Lecture 16
Spatial Data and Cartography

Colin Rundel
03/20/2017
Background
R has a rich package ecosystem for read/writing, manipulating, and analyzing geospatial data.

Some core packages:

- **sp** - core classes for handling spatial data, additional utility functions.
- **rgdal** - R interface to **gdal** (Geospatial Data Abstraction Library) for reading and writing spatial data.
- **rgeos** - R interface to **geos** (Geometry Engine Open Source) library for querying and manipulating spatial data. Reading and writing WKT.
- **raster** - classes and tools for handling spatial raster data.

See more - Spatial task view
Analysis of geospatial data in R

R has a rich package ecosystem for read/writing, manipulating, and analyzing geospatial data.

Some core packages:

- **sp**—core classes for handling spatial data, additional utility functions.
- **rgdal**—R interface to gdal (Geospatial Data Abstraction Library) for reading and writing spatial data.
- **rgeos**—R interface to geos (Geometry Engine Open Source) library for querying and manipulating spatial data. Reading and writing WKT.
- **sf**—Combines the functionality of **sp**, **rgdal**, and **rgeos** into a single package based on tidy principles.
- **raster**—classes and tools for handling spatial raster data.

See more - Spatial task view
Installing sf

The sf package is currently under active development and is evolving rapidly. The version on CRAN should be reasonably up to date, but the most current version is always available from github.

Difficulty comes from requirements for external libraries (geos, gdal, and proj4).

- **Windows** - installing from source works when Rtools is installed (system requirements are downloaded from rwinlib)
- **MacOS** - install dependencies via homebrew:

```bash
brew tap osgeo/osgeo4mac && brew tap --repair
brew install proj
brew install geos
brew install udunits
brew unlink gdal
brew install gdal2
```

- **Linux** - Install development packages for GDAL (>= 2.0.0), GEOS (>= 3.3.0) and Proj.4 (>= 4.8.0) from your package manager of choice.
Simple Features

- Point
- Linestring
- Polygon
- Polygon w/ Hole(s)
- Multipoint
- Multilinestring
- Multipolygon
- Multipolygon w/ Hole(s)
Geometry Collection

Point, Multipoint, Multilinestring, Polygon
Reading and writing geospatial data via sp

- **maptools**
  - `readShapePoints` / `writeShapePoints` - Shapefile w/ points
  - `readShapeLines` / `writeShapeLines` - Shapefile w/ lines
  - `readShapePoly` / `writeShapePoly` - Shapefile w/ polygons
  - `readShapeSpatial` / `writeShapeSpatial` - Shapefile

- **rgdal**
  - `readOGR` / `writeOGR` - Shapefile, GeoJSON, KML, ...

- **rgeos**
  - `readWKT` / `writeWKT` - Well Known Text

- **sf**
  - `st_read` / `st_write` - Shapefile, GeoJSON, KML, ...
Reading and writing geospatial data via sp

- maptools
  - `readShapePoints` / `writeShapePoints` - Shapefile w/ points
  - `readShapeLines` / `writeShapeLines` - Shapefile w/ lines
  - `readShapePoly` / `writeShapePoly` - Shapefile w/ polygons
  - `readShapeSpatial` / `writeShapeSpatial` - Shapefile
- rgdal
  - `readOGR` / `writeOGR` - Shapefile, GeoJSON, KML, ...
- rgeos
  - `readWKT` / `writeWKT` - Well Known Text
- sf
  - `st_read` / `st_write` - Shapefile, GeoJSON, KML, ...
Geospatial stuff is complicated
Projections

Lat/Long (epsg:4326)

Google / Web Mercator (epsg:3857)

Lambert Conformal Conic:

