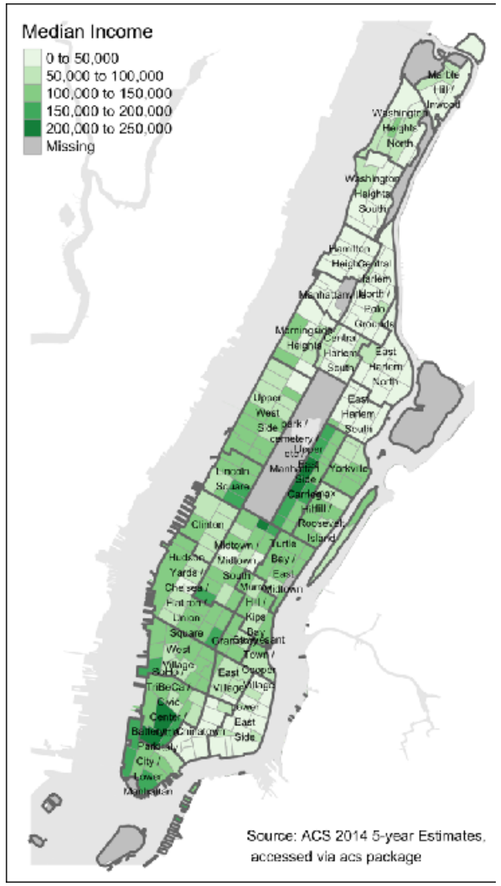
Working with Geospatial Data in R

Reading in spatial data

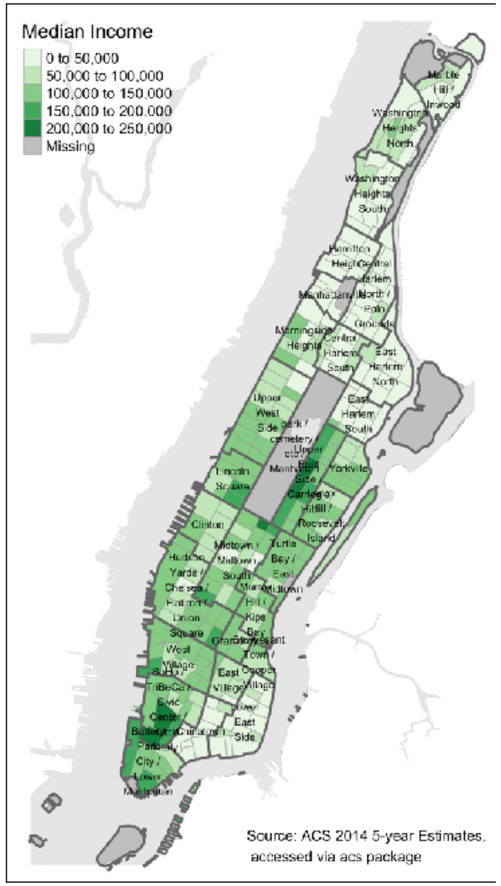
Median incomes in New York County

- Census tracts are areas with roughly the same number of people
- Spatial objects:
  - Census tract polygons
  - Larger neighborhood polygons
  - Areas of water polygons



Procedure

- Read in shape files that describe neighborhoods and waterways
- Match up two different coordinate systems
- Merge data from a data frame into a SpatialPolygonsDataFrame
- Polish a map to be publication ready



Reading in a shape file

- Vector data: data described by points, lines, polygons
- Shape file is the most common format

Reading in a shape file

```
> library(rgdal) # rgdal::readOGR() reads in vector formats
> library(sp)
> dir()
[1] "water"

> dir("water")
[1] "water-areas.dbf" "water-areas.prj" "water-areas.shp"
[4] "water-areas.shx"

> water <- readOGR("water", "water-areas")
OGR data source with driver: ESRI Shapefile
Source: "water", layer: "water-areas"
with 20 features
It has 5 fields
```

Checking the result

```
> summary(water)
Object of class SpatialPolygonsDataFrame
Coordinates:
         min       max
x -74.04731 -73.90866
y  40.68419  40.88207
Is projected: FALSE
…

> plot(water)
```

