











# 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

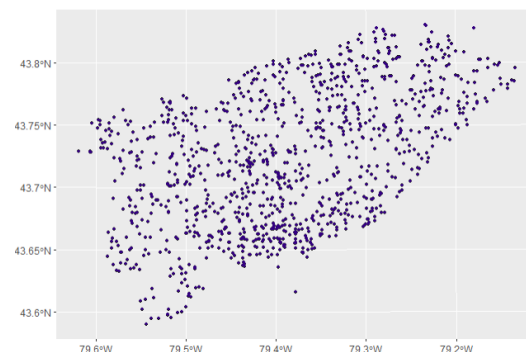
-  `st_contains(x, y, ...)` Identifies if x is within y (i.e. point within polygon)
-  `st_covered_by(x, y, ...)` Identifies if x is completely within y (i.e. polygon completely within polygon)
-  `st_covers(x, y, ...)` Identifies if any point from x is outside of y (i.e. polygon outside polygon)
-  `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

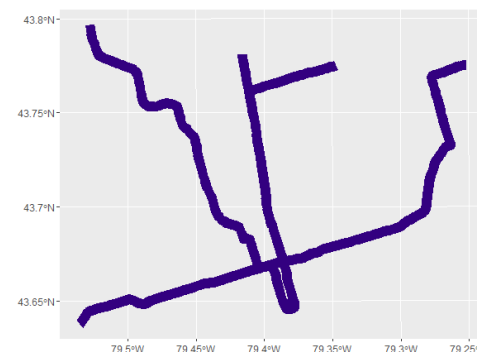
## Geometry creation

-  `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



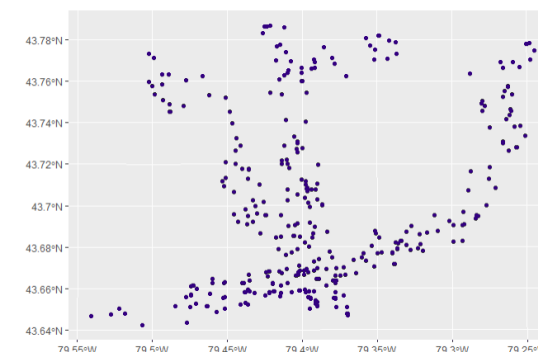
ggplot() +  
geom\_sf(data = schools)

+



ggplot() +  
geom\_sf(data = subway)

=>



ggplot() +  
geom\_sf(data = st\_intersection(schools, st\_buffer(subway, 1000)))