Alberts Equal Area

Robinson

Mollweide
Want to fly from the Western most point in the US to the Eastern most point?
Relationships
How do we define the distance between A and B, A and C, or B and C?
Using sf
Example data

```r
nc = st_read("data/gis/nc_counties/", quiet=TRUE, stringsAsFactors=FALSE)
air = st_read("data/gis/airports/", quiet=TRUE, stringsAsFactors=FALSE)
hwy = st_read("data/gis/us_interstates/", quiet=TRUE, stringsAsFactors=FALSE)
```

```r
head(nc)
## Simple feature collection with 6 features and 8 fields
## geometry type: MULTIPOLYGON
## dimension: XY
## bbox: xmin: -81.74178 ymin: 36.07215 xmax: -75.77323 ymax: 36.58815
## epsg (SRID): 4269
## proj4string: +proj=longlat +datum=NAD83 +no_defs
## AREA PERIMETER COUNTYP010 STATE COUNTY FIPS
## 1 0.11175964 1.610396 1994 NC Ashe County 37009
## 2 0.06159483 1.354829 1996 NC Alleghany County 37005
## 3 0.14023009 1.769388 1998 NC Surry County 37171
## 4 0.08912401 1.425249 1999 NC Gates County 37073
## 5 0.06865730 4.428217 2000 NC Currituck County 37053
## 6 0.11859434 1.404309 2001 NC Stokes County 37169
## STATE_FIPS SQUARE_MIL geometry
## 1 37 429.350 MULTIPOLYGON(((-81.65648656...
## 2 37 236.459 MULTIPOLYGON(((-81.30999399...
## 3 37 538.863 MULTIPOLYGON(((-80.71416384...
## 4 37 342.340 MULTIPOLYGON(((-76.91183250...
## 5 37 263.871 MULTIPOLYGON(((-75.82777792...
## 6 37 455.793 MULTIPOLYGON(((-80.43314893...
```
## Simple feature collection with 6 features and 16 fields
## geometry type: POINT
## dimension:   XY
## bbox:      xmin: -118.1506   ymin: 27.49748  xmax: -72.04514  ymax: 46.25149
## epsg (SRID): 4269
## proj4string: +proj=longlat +datum=NAD83 +no_defs

<table>
<thead>
<tr>
<th>AIRPRTX010</th>
<th>FEATURE</th>
<th>ICAO</th>
<th>IATA</th>
<th>AIRPT_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>AIRPORT</td>
<td>KGON</td>
<td>GON</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>AIRPORT</td>
<td>K6S5</td>
<td>6S5</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>AIRPORT</td>
<td>KMHV</td>
<td>MHV</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>AIRPORT</td>
<td>KSEE</td>
<td>SEE</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>AIRPORT</td>
<td>KFPR</td>
<td>FPR</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>AIRPORT</td>
<td>KRYY</td>
<td>RYY</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CITY</th>
<th>STATE</th>
<th>STATE_FIPS</th>
<th>COUNTY</th>
<th>FIPS</th>
<th>TOT_ENP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROTON</td>
<td>CT</td>
<td>09</td>
<td>NEW LONDON</td>
<td>09011</td>
<td>75</td>
</tr>
<tr>
<td>HAMILTON</td>
<td>MT</td>
<td>30</td>
<td>RAVALLI</td>
<td>30081</td>
<td>112</td>
</tr>
<tr>
<td>MOJAVE</td>
<td>CA</td>
<td>06</td>
<td>KERN</td>
<td>06029</td>
<td>135</td>
</tr>
<tr>
<td>SAN DIEGO/EL CAJON</td>
<td>CA</td>
<td>06</td>
<td>SAN DIEGO</td>
<td>06073</td>
<td>30</td>
</tr>
<tr>
<td>FORT PIERCE</td>
<td>FL</td>
<td>12</td>
<td>ST LUCIE</td>
<td>12111</td>
<td>33</td>
</tr>
<tr>
<td>ATLANTA</td>
<td>GA</td>
<td>13</td>
<td>COBB</td>
<td>13067</td>
<td>110</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LATITUDE</th>
<th>LONGITUDE</th>
<th>ELEV</th>
<th>ACT_DATE</th>
<th>CNTL_TWR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 41.33006</td>
<td>-72.04514</td>
<td>9</td>
<td>04/1940</td>
<td>Y</td>
</tr>
<tr>
<td>2 46.25149</td>
<td>-114.12554</td>
<td>3642</td>
<td>04/1940</td>
<td>N</td>
</tr>
<tr>
<td>3 35.05864</td>
<td>-118.15056</td>
<td>2801</td>
<td>04/1940</td>
<td>Y</td>
</tr>
<tr>
<td>4 32.82622</td>
<td>-116.97244</td>
<td>388</td>
<td>12/1942</td>
<td>Y</td>
</tr>
<tr>
<td>5 27.49748</td>
<td>-80.37263</td>
<td>24</td>
<td>&lt;NA&gt;</td>
<td>Y</td>
</tr>
<tr>
<td>6 34.01316</td>
<td>-84.59706</td>
<td>1041</td>
<td>12/1942</td>
<td>Y</td>
</tr>
</tbody>
</table>

```r

