

Describing landscapes: calculating forest amount and isolation for Atlantic Forest sampling points

Bernardo Niebuhr, Mauricio Vancine

2022-02-23

Contents

Introduction	1
Loading data	1
Calculating changes in forest amount and isolation	2
Pollination networks	2
Calculate metrics and changes	2
Plot and identify changes	4
Join with spatial data and save output	8
Plot spatial data	9
Furgivory networks	12
Calculate metrics and changes	12
Plot and identify changes	14
Join with spatial data and save output	18
Plot spatial data	19
A few things to check	22

Introduction

Here we calculate the change in forest amount and isolation between 1985 and 2020 within several extents around each sampling point where interaction networks were sampled within the Atlantic Forest in Brazil. We consider both frugivory/seed dispersal networks (Bello et al. 2017) and pollination networks (REF, *in prep*). The aim is to describe the landscape changes at multiple scales for these sampling points, so that it is possible to select which ones to include when studying extinction debts (and credits) and changes in the ecological communities and their interaction networks.

Loading data

Here we load the packages and the landscape and interaction data. To assess changes in forest amount, we used the land use maps from MapBiomas, from the years 1985 and 2020, and consider the class forest (detail the procedure here).

```
# library(devtools)
# devtools::install_github("bniebuhr/landscapemetrics", ref = "multibuffer")
# devtools::install_github("bniebuhr/landscapetools", ref = "multibuffer")

# packages
library(dplyr)
library(readxl)
library(ggplot2)
```

```

library(forcats)

library(raster)
library(sf)
library(landscapetools)
library(landscapemetrics)
library(tmap)
library(tmaptools)

# rasters
forest_1985 <- raster::raster("data/mapbiomas_v06_1985_forest.tif")
forest_2020 <- raster::raster("data/mapbiomas_v06_2020_forest.tif")
# plot(forest_1985)

# CRS
crs_atlantic <- raster::crs(forest_1985)
# resolution of the map in meters
res_orig <- raster::res(forest_1985)[1]
res_m <- 30

# datasets
pol <- readxl::read_xlsx("data/Atlantic_forest_floral_visitor_data_points.xlsx") %>%
  dplyr::rename(x = longitude, y = latitude) %>%
  sf::st_as_sf(coords = c("x", "y"), crs = crs_atlantic)
fru <- readxl::read_xlsx("data/Atlantic_forest_frugivory_data_points.xlsx") %>%
  dplyr::rename(x = longitude, y = latitude) %>%
  sf::st_as_sf(coords = c("x", "y"), crs = crs_atlantic)

```

Calculating changes in forest amount and isolation

To calculate the change in forest amount around each sampling point, we followed these steps:

- Create a series of buffers, with extent from 250 m to 10 km .
- Calculate the change in absolute area of forest (in hectares) within each buffer, for each point, between 1985 and 2020.
- Calculate the change in proportion of forest within each buffer, for each point, between 1985 and 2020.
- Calculate the change in isolation, both absolute and proportional, between 1985 and 2020. Isolation was calculated as the mean distance between nearest neighbor forest patches.
- Classify points where the changes in forest amount and isolation were the highest (> 3% of change).

Pollination networks

Calculate metrics and changes

```

# pollination

# calculate amount of forest - 1985
buffers <- c(250, 750, 1000, 1500, 2000, 3000, 4000, 5000, 7500, 10000)
tictoc::tic()
pol_1985 <- landscapetools::util_extract_multibuffer(forest_1985, pol,
  buffer_width = buffers,
  rel_freq = TRUE, point_id_text = FALSE

```

```

) %>%
  dplyr::rename(freq1985 = freq, rel_freq1985 = rel_freq) %>%
  tibble::as_tibble() %>%
  dplyr::mutate(
    id = as.numeric(as.character(id)),
    layer = as.numeric(as.character(layer))
  )
tictoc::toc() # 17 min, 10 buffer sizes, 200 points

# calculate isolation - 1985
pol_1985_enh <- landscapemetrics::scale_sample(forest_1985, pol,
  shape = "circle",
  size = buffers * res_orig / res_m,
  what = c("lsm_c_enh_mn"),
  verbose = TRUE,
  progress = TRUE
) %>%
  dplyr::mutate(size = size / res_orig * res_m)

# subset and rename columns
pol_1985_enh <- pol_1985_enh %>%
  dplyr::select(
    id = plot_id, layer = class, metric1985 = metric,
    value1985 = value, buffer = size
  )

# join
pol_1985 <- pol_1985 %>%
  dplyr::left_join(pol_1985_enh,
    by = c("id", "layer", "buffer")
  )

# calculate amount of forest - 2020
pol_2020 <- landscapetools::util_extract_multibuffer(forest_2020, pol,
  buffer_width = buffers,
  rel_freq = TRUE, point_id_text = FALSE
) %>%
  dplyr::rename(freq2020 = freq, rel_freq2020 = rel_freq) %>%
  tibble::as_tibble() %>%
  dplyr::mutate(
    id = as.numeric(as.character(id)),
    layer = as.numeric(as.character(layer))
  )

# calculate isolation - 2020
pol_1985_enh <- landscapemetrics::scale_sample(forest_2020, pol,
  shape = "circle",
  size = buffers * res_orig / res_m,
  what = c("lsm_c_enh_mn"),
  verbose = TRUE,
  progress = TRUE
) %>%
  dplyr::mutate(size = size / res_orig * res_m)

# subset and rename columns
pol_2020_enh <- pol_2020_enh %>%

```

```

dplyr::select(
  id = plot_id, layer = class, metric2020 = metric,
  value2020 = value, buffer = size
)

# join
pol_2020 <- pol_2020 %>%
  dplyr::left_join(pol_2020_enn,
    by = c("id", "layer", "buffer")
  )

# check
all(pol_1985$id == pol_2020$id)

# calculate changes
pol_forest_change <- pol_1985 %>%
  dplyr::bind_cols(
    pol_2020 %>%
      dplyr::select(contains("2020"))
  ) %>%
  dplyr::mutate(
    change_area_ha = (freq2020 - freq1985) * res_m**2 / 10e4,
    change_area_prop = (rel_freq2020 - rel_freq1985),
    change_iso = value2020 - value1985,
    change_iso_prop = (value2020 - value1985) / value1985
  )

# export table of changes
pol_forest_change %>%
  readr::write_csv(file = "output/polinization_forest_change.csv")

