ESM 270 Week 3 Marxan

Cameryn Brock

4/28/2020

```r
library(here)
library(janitor)
library(prioritizr)
library(sf)
library(slam)
library(gurobi)
library(ggmap)
library(patchwork)
library(tidyverse)
```

Information on the `prioritizr` package is at https://mran.microsoft.com/snapshot/2018-03-04/web/packages/prioritizr/vignettes/quick__start.html.

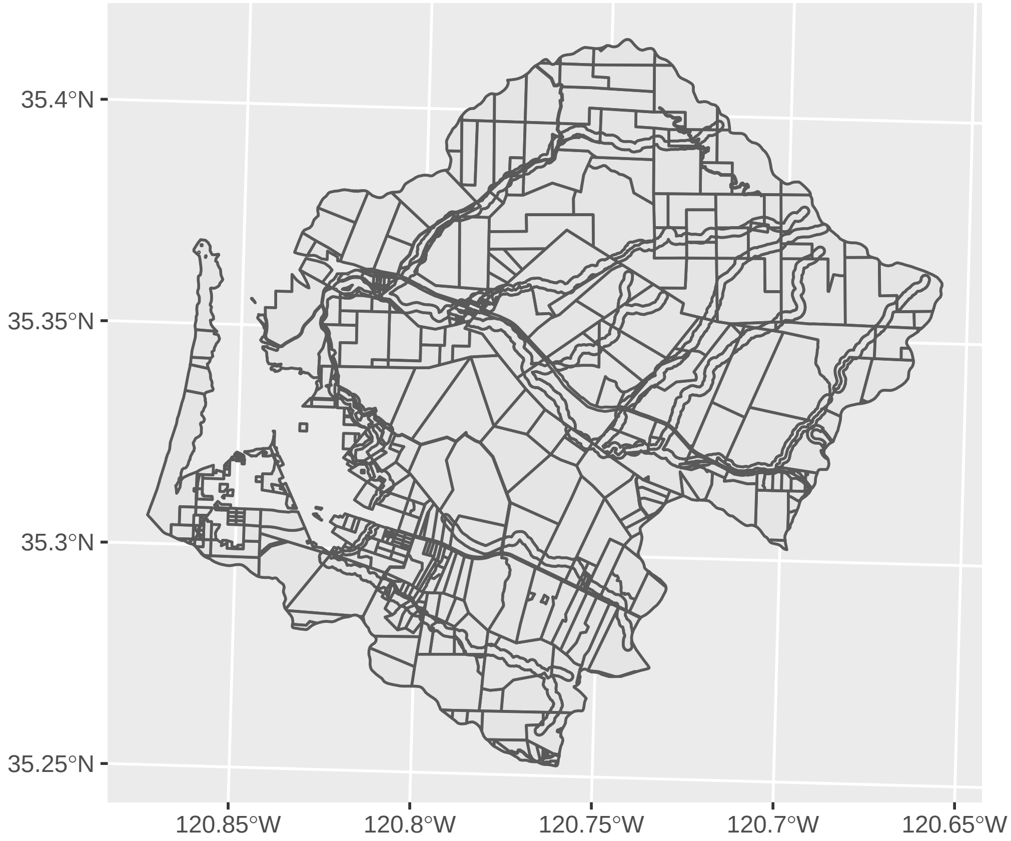
A really useful example for summed solutions is under “Selection Frequencies” here: https://cran.r-project.org/web/packages/prioritizr/vignettes/tasmania.html

Note: for `solve()` function, must have a solver installed, which for the the summing/multiple runs process, must be `gurobi`. If you don’t have it, it requires getting an academic license and is generally a hassle which required a lot of troubleshooting for me… but the instructions below should make it quicker for you if you decide to.

