

Analyses v2

Cape vs SWA

Ruan van Mazijk

2019-06-11

Preamble/outline

Here I layout the “new”, second incarnation of the analyses as discussed over the course of May/June 2019, following the first draft of the manuscript.

The proposed “story” of questions for the analyses as follows:

1. *Is SWA more heterogeneous environmentally than the Cape, and does the scale of that heterogeneity differ to that of the Cape?*
2. *Do the regions differ w.r.t. the species richness of both HDS and QDS cells, and, for HDS cells’ richness (S_{HDS}), does the explanatory power of mean QDS richness (S_{QDS}) and turnover (T_{QDS}) differ between the regions?*
3. *Does heterogeneity explain differences in richness and turnover between the regions?*

1. Environmental heterogeneity & scale

Is the Cape for environmental heterogeneity more heterogeneous than SWA, and does the scale of that heterogeneity differ to that of SWA?

For this question, environmental roughness in both regions was calculated, in moving 3 x 3 cell windows, as the average absolute difference between cells and their (usually) 8 neighbours. Alternatively, for a focal cell x^* , the roughness is based on $x_1, x_2, \dots, x_i, \dots, x_8$ neighbour cells as:

$$Roughness(x^*) = f \begin{pmatrix} x_1 & x_2 & x_3 \\ x_4 & x^* & x_5 \\ x_6 & x_7 & x_8 \end{pmatrix} = \frac{1}{n} \sum_{i=1}^n |x^* - x_i|$$

In R, I have implemented this as follows:

```
roughness <- function(x) {  
  raster::focal(x, matrix(1, nrow = 3, ncol = 3), function(x) {  
    focal_cell <- x[5]  
    neighbour_cells <- x[  
      !is.na(x) &  
      !is.nan(x) &  
      x != focal_cell  
    ]  
    ifelse(!is.na(focal_cell) & !is.nan(focal_cell),  
           mean(abs(focal_cell - neighbour_cells)),  
           NA  
    )  
  })  
}
```

1.1. Ordinating environmental heterogeneity

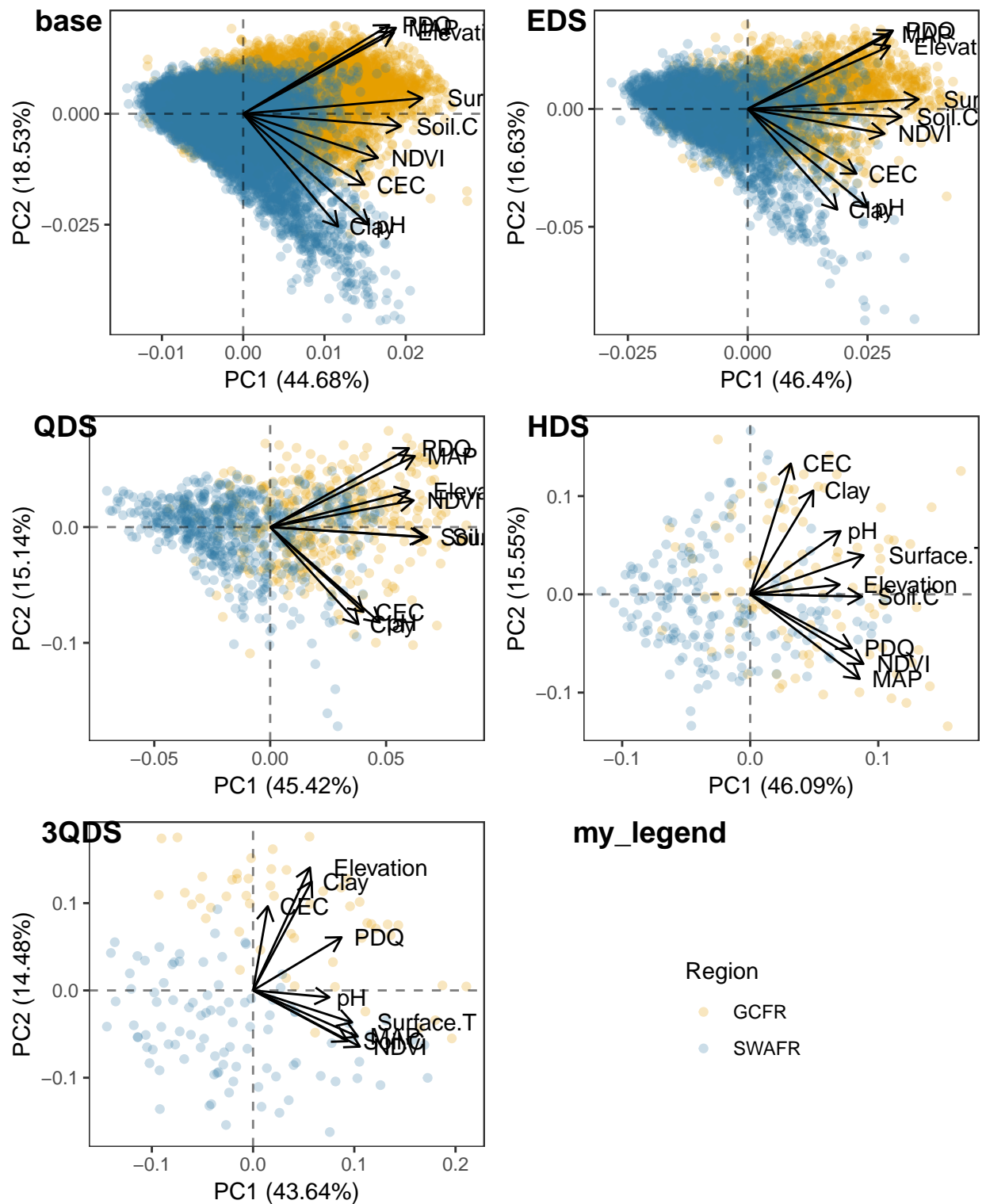
Here, I $\log(x + 1)$ -transform and perform a scaled and centred PCA of the roughness values as defined above, at each of five spatial scales: the base data resolution ($0.05^\circ \times 0.05^\circ$), eighth- (EDS), quarter- (QDS), half- (HDS) and three-quarter-degree-squares (3QDS).

The data matrices for each scale for this PCA look like this (and obviously the “region” column was excluded from the PCA proper):

Table 1: Portions of the data matrices used in the PCA for this analysis.

region	Elevation	MAP	PDQ	Surface.T	NDVI	CEC	Clay	Soil.C	pH
GCFR	5.19	2.52	0.72	1.32	15.13	1.14	1.2	2.46	1.36
GCFR	5	2.7	0.61	1.16	15.01	1.11	1.11	1.74	1.83
GCFR	4.86	2.55	0.72	1.17	15.08	1.18	1.4	1.79	1.65
GCFR	4.44	2.58	0.89	1.1	14.83	1.32	1.12	2.12	2.03
GCFR	4.62	1.98	0.91	0.71	14.35	1.35	0.86	1.54	1.46
GCFR	4.15	2.23	0.62	1.04	13.64	1	0.99	1.21	1.33
...
SWAFR	3.27	2.77	1.1	0.71	14.91	0.31	1.19	1.59	0.48
SWAFR	2.36	2.41	1.15	0.7	14.28	0.67	1.29	2.03	1.3
SWAFR	2.86	1.98	1.17	1.09	13.58	0.73	2.27	2.4	2.58
SWAFR	2.29	2.65	1.09	1	14.83	0.58	1.3	3.04	1.58
SWAFR	2.61	3.22	1.44	0.61	15.34	0.5	1.6	2.26	1.89
SWAFR	2.39	2.56	1.78	0.49	16.07	0.45	2.64	2	3.14

Plot PC1 vs PC2 at each scale:



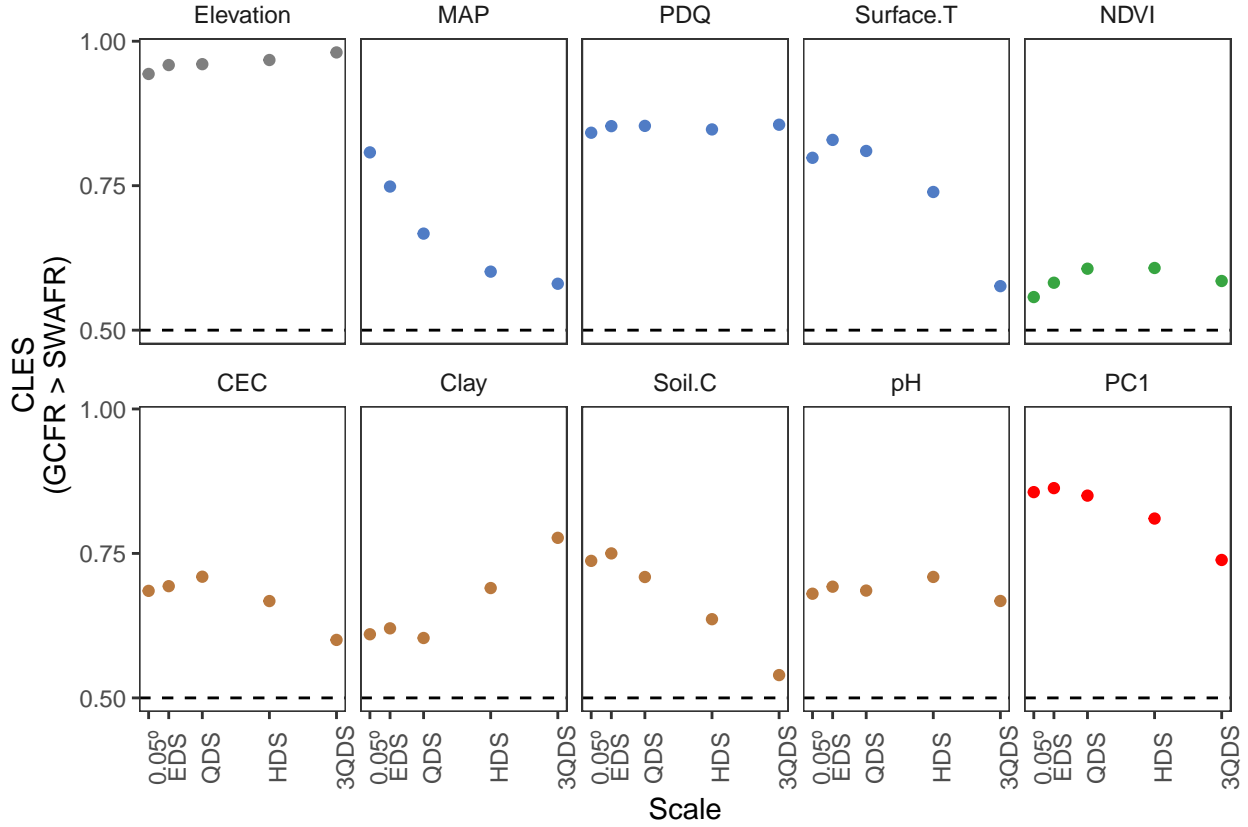
Plot maps of exemplary roughness variables and PC1

