

```

#####
## 0. Setup R
#####

install.packages(c("sf", "tidyverse", "sp", "rgeos", "rgdal",
"raster",
                    "units", "prioritizr", "prioritizrdata",
"Rsymphony",
                    "mapview", "assertthat", "velox", "remotes",
                    "gridExtra", "data.table", "readxl"))

#####
## 1. Using prioritizr for systematic conservation planning
#####

# Example using data from British Columbia
# https://prioritizr.net/articles/saltspring.html

library(rgdal)
library(data.table)
library(readxl)
library(prioritizr)
library(prioritizrdata)

# load planning unit data
data(salt_pu)

# load conservation feature data
data(salt_features)

# print planning unit data
print(salt_pu)

# plot map showing the planning units costs on a log-scale
plot(log(salt_pu), main = "Planning unit costs (log)")

# print features
print(salt_features)

# plot map showing the distribution of the features
plot(salt_features, main = names(salt_features))

# Formulating the Problem

# create problem
p1 <- problem(salt_pu, salt_features) %>%
  add_min_set_objective() %>%

```

```

add_relative_targets(0.17) %>%
add_binary_decisions() %>%
add_default_solver(gap = 0)

# print problem
print(p1)

# solve the problem
s1 <- solve(p1)

plot(s1)

# Adding connectivity
titles2 <- c() # create vector to store plot titles
s2 <- stack() # create empty stack to store solutions

# no connectivity requirement
titles2 <- c(titles2, "No connectivity")
s2 <- addLayer(s2, solve(p1))

# impose small penalty for fragmented solutions
titles2 <- c(titles2, "Boundary penalty (low)")
p2 <- p1 %>% add_boundary_penalties(50, 0.5)
s2 <- addLayer(s2, solve(p2))

# impose high penalty for fragmented solutions
titles2 <- c(titles2, "Boundary penalty (high)")
p3 <- p1 %>% add_boundary_penalties(500, 0.5)
s2 <- addLayer(s2, solve(p3))

# plot solutions
plot(s2, main = titles2, breaks = c(0, 0.5, 1), col = c("grey70",
"darkgreen"))

```