1. Download gurobi (https://www.gurobi.com/downloads/gurobi-optimizer-eula/)
2. Request an academic license (https://www.gurobi.com/downloads/end-user-license-agreement-academic/)
3. Verify license in terminal by copying and pasting information in your acount > your licenses
4. Must install r package with`install.packages('/Library/gurobi902/mac64/R/gurobi_9.0-2_R_3.6.1.tgz', repos=NULL)` (if on a mac and downloaded the most recent version of gurobi, don’t download newest version of R! (4.0.0)).
5. must install slam package with ‘install.packages(“slam”)
6. add gurobi and slam libraries

Read in Morro Bay data

```r
# species and pu data
# all csv files are unaltered from the xlsx files in the R drive. Just saved as csvs.

spec <- read_csv("MorroBay_spec.csv") %>%
  head(140) %>%  # read in extra blank rows
  rename("amount" = "target") # target is called "prop" (relative) or "amount" (absolute)

pu <- read_csv("MorroBay_pu.csv") %>%
  select(1:3) # read in extra blank column

puvsp <- read_csv("MorroBay_puvspr.csv") %>%
  select(1:3) %>%
  head(11849)

status <- read_csv("spec_name_status.csv") %>%
  select(1:3) %>%
  head(140)

# polygons

parcels <- read_sf(dsn = here("MorroBay_data"), layer = "MorroBay_parcels") %>%
  clean_names()

ggplot(data = parcels) +
  geom_sf()
```



```
# create df with pus and polygons
```

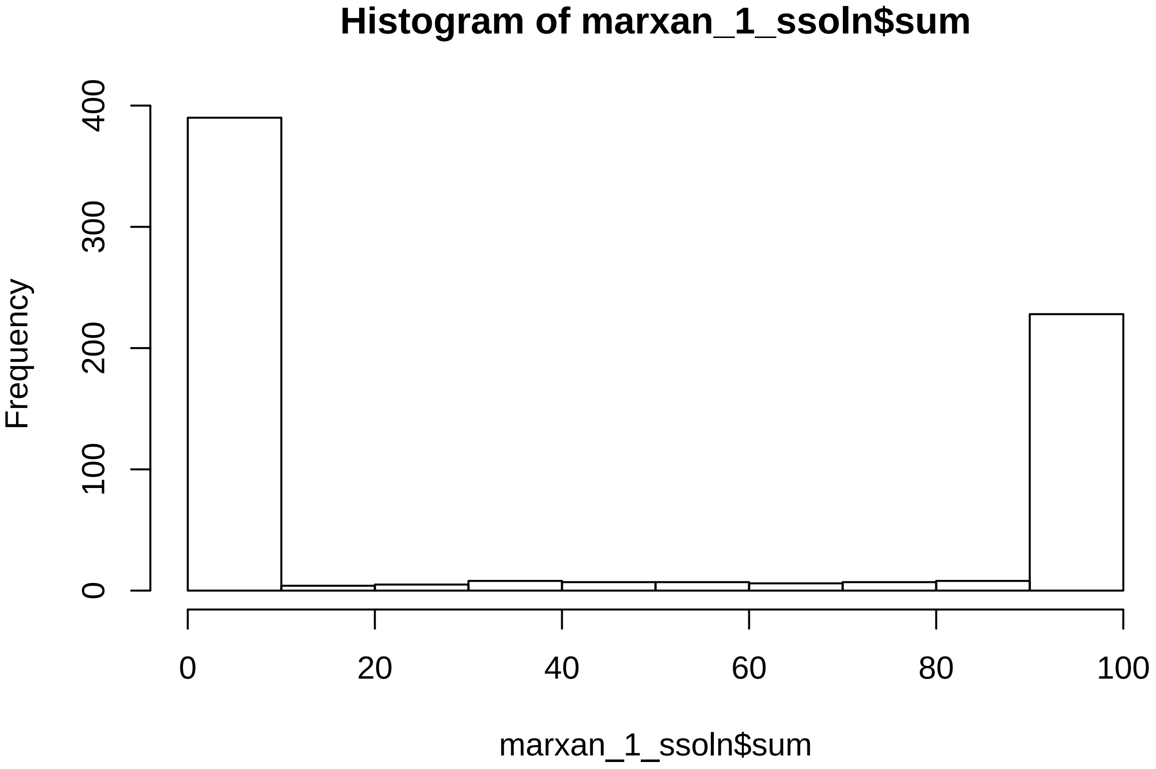
Run Marxan problem

```
# marxan_problem(), more 'canned' approach apparently, but seems good for our purposes. If more customi

marxan_1 <- marxan_problem(x = pu,
                          spec = spec,
                          puvspr = puvsp,
                          bound = NULL,
                          blm = 0)
```

```r
marxan_1_problem <- marxan_1 %>%
  add_gurobi_solver(gap = 0.15) %>%
  add_pool_portfolio(method = 2, number_solutions = 100) # see method meaning under ?add_pool_portfolio

print(marxan_1_problem)

marxan_1_soln <- solve(marxan_1_problem)
```

