Quickly visualize solutions

2023-07-26

Take a glimpse at one of the solutions (reproduce Figure 1 panel B)!

Read in one of the solution

```
sol <- list.files("../data-formatted/sol/", pattern = "globiomICflat_highs_ref.csv", full.names = T)
solution <- read_csv(sol[[1]]) |>
  dplyr::select(id, solution_1_z1:solution_1_z26)
```

```
## Rows: 41046 Columns: 53
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (53): z1, z2, z3, z4, z5, z6, z7, z8, z9, z10, z11, z12, z13, z14, z15, ...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
zone_id <- read_csv("../data-formatted/zone_id.csv")
```

```
## Rows: 26 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): zone
## dbl (1): id
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
zone_id$zone
```

```
##  [1] "Cropland_low_restore"                "Cropland_med_restore"
##  [3] "Grassland_natural_restore"           "HeathlandShrub_natural_restore"
##  [5] "MarineTransitional_natural_restore"  "Pasture_low_restore"
##  [7] "SparseVeg_natural_restore"           "Wetlands_natural_restore"
##  [9] "WoodlandForest_multi_restore"        "WoodlandForest_primary_restore"
## [11] "WoodlandForest_prod_restore"         "WoodlandForest_multi_production"
## [13] "WoodlandForest_prod_production"      "Pasture_high_production"
## [15] "Pasture_low_production"              "Cropland_med_production"
## [17] "Cropland_low_production"             "Cropland_high_production"
## [19] "HeathlandShrub_natural_conserve"     "Grassland_natural_conserve"
## [21] "SparseVeg_natural_conserve"          "Wetlands_natural_conserve"
## [23] "RiversLakes_natural_conserve"        "MarineTransitional_natural_conserve"
## [25] "WoodlandForest_primary_conserve"     "Urban_urban_lockin"
```

```
colnames(solution) <- c("id", (zone_id$zone))
```

```
solution_table <- pu_in_EU |> rename(id = pu) |> #plot_data |>
  left_join(PU_template) |>
  dplyr::select(-id) |>
  rename(id = EU_id) |>
  left_join(solution) |>
  pivot_longer(-c(nuts2id:geometry))
```

```
## Joining with `by = join_by(id)`
## Joining with `by = join_by(id)`
```

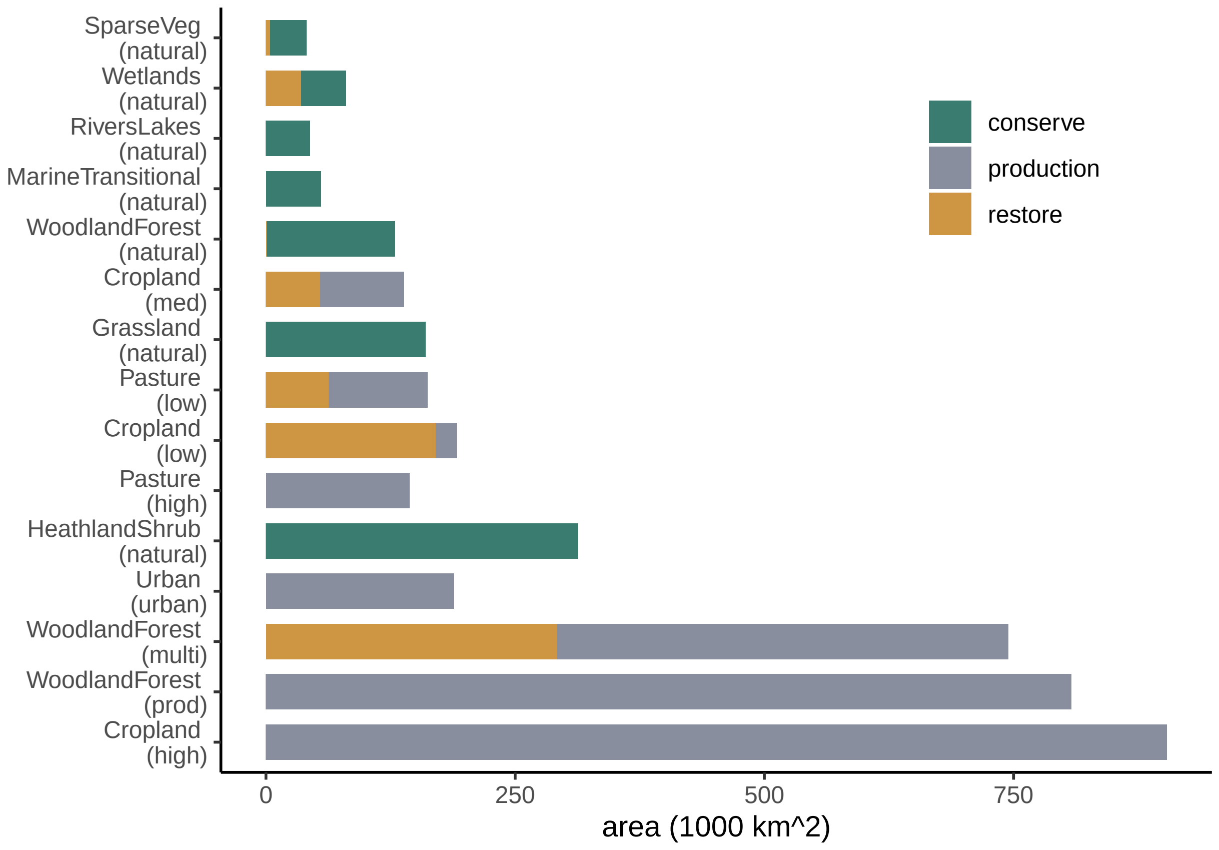
```
solution_table_plot <- solution_table |>
  mutate(zone = name) |>
  separate(name, c('maes_label', 'intensity', 'action'), sep = "_")
```