Reading in a raster files

```
> # rgdal::readGDAL() reads in raster formats to sp objects
> library(rgdal)

> # raster::raster() reads in raster formats to raster objects
> library(raster)

> dir()
[1] "usgrid_data_2000"   "usgrid_data_2000_1"

> dir("usgrid_data_2000")
[1] "metadata"                    "usarea00.tif"
[3] "usba00.tif"                  "usfb00.tif"
[5] "usgrid-2000-variables.xls" "usp2500.tif"
[7] "uspop300.tif"                "uspov00.tif"
[9] "uspvp00.tif"

> total_pop <- raster("usgrid_data_2000/uspop300.tif")
```

Checking the result

```
> total_pop

class       : RasterLayer
dimensions  : 3120, 7080, 22089600  (nrow, ncol, ncell)
resolution  : 0.008333333, 0.008333333  (x, y)
extent      : -125, -66, 24, 50  (xmin, xmax, ymin, ymax)
coord. ref. : +proj=longlat +datum=NAD83 +no_defs +ellps=GRS80
+towgs84=0,0,0
data source : /Users/wickhamc/Documents/Projects/courses-
visualizing-geospatial-data-in-r/data/census_grids/
usgrid_data_2000/uspop300.tif
names       : uspop300
values      : 0, 65535  (min, max)
```

Working with Geospatial Data in R

Let’s practice!

Working with Geospatial Data in R

Coordinate reference systems (CRS)

proj4string()

```
> proj4string(countries_spdf)
[1] "+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84
+towgs84=0,0,0"
```

Good for global datasets

```
> proj4string(water)
[1] "+proj=longlat +datum=NAD83 +no_defs +ellps=GRS80
+towgs84=0,0,0"
```

Common for United States datasets

```
> proj4string(nyc_tracts)
[1] "+proj=longlat +datum=NAD83 +no_defs +ellps=GRS80
+towgs84=0,0,0"

> proj4string(neighborhoods)
[1] "+proj=lcc +lat_1=40.66666666666666 +lat_2=41.03333333333333
+lat_0=40.16666666666666 +lon_0=-74 +x_0=300000 +y_0=0
+datum=NAD83 +units=us-ft +no_defs +ellps=GRS80 +towgs84=0,0,0"
```

Lambert conformal conic projection

Setting CRS

```
> x <- SpatialPoints(data.frame(-123.2620, 44.5646))
> x
class       : SpatialPoints
features    : 1
extent      : -123.262, -123.262, 44.5646, 44.5646  (xmin, xmax,
ymin, ymax)
coord. ref. : NA

> proj4string(x) <- "+proj=longlat +datum=WGS84 +no_defs
                     +ellps=WGS84 +towgs84=0,0,0"
> x
class       : SpatialPoints
features    : 1
extent      : -123.262, -123.262, 44.5646, 44.5646  (xmin, xmax,
ymin, ymax)
coord. ref. : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84
+towgs84=0,0,0
```

Transforming CRS

```
rgdal::spTransform(x, CRSobj, ...)
```

```
> spTransform(x, "+proj=lcc +lat_1=40.66666666666666
                  +lat_2=41.03333333333333 +lat_0=40.16666666666666
                  +lon_0=-74 +x_0=300000 +y_0=0 +datum=NAD83
                  +units=us-ft +no_defs +ellps=GRS80 +towgs84=0,0,0")
```

Transforming CRS

```
rgdal::spTransform(x, CRSobj, ...)
```

```
> spTransform(x, proj4string(neighborhoods))



class        : SpatialPoints
features     : 1
extent       : -11214982, -11214982, 5127323, 5127323  (xmin, xmax,
ymin, ymax)
coord. ref.  : +proj=lcc +lat_1=40.66666666666666
               +lat_2=41.03333333333333 +lat_0=40.16666666666666
               +lon_0=-74 +x_0=300000 +y_0=0 +datum=NAD83
               +units=us-ft +no_defs +ellps=GRS80 +towgs84=0,0,0
```

Working with Geospatial Data in R

Let’s practice!