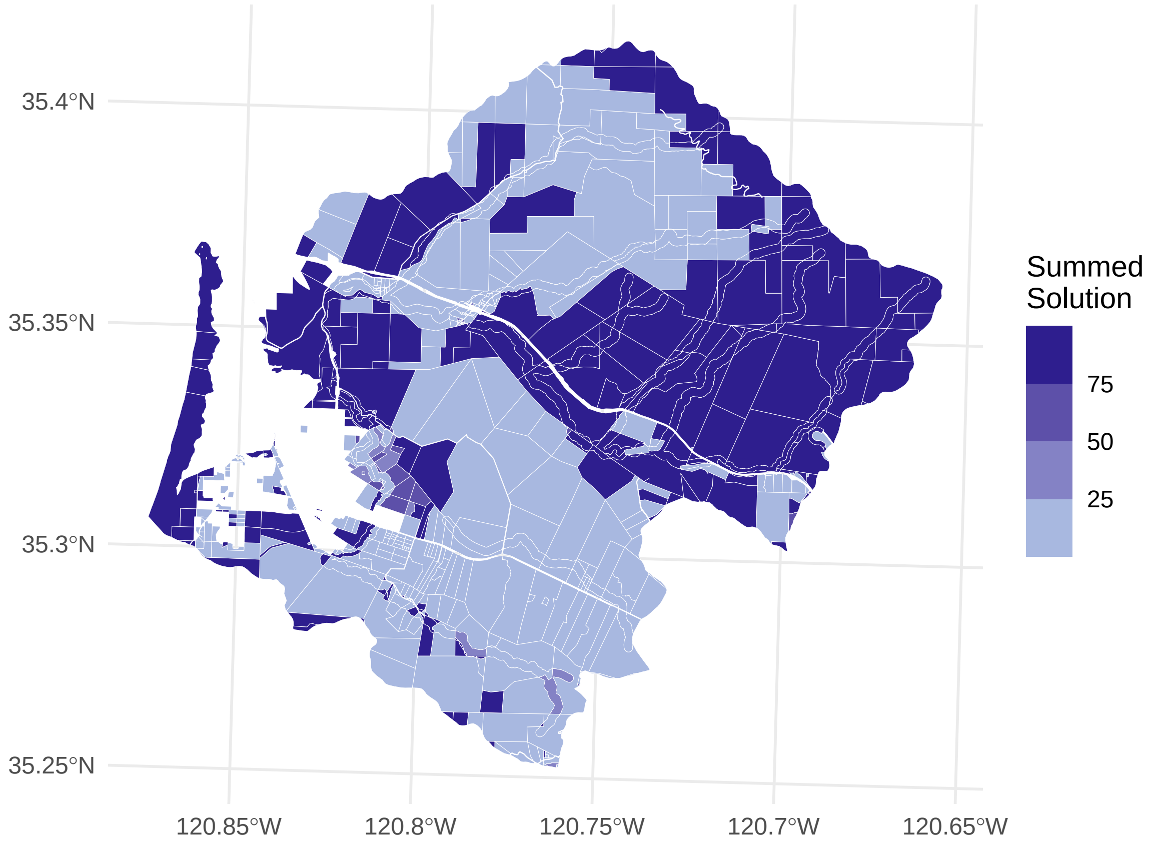
```
## Gurobi Optimizer version 9.0.2 build v9.0.2rc0 (mac64)
## Optimize a model with 140 rows, 670 columns and 11849 nonzeros
## Model fingerprint: 0xd2c5bccb
## Variable types: 0 continuous, 670 integer (670 binary)
## Coefficient statistics:
##   Matrix range     [1e+00, 1e+00]
##   Objective range  [2e+00, 5e+06]
##   Bounds range     [1e+00, 1e+00]
##   RHS range        [3e-01, 2e+02]
## Found heuristic solution: objective 4.238988e+07
## Presolve removed 116 rows and 205 columns
## Presolve time: 0.01s
## Presolved: 24 rows, 465 columns, 3066 nonzeros
## Variable types: 0 continuous, 465 integer (465 binary)
## Presolve removed 1 rows and 0 columns
## Presolved: 23 rows, 465 columns, 3058 nonzeros
##
##
## Root relaxation: objective 2.840941e+07, 14 iterations, 0.00 seconds
##
##     Nodes    |    Current Node    |     Objective Bounds      |     Work
##  Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
##
## *    0     0               0    2.840941e+07 2.8409e+07  0.00%     -    0s
##
## Optimal solution found at node 0 - now completing solution pool...
##
##     Nodes    |    Current Node    |      Pool Obj. Bounds     |     Work
##              |                    |   Worst                   |
##  Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
##
##      0     0          -    0               - 2.8409e+07      -      -    0s
##      0     0          -    0               - 2.8409e+07      -      -    0s
##      0     2          -    0               - 2.8409e+07      -      -    0s
##
## Explored 1958 nodes (383 simplex iterations) in 0.10 seconds
## Thread count was 1 (of 4 available processors)
##
## Solution count 100: 2.84094e+07 2.84097e+07 2.842e+07 ... 3.34128e+07
##
## Optimal solution found (tolerance 1.50e-01)
## Best objective 2.840941400000e+07, best bound 2.840941400000e+07, gap 0.0000%
```

```r
# sum solutions

marxan_1_ssoln <- marxan_1_soln %>%
  mutate(sum = rowSums(.[6:105])) %>%
  select(id, cost, status, locked_in, locked_out, sum)

hist(marxan_1_ssoln$sum)
```