TODO

1.2. Effect size of Cape vs SWA heterogeneity

I calculated the *CLES* of Cape > SWA untransformed roughness and PC1 values (from the analysis above).

Let's plot *CLES* vs scale for each variable:



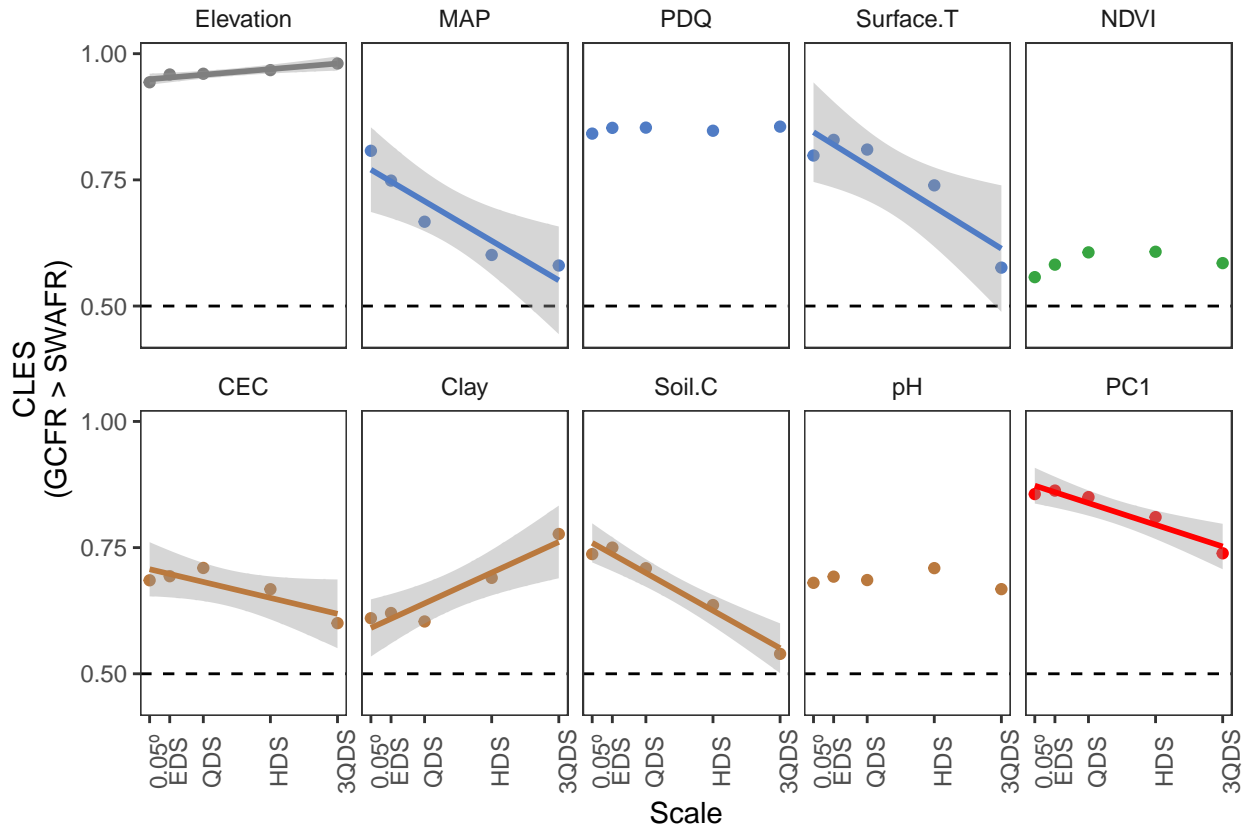
Let's regress *CLES* against scale for each of these panels:

Table 2: Slopes, significances and R^2 -values from regressions of *CLES* against each form of environmental roughness.

variable	slope	p.value	r.squared	sig	slope_sign
MAP	-0.313	0.020	0.874	*	-
Surface.T	-0.330	0.026	0.848	*	-
Soil.C	-0.298	0.003	0.967	*	-
PC1	-0.172	0.010	0.922	*	-
Elevation	0.044	0.016	0.891	*	+
Clay	0.243	0.013	0.902	*	+
CEC	-0.126	0.063	0.735	.	-
PDQ	0.010	0.387	0.253		
NDVI	0.032	0.459	0.193		
pH	-0.010	0.756	0.037		

CEC is on the verge of significant, so I will still plot it ($P = 0.063$). Let's plot those significant fits then:

Warning: Removed 15 rows containing non-finite values (stat_smooth).



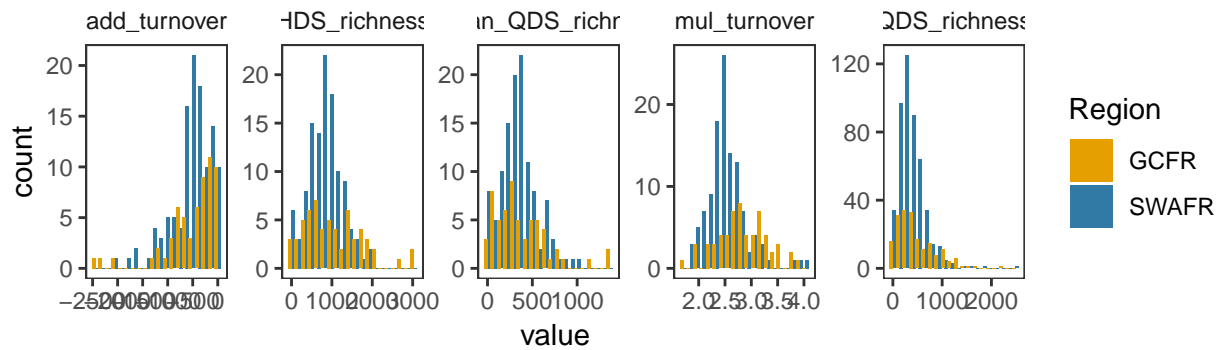
2. Species richness & turnover

Do the regions differ w.r.t. the species richness of both HDS and QDS cells, and, for HDS cells' richness (S_{HDS}), does the explanatory power of mean QDS richness (S_{QDS}) and turnover (T_{QDS}) differ between the regions?

To tackle this question, I compare measures of species richness and turnover between the regions in both univariate and multivariate/model analyses.

2.1. Univariate comparisons

What do the data look like?



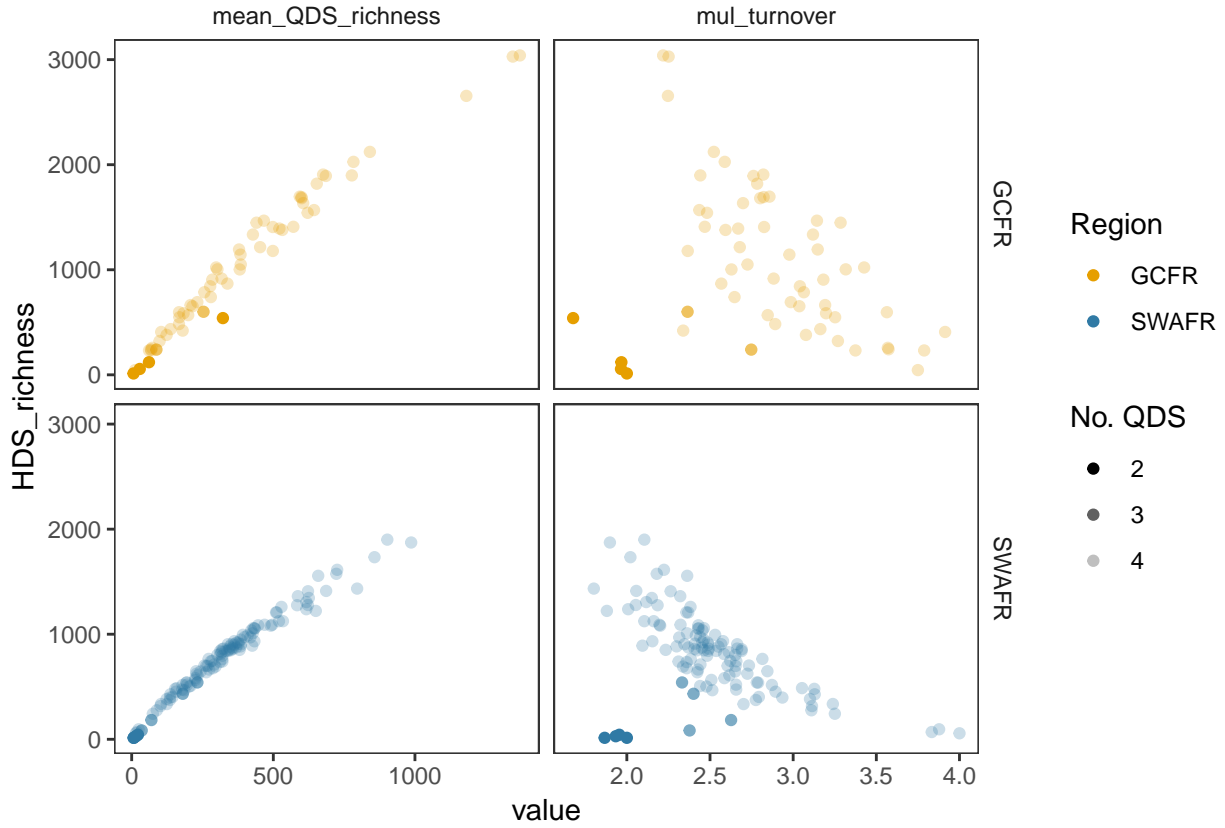
Are there significant differences between GCFR and SWAFR values?

