"Learning GIS with ice-cream parlors"

Open Source in GIS conference • June 9th, 2021

Amit Levinson •

About me

- Sociology MA Graduate, BGU
- Been using R Programming language for < 2 years
- Had no idea what is GIS prior to using R
- Enjoy:
  - Open source tools (R, Linux, Git, MySQL)
  - Blogging
  - Learning new things
  - 🍪 & 🥛

"Regular" data

How I first encountered and learn to work with data:

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Spatial data*

How I encountered spatial data (e.g. `.shp`):

```
head(shelters[,2])
```

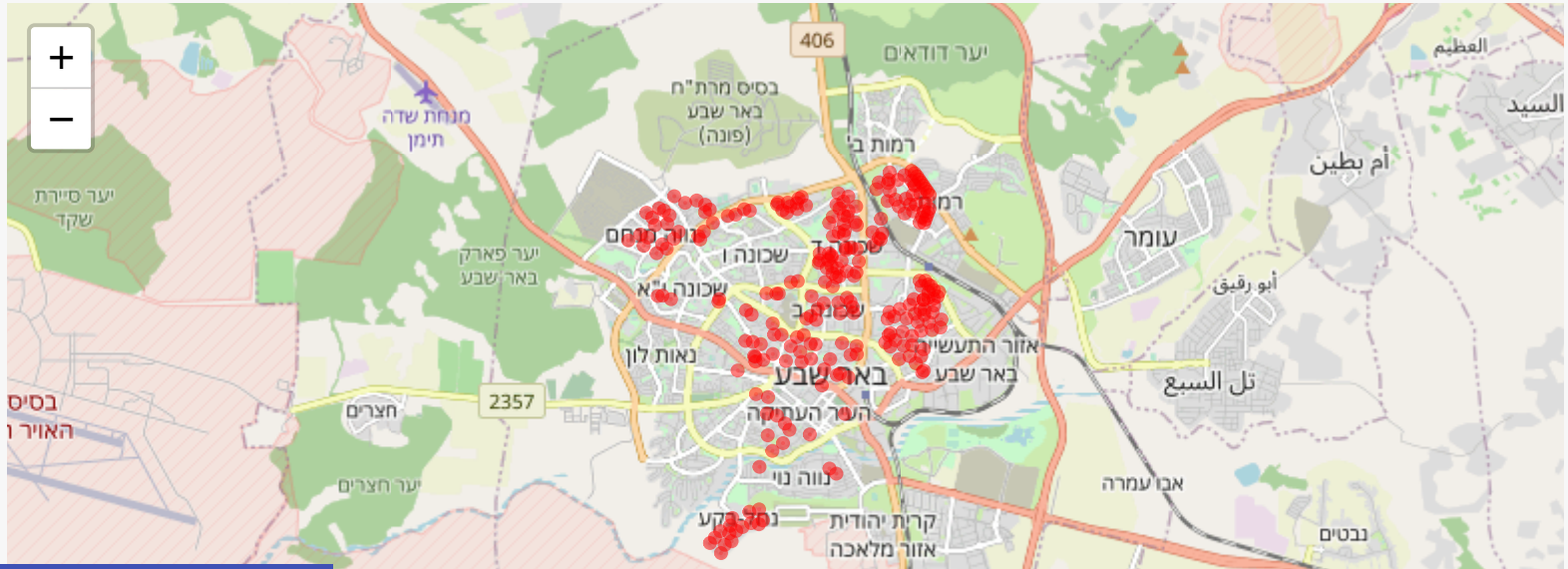
```
## Simple feature collection with 6 features and 0 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 181680.1 ymin: 574027.4 xmax: 181972.9 ymax: 574304
## Projected CRS: Israel 1993 / Israeli TM Grid
## # A tibble: 6 x 1
##              geometry
##           <POINT [m]>
## 1 (181710.6 574217.8)
## 2   (181680.1 574304)
## 3   (181820.6 574243)
## 4 (181819.3 574027.4)
## 5 (181972.9 574046.2)
## 6 (181912.3 574151.1)
```

* To reflect my learning process, I address spatial data here as in a vector format (and not, e.g., raster)

Returning to my comfort zone

Used a `.csv` file instead:

```
## # A tibble: 6 x 2
##     lat   lon
##   <dbl> <dbl>
## 1  31.3  34.8
## 2  31.3  34.8
## 3  31.3  34.8
## 4  31.3  34.8
## 5  31.3  34.8
## 6  31.3  34.8
```