```

```

```

```

```

```

```

```

```

```

```

```r

```
head(hwy)
## Simple feature collection with 6 features and 3 fields
## geometry type: MULTILINESTRING
## dimension: XY
## bbox: xmin: -1910156 ymin: 3264168 xmax: 1591535 ymax: 5340953
## epsg (SRID): 26915
## proj4string: +proj=utm +zone=15 +datum=NAD83 +units=m +no_defs
## ROUTE_NUM DIST_MILES DIST_KM geometry
##   1   I10   2449.12  3941.48 MULTILINESTRING((-1881199.8...
##   2   I105  20.75   33.39 MULTILINESTRING((-1910155.9...
##   3   I110  41.42   66.65 MULTILINESTRING((1054138.60...
##   4   I115   1.58   2.55 MULTILINESTRING((-1013795.8...
##   5   I12   85.32  137.31 MULTILINESTRING((680741.744...
##   6   I124  1.73   2.79 MULTILINESTRING((1201467.26...
str(nc)
## Classes 'sf' and 'data.frame': 100 obs. of 9 variables:
## $ AREA : num 0.1118 0.0616 0.1402 0.0891 0.0687 ...
## $ PERIMETER : num 1.61 1.35 1.77 1.43 4.43 ...
## $ COUNTYP010: num 1994 1996 1998 1999 2000 ...
## $ STATE : chr "NC" "NC" "NC" "NC" ...
## $ COUNTY : chr "Ashe County" "Alleghany County" "Surry County" "Gates County" ...
## $ FIPS : chr "37009" "37005" "37171" "37073" ...
## $ STATE_FIPS: chr "37" "37" "37" "37" ...
## $ SQUARE_MIL: num 429 236 539 342 264 ...
## $ geometry : List of 100 , printing List of 1
## ..$ :List of 1
## .. ..$ : num [1:1030, 1:2] -81.7 -81.7 -81.7 -81.6 -81.6 ...
## ..- attr(*, "class")= chr "XY" "MULTIPOLYGON" "sfg"
## - attr(*, "sf_column")= chr "geometry"
## - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",..: NA NA NA NA NA NA NA NA
## ..- attr(*, "names")= chr "AREA" "PERIMETER" "COUNTYP010" "STATE" ...

class(nc)
## [1] "sf" "data.frame"

class(nc$geometry)
## [1] "sfc_MULTIPOLYGON" "sfc"

class(nc$geometry[[1]])
## [1] "XY" "MULTIPOLYGON" "sfg"
Projections

\texttt{st_crs(nc)$proj4string}
## [1] "+proj=longlat +datum=NAD83 +no_def\text{s}"

\texttt{st_crs(air)$proj4string}
## [1] "+proj=longlat +datum=NAD83 +no_def\text{s}"