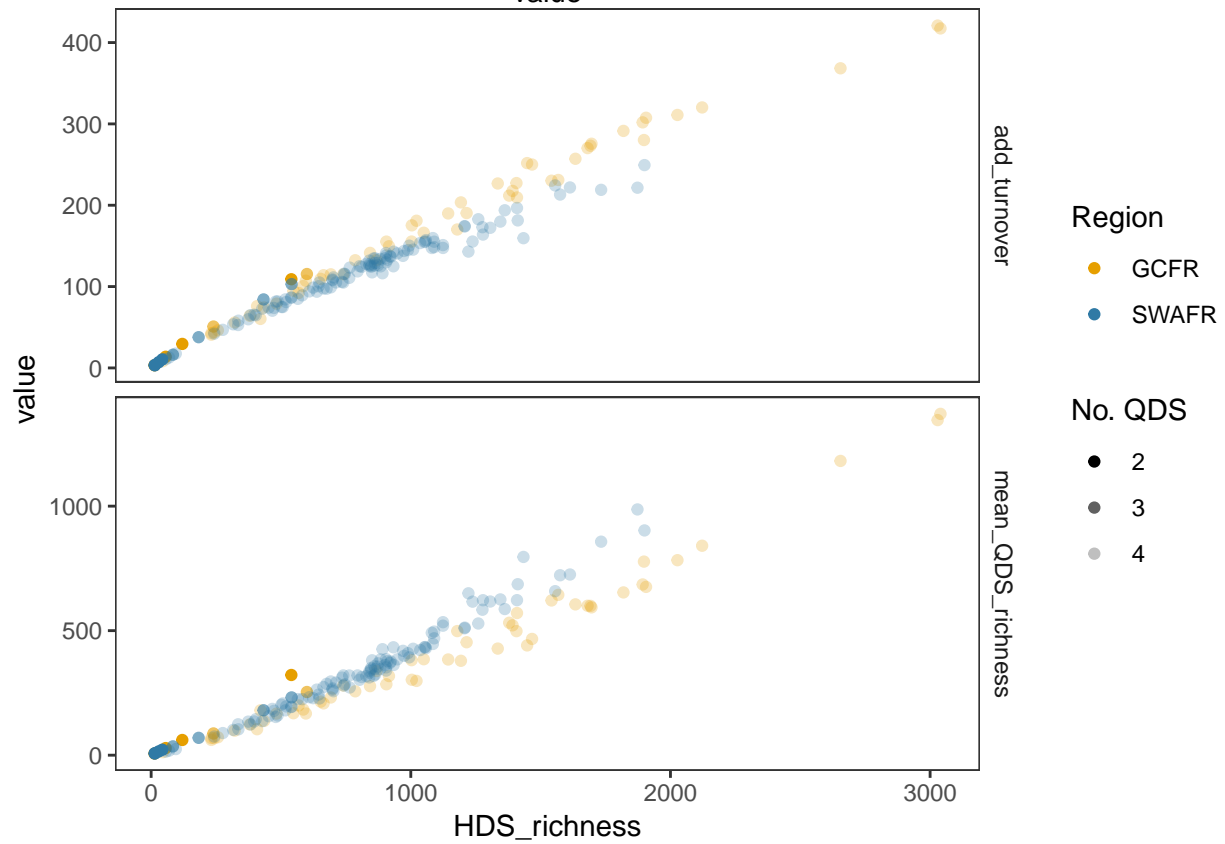
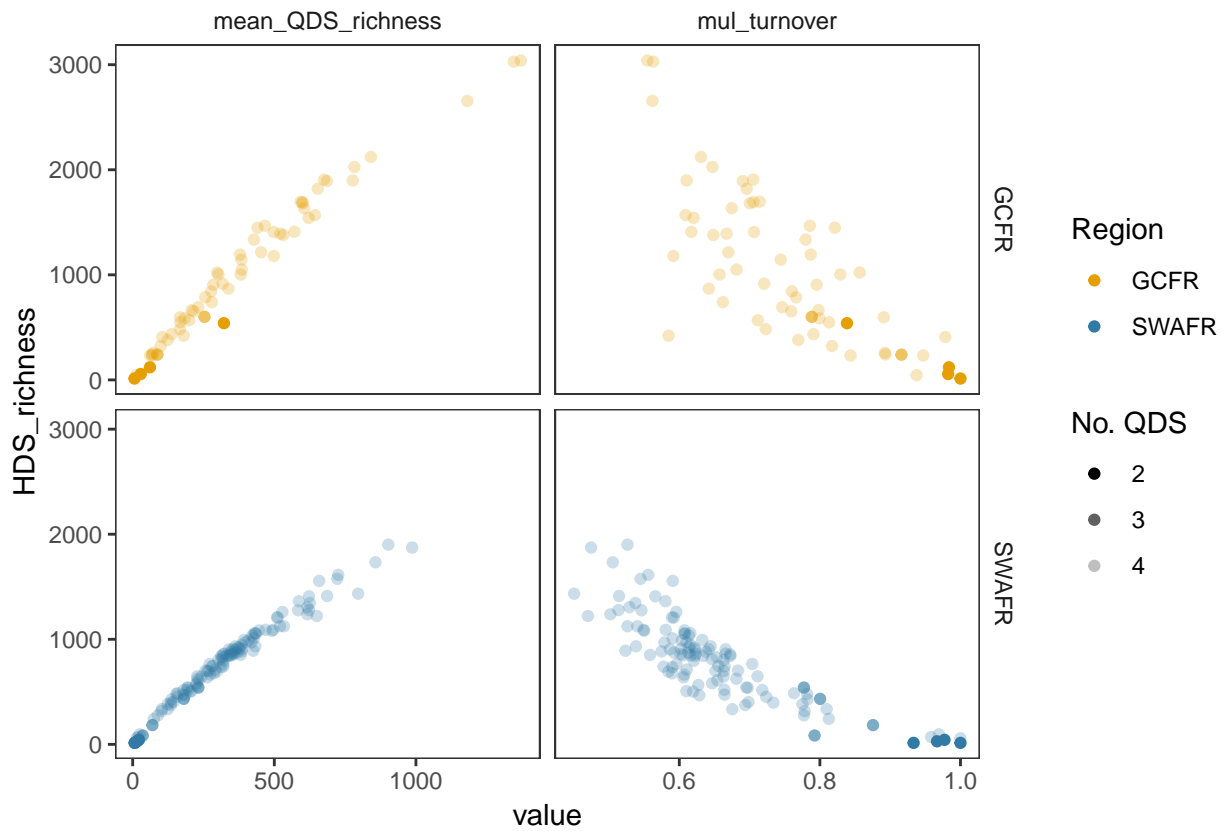
Table 3: Results of Mann-Whitney U -tests and the $CLES$ of GCFR vs SWAFR QDS and HDS richness and multiplicative turnover.

metric	U	P	sig	CLES_value
add_turnover	2918.5	0.0950755		0.5762319
HDS_richness	2864.5	0.0659228		0.5844928
mean_QDS_richness	3302.5	0.6440151		0.5210145
mul_turnover	1810.5	0.0000003	*	0.7375362
QDS_richness	38845.5	0.0019324	*	0.5761360