Working with Geospatial Data in R

Adding data to spatial objects

Income data from ACS

```
> str(nyc_income)
'data.frame':288 obs. of  6 variables:
 $ name    : chr  "Census Tract 1, New York County, New York"
"Census Tract 2.01, New York County, New York" "Census Tract 2.02,
New York County, New York" "Census Tract 5, New York County, New
York" ...
 $ state   : int  36 36 36 36 36 36 36 36 36 36 ...
 $ county  : int  61 61 61 61 61 61 61 61 61 61 ...
 $ tract   : chr  "000100" "000201" "000202" "000500" ...
 $ estimate: num  NA 23036 29418 NA 18944 ...
 $ se      : num  NA 3083 1877 NA 1442 ...
```

estimate of median income in this tract

standard error of estimate

Tract polygon data

```
> str(nyc_tracts, max.level = 2)
Formal class 'SpatialPolygonsDataFrame' [package "sp"] with 5
slots
  ..@ data       :'data.frame': 288 obs. of  9 variables:
  ..@ polygons   :List of 288
  .. .. [list output truncated]
  ..@ plotOrder  : int [1:288] 175 225 97 209 249 232 208 247 244
207 ...
  ..@ bbox       : num [1:2, 1:2] -74 40.7 -73.9 40.9
  .. ..- attr(*, "dimnames")=List of 2
  ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
```

Tract polygon data

```
> str(nyc_tracts@data)
'data.frame':288 obs. of  9 variables:
 $ STATEFP : chr  "36" "36" "36" "36" ...
 $ COUNTYFP: chr  "061" "061" "061" "061" ...
 $ TRACTCE : chr  "001401" "002201" "003200" "004000" ...
 $ AFFGEOID: chr  "1400000US36061001401" "1400000US36061002201"
"1400000US36061003200" "1400000US36061004000" ...
 $ GEOID   : chr  "36061001401" "36061002201" "36061003200"
"36061004000" ...
 $ NAME    : chr  "14.01" "22.01" "32" "40" ...
 $ LSAD    : chr  "CT" "CT" "CT" "CT" ...
 $ ALAND   : num  93510 161667 217682 178340 124447 ...
 $ AWATER  : num  0 0 0 0 0 0 0 0 0 0 ...
```

- Goal: Add the estimated median income to this data frame

Correspondence between polygons and data

```
> four_tracts
class       : SpatialPolygons
features    : 4
extent      : -73.99022, -73.97875, 40.71413, 40.73329  (xmin,
xmax, ymin, ymax)
coord. ref. : +proj=longlat +datum=NAD83 +no_defs +ellps=GRS80
+towgs84=0,0,0

> sapply(four_tracts@polygons, function(x) x@ID)
[1] "156" "157" "158" "159"

> four_data
    TRACTCE
159  004000
158  003200
157  002201
156  001401
```

Correspondence between polygons and data

```
SpatialPolygonsDataFrame(Sr, data, match.ID = TRUE)
```

```
> SpatialPolygonsDataFrame(four_tracts, four_data)

class       : SpatialPolygonsDataFrame
features    : 4
extent      : -73.99022, -73.97875, 40.71413, 40.73329  (xmin,
xmax, ymin, ymax)
coord. ref. : +proj=longlat +datum=NAD83 +no_defs +ellps=GRS80
+towgs84=0,0,0
variables   : 1
names       : TRACTCE
min values  :  001401
max values  :  004000
```

Correspondence between polygons and data

```
SpatialPolygonsDataFrame(Sr, data, match.ID = TRUE)
```

```
> SpatialPolygonsDataFrame(four_tracts, four_data)@data


    TRACTCE
156  001401
157  002201
158  003200
159  004000
```

Correspondence between polygons and data

```
SpatialPolygonsDataFrame(Sr, data, match.ID = TRUE)
```

```
> SpatialPolygonsDataFrame(four_tracts, four_data,
                           match.ID = FALSE)@data

    TRACTCE
159  004000
158  003200
157  002201
156  001401
```

correspondence is now lost!

Adding new data

- Once created, no checks that data stay matched to polygons
- Recreate object being very careful to match polygons to the right rows
- `sp::merge()`, `merge()` for `sp` objects

Working with Geospatial Data in R

Let’s practice!