```

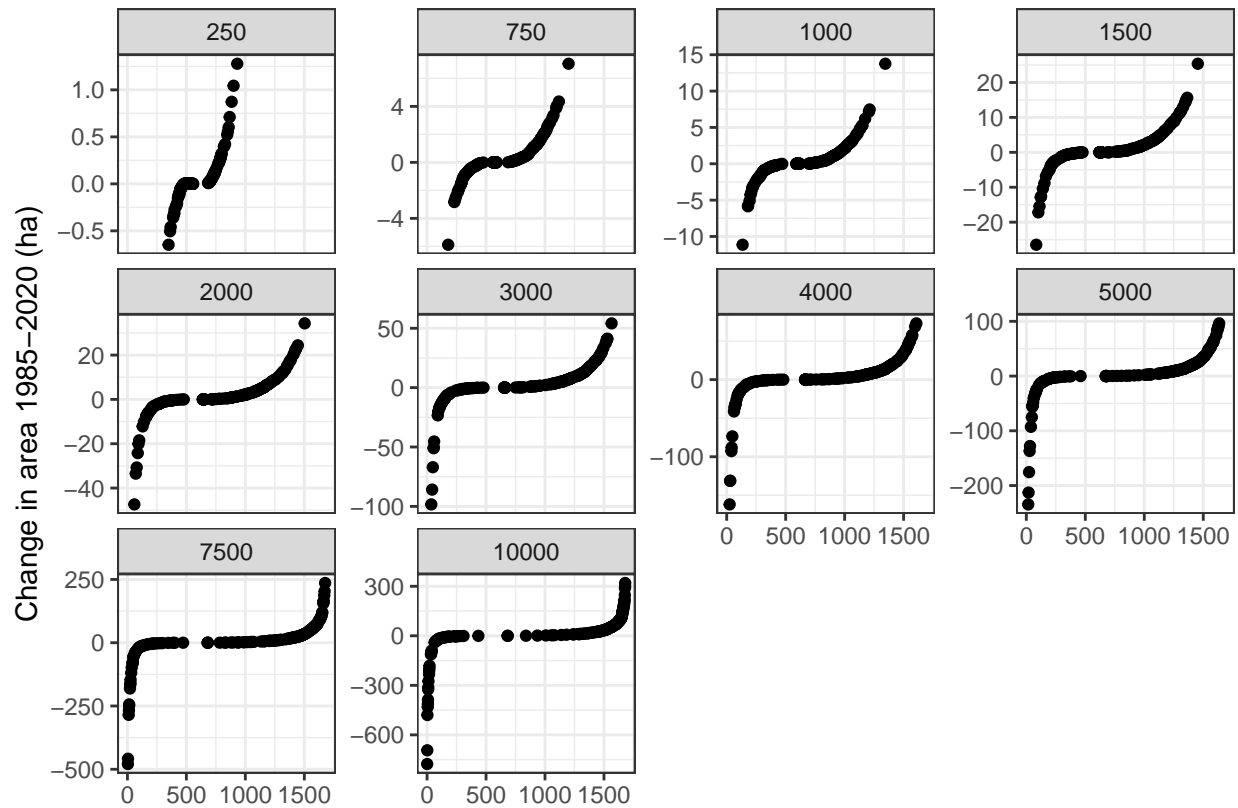
Plot and identify changes

Below we plot the absolute change in forest area (in hectares) and the change in the proportion of forests. Please note that the y axes follow different scales. In both plots the points were sorted according to the value of absolute change in forest area (in the x axis).

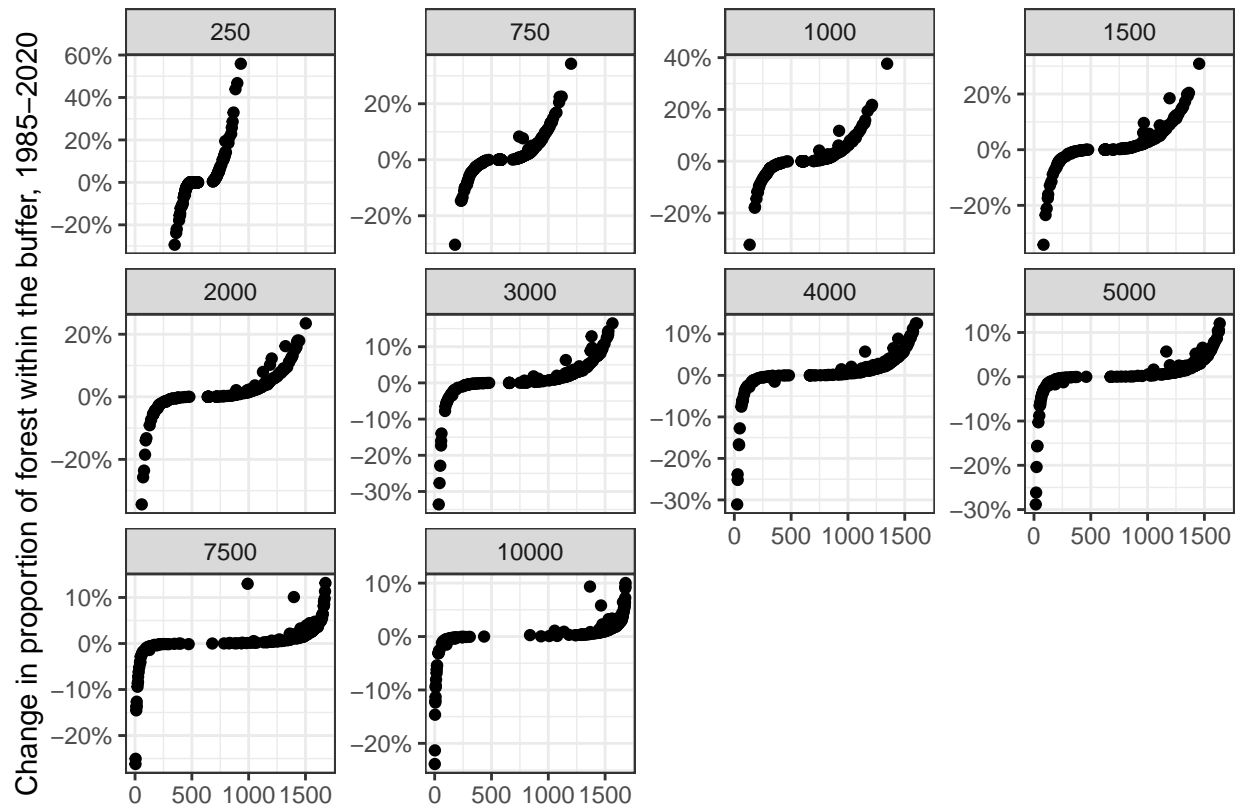
```

# plot - total area
pol_forest_change %>%
  dplyr::filter(layer == 1) %>%
  dplyr::arrange(change_area_ha) %>%
  tibble::rowid_to_column(var = "rown") %>%
  ggplot(aes(x = rown, y = change_area_ha)) +
  geom_point() +
  facet_wrap(~buffer, scales = "free_y") +
  labs(y = "Change in area 1985-2020 (ha)", x = "") +
  theme_bw()

```

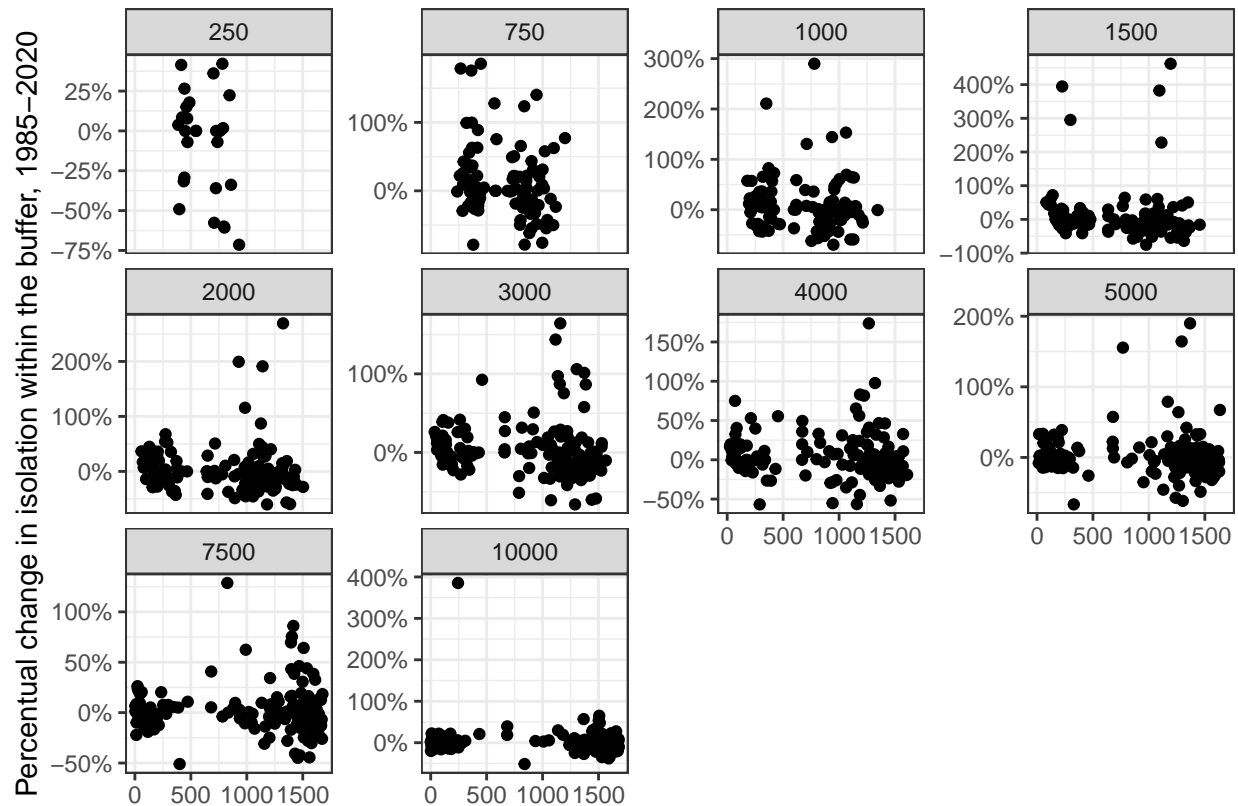


```
# plot - proportion area
pol_forest_change %>%
  dplyr::filter(layer == 1) %>%
  dplyr::arrange(change_area_ha) %>%
  tibble::rowid_to_column(var = "rown") %>%
  ggplot(aes(x = rown, y = change_area_prop)) +
  geom_point() +
  facet_wrap(~buffer, scales = "free_y") +
  labs(y = "Change in proportion of forest within the buffer, 1985-2020", x = "") +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  theme_bw()
```