2.2. Multivariate comparisons & models

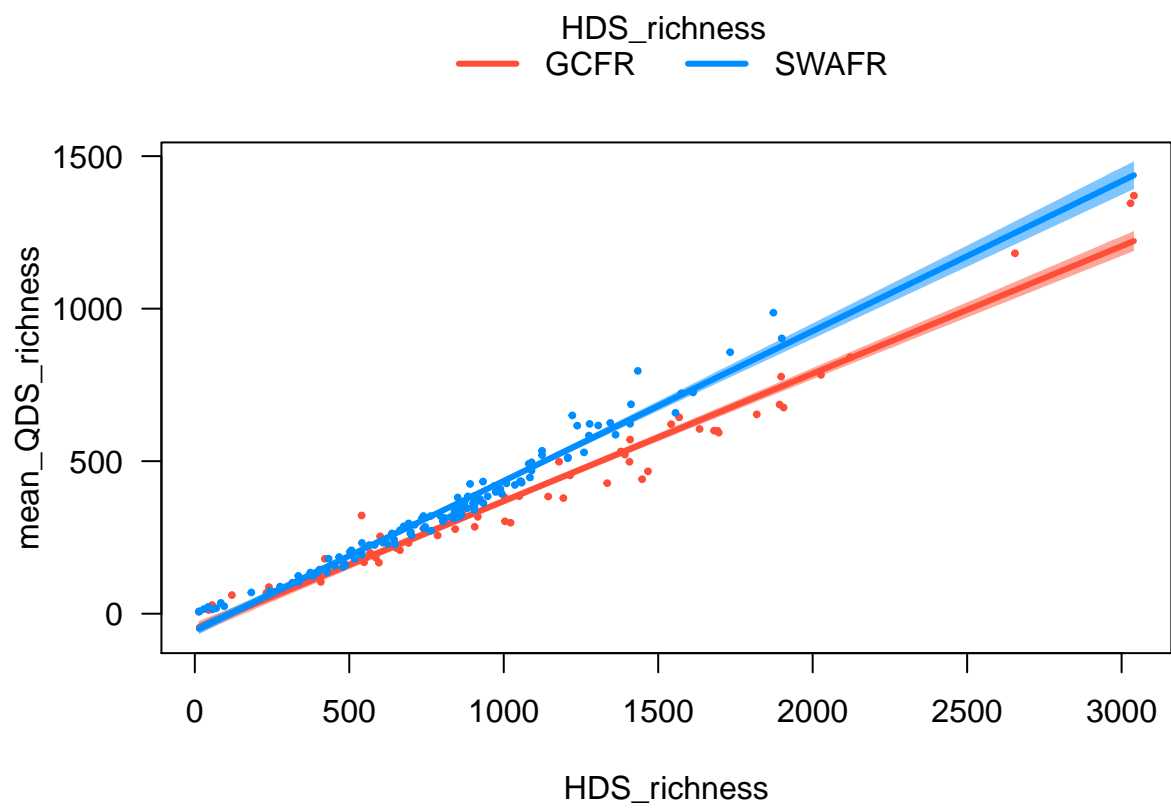
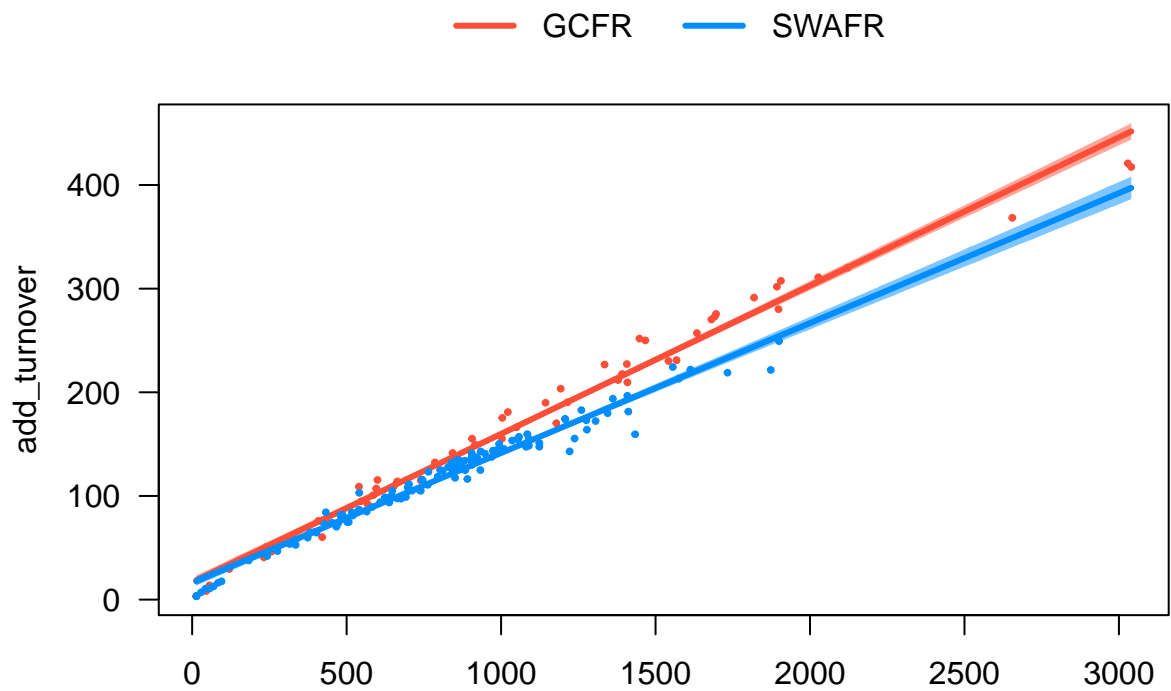
Before I get stuck in with modelling HDS richness as a function of QDS richness and turnover, let's wrap our heads around the dataset first, with special attention to how the number of constituent QDS in HDS affects the patterns:

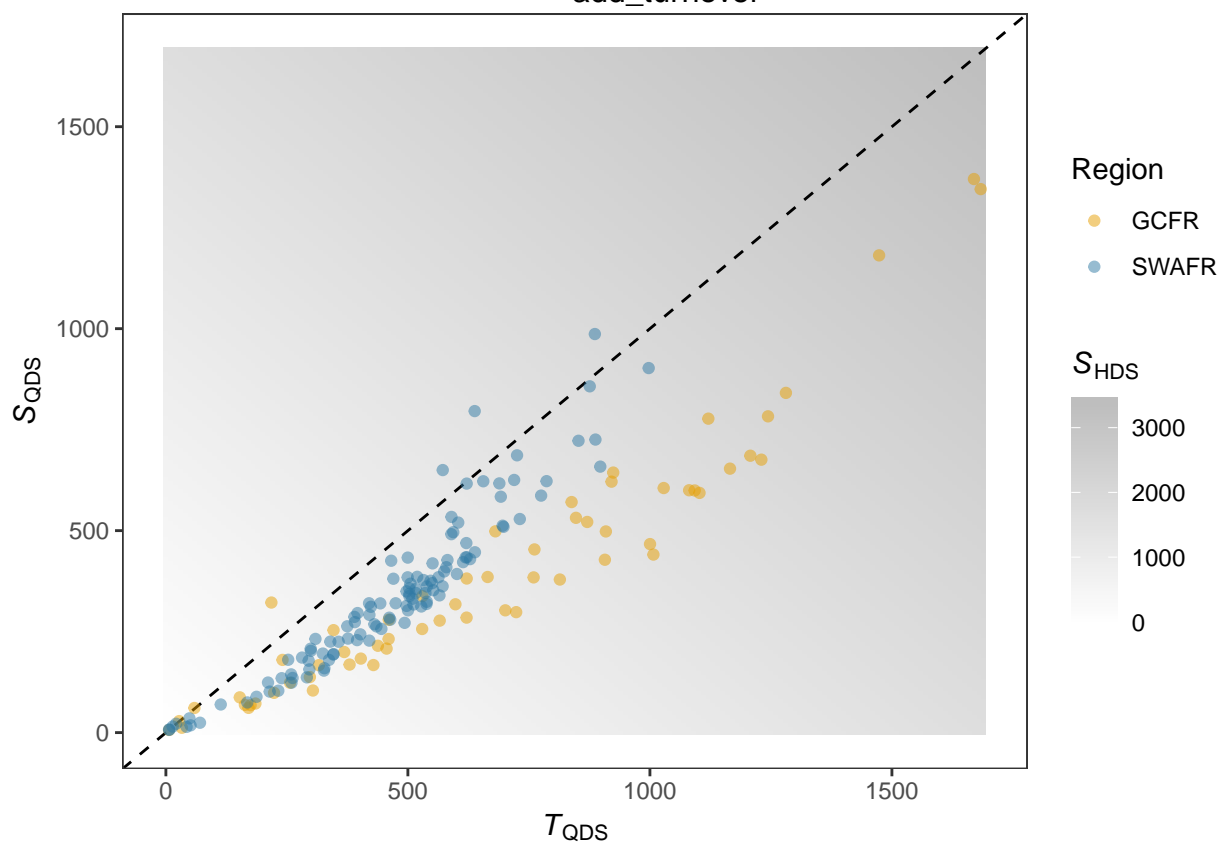
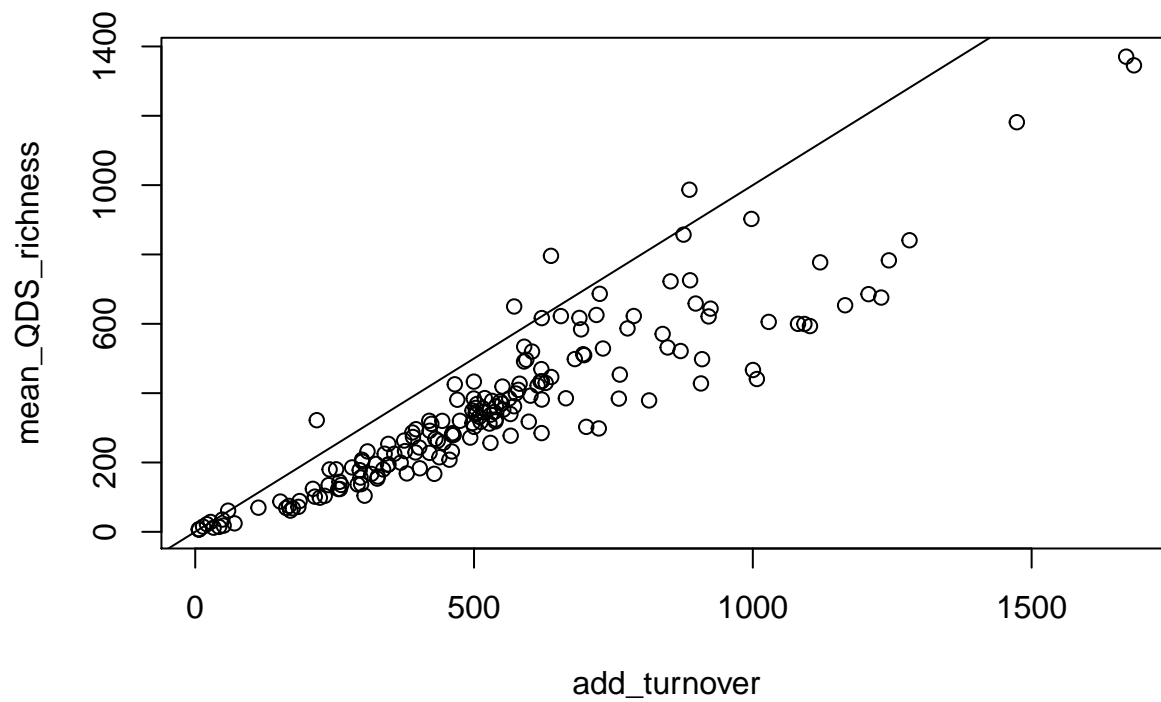