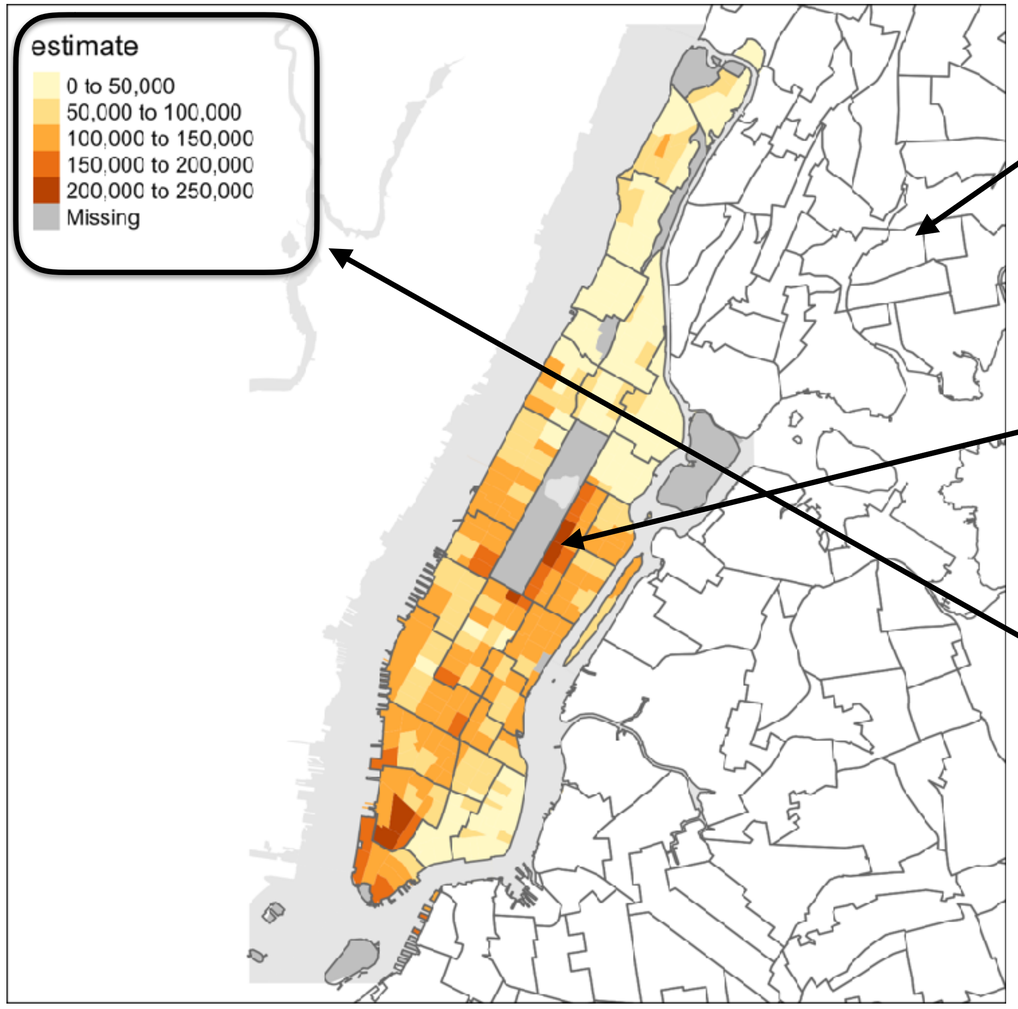
Working with Geospatial Data in R

Polishing a map

Polishing a map

- Remove distractions, let data shine
- **Useful** spatial context
- Like any plot: check legend, title, and labels for readability
- Add annotations:
  - highlight important points
  - attribute data sources