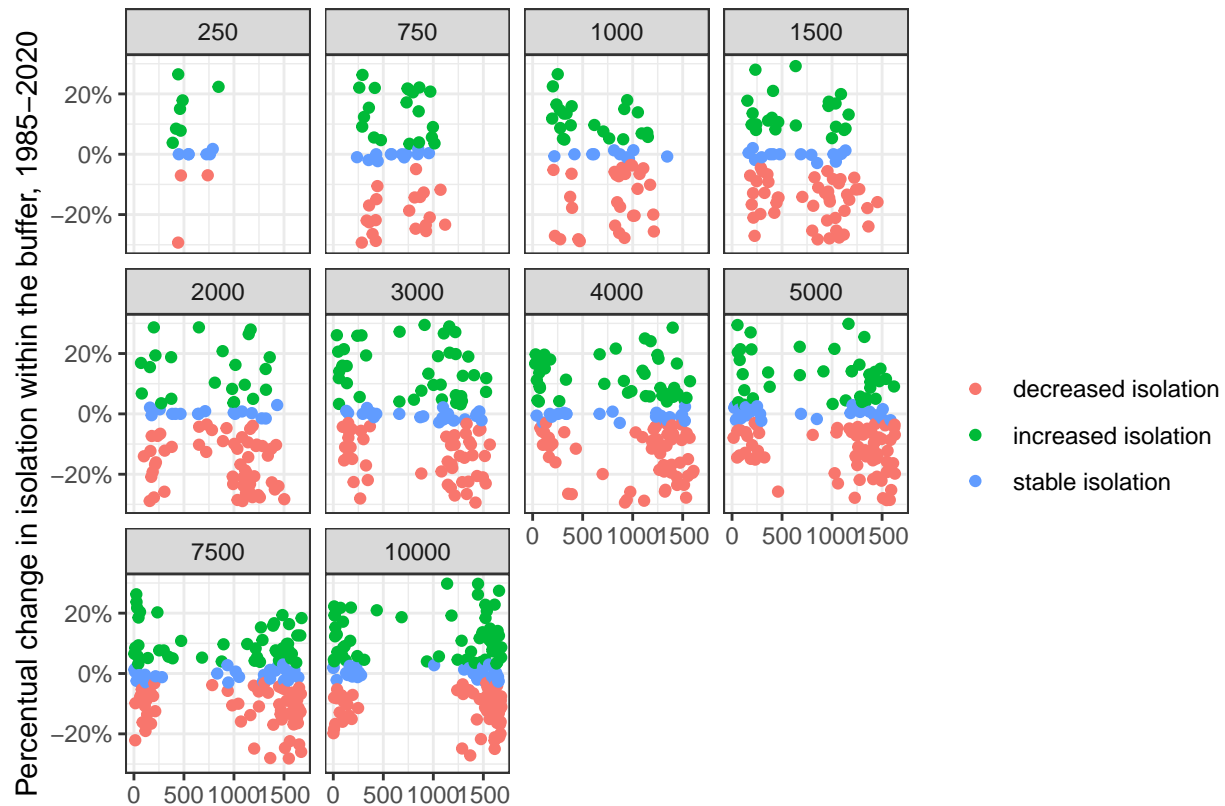
Now we make similar plots for the changes in isolation at each landscape extent.

```
# plot - proportion isolation
pol_forest_change %>%
  dplyr::filter(layer == 1) %>%
  dplyr::arrange(change_area_ha) %>%
  tibble::rowid_to_column(var = "rown") %>%
  ggplot(aes(x = rown, y = change_iso_prop)) +
  geom_point() +
  facet_wrap(~buffer, scales = "free_y") +
  labs(y = "Percentual change in isolation within the buffer, 1985-2020", x = "") +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  theme_bw()
```



The following plot shows a zoomed version of the previous one, focusing on the changes < than 30%. The different colors show points with increased isolation (change in isolation > 3%), decreased isolation (change in isolation < -3%), and points with stable isolation (absolute change in isolation < 3%).

```
# plot - proportion isolation - a closer look
pol_forest_change %>%
  dplyr::filter(layer == 1) %>%
  dplyr::arrange(change_area_ha) %>%
  tibble::rowid_to_column(var = "rown") %>%
  ggplot(aes(
    x = rown, y = change_iso_prop,
    color = case_when(
      change_iso_prop > 0.03 ~ "increased isolation",
      change_iso_prop < -0.03 ~ "decreased isolation",
      TRUE ~ "stable isolation"
    )
  )) +
  geom_point() +
  facet_wrap(~buffer) +
  labs(
    y = "Percentual change in isolation within the buffer, 1985-2020", x = "",
    color = ""
  ) +
  scale_y_continuous(limits = c(-0.3, 0.3), labels = scales::percent_format(accuracy = 1)) +
  theme_bw()
```



Join with spatial data and save output

Now we join this information with the sampling point positions and export it as a csv, as well as a shapefile. We classify all points where the proportion of forest change was $< 3\%$ (at the highest extent, 10 km buffer, when comparing 1985 and 2020) as stable, and the others as “forest loss” and “forest gain”. The same is done for isolation: all points with a proportional change in isolation $< 3\%$ (at the 10 km extent) were considered as “stable isolation”, and the others were classified as either increased isolation or decreased isolation. Finally, the sampling points were classified according to both forest amount and isolation change.

```
# join tables
pol <- pol %>%
  dplyr::left_join(
    pol_forest_change %>%
      dplyr::filter(layer == 1 & buffer == 10000) %>%
      dplyr::select(-layer) %>%
      dplyr::mutate(id = as.numeric(id)),
    by = c("points" = "id")
  ) %>%
  dplyr::mutate(
    change_amount = case_when(
      change_area_prop < -0.03 ~ "forest loss",
      change_area_prop < 0.03 ~ "stable forest",
      TRUE ~ "forest gain"
    ),
    change_isolation = case_when(
      change_iso_prop > 0.03 ~ "increased isolation",
      change_iso_prop < -0.03 ~ "decreased isolation",
      TRUE ~ "stable isolation"
    )
  )
```



```

    ),
    change_amount_iso = case_when(
      (change_amount == "forest loss" & change_isolation == "increased isolation") ~ "Forest cover loss and increased isolation",
      (change_amount == "forest loss" & change_isolation != "increased isolation") ~ "Forest cover loss and no increased isolation",
      (change_amount == "stable forest" & change_isolation == "increased isolation") ~ "Stable forest cover and increased isolation",
      (change_amount == "stable forest" & change_isolation != "increased isolation") ~ "Stable forest cover and no increased isolation",
      (change_amount == "forest gain" & change_isolation == "decreased isolation") ~ "Forest cover gain and decreased isolation",
      (change_amount == "forest gain" & change_isolation != "decreased isolation") ~ "Forest cover gain and no decreased isolation"
    )
  )
)

# How many forest loss points do we have?
pol %>%
  dplyr::filter(change_amount == "forest loss") %>%
  nrow()

```

```
## [1] 16
```

Here we see we have only 16 points with forest loss. But considering the possibility of increased isolation, there are more points with combinations between changes in forest amount and isolation, that might be interesting for the analyses:

```
table(pol$change_amount_iso)
```

```
##
## Forest cover gain and no reduction in isolation
##                               15
##   Forest cover gain and reduction in isolation
##                               22
##   Forest cover loss and increased isolation
##                               8
##   Forest cover loss and no increased isolation
##                               8
##   Stable forest cover and increased isolation
##                               42
##   Stable forest cover and no increased isolation
##                               73