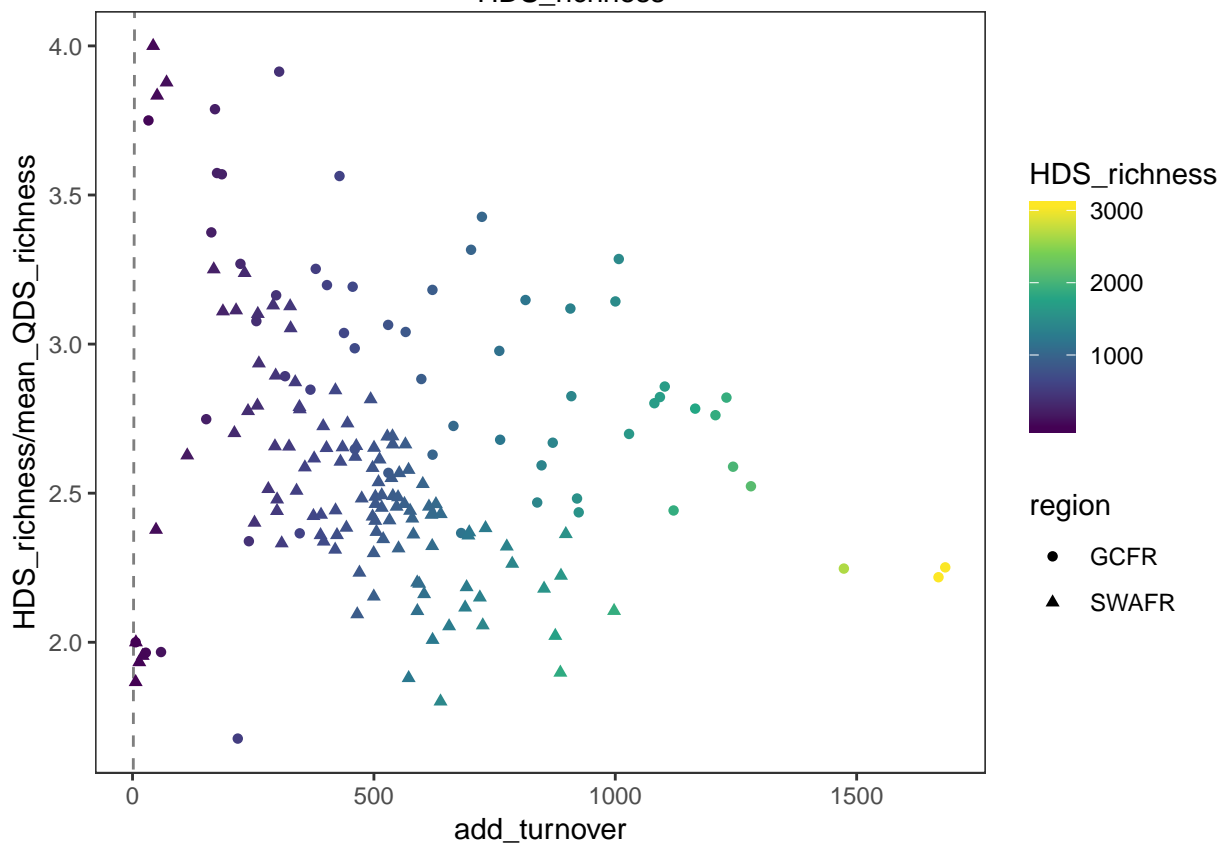
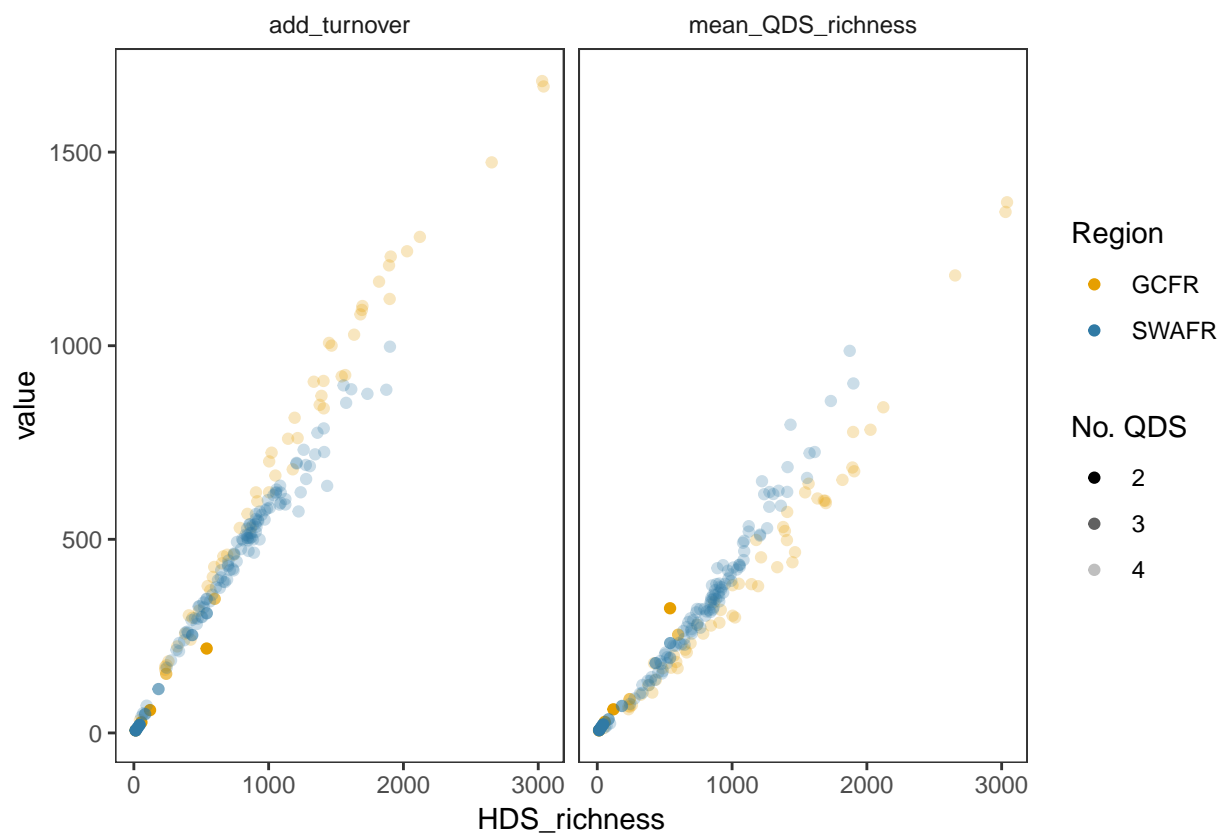