Critiquing our map



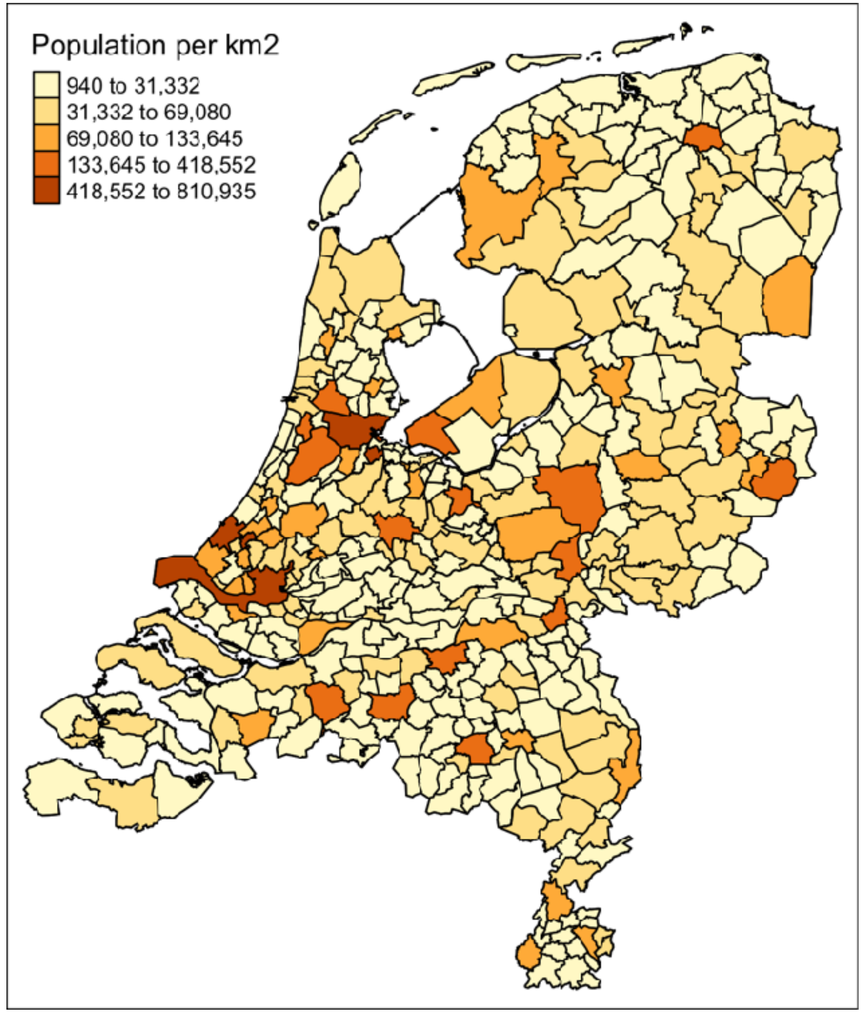
Extraneous neighborhoods are distracting