```

```

# save shapefile
sf::st_write(pol,
  dsn = "output/Atlantic_forest_floral_visitor_data_points_forest_change.gpkg",
  delete_dsn = TRUE
)

```

Plot spatial data

We finally plot the spatial data.

```

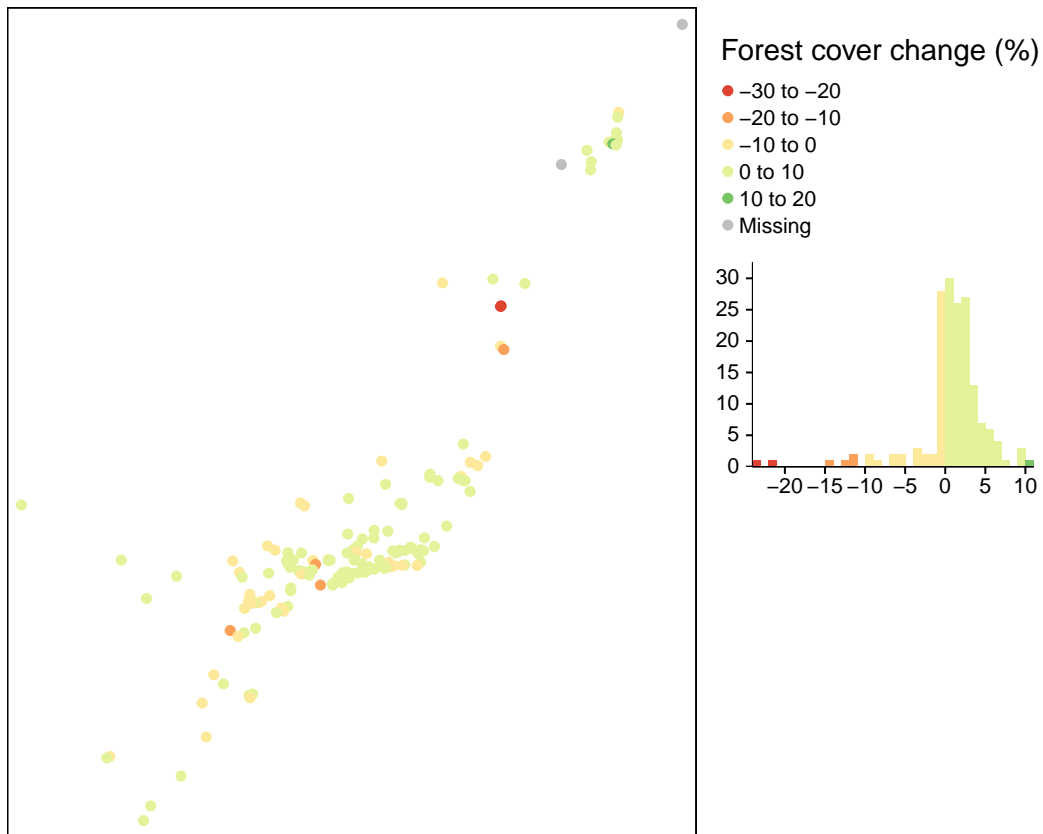
# forest change
pol %>%
  dplyr::mutate(change_area_prop = 100 * change_area_prop) %>%
  tm_shape() +
  tm_dots(
    col = "change_area_prop", size = 0.1,
    legend.hist = T, n = 4,
    title = "Forest cover change (%)"
  ) +
  tm_layout(

```

```

legend.outside = T,
legend.hist.width = 1
)

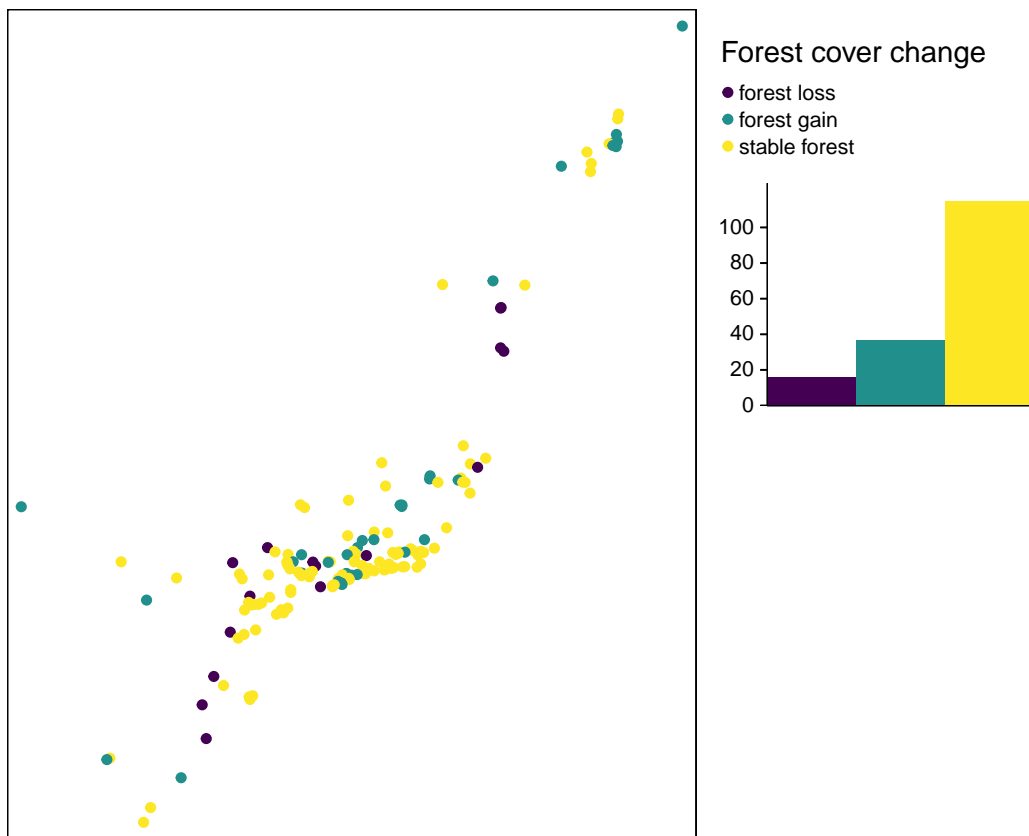
```



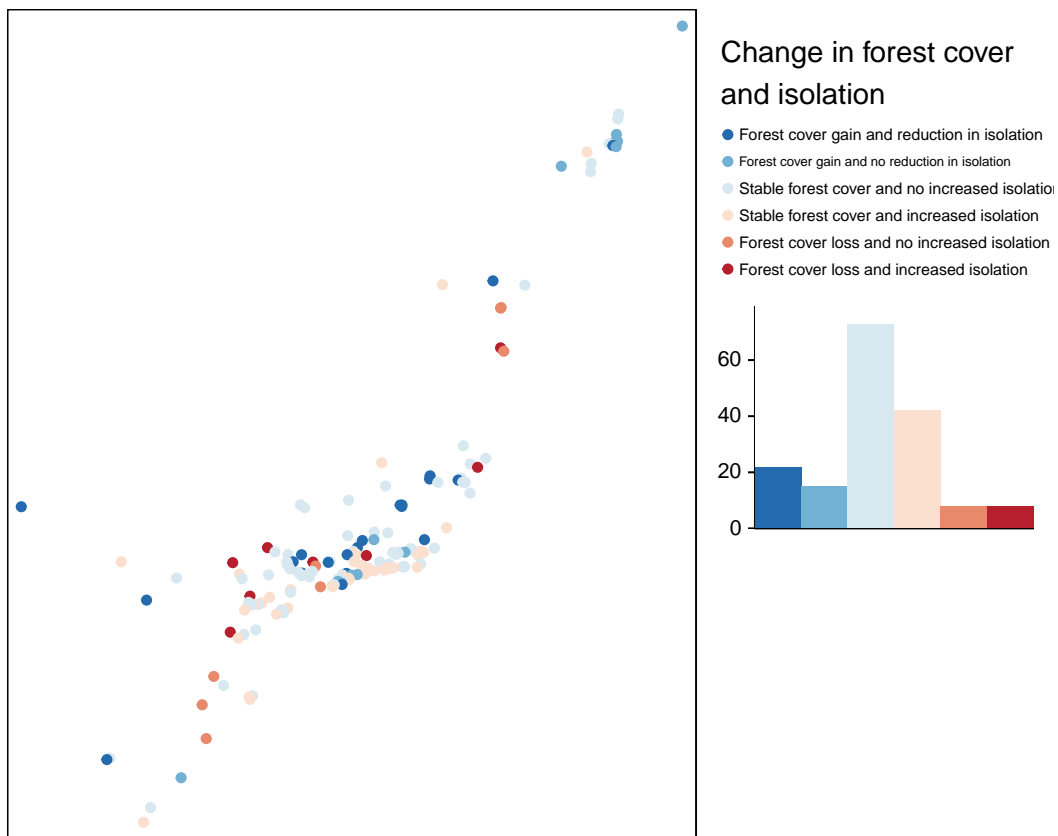
```

# forest amount
pol %>%
  dplyr::mutate(change_amount = factor(change_amount,
    levels = c(
      "forest loss", "forest gain",
      "stable forest"
    )
  )) %>%
  tm_shape() +
  tmap::tm_dots(
    col = "change_amount", size = 0.1,
    palette = "viridis",
    title = "Forest cover change",
    legend.hist = T
  ) +
  tm_layout(
    legend.outside = T,
    legend.hist.width = 1
  )