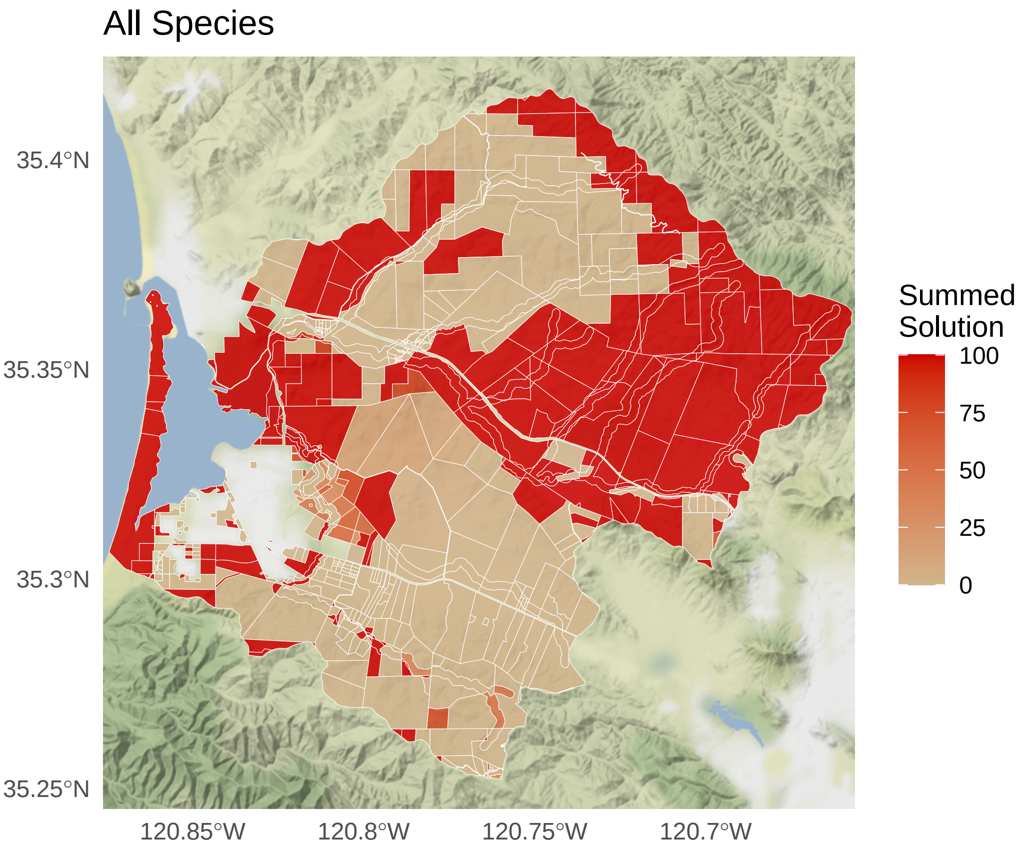


Join parcels polygons with output

```
parcels_marxan_1 <- inner_join(parcels, marxan_1_ssoln, by = "id")

ggplot(data = parcels_marxan_1) +
  geom_sf(aes(fill = sum),
          color = "white",
          size = 0.05) +
  scale_fill_binned(low = "slategray2",
                    high = "navy") +
  labs(fill = "Summed \nSolution") +
  theme_minimal()
```



```
# add basemap

morrobay <- get_map(location = c(lon = -120.7665, lat = 35.335),
                    zoom = 12,
                    maptype = "terrain-background",
                    source = "google")

# see different ggmap maptypes here: https://www.nceas.ucsb.edu/sites/default/files/2020-04/ggmapCheats
# - background means no references, omit if want references

ggmap(morrobay) +
    geom_sf(data = parcels_marxan_1,
            aes(fill = sum),
          color = "white",
          size = 0.1,
          alpha = 0.85,
          inherit.aes = FALSE) +
  coord_sf(crs = st_crs(4326)) +
  scale_fill_gradient(low = "tan",
                    high = "red3") +
  labs(title = "All Species",
       fill = "Summed \nSolution",
       x = NULL,
       y = NULL) +
  theme_minimal()
```



```
ggsave("spec_graph.png")
```