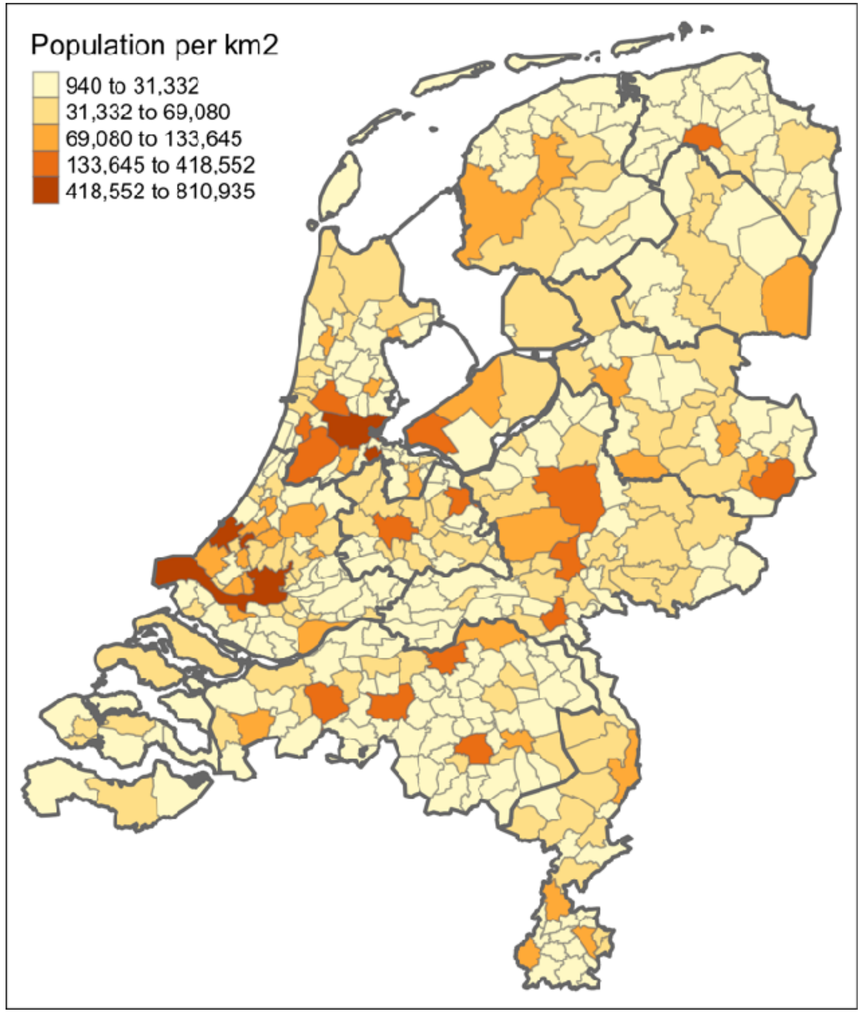
Neighborhoods should be labelled

Legend needs better title

Experiment with colors and line weights

The effect of line weights and color





Working with Geospatial Data in R

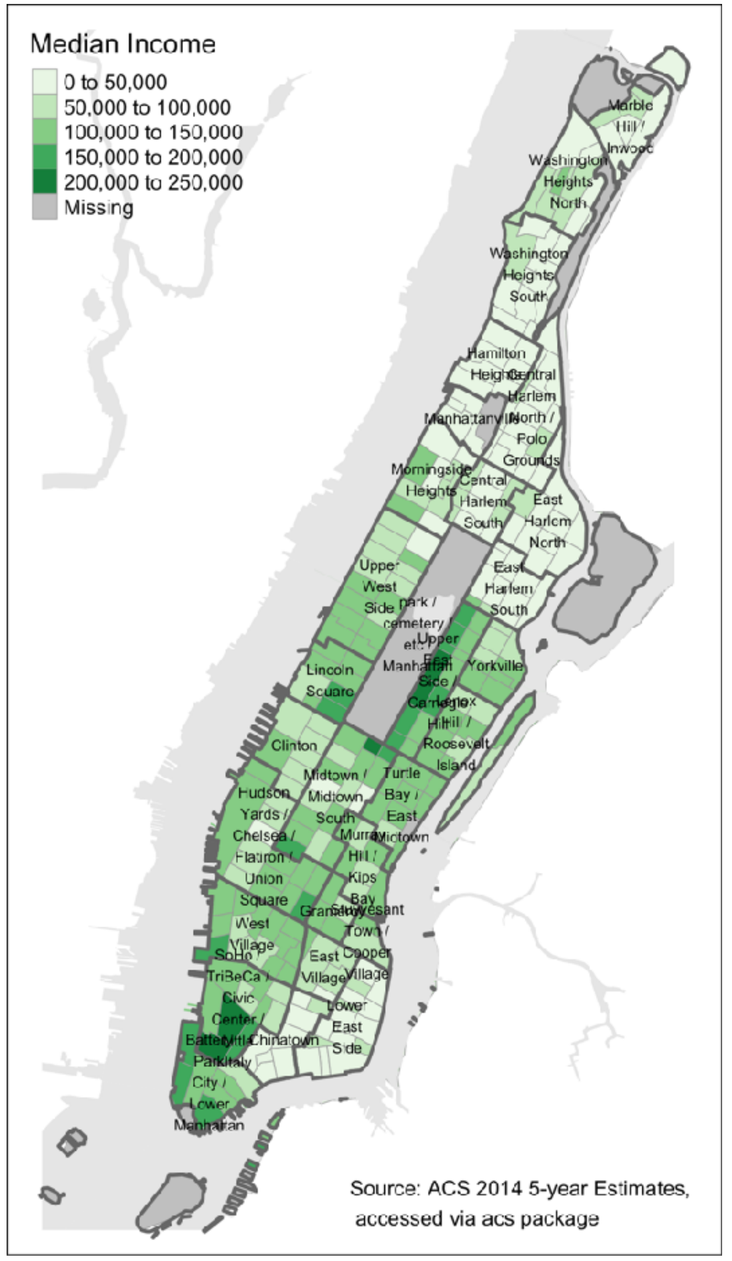
Let’s practice!