```



```
# forest amount and isolation
pol %>%
  dplyr::mutate(change_amount_iso = fct_relevel(
    change_amount_iso,
    function(x) x[c(2, 1, 6, 5, 4, 3)]
  )) %>%
  tm_shape() +
  tmap::tm_dots(
    col = "change_amount_iso", size = 0.1,
    palette = "-RdBu",
    title = "Change in forest cover\nand isolation",
    legend.hist = T
  ) +
  tm_layout(
    legend.outside = T,
    legend.hist.width = 1
  )
```



Furgivory networks

Now we do the same for frugivory networks.

Calculate metrics and changes

```
# frugivory

# calculate amount of forest - 1985
buffers <- c(250, 500, 750, 1000, 1500, 2000, 3000, 4000, 5000, 7500, 10000)
fru_1985 <- landscapetools::util_extract_multibuffer(forest_1985, fru,
  buffer_width = buffers,
  rel_freq = TRUE, point_id_text = FALSE
) %>%
  dplyr::rename(freq1985 = freq, rel_freq1985 = rel_freq) %>%
  tibble::as_tibble() %>%
  dplyr::mutate(
    id = as.numeric(as.character(id)),
    layer = as.numeric(as.character(layer))
  )

# calculate isolation - 1985
fru_1985_enh <- landscapemetrics::scale_sample(forest_1985, fru,
  shape = "circle",
  size = buffers * res_orig / res_m,
```

```

  what = c("lsm_c_enn_mn"),
  verbose = TRUE,
  progress = TRUE
) %>%
  dplyr::mutate(size = size / res_orig * res_m)

# subset and rename columns
fru_1985_enn <- fru_1985_enn %>%
  dplyr::select(
    id = plot_id, layer = class, metric1985 = metric,
    value1985 = value, buffer = size
  )

# join
fru_1985 <- fru_1985 %>%
  dplyr::left_join(fru_1985_enn,
    by = c("id", "layer", "buffer")
  )

# calculate forest amount - 2020
fru_2020 <- landscapetools::util_extract_multibuffer(forest_2020, fru,
  buffer_width = buffers,
  rel_freq = TRUE, point_id_text = FALSE
) %>%
  dplyr::rename(freq2020 = freq, rel_freq2020 = rel_freq) %>%
  tibble::as_tibble() %>%
  dplyr::mutate(
    id = as.numeric(as.character(id)),
    layer = as.numeric(as.character(layer))
  )

# calculate isolation - 2020
fru_2020_enn <- landscapemetrics::scale_sample(forest_2020, pol,
  shape = "circle",
  size = buffers * res_orig / res_m,
  what = c("lsm_c_enn_mn"),
  verbose = TRUE,
  progress = TRUE
) %>%
  dplyr::mutate(size = size / res_orig * res_m)

# subset and rename columns
fru_2020_enn <- fru_2020_enn %>%
  dplyr::select(
    id = plot_id, layer = class, metric2020 = metric,
    value2020 = value, buffer = size
  ) %>%
  tibble::as_tibble()

# join
fru_2020 <- fru_2020 %>%
  dplyr::left_join(fru_2020_enn,
    by = c("id", "layer", "buffer")
  )

# check

```

```

all(fru_1985$id == fru_2020$id)

# calculate changes
fru_forest_change <- fru_1985 %>%
  dplyr::bind_cols(
    fru_2020 %>%
      dplyr::select(contains("2020"))
  ) %>%
  dplyr::mutate(
    change_area_ha = (freq2020 - freq1985) * res_m**2 / 10e4,
    change_area_prop = (rel_freq2020 - rel_freq1985),
    change_iso = value2020 - value1985,
    change_iso_prop = (value2020 - value1985) / value1985
  )

# export table of changes
fru_forest_change %>%
  readr::write_csv(file = "output/frugivory_forest_change.csv")

```

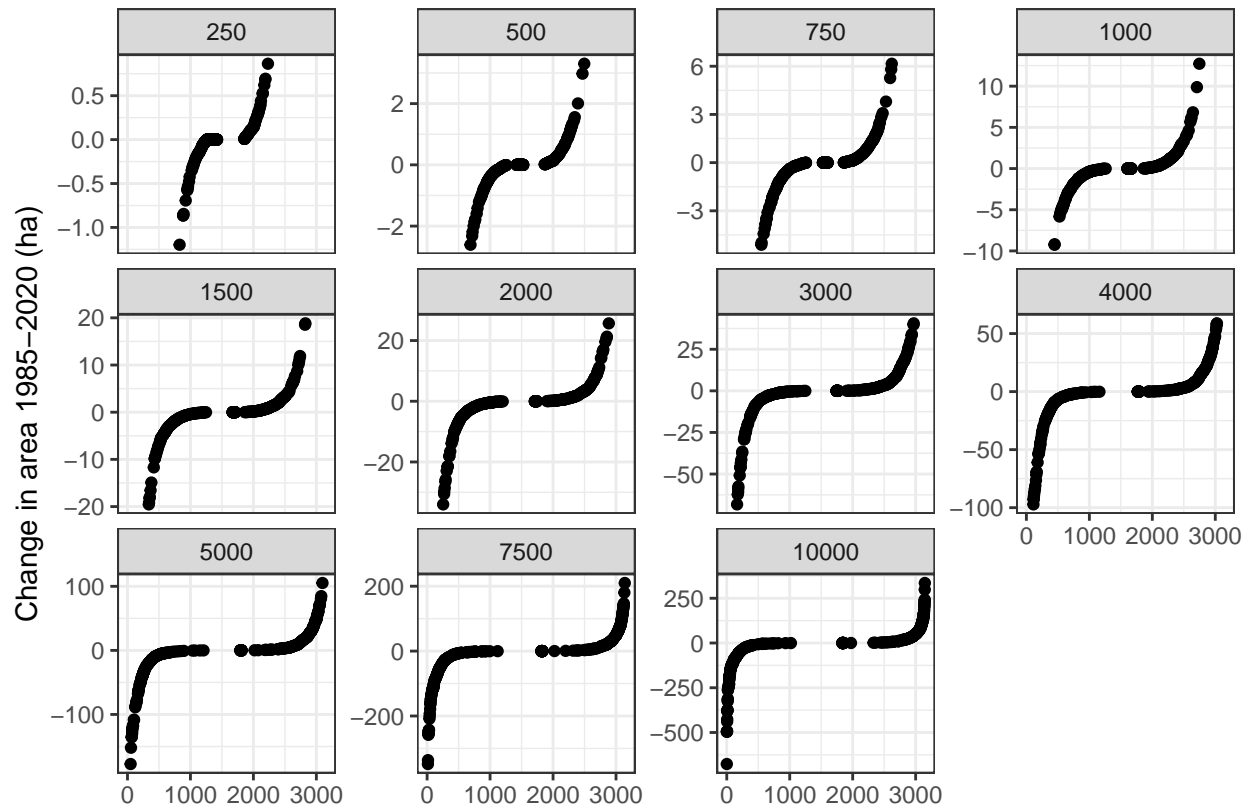
Plot and identify changes

Below we plot the absolute change in forest area (in hectares) and the change in the proportion of forests. Please note that the y axes follow different scales. In both plots the points were sorted according to the value of absolute change in forest area (in the x axis).

