Working with Geospatial Data in R

Introduction to spatial data

What is spatial data?

- Data is associated with locations
- Location described by coordinates + a coordinate reference system (CRS)
- Common CRS: longitude, latitude describes locations on the surface of the Earth

House sales in Corvallis

1112 NW 26TH ST,
CORVALLIS, OR

latitude	longitude
44.57808 N	123.2803 W

-123.2803, 44.57808

House sales in a data frame

location

data associated with this location

```
> head(sales)
        lon      lat  price    bedrooms  full_baths
1 -123.2803 44.57808 267500           5           2
2 -123.2330 44.59718 255000           3           2
3 -123.2635 44.56923 295000 ...       3           2  ...
4 -123.2599 44.59453   5000           0           1
5 -123.2632 44.53606  13950           0           2
6 -123.2847 44.59877 233000           3           2

> nrow(sales)
[1] 931
```

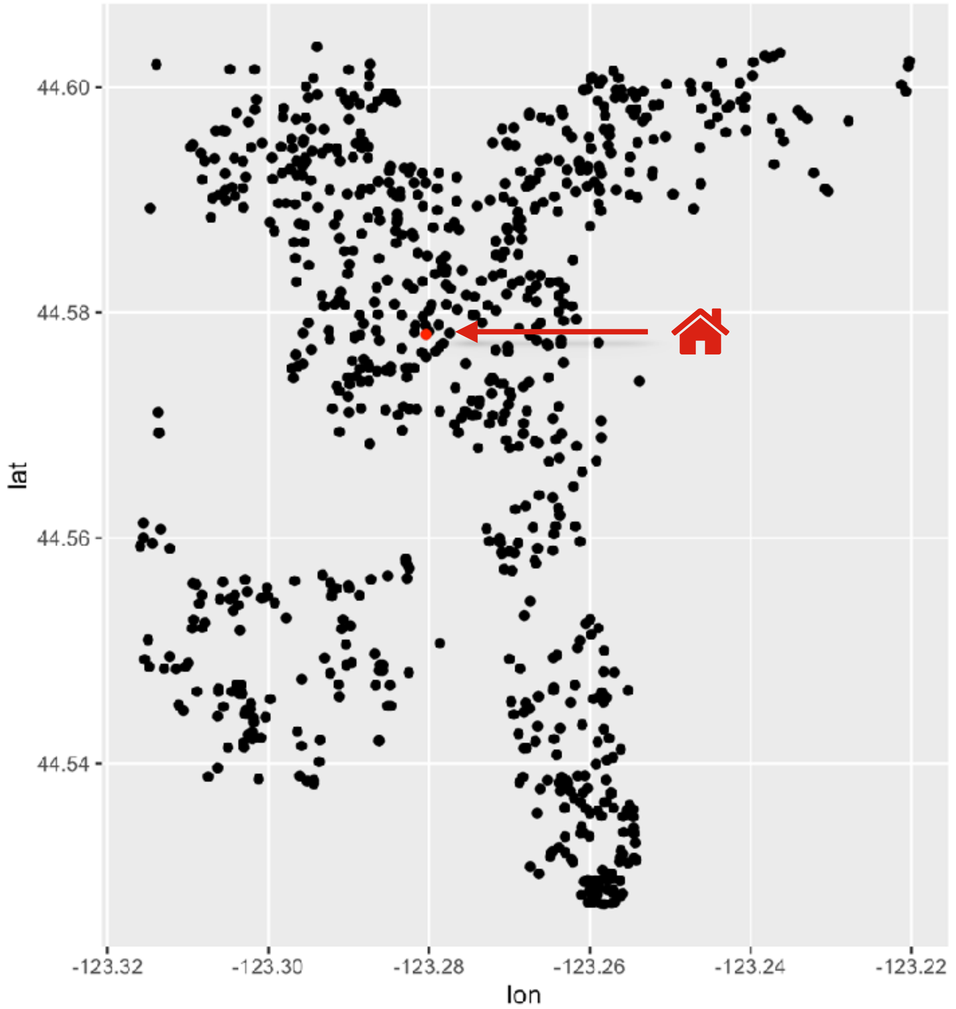
- **Point data**: locations are points, described by a single pair of coordinates