Considering endangered species

Make spec and pucsp files with only endangered & threatened species

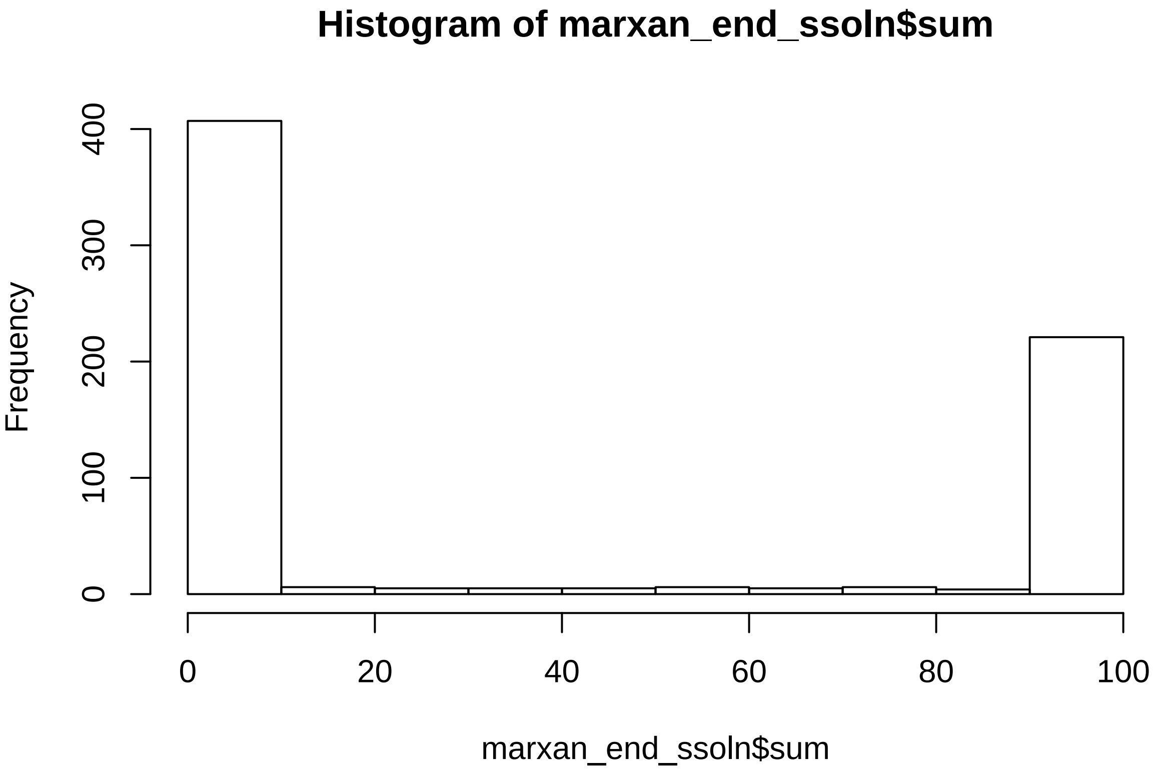
```
end_status <- status %>%
  mutate("endangered" = case_when(
    str_detect(status, pattern = "endangered") == TRUE ~ "yes",
    str_detect(status, pattern = "threatened") == TRUE ~ "yes",
    T ~ "no")) %>%
  filter(endangered == "yes")

end_spec <- merge(end_status, spec, by = "id") %>%
  select(id, amount, spf, name.x) %>%
  rename("name" = "name.x")

puvsp_id <- puvsp %>%
  rename("id" = "species")

end_puvsp <- merge(end_status, puvsp_id, by = "id") %>%
  rename("species" = "id")
```

Run Marxan problem

```
marxan_end <- marxan_problem(x = pu,
                             spec = end_spec,
                         puvspr = end_puvsp,
                         bound = NULL,
                         blm = 0)

marxan_end_problem <- marxan_end %>%
  add_gurobi_solver(gap = 0.15) %>%
  add_pool_portfolio(method = 2, number_solutions = 100)

print(marxan_end_problem)

marxan_end_soln <- solve(marxan_end_problem)
```