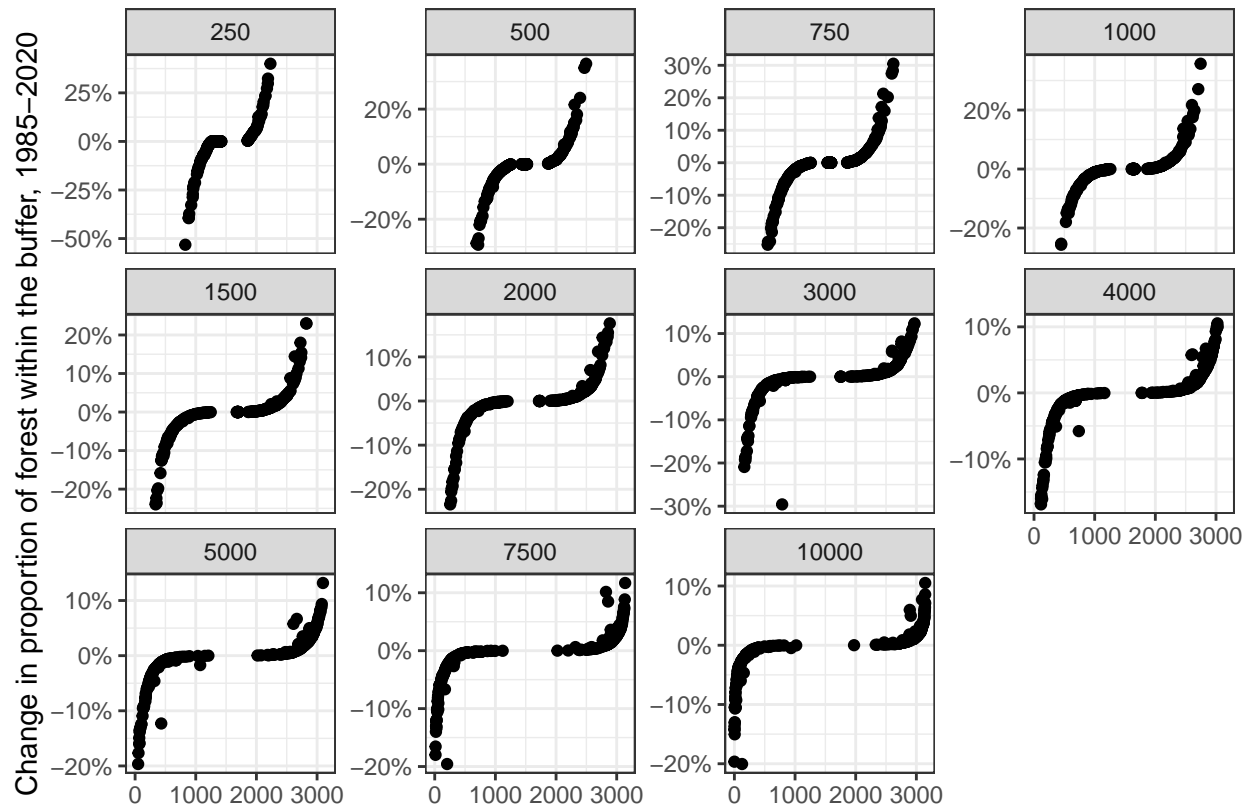
```

# plot - total area
fru_forest_change %>%
  dplyr::filter(layer == 1) %>%
  dplyr::arrange(change_area_ha) %>%
  tibble::rowid_to_column(var = "rown") %>%
  ggplot(aes(x = rown, y = change_area_ha)) +
  geom_point() +
  facet_wrap(~buffer, scales = "free_y") +
  labs(y = "Change in area 1985-2020 (ha)", x = "") +
  theme_bw()

```

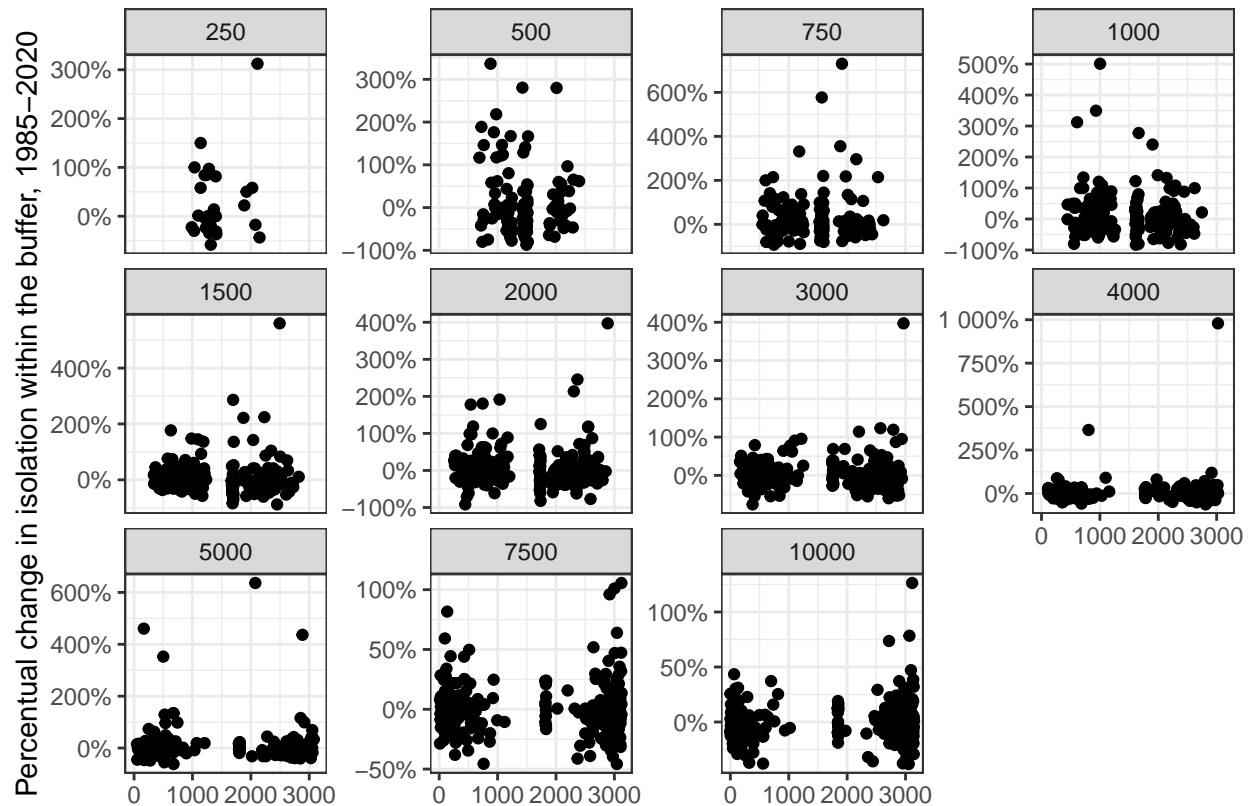


```
# plot - proportion
fru_forest_change %>%
  dplyr::filter(layer == 1) %>%
  dplyr::arrange(change_area_ha) %>%
  tibble::rowid_to_column(var = "rown") %>%
  ggplot(aes(x = rown, y = change_area_prop)) +
  geom_point() +
  facet_wrap(~buffer, scales = "free_y") +
  labs(y = "Change in proportion of forest within the buffer, 1985-2020", x = "") +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  theme_bw()
```