Working with Geospatial Data in R

Wrap up

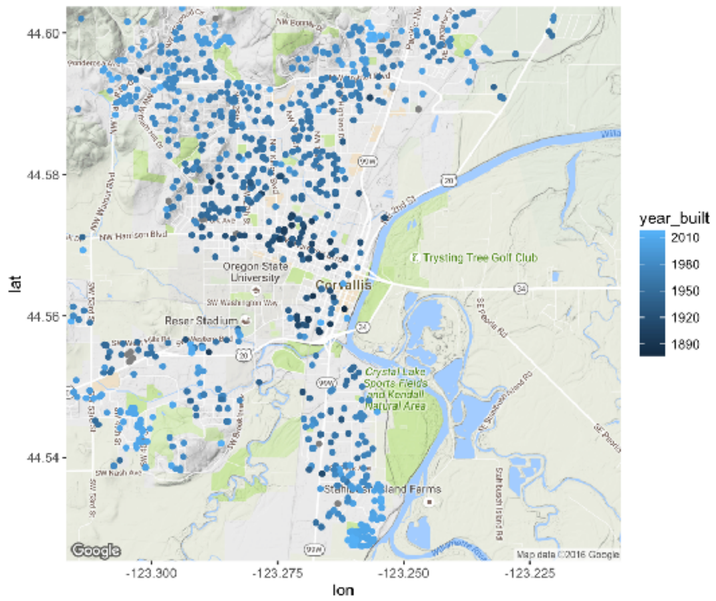
Final tweaks

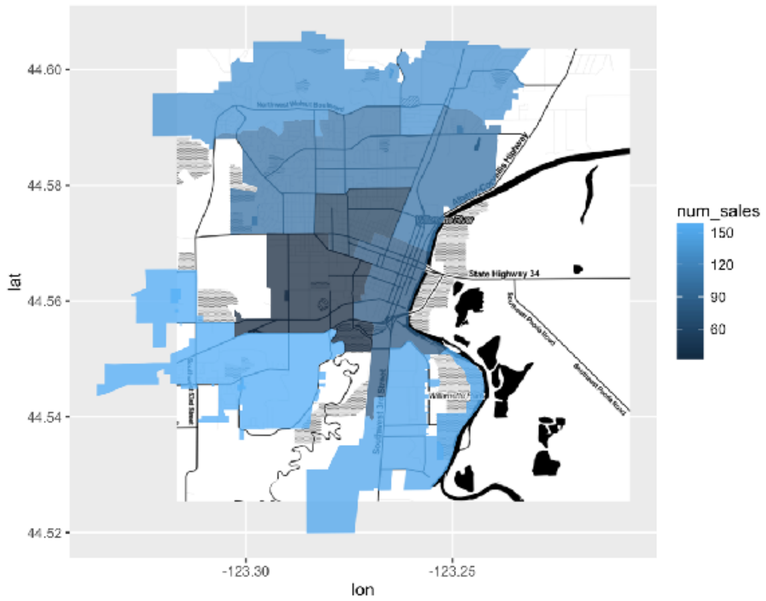
- Tweak labels "by hand"
- Add $ to legend
- Remove tiny areas