Exploring distances to the nearest ice-cream parlors

Background

- Fiddled around but never really knew what I was doing
- Participated in several days of the #30Daysmapchallenge
- Realized it's time to learn some spatial analysis

Inspiration

Dominic Royé's blog post about distances to the sea in Iceland

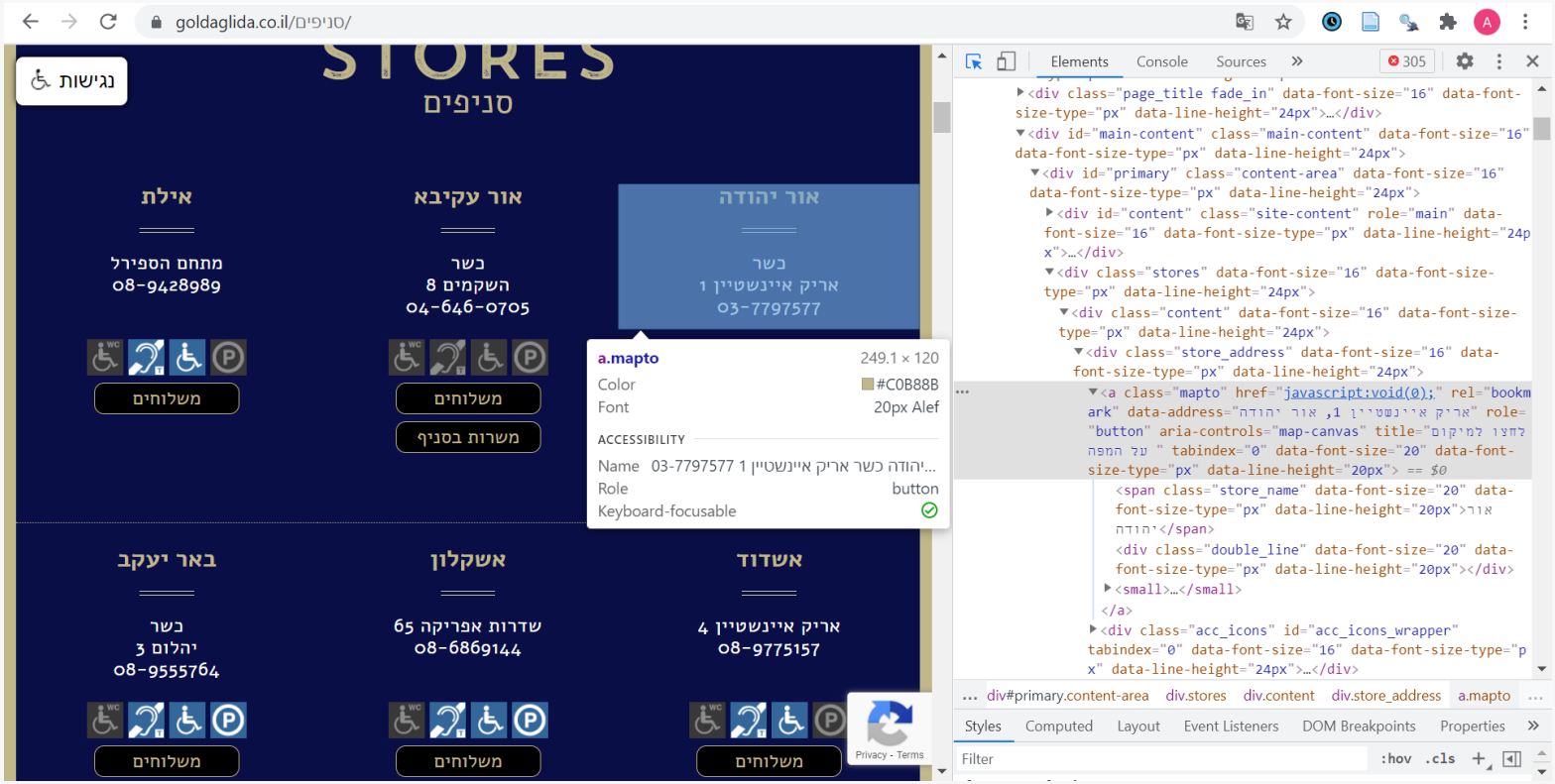
Serendipitous resource

Michael Dorman's Spatial data analysis workshop at Israel's CBS

The talk is based on an earlier blog post on the topic.