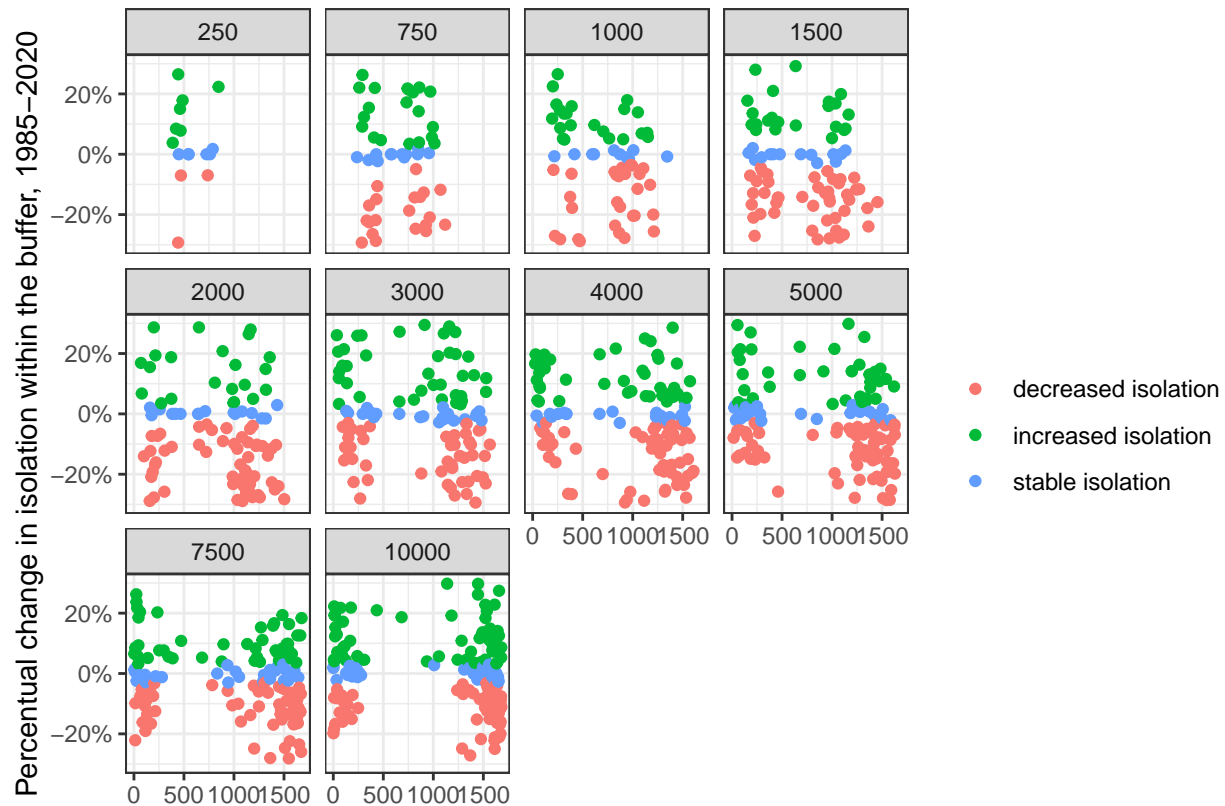
Now we make similar plots for the changes in isolation at each landscape extent.

```
# plot - proportion isolation
fru_forest_change %>%
  dplyr::filter(layer == 1) %>%
  dplyr::arrange(change_area_ha) %>%
  tibble::rowid_to_column(var = "rown") %>%
  ggplot(aes(x = rown, y = change_iso_prop)) +
  geom_point() +
  facet_wrap(~buffer, scales = "free_y") +
  labs(y = "Percentual change in isolation within the buffer, 1985-2020", x = "") +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  theme_bw()
```

The following plot shows a zoomed version of the previous one, focusing on the changes < than 30%. The different colors show points with increased isolation (change in isolation > 3%), decreased isolation (change in isolation < -3%), and points with stable isolation (absolute change in isolation < 3%).

```
# plot - proportion isolation - a closer look
pol_forest_change %>%
  dplyr::filter(layer == 1) %>%
  dplyr::arrange(change_area_ha) %>%
  tibble::rowid_to_column(var = "rown") %>%
  ggplot(aes(
    x = rown, y = change_iso_prop,
    color = case_when(
      change_iso_prop > 0.03 ~ "increased isolation",
      change_iso_prop < -0.03 ~ "decreased isolation",
      TRUE ~ "stable isolation"
    )
  )) +
  geom_point() +
  facet_wrap(~buffer) +
  labs(
    y = "Percentual change in isolation within the buffer, 1985-2020", x = "",
    color = ""
  ) +
  scale_y_continuous(limits = c(-0.3, 0.3), labels = scales::percent_format(accuracy = 1)) +
  theme_bw()
```



Join with spatial data and save output

Now we join this information with the sampling point positions and export it as a csv, as well as a shapefile. We classify all points where the proportion of forest change was $< 3\%$ (at the highest extent, 10 km buffer, when comparing 1985 and 2020) as stable, and the others as “forest loss” and “forest gain”. The same is done for isolation: all points with a proportional change in isolation $< 3\%$ (at the 10 km extent) were considered as “stable isolation”, and the others were classified as either increased isolation or decreased isolation. Finally, the sampling points were classified according to both forest amount and isolation change.

```
# join tables
fru <- fru %>%
  dplyr::left_join(
    fru_forest_change %>%
      dplyr::filter(layer == 1 & buffer == 10000) %>%
      dplyr::select(-layer) %>%
      dplyr::mutate(id = as.numeric(id)),
    by = c("points" = "id")
  ) %>%
  dplyr::mutate(
    change_amount = case_when(
      change_area_prop < -0.03 ~ "forest loss",
      change_area_prop < 0.03 ~ "stable forest",
      TRUE ~ "forest gain"
    ),
    change_isolation = case_when(
      change_iso_prop > 0.03 ~ "increased isolation",
      change_iso_prop < -0.03 ~ "decreased isolation",
      TRUE ~ "stable isolation"
    )
  )
```

```

    ),
    change_amount_iso = case_when(
      (change_amount == "forest loss" & change_isolation == "increased isolation") ~ "Forest cover loss and increased isolation",
      (change_amount == "forest loss" & change_isolation != "increased isolation") ~ "Forest cover loss and no increased isolation",
      (change_amount == "stable forest" & change_isolation == "increased isolation") ~ "Stable forest cover and increased isolation",
      (change_amount == "stable forest" & change_isolation != "increased isolation") ~ "Stable forest cover and no increased isolation",
      (change_amount == "forest gain" & change_isolation == "decreased isolation") ~ "Forest cover gain and decreased isolation",
      (change_amount == "forest gain" & change_isolation != "decreased isolation") ~ "Forest cover gain and no decreased isolation"
    )
  )
)

# How many forest loss points do we have?
fru %>%
  dplyr::filter(change_amount == "forest loss") %>%
  nrow()

```

```
## [1] 61
```

Here we see we have 61 points with forest loss in the frugivory dataset. But considering the possibility of change in isolation, we have these situation:

```
table(fru$change_amount_iso)
```

```
##
## Forest cover gain and no reduction in isolation
##                                42
##   Forest cover gain and reduction in isolation
##                                21
##   Forest cover loss and increased isolation
##                                19
##   Forest cover loss and no increased isolation
##                                42
##   Stable forest cover and increased isolation
##                                46
##   Stable forest cover and no increased isolation
##                                116

```

```

# save shapefile
sf::st_write(fru,
  dsn = "output/Atlantic_forest_frugivory_data_points_forest_change.gpkg",
  delete_dsn = TRUE
)