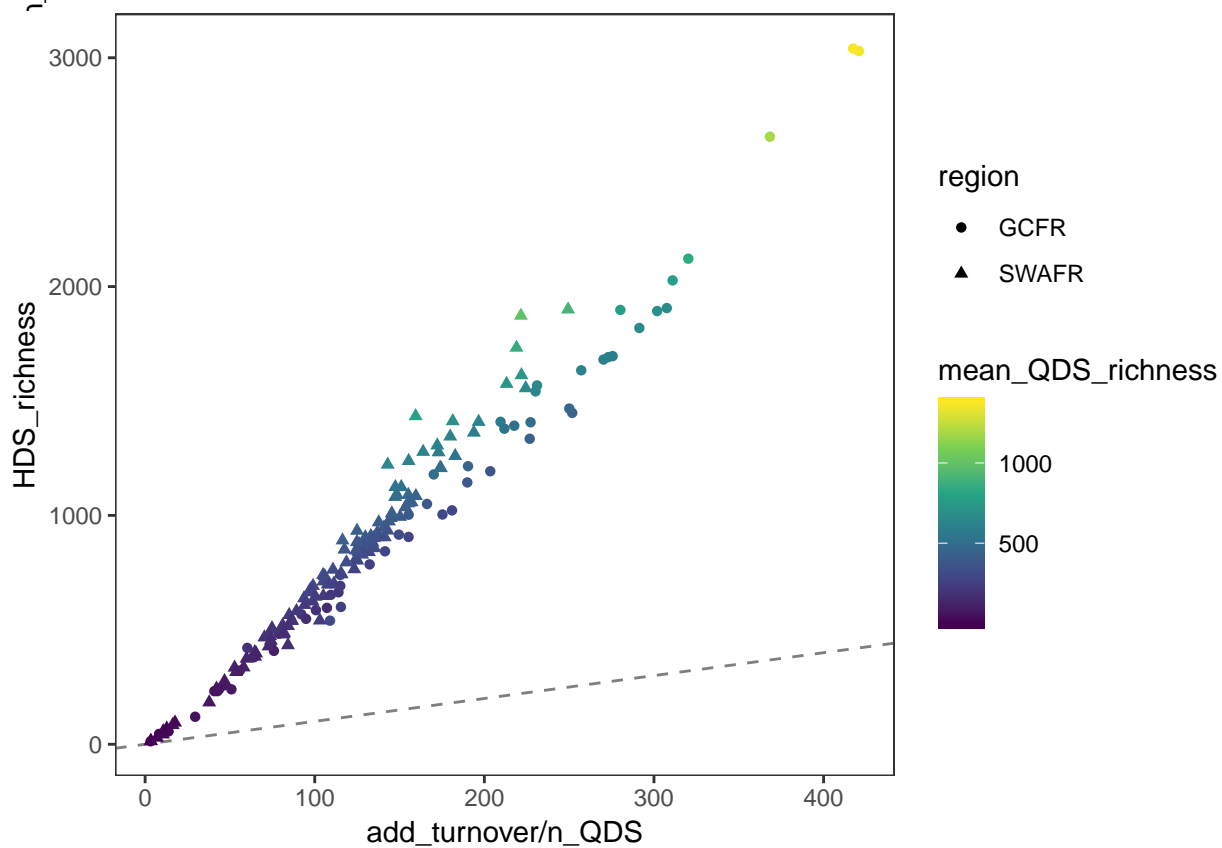
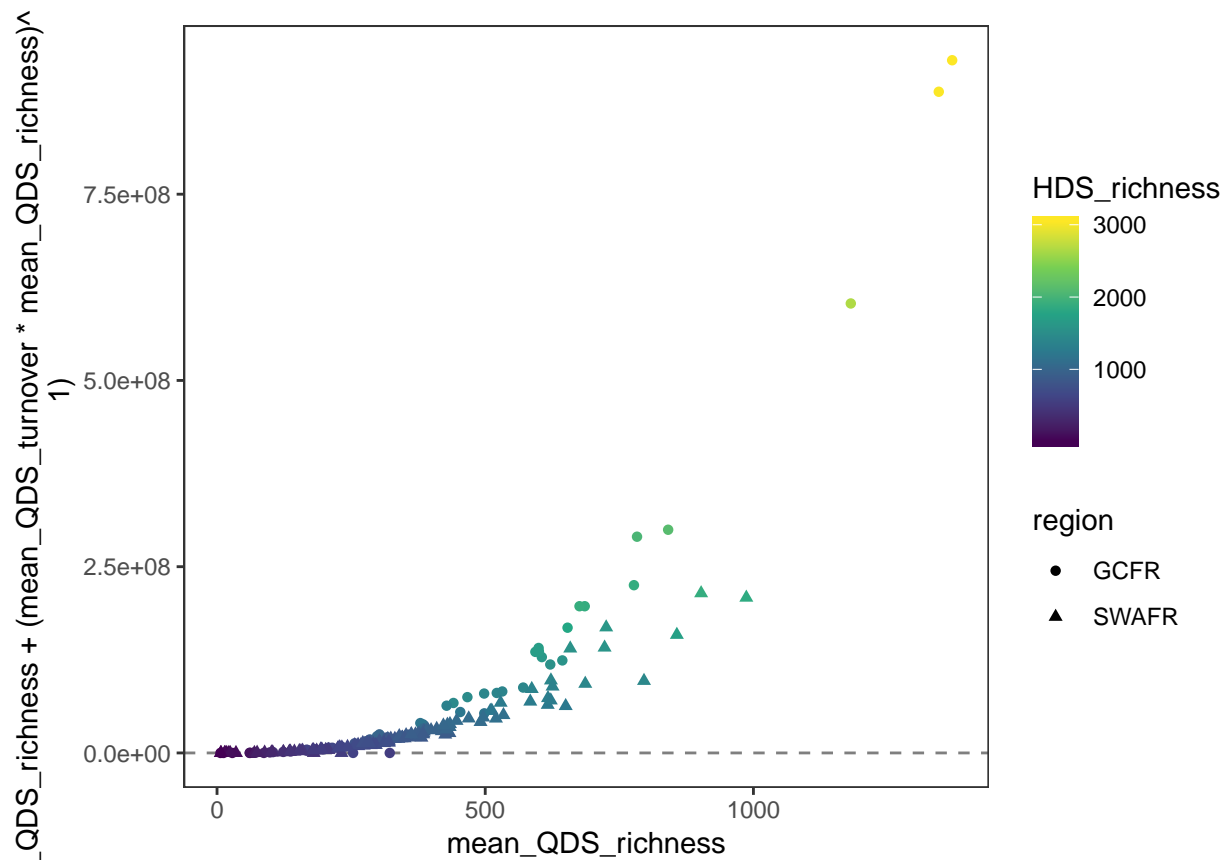
##

