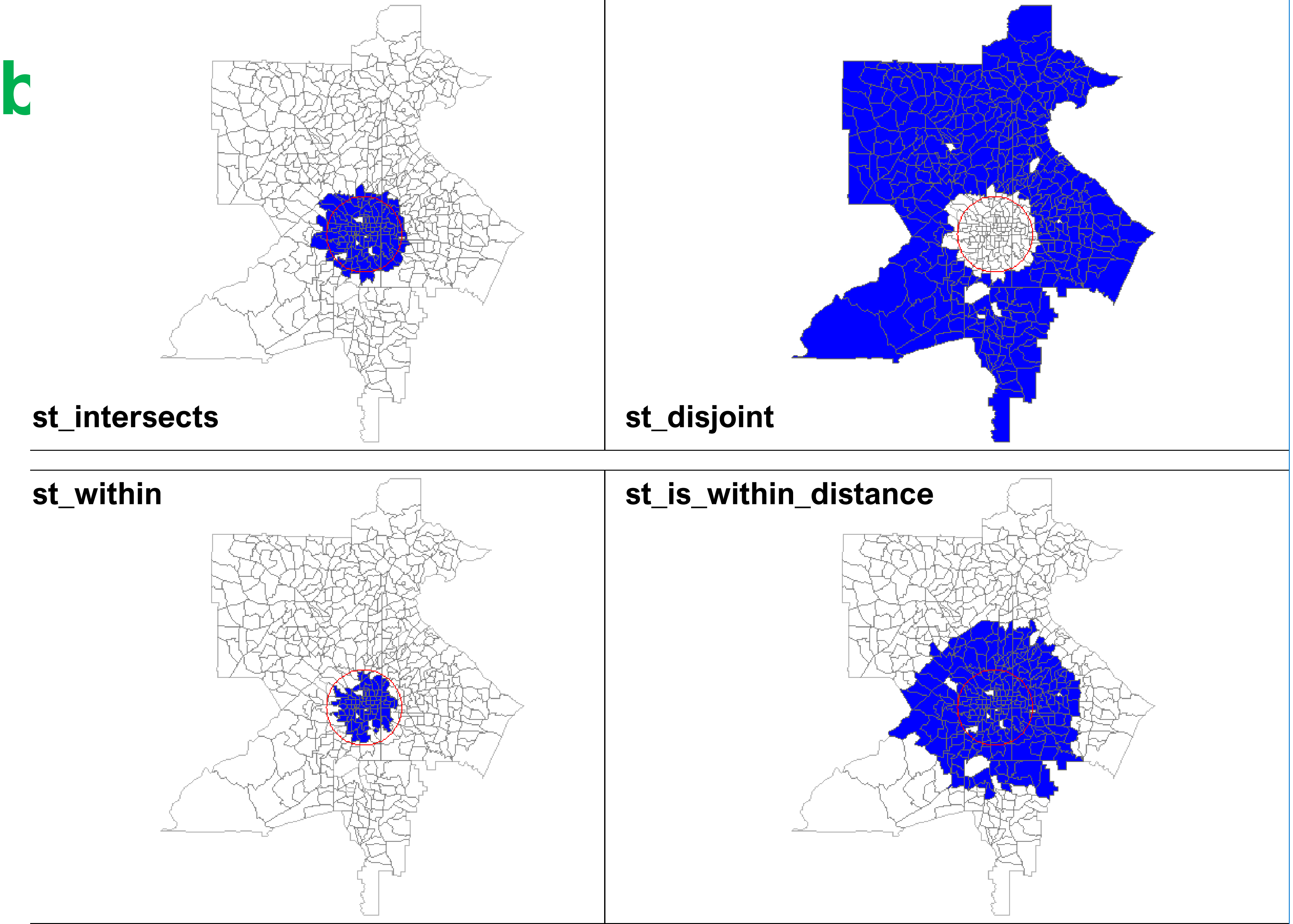


st_is_within_distance



st_is_within_distance

Sub



st_intersects

st_disjoint

st_within

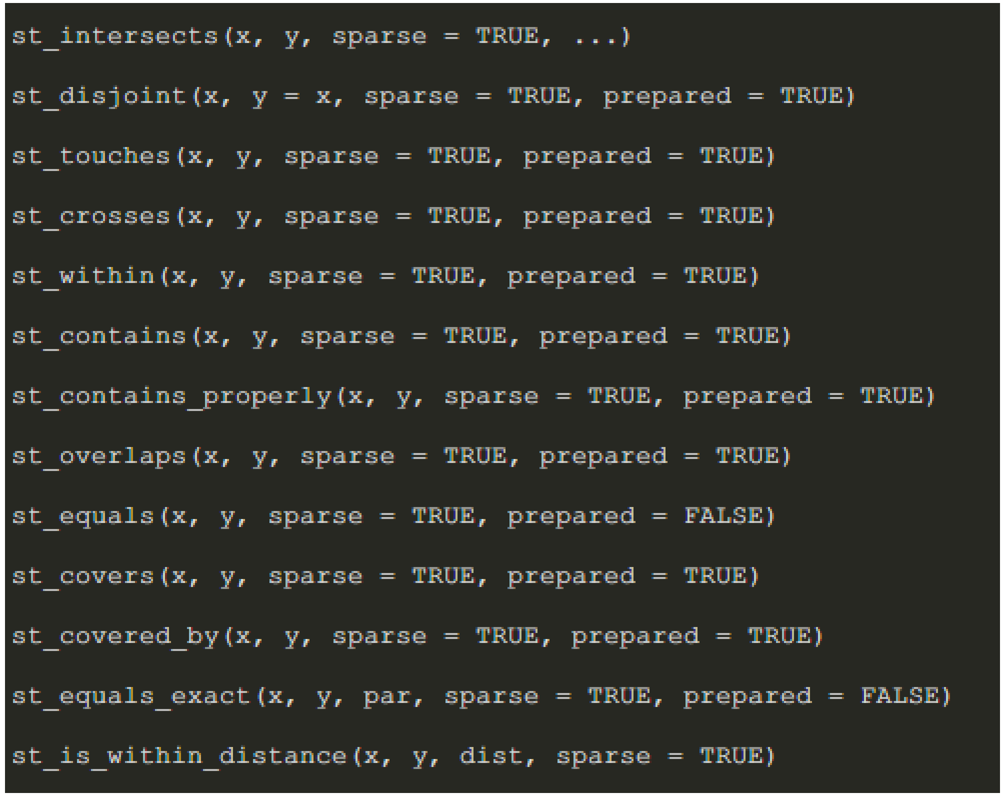
st_is_within_distance

st_is_within_distance

Subsetting

For more options other than st_intersects, etc.,

Type ?st_intersects in R console.



```
st_intersects(x, y, sparse = TRUE, ...)
st_disjoint(x, y = x, sparse = TRUE, prepared = TRUE)
st_touches(x, y, sparse = TRUE, prepared = TRUE)
st_crosses(x, y, sparse = TRUE, prepared = TRUE)
st_within(x, y, sparse = TRUE, prepared = TRUE)
st_contains(x, y, sparse = TRUE, prepared = TRUE)
st_contains_properly(x, y, sparse = TRUE, prepared = TRUE)
st_overlaps(x, y, sparse = TRUE, prepared = TRUE)
st_equals(x, y, sparse = TRUE, prepared = FALSE)
st_covers(x, y, sparse = TRUE, prepared = TRUE)
st_covered_by(x, y, sparse = TRUE, prepared = TRUE)
st_equals_exact(x, y, par, sparse = TRUE, prepared = FALSE)
st_is_within_distance(x, y, dist, sparse = TRUE)
```