Quick bar plot of zones

```
bar_plot <- as_tibble(solution_table_plot) |>
  group_by(zone, maes_label, intensity, action) |>
  summarise(area = sum(value)) |>
  mutate(action = ifelse(action == "lockin", "production", action)) |>
  ungroup() |> filter(area >0) |>
  mutate(intensity = ifelse(intensity == "primary", "natural", intensity)) |>
  mutate(name = paste0(maes_label, " \n (", intensity, ")")) |>
  ggplot(aes(x = reorder(name, -area), y = area/10, fill = action)) + geom_bar(stat="identity", width =
  theme_classic() +
  labs(x = element_blank(), y = "area (1000 km^2)") +
  theme(legend.position = c(0.8,0.8),
        legend.title = element_blank()) +
    scale_fill_manual(values = c(met.brewer(name="Kandinsky",n=4,type="continuous"))[c(1,3,2)]) + coord_
```

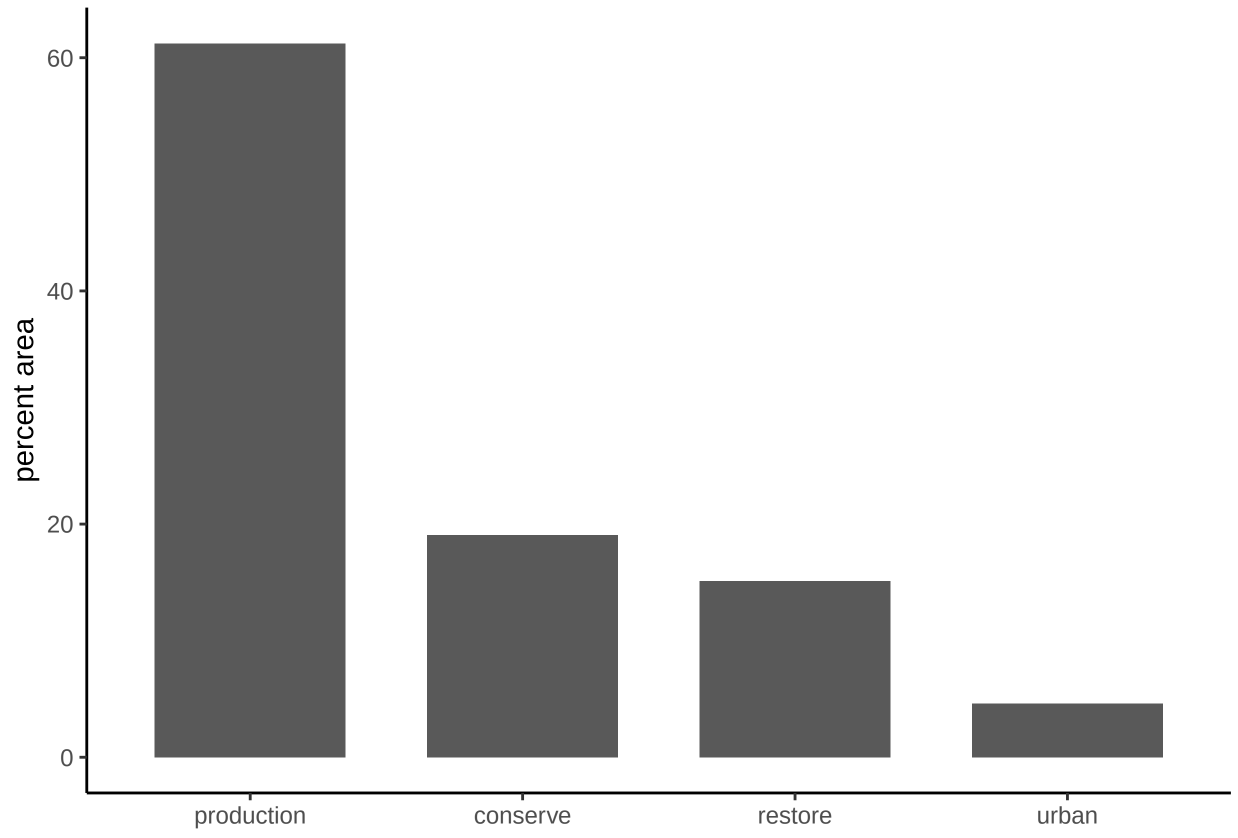
```
## `summarise()` has grouped output by 'zone', 'maes_label', 'intensity'. You can
## override using the `.groups` argument.
```

```
bar_plot
```



```r
bar_plot <- as_tibble(solution_table_plot) |>
  group_by(action) |>
  summarise(area = sum(value)) |>
  mutate(action = ifelse(action == "lockin", "urban", action)) |>
  ungroup() |> filter(area >0) |> mutate(tot_area = sum(area)) |>
  mutate(perc = area/tot_area*100) |>
  #mutate(intensity = ifelse(intensity == "primary", "natural", intensity)) |>
  #mutate(name = paste0(maes_label, " \n (", intensity, ")")) |>
  ggplot(aes(x = reorder(action, -perc), y = perc)) + geom_bar(stat="identity", width = 0.7) +