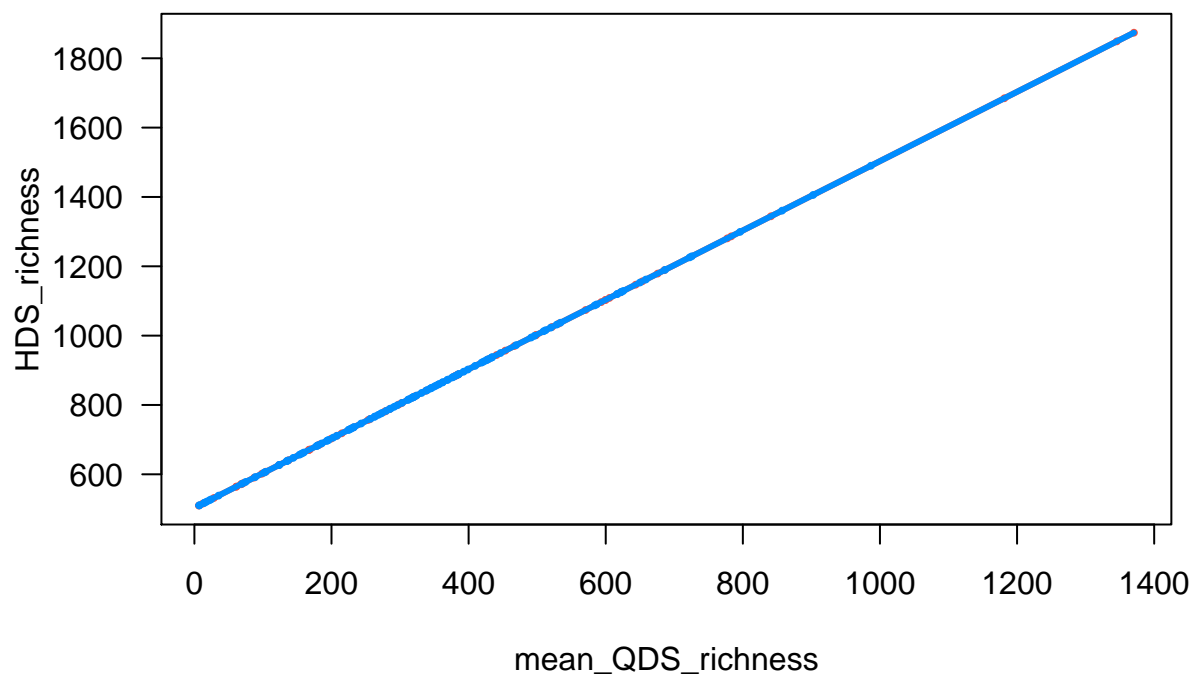
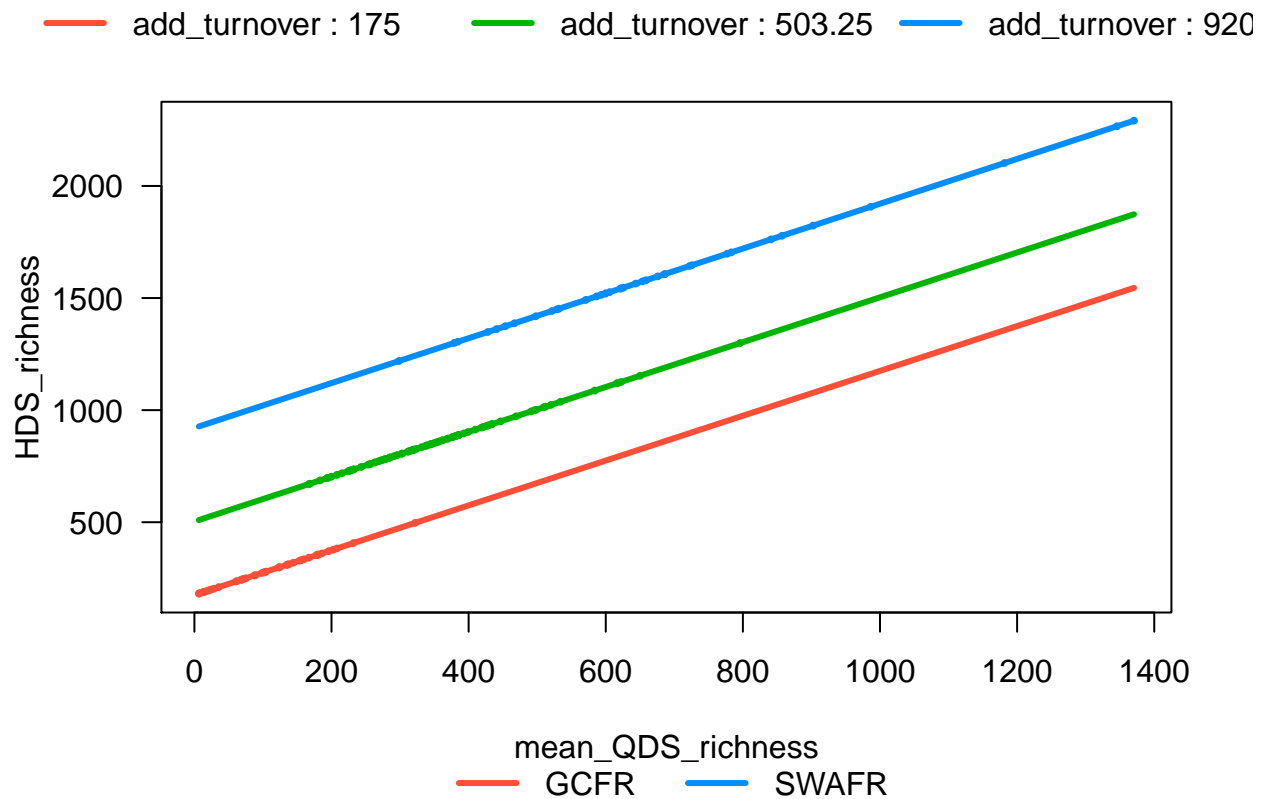
```
## Call:
## lm(formula = add_turnover ~ HDS_richness * region, data = mutate(HDS,
##   add_turnover = (HDS_richness - mean_QDS_richness)/n_QDS))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -36.362  -5.098   0.269   6.481  27.859
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    16.818779   2.378086   7.072 3.73e-11 ***
## HDS_richness     0.143049   0.001872  76.406 < 2e-16 ***
## regionSWAFR    -0.730515   3.216319  -0.227   0.821
## HDS_richness:regionSWAFR -0.017684   0.003056  -5.787 3.35e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.28 on 171 degrees of freedom
## Multiple R-squared:  0.9823, Adjusted R-squared:  0.982
## F-statistic: 3164 on 3 and 171 DF, p-value: < 2.2e-16
##
## Call:
## lm(formula = mean_QDS_richness ~ HDS_richness * region, data = HDS)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -115.532  -25.990   -5.309   19.043  148.745
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -48.99680    9.97134  -4.914 2.08e-06 ***
## HDS_richness     0.41801    0.00785  53.248 < 2e-16 ***
## regionSWAFR    -7.02383   13.48606  -0.521   0.603
## HDS_richness:regionSWAFR  0.07330    0.01281   5.720 4.65e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43.1 on 171 degrees of freedom
## Multiple R-squared:  0.9684, Adjusted R-squared:  0.9679
## F-statistic: 1749 on 3 and 171 DF, p-value: < 2.2e-16
```









```
HDS %>%
  dplyr::select(region, HDS_richness, mean_QDS_richness, mean_QDS_turnover, mul_turnover) %>%
  gather(partition, value, -region, -HDS_richness) %>%
  ggplot(aes(value, HDS_richness, colour = region)) +
    geom_smooth(method = lme4::lmer, formula = y ~ x + (x | colour)) +
    geom_point() +
```

```

    facet_grid(~partition, scales = "free_x")

ggplot(HDS, aes(mul_turnover / n_QDS, HDS_richness / n_QDS, colour = region, size = mean_QDS_richness /
  geom_smooth(method = lm) +
  geom_point()

ggplot(HDS, aes(mean_QDS_turnover, HDS_richness, colour = region, size = mean_QDS_richness)) +
  geom_smooth(method = lm) +
  geom_point()

foo <- HDS %>%
  mutate(
    HDS_richness      = HDS_richness      / n_QDS,
    mean_QDS_richness = mean_QDS_richness / n_QDS,
    mul_turnover      = mul_turnover      / n_QDS
  ) %>%
  {lm(HDS_richness ~ mean_QDS_richness * mul_turnover, .)}
summary(foo)
visreg::visreg(foo, xvar = "mean_QDS_richness", by = "mul_turnover", breaks = 1, overlay = TRUE)
visreg::visreg(foo, xvar = "mul_turnover", by = "region", overlay = TRUE)

foo <- HDS %>%
  mutate(
    HDS_richness      = HDS_richness      / n_QDS,
    mean_QDS_richness = mean_QDS_richness / n_QDS
  ) %>%
  {lm(HDS_richness ~ mean_QDS_richness * mean_QDS_turnover + mean_QDS_richness * region + mean_QDS_turnover * region, .)}
summary(foo)
visreg::visreg(foo, xvar = "mean_QDS_richness", by = "mean_QDS_turnover", breaks = 3, overlay = TRUE)
visreg::visreg(foo, xvar = "mean_QDS_richness", by = "region", overlay = TRUE)
visreg::visreg(foo, xvar = "mean_QDS_turnover", by = "region", overlay = TRUE)

foo <- HDS %>%
  mutate(
    HDS_richness      = HDS_richness      / n_QDS,
    mean_QDS_richness = mean_QDS_richness / n_QDS,
    mul_turnover      = mul_turnover      / n_QDS
  )
bar <- list(
  lme4::lmer(HDS_richness ~ mul_turnover + (1 + mul_turnover | region), foo),
  lme4::lmer(HDS_richness ~ mul_turnover + (1 + mul_turnover || region), foo),
  lme4::lmer(HDS_richness ~ mean_QDS_richness * mul_turnover + (1 + mul_turnover | region) + (1 + mean_QDS_richness * mul_turnover || region), foo)
)
summary(bar[[1]])
visreg::visreg(bar[[1]], xvar = "mul_turnover", by = "region", overlay = TRUE)
visreg::visreg(bar[[2]], xvar = "mul_turnover", by = "region", overlay = TRUE)
visreg::visreg(bar[[3]], xvar = "mul_turnover", by = "region", overlay = TRUE)
visreg::visreg(bar[[3]], xvar = "mean_QDS_richness", by = "region", overlay = TRUE)

foo <- HDS %>%
  mutate(
    HDS_richness      = HDS_richness      / n_QDS,
    mean_QDS_richness = mean_QDS_richness / n_QDS,

```

```

    mul_turnover      = mul_turnover      / n_QDS
  )
bar <- list(
  lme4::lmer(HDS_richness ~ mean_QDS_turnover + (1 + mean_QDS_turnover | region), foo),
  lme4::lmer(HDS_richness ~ mean_QDS_turnover + (1 + mean_QDS_turnover || region), foo),
  lme4::lmer(HDS_richness ~ mean_QDS_richness * mean_QDS_turnover + (1 + mean_QDS_turnover | region) +
)
summary(bar[[1]])
visreg::visreg(bar[[1]], xvar = "mean_QDS_turnover", by = "region", overlay = TRUE)
visreg::visreg(bar[[2]], xvar = "mean_QDS_turnover", by = "region", overlay = TRUE)
visreg::visreg(bar[[3]], xvar = "mean_QDS_turnover", by = "region", overlay = TRUE)
visreg::visreg(bar[[3]], xvar = "mean_QDS_richness", by = "region", overlay = TRUE)

HDS %>%
  dplyr::select(region, HDS_richness, mean_QDS_richness, n_QDS) %>%
  mutate(
    HDS_richness = HDS_richness / n_QDS,
    mean_QDS_richness = mean_QDS_richness / n_QDS
  ) %>%
  mutate(
    alpha = mean_QDS_richness / HDS_richness,
    beta = (HDS_richness - mean_QDS_richness) / HDS_richness
  ) %>%
  dplyr::select(region, HDS_richness, alpha, beta) %>%
  gather(partition, value, -region, -HDS_richness) %>%
  ggplot(aes(value, HDS_richness, colour = region)) +
    geom_point() +
    facet_grid(~partition)

HDS %>% dplyr::select(region, hdgc, HDS_richness, mean_QDS_richness, n_QDS) %>% mutate(HDS_richness = H
HDS %>% dplyr::select(region, hdgc, HDS_richness, mean_QDS_richness, n_QDS) %>% mutate(HDS_richness = H
HDS %>% dplyr::select(region, hdgc, HDS_richness, mean_QDS_richness, n_QDS) %>% mutate(HDS_richness = H
HDS %>% dplyr::select(region, hdgc, HDS_richness, mean_QDS_richness, n_QDS) %>% mutate(HDS_richness = H
HDS %>% dplyr::select(region, hdgc, HDS_richness, mean_QDS_richness, n_QDS) %>% mutate(HDS_richness = H

```

2.1. Univariate comparisons

```

ggplot(HDS, aes(HDS_richness, fill = region)) +
  geom_histogram(bins = 20, position = "dodge")
ggplot(HDS, aes(mul_turnover, fill = region)) +
  geom_histogram(bins = 20, position = "dodge")
ggplot(HDS, aes(mul_turnover, HDS_richness, colour = region)) +
  geom_smooth() +
  geom_point()
ggplot(HDS, aes(mul_turnover / n_QDS, HDS_richness / n_QDS, colour = region)) +
  geom_smooth() + #method = lm, formula = y ~ log(x)) +
  geom_point()
ggplot(HDS, aes(mean_QDS_richness, HDS_richness, colour = region)) +

```

```

    geom_smooth(method = lm) +
    geom_point()
ggplot(HDS, aes(mean_QDS_richness / n_QDS, HDS_richness / n_QDS, colour = region)) +
    geom_smooth(method = lm) +
    geom_point()
m <- lm(HDS_richness ~ mean_QDS_richness * mul_turnover + mean_QDS_richness * region + mul_turnover * r
tidy(m)
visreg::visreg(m, xvar = "mean_QDS_richness", by = "region", cond = list(mul_turnover = 0.5), overlay =
visreg::visreg(m, xvar = "mean_QDS_richness", by = "region", cond = list(region = "SWAFR"), overlay = T

```

3. Relating environmental heterogeneity to species richness & turnover

Does heterogeneity explain differences in richness and turnover between the regions?