\texttt{st_crs(hwy)$proj4string}
## [1] "+proj=utm +zone=15 +datum=NAD83 +units=m +no_def\text{s}"
UTM Zones
nc_ll = nc
air_ll = air
hwy_ll = st_transform(hwy, st_crs(nc)$proj4string)
nc_utm = st_transform(nc, st_crs(hwy)$proj4string)
air_utm = st_transform(air, st_crs(hwy)$proj4string)
hwy_utm = hwy
```r
sub = nc$COUNTY %in% c("Durham County","Wake County","Orange County")
nc[sub,]
```

```r
filter(nc, COUNTY %in% c("Durham County","Wake County","Orange County"))
```
Distance between NC counties

counties = c("Durham County", "Wake County", "Orange County")
sub = nc$COUNTY %in% counties

st_distance(nc_ll[sub, ])
## Error in st_distance(nc_ll[sub, ]): st_distance for longitude/latitude data only available for POINT geometries
st_distance(nc_utm[sub, ])
## Units: m
## [,1] [,2] [,3]
## [1,] 0.000 0 9906.327
## [2,] 0.000 0 0.000
## [3,] 9906.327 0 0.000
Distance between NC counties (centroids)

```r
nc_ll[sub, ] %>% st_centroid() %>% st_distance()
## Units: m
## [,1] [,2] [,3]
## [1,] 0.00 22185.58 52031.22
## [2,] 22185.58 0.00 34076.78
## [3,] 52031.22 34076.78 0.00

cn_utm[sub, ] %>% st_centroid() %>% st_distance()
## Units: m
## [,1] [,2] [,3]
## [1,] 0.00 22616.18 53050.15
## [2,] 22616.18 0.00 34751.60
## [3,] 53050.15 34751.60 0.00
```
Distance to the closest airport from each county?

```r
d = \texttt{st_distance}(\text{air\_utm}, \text{nc\_utm[sub,]})
d[1:5,]
## Units: m
## [,1]      [,2]      [,3]
## [1,] 846916.0  837771.1  836234.3
## [2,] 3122697.5 3146840.3 3172522.0
## [3,] 3556664.1 3584394.6 3592972.9
## [4,] 3514296.0 3540264.5 3545184.1
## [5,] 952881.7  954495.9  921201.2

\text{nearest\_airport} = \texttt{apply}(d, 2, \text{which.min})
\text{air} \%>\% \texttt{slice(\text{nearest\_airport})} \%>\% .\$\text{AIRPT\_NAME}
## [1] "RALEIGH-DURHAM INTERNATIONAL AIRPORT"
## [2] "RALEIGH-DURHAM INTERNATIONAL AIRPORT"
## [3] "RALEIGH-DURHAM INTERNATIONAL AIRPORT"
```
Geometry Predicates
<table>
<thead>
<tr>
<th></th>
<th>Interior</th>
<th>Boundary</th>
<th>Exterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interior</td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
<tr>
<td>$\dim[I(a) \cap I(b)]$</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Boundary</td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
<tr>
<td>$\dim[B(a) \cap I(b)]$</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Exterior</td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
<tr>
<td>$\dim[E(a) \cap I(b)]$</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
### Spatial Predicates

#### `st_within(a, b)`

<table>
<thead>
<tr>
<th><code>dim(a)</code></th>
<th><code>dim(b)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td></td>
</tr>
<tr>
<td>=</td>
<td></td>
</tr>
<tr>
<td>&gt;</td>
<td></td>
</tr>
</tbody>
</table>

#### `st_touches(a, b)`

<table>
<thead>
<tr>
<th><code>dim(a)</code></th>
<th><code>dim(b)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td></td>
</tr>
</tbody>
</table>

#### `st_crosses(a, b)`

<table>
<thead>
<tr>
<th><code>dim(a)</code></th>
<th><code>dim(b)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td></td>
</tr>
<tr>
<td>&gt;</td>
<td></td>
</tr>
</tbody>
</table>