```

Plot spatial data

We finally plot the spatial data. Here there seems to be some outliers that we must either check or remove.

```

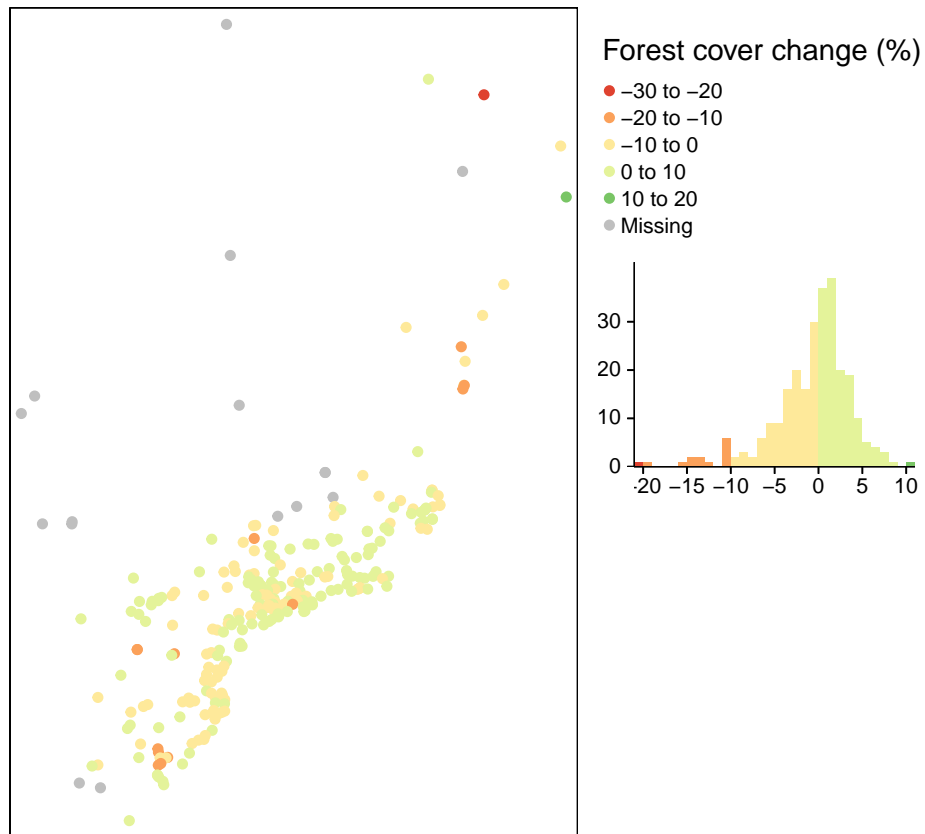
# forest change
fru %>%
  dplyr::mutate(change_area_prop = 100 * change_area_prop) %>%
  dplyr::filter(st_coordinates(.)[, 1] < -25) %>%
  tm_shape() +
  tm_dots(
    col = "change_area_prop", size = 0.1,
    legend.hist = T, n = 4,
    title = "Forest cover change (%)"
  ) +
  tm_layout(

```

```

legend.outside = T,
legend.hist.width = 1
)

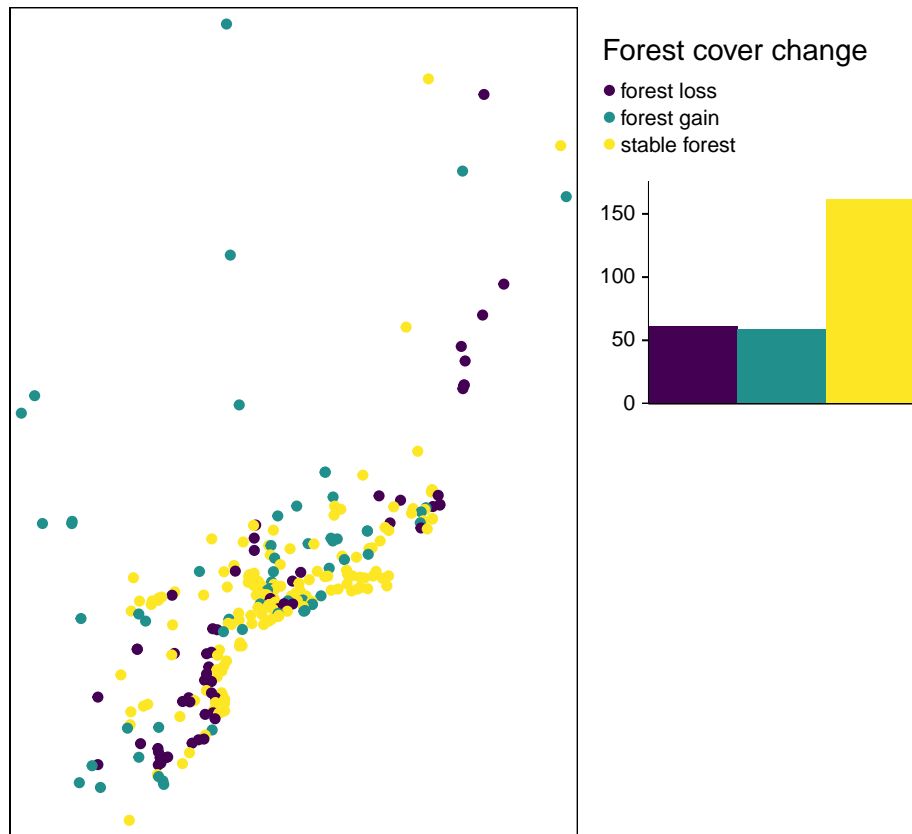
```



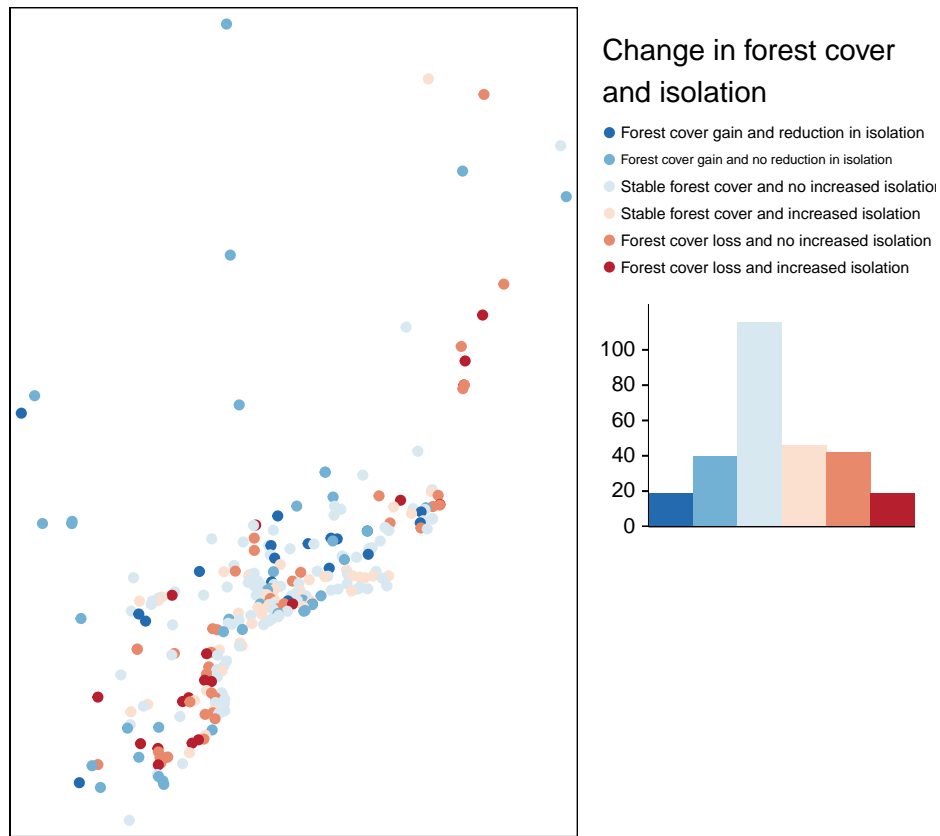
```

# forest amount
fru %>%
  dplyr::filter(st_coordinates(.)[, 1] < -25) %>%
  dplyr::mutate(change_amount = factor(change_amount,
    levels = c(
      "forest loss", "forest gain",
      "stable forest"
    )
  )) %>%
  tm_shape() +
  tmap::tm_dots(
    col = "change_amount", size = 0.1,
    palette = "viridis",
    title = "Forest cover change",
    legend.hist = T
  ) +
  tm_layout(
    legend.outside = T,
    legend.hist.width = 1
  )

```



```
# forest amount and isolation
fru %>%
  dplyr::filter(st_coordinates(.)[, 1] < -25) %>%
  dplyr::mutate(change_amount_iso = fct_relevel(
    change_amount_iso,
    function(x) x[c(2, 1, 6, 5, 4, 3)]
  )) %>%
  tm_shape() +
  tmap::tm_dots(
    col = "change_amount_iso", size = 0.1,
    palette = "-RdBu",
    title = "Change in forest cover\nand isolation",
    legend.hist = T
  ) +
  tm_layout(
    legend.outside = T,
    legend.hist.width = 1
  )
```



A few things to check

- There are some points that did not overlap with the map:
 - Pollination: point ID 152
 - Frugivory: points ID 1, 135, 159, 281, 283
- There are points with NA for isolation or isolation change - we should check.
- Any of those points above might be a problem in coding - one more reason to check.
- I did not classify all combinations of change in forest cover and isolation (e.g. forest loss with decrease in isolation). We can add that, if interesting.