Data

Where do you find ice-cream locations data?



Data collection

- Scrape the data from the website

```
## [1] "\nאור עקיבא\n\nכשרות\n8 השקמים\n04-646-0705\n\n"
```

- Clean it
- Geocode the data to get lat & long from addresses
- Reverse geocode the long & lat and verify each address.

Final output looked like this (total of 79 addresses):

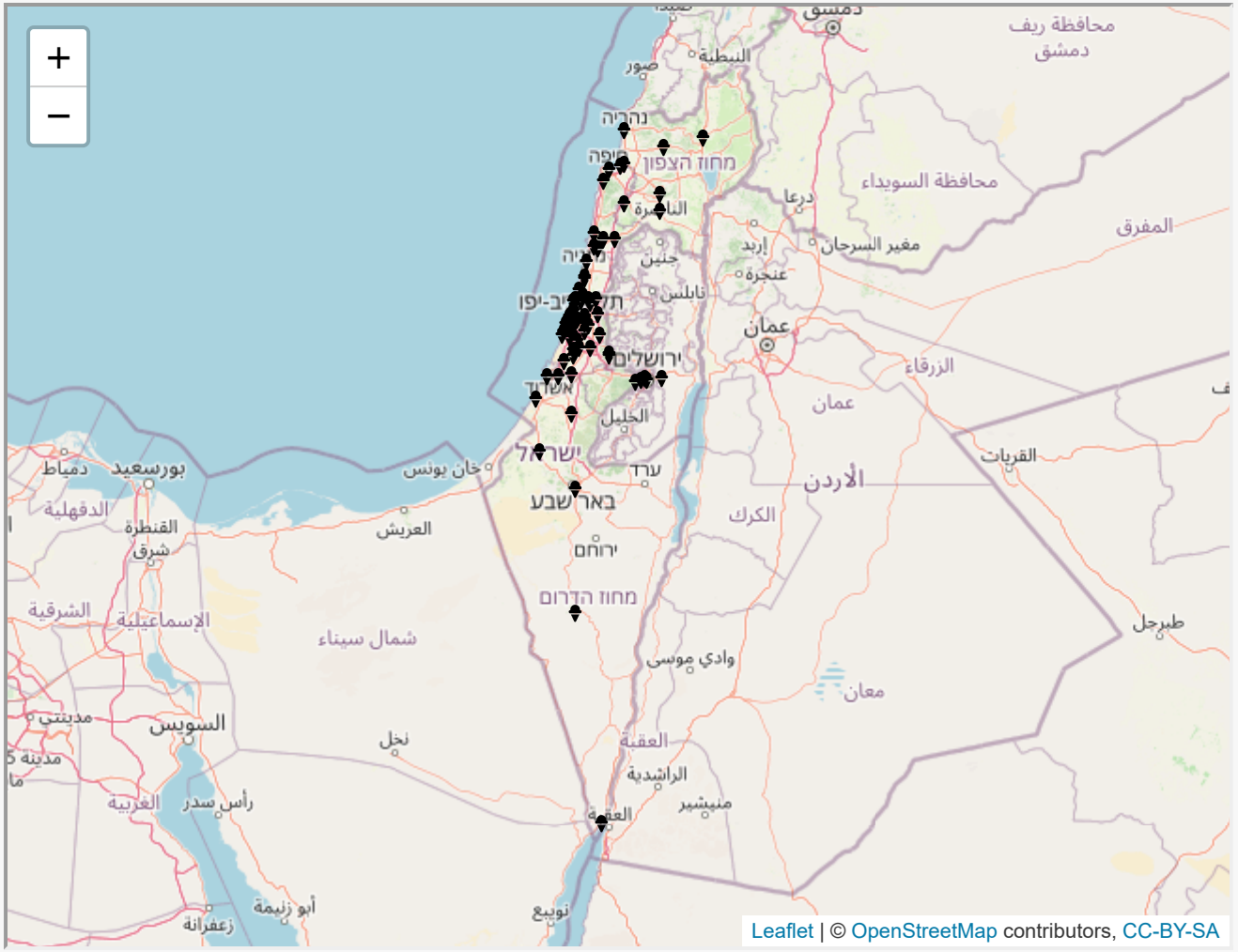
city	street	number	location	lon	lat	google_address
אור עקיבא	השקמים 8	04-646-0705	השקמים 8, אור עקיבא	34.91794	32.50436	Ha-Shikmim St 8, Or Akiva, Israel

Discrepancies were manually verified (though I probably verified everything anyway).