  theme_classic() +
  labs(x = element_blank(), y = "percent area") +
  theme(legend.position = c(0.8,0.8),
        legend.title = element_blank())

bar_plot
```



Maps of solution

```r
solution_raster <- fasterize(st_as_sf(solution_table_plot), PU_plot, field = "value", by = "zone")
solution_raster <- rast(solution_raster)
names(solution_raster)
```

```
##  [1] "Cropland_low_restore"                "Cropland_med_restore"
##  [3] "Grassland_natural_restore"           "HeathlandShrub_natural_restore"
##  [5] "MarineTransitional_natural_restore"  "Pasture_low_restore"
##  [7] "SparseVeg_natural_restore"           "Wetlands_natural_restore"
##  [9] "WoodlandForest_multi_restore"        "WoodlandForest_primary_restore"
## [11] "WoodlandForest_prod_restore"         "WoodlandForest_multi_production"
## [13] "WoodlandForest_prod_production"      "Pasture_high_production"
## [15] "Pasture_low_production"              "Cropland_med_production"
## [17] "Cropland_low_production"             "Cropland_high_production"
## [19] "HeathlandShrub_natural_conserve"     "Grassland_natural_conserve"
## [21] "SparseVeg_natural_conserve"          "Wetlands_natural_conserve"
## [23] "RiversLakes_natural_conserve"        "MarineTransitional_natural_conserve"
## [25] "WoodlandForest_primary_conserve"     "Urban_urban_lockin"
```

```r
restore <- c(1,2,6,3,4,5,7,8,9,10)
produce <- c(13:18, 26)
conserve <- c(19:25,12)
```