Displaying spatial data with ggplot2

```
> library(ggplot2)

> ggplot(sales, aes(lon, lat)) +
    geom_point()
```

- Adding some location cues would be helpful



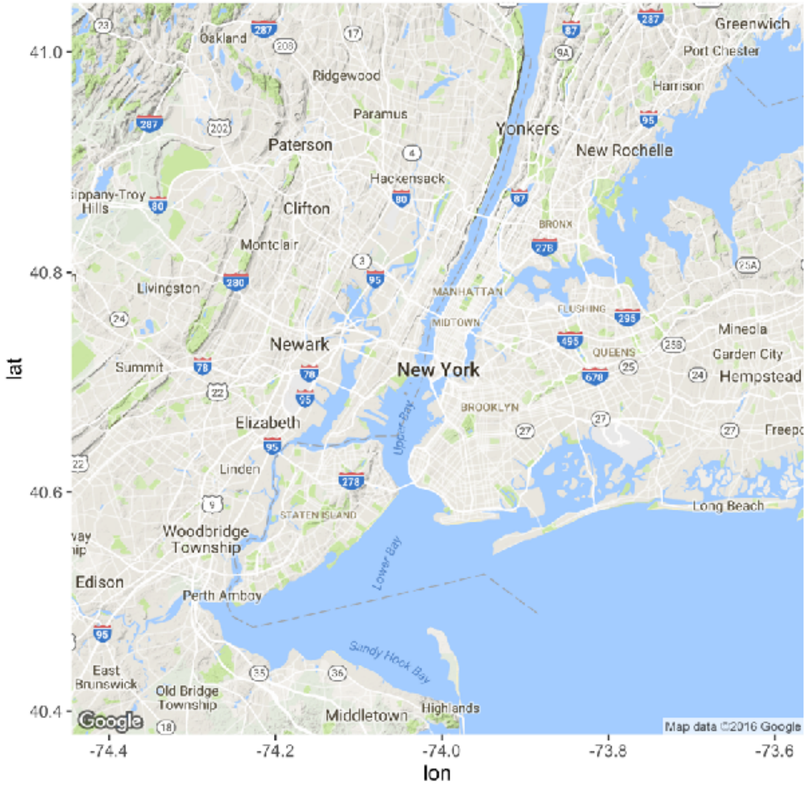
The ggmap package

```
> library(ggmap)

> # Coordinates for the location of interest
> nyc <- c(lon = -74.0059, lat = 40.7128)

> # 1. Download the relevant map
> nyc_map <- get_map(location = nyc, zoom = 10)

> # 2. Display the map
> ggmap(nyc_map)
```



Working with Geospatial Data in R

Let’s practice!

Useful `get_map()` and `ggmap()` options

Changing the map image

```
> library(ggmap)
> corvallis <- c(lon = -123.2620, lat = 44.5646)
> corvallis_map <- get_map(corvallis, zoom = 13, scale = 1)

Map from URL : http://maps.googleapis.com/maps/api/staticmap?
center=44.5646,-123.262&zoom=13&size=640x640&scale=1&
maptype=terrain&language=en-EN&sensor=false
```

- By default, `get_map()`, downloads a terrain image from Google maps

```
> corvallis_map <- get_map(corvallis, zoom = 13,
                           maptype = "terrain",
                           source = "google")
```

Other map image sources

```
> ?get_map
```

```
…
 maptype = c("terrain", "terrain-background",
             "satellite", "roadmap", "hybrid",
             "toner", "watercolor", "terrain-labels",
             "terrain-lines", "toner-2010", "toner-2011",
             "toner-background", "toner-hybrid", "toner-labels",
             "toner-lines", "toner-lite"),
 source = c("google", "osm", "stamen"),
…
```

```
> corvallis_map <- get_map(corvallis, zoom = 13,
                           maptype = "toner-2010",
                           source = "stamen")
```

Specifying default data and aesthetics

```
> ?ggmap
```

```
ggmap(ggmap, extent = "panel", base_layer, maprange = FALSE,
  legend = "right", padding = 0.02,
  darken = c(0, "black"), ...)
```

```
> ggmap(corvallis_map,
      base_layer = ggplot(sales, aes(lon, lat))) +
    geom_point() +
    facet_wrap(~ condition)
```

Changing the way the map is plotted

```
> ?ggmap
```

```
ggmap(ggmap, extent = "panel", base_layer, maprange = FALSE,
  legend = "right", padding = 0.02,
  darken = c(0, "black"), ...)
```

- `extent`: how much of the plotting area should the map take up?
- `maprange`: should the plot limits come from the map limits?