This is the worst day of my life.


This is the worst day of my life.
The worst day of your life so far.

How does our data look?



If you can't see the iframe, please refresh the page.

Some spatial manipulations

Transforming

We might want to be more explicit with our spatial data:

Geographic

```
## Simple feature collection with 1 feature and 0 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 34.86002 ymin: 32.02051 xmax: 34.86002 ymax: 32.02051
## Geodetic CRS:  WGS 84
## # A tibble: 1 x 1
##              geometry
##           <POINT [°]>
## 1 (34.86002 32.02051)
```

Projected

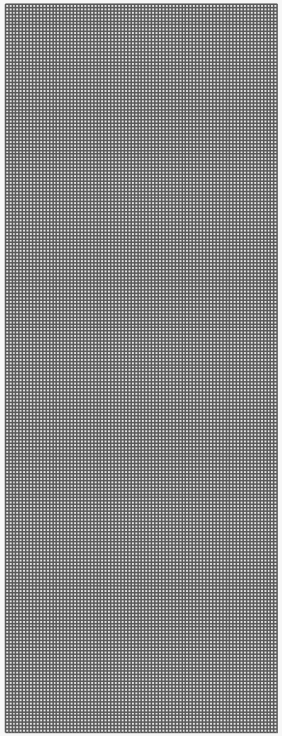
```
## Simple feature collection with 1 feature and 0 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 186911.1 ymin: 658636.9 xmax: 186911.1 ymax: 658636.9
## Projected CRS: Israel 1993 / Israeli TM Grid
## # A tibble: 1 x 1
##              geometry
##           <POINT [m]>
```

Grids

We want to calculate distances to Golda, but **from where?**

Original map

Grid layout (2km^2)