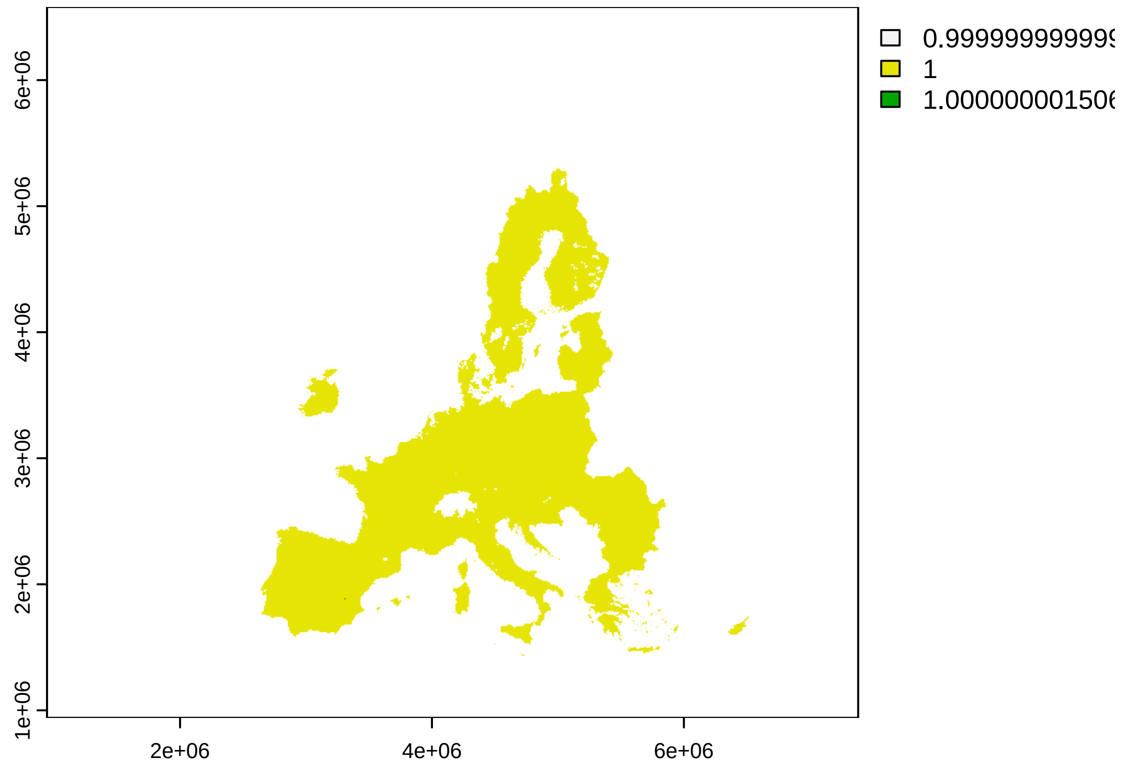
Color stuff

```
colors <- c("grey", met.brewer(name="VanGogh3",n=20,type="continuous"))
myPal <- colors
myTheme <- rasterTheme(region = myPal)
```

```
restore <- solution_raster[[restore]]
conserve <- solution_raster[[conserve]]
produce <- solution_raster[[produce]]
```

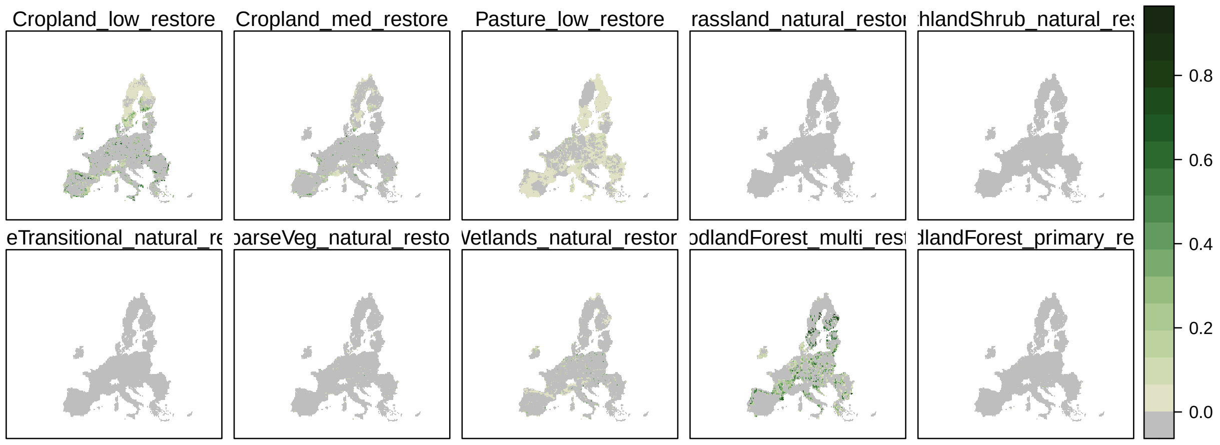
Just a check to make sure the solutions are summing to 1…

```
solu_sum <- app(solution_raster, 'sum')
plot(solu_sum)
```