Working with Geospatial Data in R

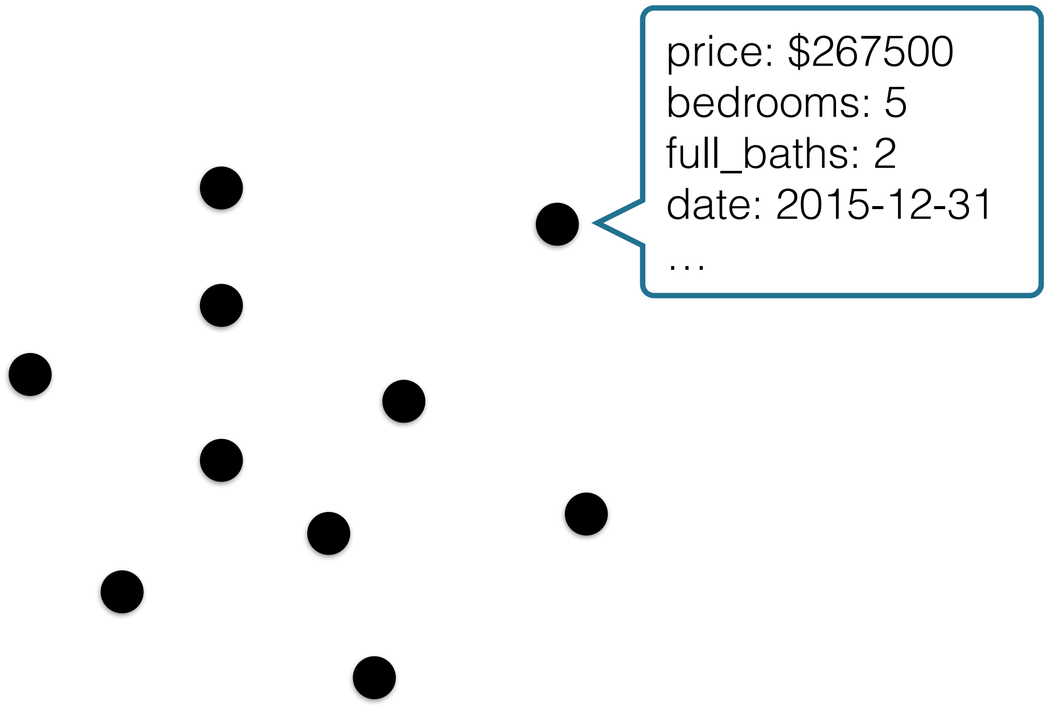
Let’s practice!