Israel as grid

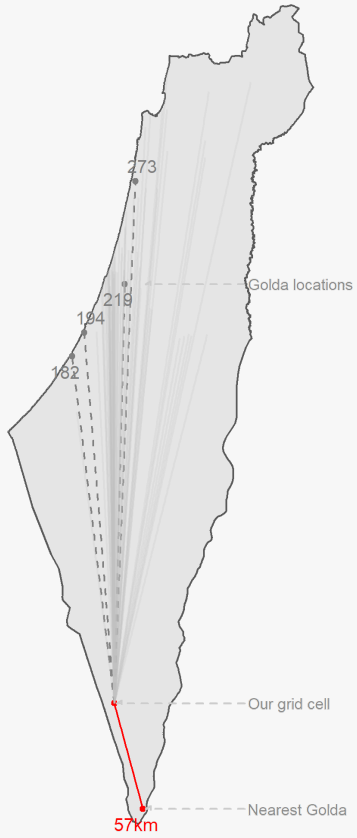
Distances

We have our grid cells, how do we identify the **nearest ice-cream location?**

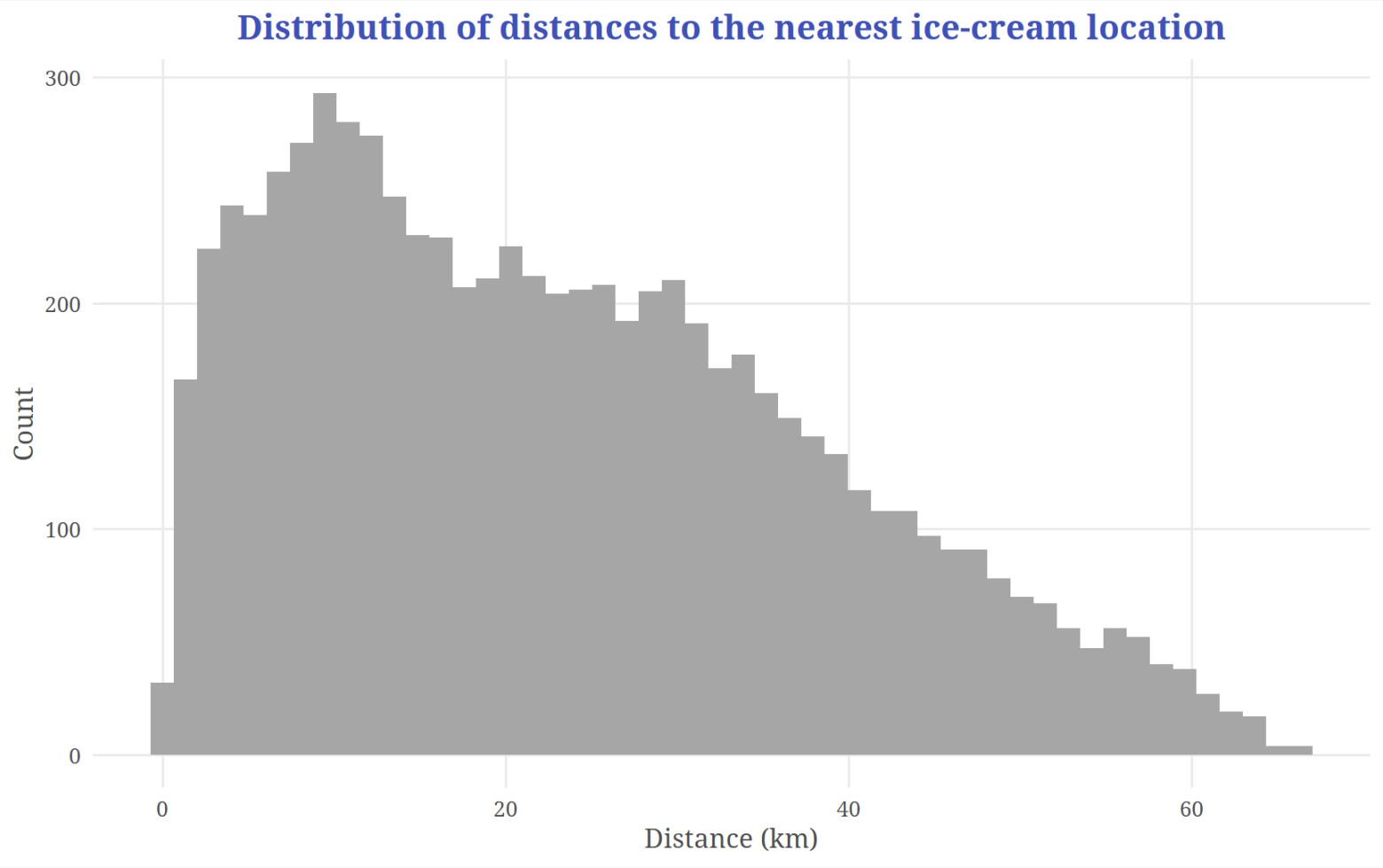
Example for 1 grid cell to several Goldas:

```
example_distances <- st_distance(
# Print 10:
head(set_units(example_distances,
```

```
## Units: [km]
##           [,1]
##  [1,] 218.88346
##  [2,] 272.69260
##  [3,]  57.20938
##  [4,] 194.03313
##  [5,] 182.41663
##  [6,] 209.48198
##  [7,] 131.61719
##  [8,] 256.71129
##  [9,] 218.63946
## [10,] 225.91495
```



Distances


Distribution of distances to the nearest ice-cream location
300
200
100
0
Count
0
20
40
60
Distance (km)