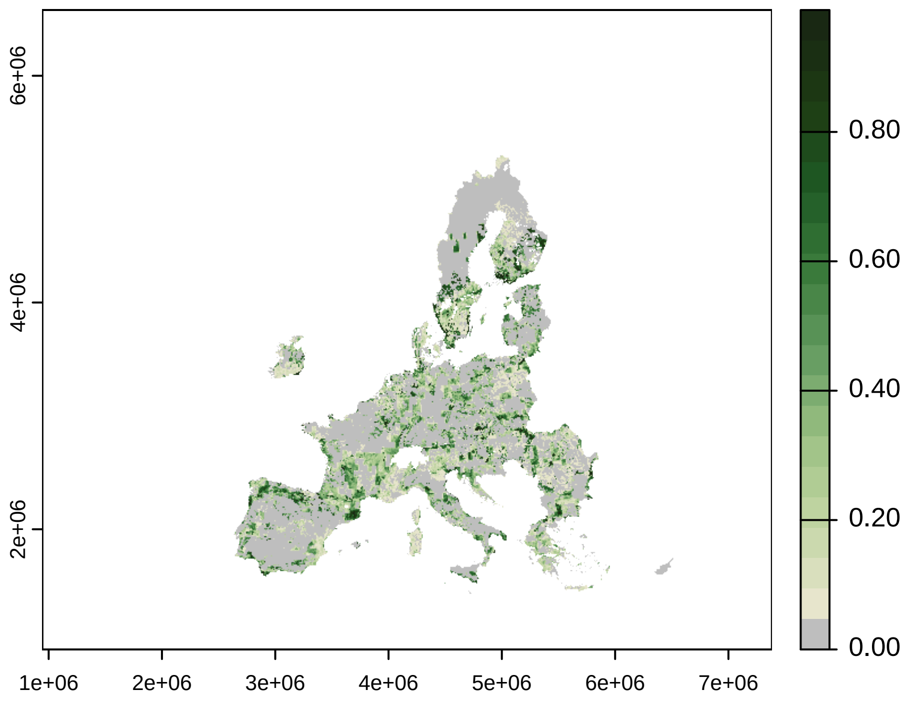


Check out restoration zones and overall restoration

```
levelplot(restore, par.settings = myTheme,xlab=NULL, ylab=NULL, scales=list(draw=FALSE))
```

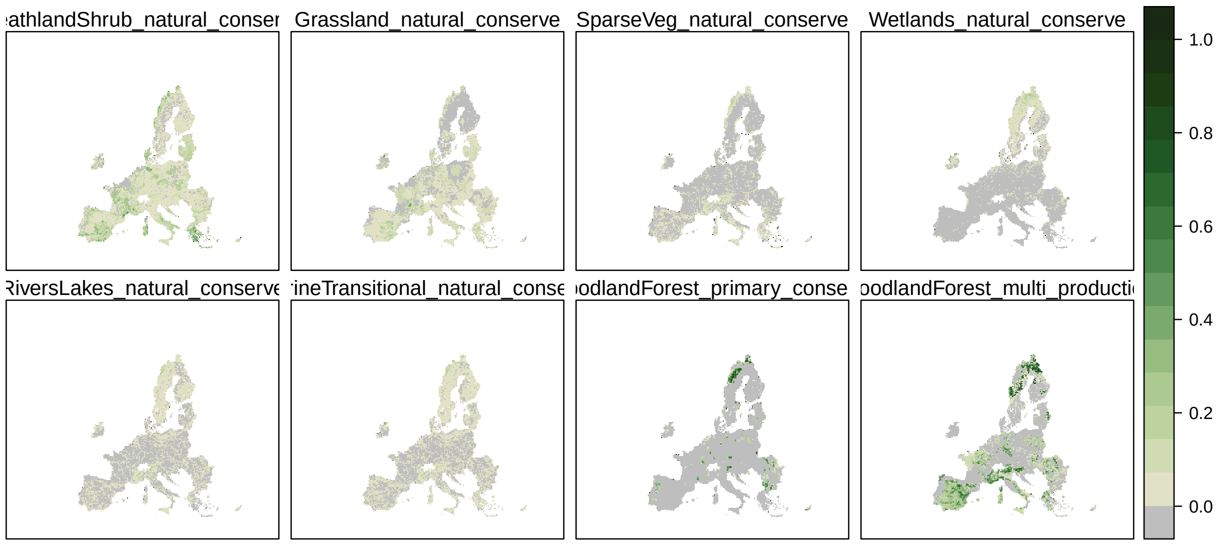


```r
rest_agg <- app(restore, sum)
plot(rest_agg, col = colors)
```