#### `st_overlaps(a, b)`

<table>
<thead>
<tr>
<th><code>dim(a)</code></th>
<th><code>dim(b)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td></td>
</tr>
<tr>
<td>∈ {0, 2}</td>
<td></td>
</tr>
<tr>
<td>= 1</td>
<td></td>
</tr>
</tbody>
</table>
Sparse vs Full Results

```r
st_intersects(nc[20:30,], air) %>% str()
## List of 11
## $ : int(0)
## $ : int(0)
## $ : int(0)
## $ : int(0)
## $ : int(0)
## $ : int 268
## $ : int 717
## $ : int(0)
## $ : int(0)
## $ : int(0)
## $ : int(0)
```

```r
st_intersects(nc, air, sparse=FALSE) %>% str()
## logi [1:100, 1:940] FALSE FALSE FALSE FALSE FALSE FALSE ...
```
Which counties have airports?

```r
nc_air = st_intersects(nc, air)
has_air = map_lgl(nc_air, ~ length(.) > 0)
nc %>% slice(which(has_air)) %>% .$COUNTY
## [1] "Forsyth County"   "Guilford County"   "Dare County"
## [4] "Wake County"       "Pitt County"      "Catawba County"
## [7] "Buncombe County"   "Wayne County"     "Mecklenburg County"
## [10] "Moore County"      "Cabarrus County" "Lenoir County"
## [13] "Craven County"     "Cumberland County" "Onslow County"
## [16] "New Hanover County"
```

```r
air_in_nc = nc_air %>% unlist() %>% unique()
air %>% slice(air_in_nc) %>% .$AIRPT_NAME
## [1] "SMITH REYNOLDS AIRPORT"
## [2] "PIEDMONT TRIAD INTERNATIONAL AIRPORT"
## [3] "DARE COUNTY REGIONAL AIRPORT"
## [4] "RALEIGH-DURHAM INTERNATIONAL AIRPORT"
## [5] "PITT-GREENVILLE AIRPORT"
## [6] "HICKORY REGIONAL AIRPORT"
## [7] "ASHEVILLE REGIONAL AIRPORT"
## [8] "SEYMOUR JOHNSON AIR FORCE BASE"
## [9] "CHARLOTTE/DOUGLAS INTERNATIONAL AIRPORT"
## [10] "MOORE COUNTY AIRPORT"
```
plot(st_geometry(nc))
plot(st_geometry(nc[has_air,]), add=TRUE, col="lightblue")
plot(st_geometry(air[air_in_nc,]), add=TRUE, pch=16, col="blue")
Adjacency matrix of counties

```r
nc = nc[order(nc$COUNTY),]
adj = st_touches(nc, sparse=FALSE)

str(adj)
## logi [1:100, 1:100] FALSE FALSE FALSE FALSE FALSE FALSE ...

durham = which(nc$COUNTY == "Durham County")
nc %>% slice(which(adj[durham,])) %>% .$COUNTY
## [1] "Chatham County"  "Granville County"  "Orange County"
## [4] "Person County"  "Wake County"
```
library(corrplot)
rownames(adj) = str_replace(nc$COUNTY, " County", "")
colnames(adj) = str_replace(nc$COUNTY, " County", "")
corrplot(adj[1:20,1:20], method="color", type="full", tl.col="black", cl.pos = "n")
Which counties have the most neighbors?

```r
most_neighbors = rowSums(adj) == max(rowSums(adj))
plot(st_geometry(nc))
plot(st_geometry(nc[most_neighbors,]), add=TRUE, col="lightblue")
```

```r
nc$COUNTY[most_neighbors]
## [1] "Iredell County" "Moore County"
```
Which counties have the least neighbors?

least_neighbors = rowSums(adj)==min(rowSums(adj))
plot(st_geometry(nc))
plot(st_geometry(nc[least_neighbors,]), add=TRUE, col="lightblue")

nc$COUNTY[least_neighbors]
## [1] "Chowan County"  "Clay County"  "Currituck County"
## [4] "Dare County"    "New Hanover County" "Pamlico County"
## [7] "Polk County"    "Tyrrell County"