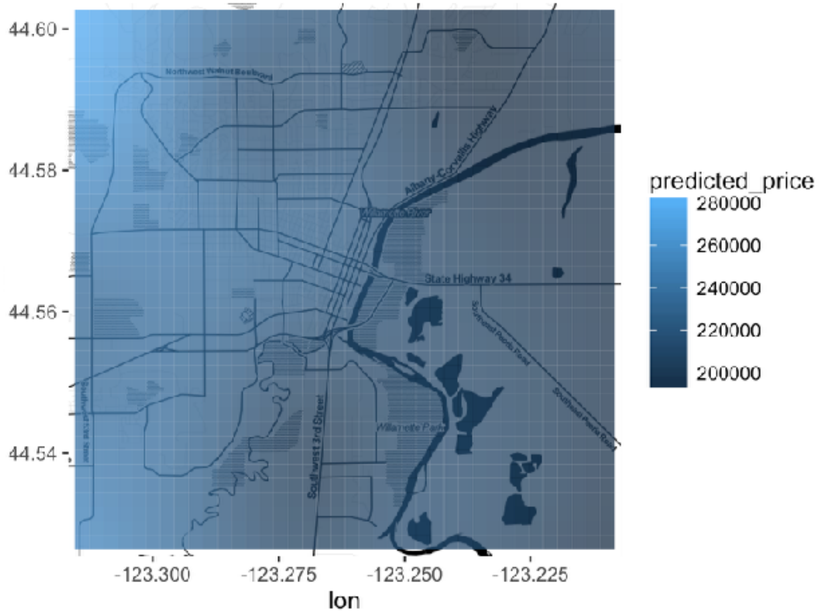


Chapter 1

- Types of spatial data: point, line, polygon and raster
- Adding context using ggmap

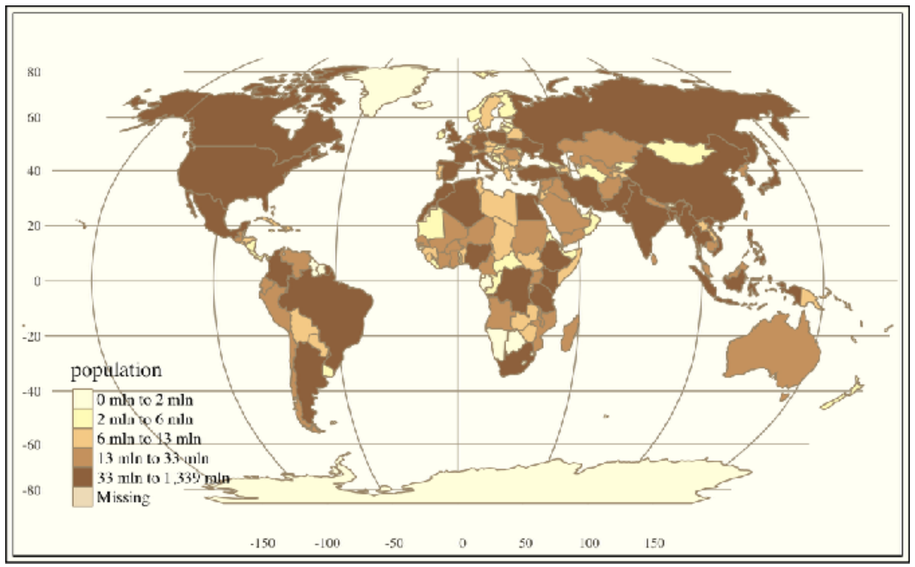


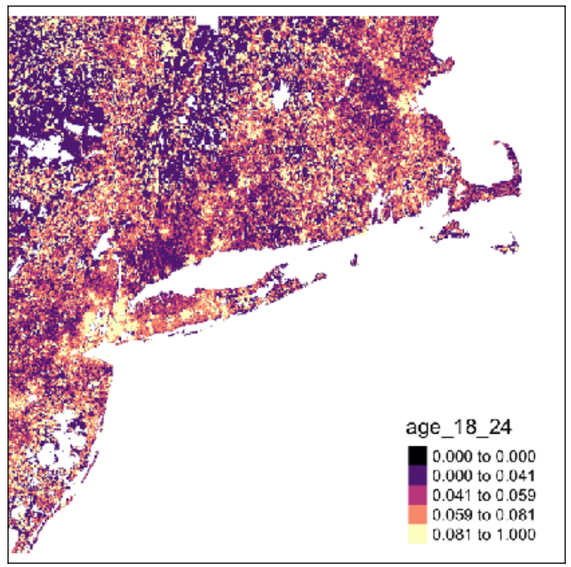




Chapters 2 & 3

- Spatial classes provided by `sp` and `raster`
- S4 objects
- `tmap` for displaying spatial data





Chapter 4

- Reading in spatial data
- Transforming coordinate systems
- Adding data to `Spatial___DataFrame` objects
- Polishing a map

Thank you!