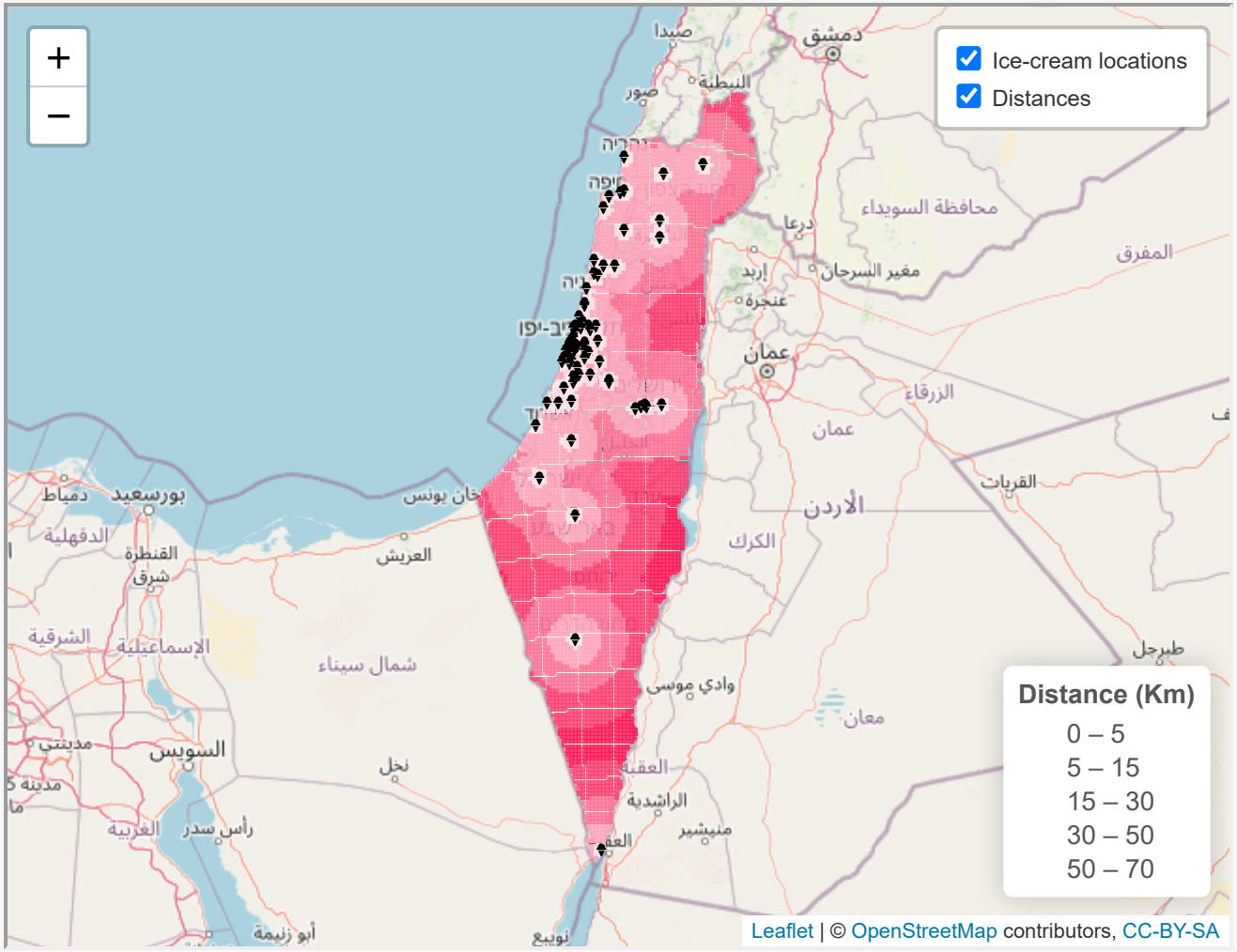
Cosmetics 🪄

```
## Rows: 7,375
## Columns: 7
## $ geometry.x  <POLYGON [°]> POLYGON ((34.90188 29.49693..., POLYGON ((34.91
## $ distance_km <dbl> 7.871391, 7.332709, 6.613721, 7.363470, 6.918497, 5.722
## $ location_id <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3
## $ city        <chr> "א" ,"אילת" ,"אילת" ,"אילת" ,"אילת" ,"אילת" ,"אילת" ,"אילת~
## $ street      <chr> "מתחם הספירל" ,"מתחם הספירל" ,"מתחם הספירל" ,"מתחם הספירל"~
## $ number      <chr> "08-9428989", "08-9428989", "08-9428989", "08-9428989",
## $ geometry.y  <POINT [°]> POINT (34.95968 29.54952), POINT (34.95968 29.549
```

```
 create_label_distances <- function(km, street, city){
   glue("
   <div style='text-align:left;'>
   You are <span style='font-size:13px;'><b>{round(km, 1)}</b></span> km
   <div style='text-align:right;'>
        {street}, {city}</div>") %>%
     HTML()}
```

```
## [[1]]
## <div style='text-align:left;'>
## You are <span style='font-size:13px;'><b>7.9</b></span> km from the nearest
## <div style='text-align:right;'>
##      מתחם הספירל, אילת</div>
```

Et Voila



If you can't see the iframe, please refresh the page.

How does this relate to open source

Feedback

👎 Don't I have anything better to blog about?

> Someone will always have what to say about your work

Positive

👍 Someone using the code to look at hamburger chains across Sydney, Aus.

👍 Micha Silver's suggestion to use a raster instead of a vector object

Personal reasons

- GIS was available in an already used OS platform (R)
- Free
- Can use/change others' code freely
- Pushes to high standards knowing others see/read your code
- Get to contribute back
- **The community!**

Thank you!