CHRISTOPHER PRENER, PH.D.
SPRING, 2018

WEEK 09
LECTURE 08C

INTRO TO GIS$_{C}$

ACCESSING SPATIAL DATA IN R

AGENDA

1. States & Counties
2. Census Geography
3. Other Features

1 STATES & COUNTIES

DOWNLOAD STATES DATA

f(x)

```
states(cb = FALSE, resolution = “res”)
```

Parameters:

- cb w to
  TRUE
- reso s are
  “500k”, “5m”, and “20m”. The “20m” data are the most generalized (least accurate).

Available in `tigris`
Download via CRAN

DOWNLOAD STATES DATA

f(x)

```
states(cb = FALSE, resolution = ‘‘res’’)
```

Parameters:

- `cb` will give us the most accurate spatial data if set to `FALSE`, if set to `TRUE` it will give us *generalized* data at one of three resolutions

- `resolution` is only needed if `cb = TRUE`; the three possible values are ‘‘`500k`’’, ‘‘`5m`’’, and ‘‘`20m`’’. The ‘‘`20m`’’ data are the most generalized (least accurate).

DOWNLOAD STATES DATA

f(x)

```
states(cb = FALSE, resolution = “res”)
```

Download state data at generalization of 1:500,000:

```
> states <- states(cb = TRUE, resolution = “500k”)
```

Remember, `resolution` is only needed if `cb = TRUE`!

CONVERT SP OBJECTS TO SF

f(x) `st_as_sf(spData)`

Parameters:

- *spda*

Available in `tigris`
Download via CRAN

CONVERT SP OBJECTS TO SF

f(x) `st_as_sf(`*spData*`)`

Parameters:

- *spdata* should be a sp spatial object

CONVERT SP OBJECTS TO SF

f(x)

```
st_as_sf(spData)
```

Download state data then convert to `sf`:

```
> states <- states(cb = TRUE, resolution = ‘‘500k’’)
> states <- st_as_sf(states)
```

If you are unsure if an object is sp, look at the global environment - the description will say “Formal class Spatial…”

DOWNLOAD COUNTIES DATA

f(x)

```
counties(state = ‘‘state’’, cb = FALSE, resolution = ‘‘res’’)
```

Parameters:

- All c the
  stat
  - s
    ‘‘Missouri’’ or ‘‘MO’’), or the numeric FIPS code value (e.g. 29)

Available in `tigris`
Download via CRAN

DOWNLOAD COUNTIES DATA

f(x) `counties(state = “state”, cb = FALSE, resolution = “res”)`

Parameters:

- All of the parameters for `counties()` are the same as they are for the `states()`, except for:
  - *state* is a string with a state name or USPS abbreviation (e.g. “Missouri” or “MO”), or the numeric FIPS code value (e.g. 29)

DOWNLOAD COUNTIES DATA

f(x)

```
counties(state = ‘‘state’’, cb = FALSE, resolution = ‘‘res’’)
```

Download Missouri county data at generalization of 1:2,000,000:

```
> moCounties <- counties(state = ‘‘Missouri’’,
    cb = TRUE, resolution = ‘‘2m’’)
```

Remember, `resolution` is only needed if `cb = TRUE`!

2 CENSUS GEOGRAPHY

DOWNLOAD CENSUS GEOGRAPHIES

f(x)

```
tracts(state = “state”, county = “county”, cb = FALSE,
    resolution = “res”)
```

Parameters:

- All c
  bloc ties()
  exce

Available in tigris
Download via CRAN

- *county* should be the name or FIPS code for a particular county

DOWNLOAD CENSUS GEOGRAPHIES

f(x)

```
tracts(state = “state”, county = “county”, cb = FALSE,
    resolution = “res”)
```

Parameters:

- All of the parameters for `tracts()` and its related functions - `block_groups()` and `blocks()` - are the same as they are for `counties()` except for:
  - *county* should be the name or FIPS code for a particular county

DOWNLOAD CENSUS GEOGRAPHIES

```
tracts(state = “state”, county = “county”, cb = FALSE,
    resolution = “res”)
```

Download St. Louis city tract data that are not generalized:

```
> stlTracts <- tracts(state = 29, county = 510)
```

I generally do not generalize sub-county spatial data.

3 OTHER FEATURES

OTHER FEATURES

- Bodies of water, polygons (by county) - `area_water()`
- Bodies of water, lines (by county) - `linear_water()`
- U.S. national boundary - `nation()`
- U.S. coastline - `coastline()`
- U.S. Congressional Districts - `congressional_districts()`
- Landmarks (by state) - `landmarks()`
- Native American reservations and trust areas - `native_areas()`

OTHER FEATURES

- Roads
  - Interstate and state highways - `primary_roads()`
  - Highways and major multi-lane roadways (by state) - `primary_secondary_roads()`
  - All roads (by county) - `roads()`
- Railways - `rails()`
- State legislative districts (by state) - `state_legislative_districts()`
- Urban areas - `urban_areas()`