Working with Geospatial Data in R

Common types of spatial data

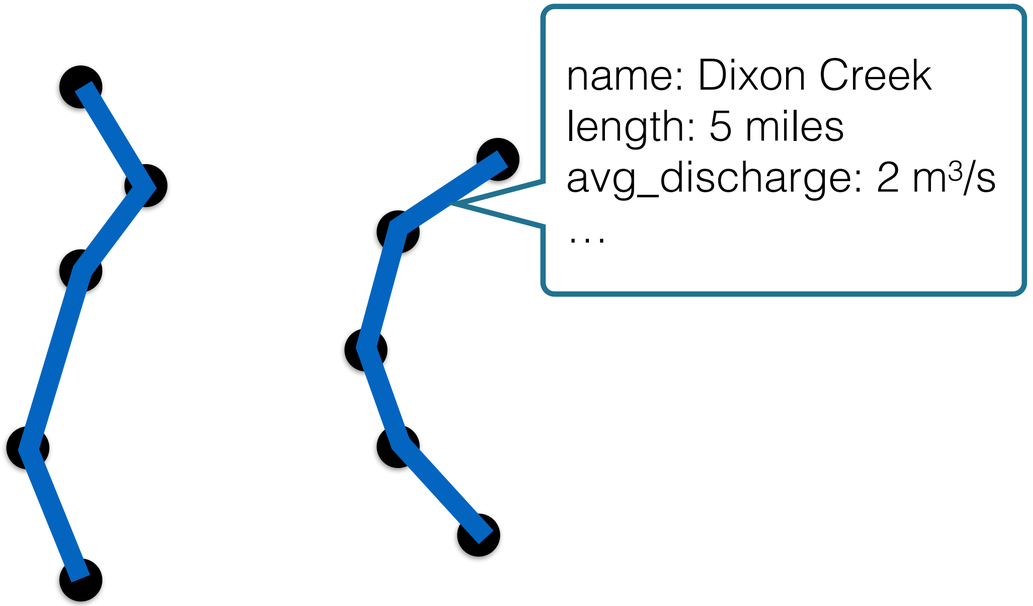
Types of spatial data

- Point



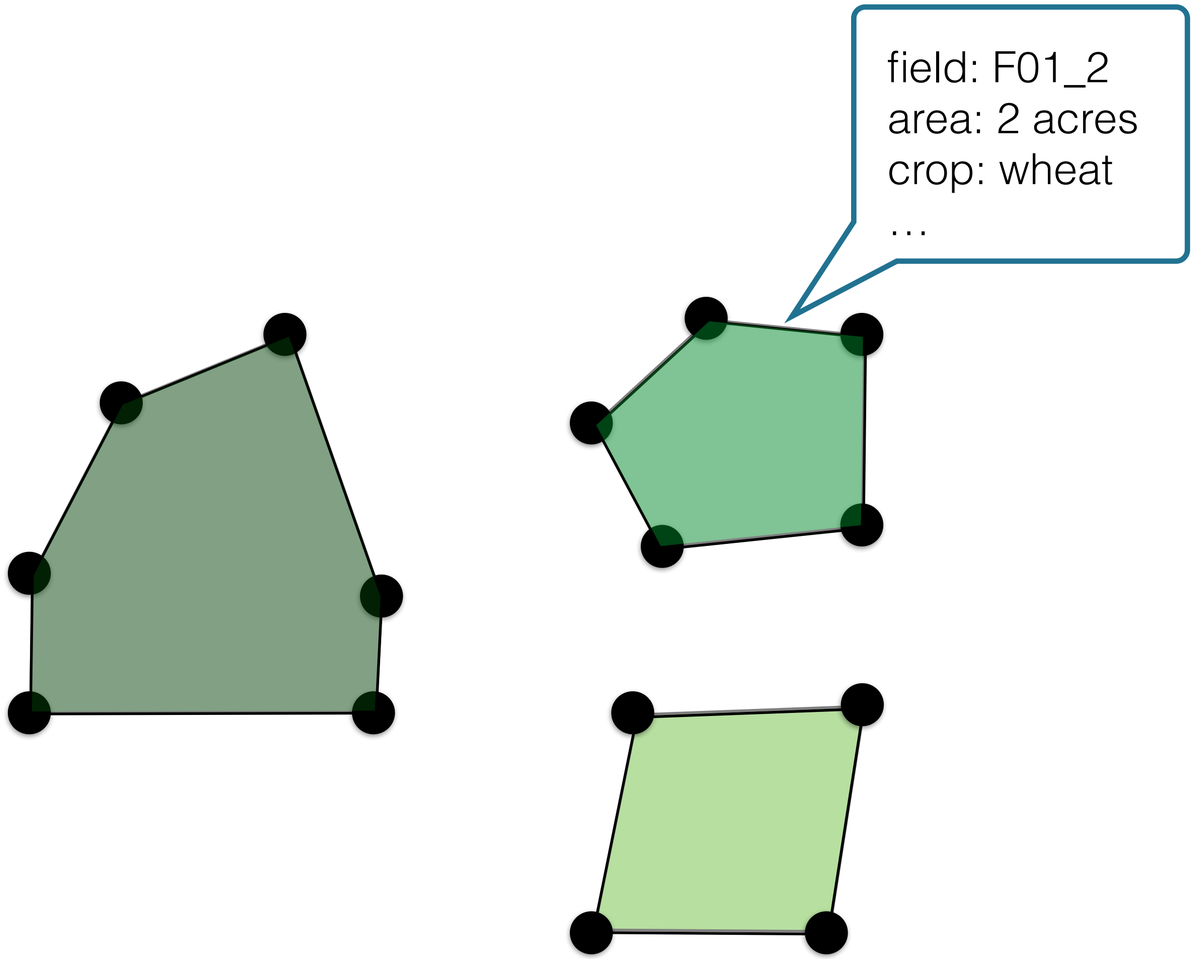
Types of spatial data