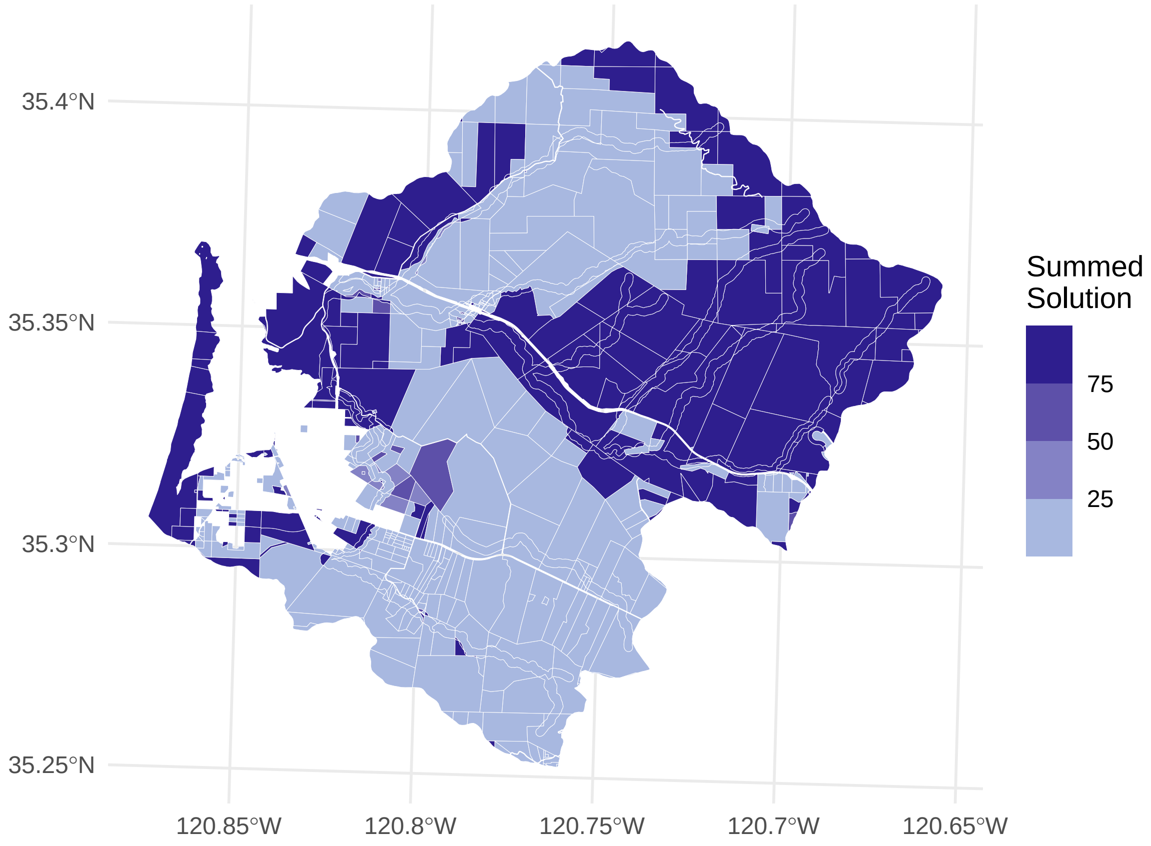
```
## Gurobi Optimizer version 9.0.2 build v9.0.2rc0 (mac64)
## Optimize a model with 18 rows, 670 columns and 970 nonzeros
## Model fingerprint: 0x6e88f508
## Variable types: 0 continuous, 670 integer (670 binary)
## Coefficient statistics:
##   Matrix range     [1e+00, 1e+00]
##   Objective range  [2e+00, 5e+06]
##   Bounds range     [1e+00, 1e+00]
##   RHS range        [9e-01, 1e+02]
## Found heuristic solution: objective 4.174920e+07
## Presolve removed 14 rows and 203 columns
## Presolve time: 0.00s
## Presolved: 4 rows, 467 columns, 495 nonzeros
## Variable types: 0 continuous, 467 integer (467 binary)
## Found heuristic solution: objective 3.534249e+07
## Presolved: 4 rows, 467 columns, 495 nonzeros
##
##
## Root relaxation: objective 2.677119e+07, 1 iterations, 0.00 seconds
##
##     Nodes    |    Current Node    |     Objective Bounds      |     Work
##  Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
##
## *    0     0               0    2.677119e+07 2.6771e+07  0.00%     -    0s
##
## Optimal solution found at node 0 - now completing solution pool...
##
##     Nodes    |    Current Node    |      Pool Obj. Bounds     |     Work
##              |                    |   Worst                   |
##  Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
##
##      0     0          -    0               - 2.6771e+07      -     -    0s
##      0     0          -    0               - 2.6771e+07      -     -    0s
##      0     2          -    0               - 2.6771e+07      -     -    0s
##
## Explored 4914 nodes (1716 simplex iterations) in 0.33 seconds
## Thread count was 1 (of 4 available processors)
##
## Solution count 100: 2.67712e+07 2.67732e+07 2.67736e+07 ... 3.14836e+07
##
## Optimal solution found (tolerance 1.50e-01)
## Best objective 2.677119300000e+07, best bound 2.677119300000e+07, gap 0.0000%
```

```
# sum solutions

marxan_end_ssoln <- marxan_end_soln %>%
  mutate(sum = rowSums(.[6:105])) %>%
  select(id, cost, status, locked_in, locked_out, sum)

hist(marxan_end_ssoln$sum)
```



