











Spatial manipulation with sf: : CHEAT SHEET













The sf package provides a set of tools for working with geospatial vectors, i.e. points, lines, polygons, etc.









Geometric confirmation

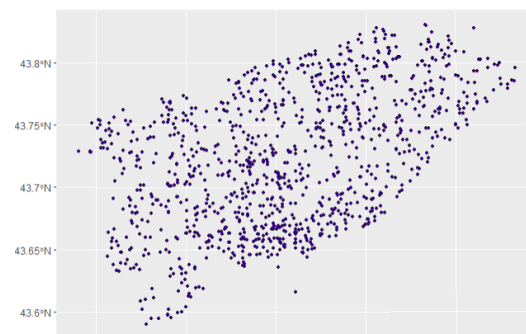
-  `st_contains(x, y, ...)` Identifies if x is within y (i.e. point within polygon)
-  `st_covered_by(x, y, ...)` Identifies if x is completely within y (i.e. polygon completely within polygon)
-  `st_covers(x, y, ...)` Identifies if any point from x is outside of y (i.e. polygon outside polygon)
-  `st_crosses(x, y, ...)` Identifies if any geometry of x have commonalities with y
-  `st_disjoint(x, y, ...)` Identifies when geometries from x do not share space with y
-  `st_equals(x, y, ...)` Identifies if x and y share the same geometry
-  `st_intersects(x, y, ...)` Identifies if x and y geometry share any space
-  `st_overlaps(x, y, ...)` Identifies if geometries of x and y share space, are of the same dimension, but are not completely contained by each other
-  `st_touches(x, y, ...)` Identifies if geometries of x and y share a common point but their interiors do not intersect
-  `st_within(x, y, ...)` Identifies if x is in a specified distance to y

Geometric operations

-  `st_boundary(x)` Creates a polygon that encompasses the full extent of the geometry
-  `st_buffer(x, dist, nQuadSegs)` Creates a polygon covering all points of the geometry within a given distance
-  `st_centroid(x, ..., of_largest_polygon)` Creates a point at the geometric centre of the geometry
-  `st_convex_hull(x)` Creates geometry that represents the minimum convex geometry of x
-  `st_line_merge(x)` Creates linestring geometry from sewing multi linestring geometry together
-  `st_node(x)` Creates nodes on overlapping geometry where nodes do not exist
-  `st_point_on_surface(x)` Creates a point that is guaranteed to fall on the surface of the geometry
-  `st_polygonize(x)` Creates polygon geometry from linestring geometry
-  `st_segmentize(x, dfMaxLength, ...)` Creates linestring geometry from x based on a specified length
-  `st_simplify(x, preserveTopology, dTolerance)` Creates a simplified version of the geometry based on a specified tolerance

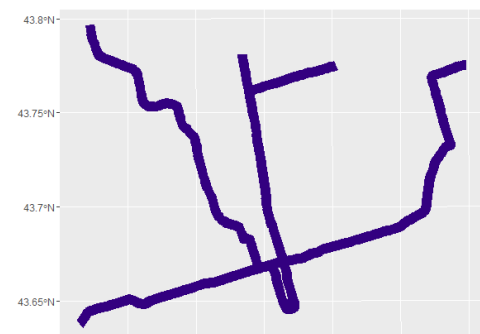
Geometry creation

-  `st_triangulate(x, dTolerance, bOnlyEdges)` Creates polygon geometry as triangles from point geometry
-  `st_voronoi(x, envelope, dTolerance, bOnlyEdges)` Creates polygon geometry covering the envelope of x, with x at the centre of the geometry
-  `st_point(x, c(numeric vector), dim = "XYZ")` Creating point geometry from numeric values
-  `st_multipoint(x = matrix(numeric values in rows), dim = "XYZ")` Creating multi point geometry from numeric values
-  `st_linestring(x = matrix(numeric values in rows), dim = "XYZ")` Creating linestring geometry from numeric values
-  `st_multilinestring(x = list(numeric matrices in rows), dim = "XYZ")` Creating multi linestring geometry from numeric values
-  `st_polygon(x = list(numeric matrices in rows), dim = "XYZ")` Creating polygon geometry from numeric values
-  `st_multipolygon(x = list(numeric matrices in rows), dim = "XYZ")` Creating multi polygon geometry from numeric values



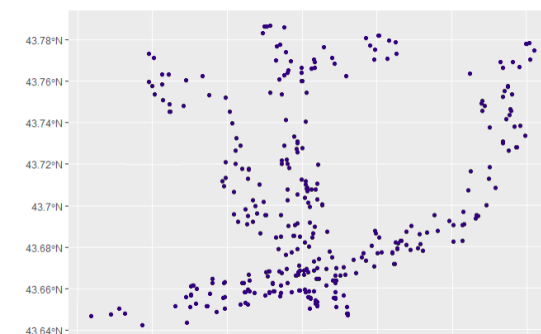
`ggplot() +
geom_sf(data = schools)`

+



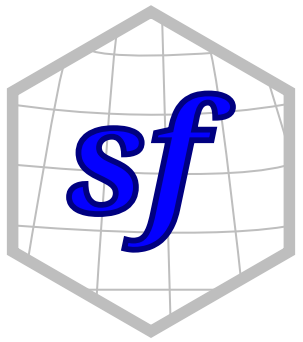
`ggplot() +
geom_sf(data = subway)`

=>










`ggplot() +
geom_sf(data = st_intersection(schools, st_buffer(subway, 1000)))`

Spatial manipulation with sf: : CHEAT SHEET



The sf package provides a set of tools for working with geospatial vectors, i.e. points, lines, polygons, etc.

Geometry operations

-  **st_contains(x, y, ...)** Identifies if x is within y (i.e. point within polygon)
-  **st_crop(x, y, ..., xmin, ymin, xmax, ymax)** Creates geometry of x that intersects a specified rectangle
-  **st_difference(x, y)** Creates geometry from x that does not intersect with y
-  **st_intersection(x, y)** Creates geometry of the shared portion of x and y
-  **st_sym_difference(x, y)** Creates geometry representing portions of x and y that do not intersect
-  **st_snap(x, y, tolerance)** Snap nodes from geometry x to geometry y
-  **st_union(x, y, ..., by_feature)** Creates multiple geometries into a single geometry, consisting of all geometry elements

Geometric measurement

- st_area(x)** Calculate the surface area of a polygon geometry based on the current coordinate reference system
- st_distance(x, y, ..., dist_fun, by_element, which)** Calculates the 2D distance between x and y based on the current coordinate system
- st_length(x)** Calculates the 2D length of a geometry based on the current coordinate system

Misc operations

- st_cast(x, to, ...)** Change x geometry to a different geometry type
- st_coordinates(x, ...)** Creates a matrix of coordinate values from x
- st_crs(x, ...)** Identifies the coordinate reference system of x
- st_join(x, y, join, FUN, suffix, ...)** Performs a spatial left or inner join between x and y
- st_make_grid(x, cellsize, offset, n, crs, what)** Creates rectangular grid geometry over the bounding box of x
- st_nearest_feature(x, y)** Creates an index of the closest feature between x and y
- st_nearest_points(x, y, ...)** Returns the closest point between x and y
- st_transform(x, crs, ...)** Convert coordinates of x to a different coordinate reference system