- Point
- Line



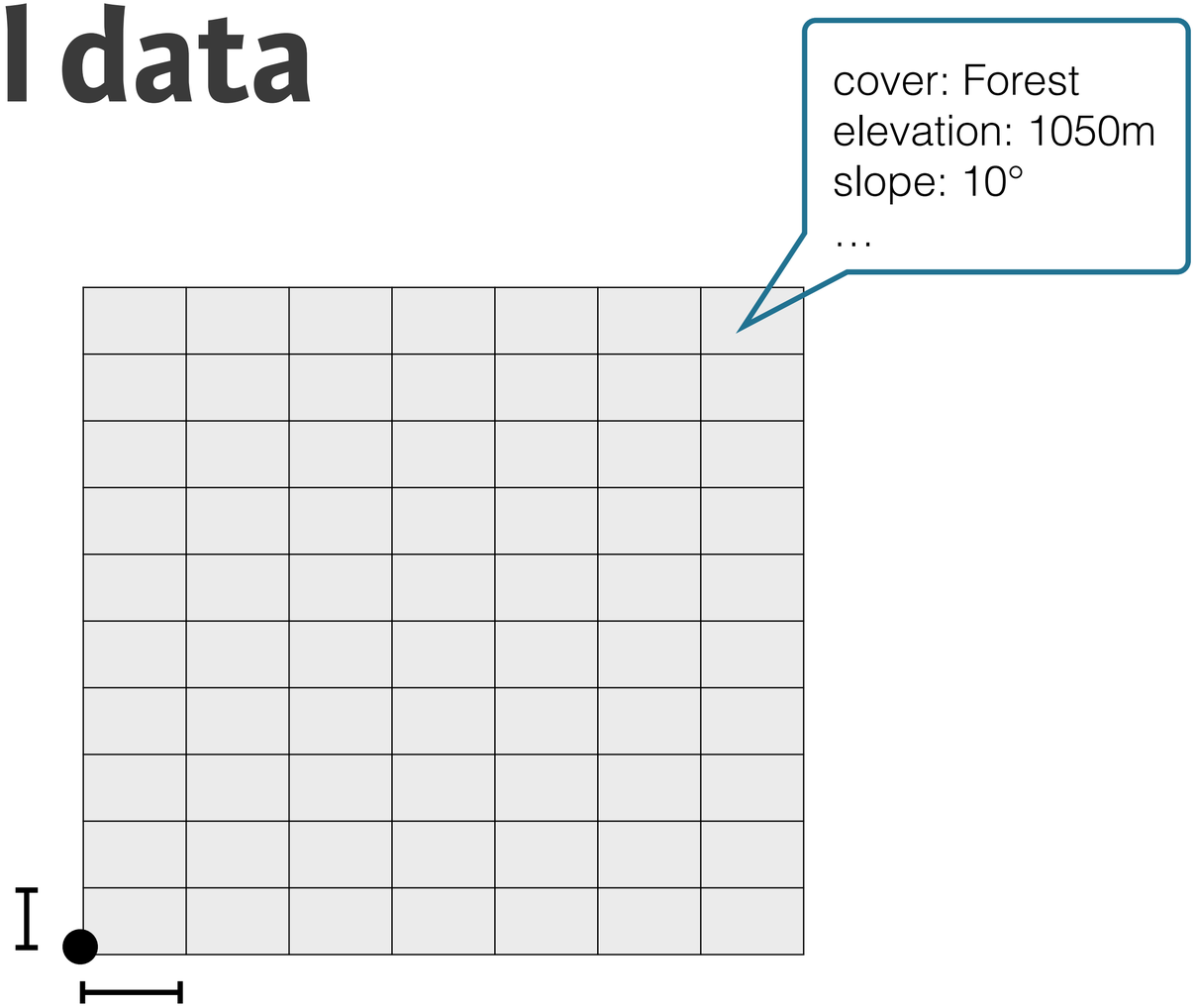
Types of spatial data

- Point
- Line
- Polygon



Types of spatial data

- Point
- Line
- Polygon
- Raster (a.k.a Gridded)