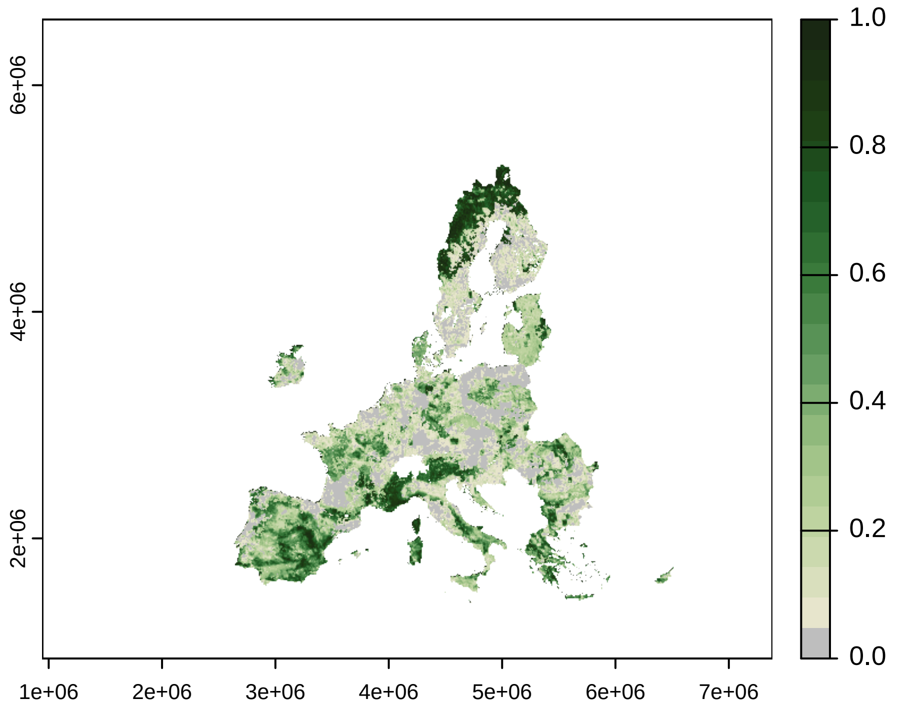


Check out conservation zones and overall conservation

```
levelplot(conserve, par.settings = myTheme,xlab=NULL, ylab=NULL, scales=list(draw=FALSE))
```

