

Inherent soil quality experiment

Serge Wiltshire

Abstract

Exploring the impact of a distribution of a persistent inherent soil quality factor.

Keywords: agriculture, ecosystem services, tipping points, diversification practices, decision-making

```
# set up the MDP
actions <- seq(0, 1, length = 100)
states <- seq(0, 1.5, length = 100)
params <- list(
  benefit = 1.56, #orig. benefit = 1.57,
  cost = 1,
  sigma = 0.1,
  r = 0.1,
  discount = 0.97,
  Tmax = 20)
transition_fn <- function(s, a, params) s + params$r * (a - s)
utility_fn <- function(s, a, params) params$benefit * s - params$cost * a
model <- continuous_model(states, actions, params, transition_fn, utility_fn)

# solve the MDP
base_soln <- MDPtoolbox::mdp_value_iteration(model$P, model$U, params$discount)
```

```
## [1] "MDP Toolbox: iterations stopped, epsilon-optimal policy found"
```

```
base_soln_df <- tibble(state = states,
  action = actions[base_soln$policy])

#write_csv(base_soln_df, "../model-data/base_soln_df.csv.gz")
```

```
reps <- 500
init <- truncnorm::rtruncnorm(reps, 0, 1, 0.5, 0.2) %>%
  map_int(function(x) which.min(abs(x - states)))
```

```
sim <- function(soln,
  Tmax = params$Tmax,
  x0 = init){
  map_dfr(1:length(x0),
    function(i) sim_mdp(model$P, model$U, soln$policy, params$discount, x0 = x0[i], Tmax = Tmax)
    .id = "reps")
}
sims_baseline <- sim(base_soln)
```

```

ggarrange(
  sim_plot_ts(sims_baseline, title = ggtitle("A. ES state over time"), rmarmod = -5),
  sim_plot_dens(sims_baseline, title = ggtitle("B. Init. and final state"), tpos = "panel", lab_lo_pee
  widths = c(2.75, 2), nrow = 1, ncol = 2
)

```

```

6 ## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family
7 ## 'Roboto Condensed' not found in PostScript font database
8
9 ## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family
10 ## 'Roboto Condensed' not found in PostScript font database
11
12 ## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family
13 ## 'Roboto Condensed' not found in PostScript font database
14
15 ## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family
16 ## 'Roboto Condensed' not found in PostScript font database
17
18 ## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family
19 ## 'Roboto Condensed' not found in PostScript font database
20
21 ## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family
22 ## 'Roboto Condensed' not found in PostScript font database
23
24 ## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family
25 ## 'Roboto Condensed' not found in PostScript font database
26
27 ## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family
28 ## 'Roboto Condensed' not found in PostScript font database
29
30 ## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family
31 ## 'Roboto Condensed' not found in PostScript font database
32
33 ## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family
34 ## 'Roboto Condensed' not found in PostScript font database
35
36 ## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family
37 ## 'Roboto Condensed' not found in PostScript font database
38
39 ## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family
40 ## 'Roboto Condensed' not found in PostScript font database
41
42 ## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family
43 ## 'Roboto Condensed' not found in PostScript font database
44
45 ## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family
46 ## 'Roboto Condensed' not found in PostScript font database
47
48 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
49 ## family 'Roboto Condensed' not found in PostScript font database
50
51 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font

```

```

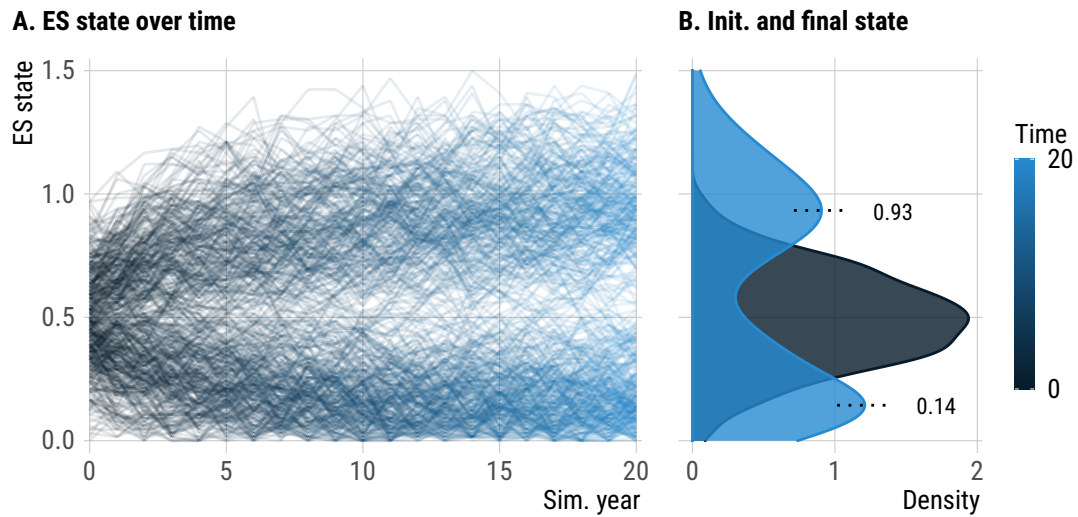
51 ## family 'Roboto Condensed' not found in PostScript font database
52
53 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
54 ## family 'Roboto Condensed' not found in PostScript font database
55
56 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
57 ## family 'Roboto Condensed' not found in PostScript font database
58
59 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
60 ## family 'Roboto Condensed' not found in PostScript font database
61
62 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
63 ## family 'Roboto Condensed' not found in PostScript font database
64
65 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
66 ## family 'Roboto Condensed' not found in PostScript font database
67
68 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
69 ## family 'Roboto Condensed' not found in PostScript font database
70
71 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
72 ## family 'Roboto Condensed' not found in PostScript font database
73
74 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
75 ## family 'Roboto Condensed' not found in PostScript font database
76
77 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
78 ## family 'Roboto Condensed' not found in PostScript font database
79
80 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
81 ## family 'Roboto Condensed' not found in PostScript font database
82
83 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
84 ## family 'Roboto Condensed' not found in PostScript font database
85
86 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
87 ## family 'Roboto Condensed' not found in PostScript font database
88
89 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
90 ## family 'Roboto Condensed' not found in PostScript font database
91
92 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
93 ## family 'Roboto Condensed' not found in PostScript font database
94
95 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
96 ## family 'Roboto Condensed' not found in PostScript font database
97
98 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
99 ## family 'Roboto Condensed' not found in PostScript font database
100
101 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
102 ## family 'Roboto Condensed' not found in PostScript font database

```

```

103 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
104 ## family 'Roboto Condensed' not found in PostScript font database
105
106 ## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
107 ## family 'Roboto Condensed' not found in PostScript font database
108

```



109

```

state99th <- sims_baseline %>%
  mutate(state = states[state]) %>%
  pull(state) %>%
  quantile(.99) %>%
  unname()

tpt <- 0
for(si in 1:length(states)-1) {
  if(base_soln_df$action[si] < .1 && base_soln_df$action[si+1] > .9) {
    tpt <- base_soln_df$state[si] + ((base_soln_df$state[2] - base_soln_df$state[1]) / 2)
  }
}

soln_plot(base_soln_df %>% filter(state <= state99th), tpt = tpt)

```