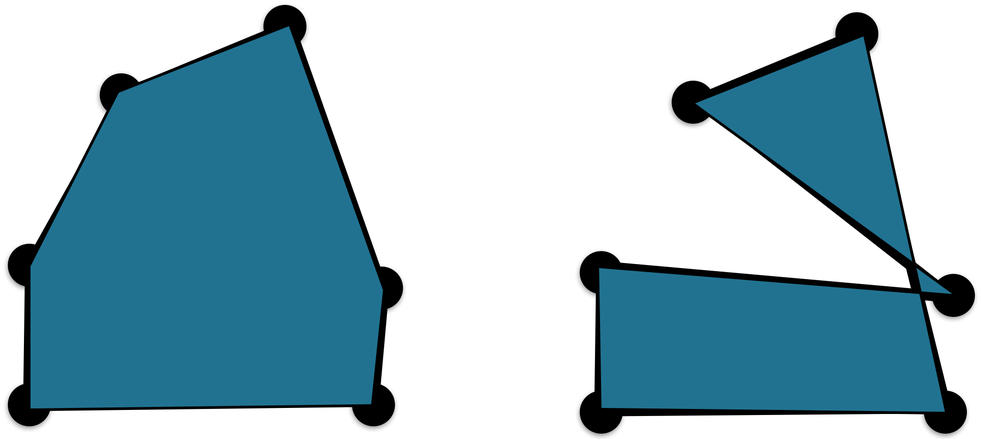
House prices by ward

- Wards are areas that have roughly equal numbers of people
- Can be described by polygons

```
> head(ward_sales)
  ward       lon      lat group order num_sales avg_price
1    1 -123.3128 44.56531   0.1     1       159  311626.9
2    1 -123.3122 44.56531   0.1     2       159  311626.9
3    1 -123.3121 44.56531   0.1     3       159  311626.9
4    1 -123.3119 44.56531   0.1     4       159  311626.9
5    1 -123.3119 44.56485   0.1     5       159  311626.9
6    1 -123.3119 44.56430   0.1     6       159  311626.9
```

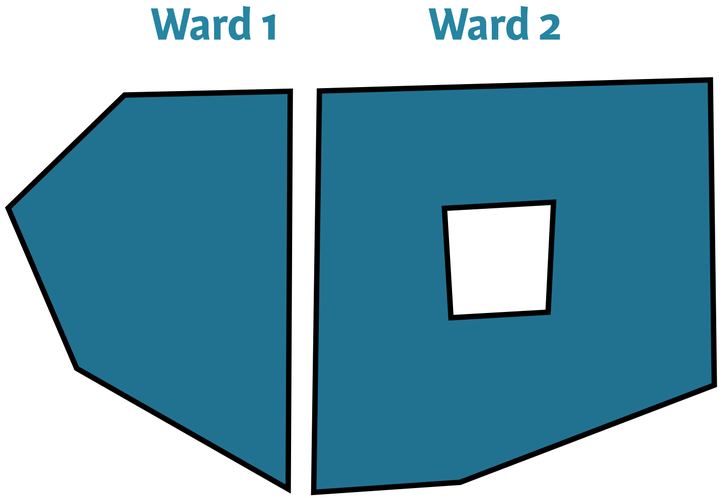
Drawing polygons is tricky

- Order matters



Drawing polygons is tricky

- Order matters
- Some areas may need more than one polygon



Predicted house prices

```
> head(preds)
        lon      lat predicted_price
1 -123.3168 44.52539        258936.2
2 -123.3168 44.52740        257258.4
3 -123.3168 44.52940        255543.1
4 -123.3168 44.53141        253791.0
5 -123.3168 44.53342        252002.4
6 -123.3168 44.53542        250178.7
```

Working with Geospatial Data in R

Let’s practice!