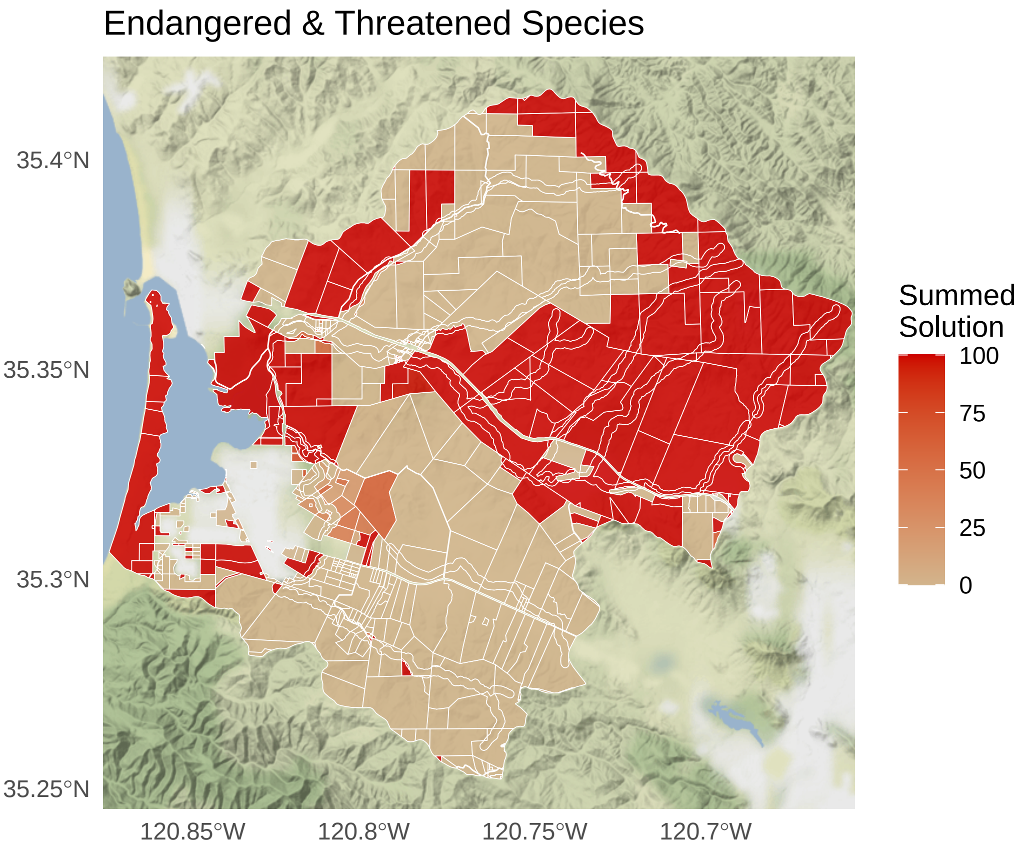
Join parcels polygons with output

```
parcels_marxan_end <- inner_join(parcels, marxan_end_ssoln, by = "id")

ggplot(data = parcels_marxan_end) +
  geom_sf(aes(fill = sum),
          color = "white",
          size = 0.05) +
  scale_fill_binned(low = "slategray2",
                    high = "navy") +
  labs(fill = "Summed \nSolution") +
  theme_minimal()
```



Map

```
ggmap(morrobay) +
    geom_sf(data = parcels_marxan_end,
            aes(fill = sum),
          color = "white",
          size = 0.15,
          alpha = 0.85,
          inherit.aes = FALSE) +
  coord_sf(crs = st_crs(4326)) +
  scale_fill_gradient(low = "tan",
                      high = "red3") +
  labs(title = "Endangered & Threatened Species",
       fill = "Summed \nSolution",
       x = NULL,
       y = NULL) +
  theme_minimal()
```



```r
ggsave("end_graph.png")
```