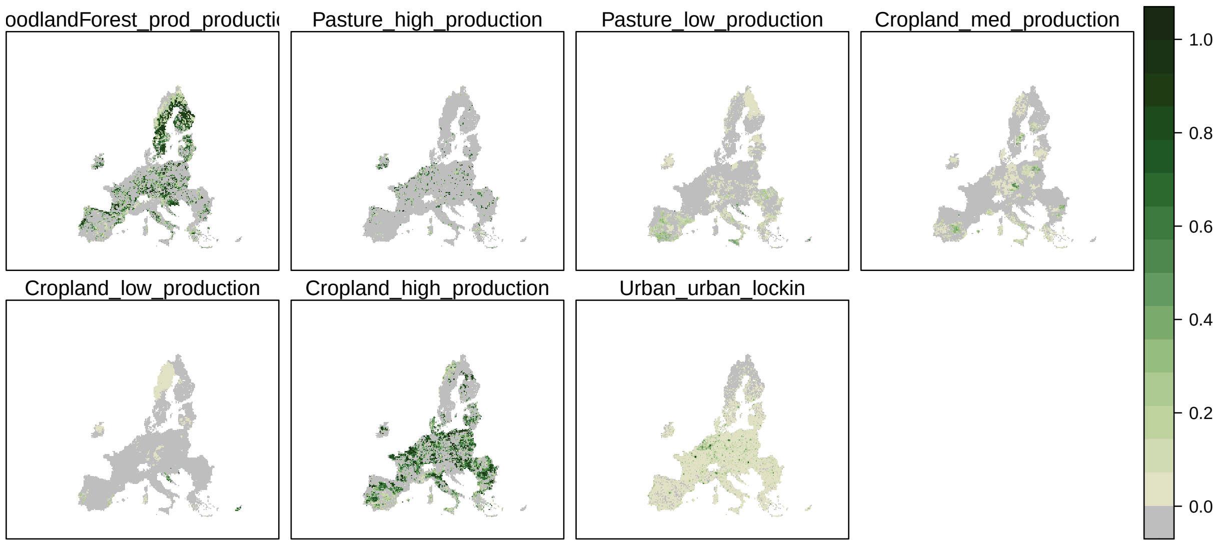

athlandShrub_natural_conser
Grassland_natural_conserve
SparseVeg_natural_conserve
Wetlands_natural_conserve
RiversLakes_natural_conserve
ineTransitional_natural_conse
odlandForest_primary_conse
oodlandForest_multi_producti
1.0
0.8
0.6
0.4
0.2
0.0

```
conserve_agg <- app(conserve, sum)
plot(conserve_agg, col = colors)
```

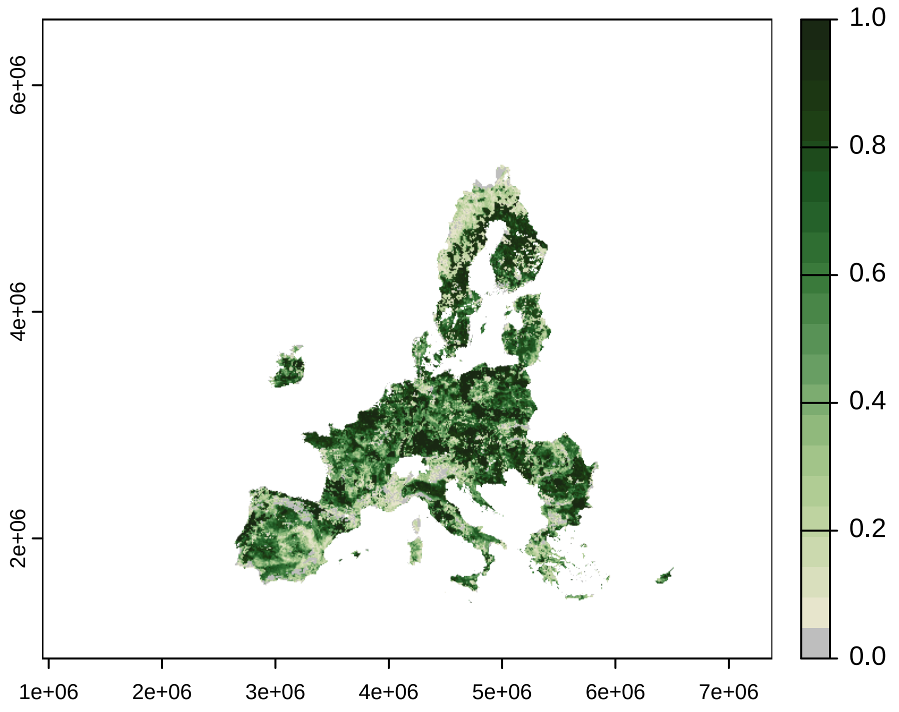


Check out production zones and overall production

```
levelplot(produce, par.settings = myTheme,xlab=NULL, ylab=NULL, scales=list(draw=FALSE))
```



```
prod_agg <- app(produce, sum)
plot(prod_agg, col = colors)
```


6e+06
4e+06
2e+06
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
7e+06
1.0
0.8
0.6
0.4
0.2
0.0