...

```

library(here)
source(here("analyses-May-2019/setup.R"))
HDS <- read_csv(here("outputs/QDS_data_cells.csv"))
QDS <- read_csv(here("outputs/EDS_data_cells.csv"))

# Log roughness values to match that PCAs were done on logged data
HDS[, str_which(names(HDS), "roughness")] %<>% log()
QDS[, str_which(names(QDS), "roughness")] %<>% log()

# (.) Explore normality of data -----

non_normal_vars <- HDS[, str_which(names(HDS), "(region|mean_value)")] %>%
  split(.$region) %>%
  map(dplyr::select, -region) %>%
  map(map, shapiro.test) %>%
  map(map_df, tidy, .id = "variable") %>%
  bind_rows(.id = "region") %>%
  as_tibble() %>%
  dplyr::select(-statistic, -method) %>%
  mutate(sig = p.value <= 0.05) %>%
  filter(sig)

HDS[, str_which(names(HDS), "(region|mean_value)")] %>%
  split(.$region) %>%
  map(dplyr::select, -region) %>%
  map(map, log) %>% # !!!
  map(map, shapiro.test) %>%
  map(map_df, tidy, .id = "variable") %>%
  bind_rows(.id = "region") %>%
  as_tibble() %>%
  dplyr::select(-statistic, -method) %>%
  mutate(sig = p.value <= 0.05) %>%
  filter(sig)

# Conclusion: Logging can't solve everything!

```



```

# (a) Separate-regions models with combinations of variables -----
# .... HDS -----
# ..... GCFR -----

GCFR_HDS <- filter(HDS, region == "GCFR")

predictor_names <- HDS %>%
  {names(.)[str_which(names(.), "(roughness|mean_value)")] } %>%
  paste(collapse = " + ")

m_all_richness <- lm(glue("HDS_richness ~ {predictor_names}"), GCFR_HDS)
m_all_turnover <- lm(glue("mul_turnover ~ {predictor_names}"), GCFR_HDS)
summary(m_all_richness)
summary(m_all_turnover)

m_all_richness_step <- step(m_all_richness)
m_all_turnover_step <- step(m_all_turnover)
summary(m_all_richness_step)
summary(m_all_turnover_step)

# PC1 only model
m_PC1_richness <- lm(HDS_richness ~ PC1, GCFR_HDS)
m_PC1_turnover <- lm(mul_turnover ~ PC1, GCFR_HDS)
summary(m_PC1_richness)
summary(m_PC1_turnover)
visreg::visreg(m_PC1_richness)
visreg::visreg(m_PC1_turnover)

# ..... SWAFR -----

SWAFR_HDS <- filter(HDS, region == "SWAFR")

predictor_names <- HDS %>%
  {names(.)[str_which(names(.), "(roughness|mean_value)")] } %>%
  paste(collapse = " + ")

m_all_richness <- lm(glue("HDS_richness ~ {predictor_names}"), SWAFR_HDS)
m_all_turnover <- lm(glue("mul_turnover ~ {predictor_names}"), SWAFR_HDS)
summary(m_all_richness)
summary(m_all_turnover)

m_all_richness_step <- step(m_all_richness)
m_all_turnover_step <- step(m_all_turnover)
summary(m_all_step)
summary(m_all_turnover_step)

# PC1 only model
m_PC1_richness <- lm(HDS_richness ~ PC1, SWAFR_HDS)
m_PC1_turnover <- lm(mul_turnover ~ PC1, SWAFR_HDS)
summary(m_PC1_richness)
summary(m_PC1_turnover)

```

```

visreg::visreg(m_PC1_richness)
visreg::visreg(m_PC1_turnover)

# .... QDS -----

# ..... GCFR -----

GCFR_QDS <- filter(QDS, region == "GCFR")

predictor_names <- QDS %>%
  {names(.)[str_which(names(.), "(roughness|mean_value)")] } %>%
  paste(collapse = " + ")

m_all <- lm(glue("QDS_richness ~ {predictor_names}"), GCFR_QDS)
summary(m_all)

m_all_step <- step(m_all)
summary(m_all_step)

# PC1 only model
m_PC1 <- lm(QDS_richness ~ PC1, GCFR_QDS)
summary(m_PC1)
visreg::visreg(m_PC1)

# ..... SWAFR -----

SWAFR_QDS <- filter(QDS, region == "SWAFR")

predictor_names <- QDS %>%
  {names(.)[str_which(names(.), "(roughness|mean_value)")] } %>%
  paste(collapse = " + ")

m_all <- lm(glue("QDS_richness ~ {predictor_names}"), SWAFR_QDS)
summary(m_all)

m_all_step <- step(m_all)
summary(m_all_step)

# PC1 only model
m_PC1 <- lm(QDS_richness ~ PC1, SWAFR_QDS)
summary(m_PC1)
visreg::visreg(m_PC1)

# (b) Combined-regions models with individual variables -----

# .... HDS -----

# ..... Richness -----

predictor_names <- names(HDS)[
  str_which(names(HDS), "(roughness|mean_value)")
]

```

```

HDS_richness_models_no_region <- predictor_names %>%
  map(~lm(glue("HDS_richness ~ {.x}"), HDS)) %>%
  set_names(predictor_names) %>%
  {tibble(predictor = names(.), model = .)} %>%
  mutate(
    slope_p_value = map_dbl(model, ~tidy(.x)$p.value[[2]]),
    r_squared      = map_dbl(model, ~glance(.x)$r.squared),
    slope_sig      = ifelse(slope_p_value <= 0.05, "*", ""),
    plot           = map(model, visreg::visreg, gg = TRUE)
  )
HDS_richness_models_add_region <- predictor_names %>%
  map(~lm(glue("HDS_richness ~ {.x} + region"), HDS)) %>%
  set_names(predictor_names) %>%
  {tibble(predictor = names(.), model = .)} %>%
  mutate(
    slope_p_value = map_dbl(model, ~tidy(.x)$p.value[[2]]),
    region_p_value = map_dbl(model, ~tidy(.x)$p.value[[3]]),
    r_squared      = map_dbl(model, ~glance(.x)$r.squared),
    slope_sig      = ifelse(slope_p_value <= 0.05, "*", ""),
    region_sig     = ifelse(region_p_value <= 0.05, "*", ""),
    plot           = map(model, visreg::visreg, gg = TRUE)
  )
HDS_richness_models_int_region <- predictor_names %>%
  map(~lm(glue("HDS_richness ~ {.x} * region"), HDS)) %>%
  set_names(predictor_names) %>%
  {tibble(predictor = names(.), model = .)} %>%
  mutate(
    #model_tidy      = map(model, tidy),
    #foo              = map_chr(map(model_tidy, "term"), paste, collapse = ", ")
    slope_p_value = map_dbl(model, ~tidy(.x)$p.value[[2]]),
    region_p_value = map_dbl(model, ~tidy(.x)$p.value[[3]]),
    int_p_value   = map_dbl(model, ~tidy(.x)$p.value[[4]]),
    r_squared      = map_dbl(model, ~glance(.x)$r.squared),
    slope_sig      = ifelse(slope_p_value <= 0.05, "*", ""),
    region_sig     = ifelse(region_p_value <= 0.05, "*", ""),
    int_sig        = ifelse(int_p_value <= 0.05, "*", ""),
    plot           = map2(model, predictor,
      ~ visreg::visreg(.x,
        xvar = .y,
        by = "region",
        overlay = TRUE,
        gg = TRUE
      )
    )
  )

cowplot::plot_grid(plotlist = HDS_richness_models_no_region$plot)
cowplot::plot_grid(plotlist = HDS_richness_models_add_region$plot)
cowplot::plot_grid(plotlist = HDS_richness_models_int_region$plot)

# ..... PC1 only model -----
# Richness

```

```

m1 <- lm(HDS_richness ~ PC1,          HDS)
m2 <- lm(HDS_richness ~ PC1 + region, HDS)
m3 <- lm(HDS_richness ~ PC1 : region, HDS)
m4 <- lm(HDS_richness ~ PC1 * region, HDS)

AIC(m1, m2, m3, m4)

visreg::visreg(m1)
visreg::visreg(m2, xvar = "PC1", by = "region", overlay = TRUE)
visreg::visreg(m3, xvar = "PC1", by = "region", overlay = TRUE)
visreg::visreg(m4, xvar = "PC1", by = "region", overlay = TRUE)

tidy(m1)
glance(m1)

# Richness (QDS)

m1 <- lm(QDS_richness ~ PC1,          QDS)
m2 <- lm(QDS_richness ~ PC1 + region, QDS)
m3 <- lm(QDS_richness ~ PC1 : region, QDS)
m4 <- lm(QDS_richness ~ PC1 * region, QDS)

AIC(m1, m2, m3, m4)

visreg::visreg(m1)
visreg::visreg(m2, xvar = "PC1", by = "region", overlay = TRUE)
visreg::visreg(m3, xvar = "PC1", by = "region", overlay = TRUE)
visreg::visreg(m4, xvar = "PC1", by = "region", overlay = TRUE)

tidy(m1)
glance(m1)

# Turnover

m1 <- lm(mul_turnover ~ PC1,          HDS)
m2 <- lm(mul_turnover ~ PC1 + region, HDS)
m3 <- lm(mul_turnover ~ PC1 : region, HDS)
m4 <- lm(mul_turnover ~ PC1 * region, filter(HDS, n_QDS == 4))
m5 <- lm(mul_turnover ~ PC1 * region + n_QDS, HDS) # uh oh!

AIC(m1, m2, m3, m4)

visreg::visreg(m1)
visreg::visreg(m2, xvar = "PC1", by = "region", overlay = TRUE)
visreg::visreg(m3, xvar = "PC1", by = "region", overlay = TRUE)
visreg::visreg(m4, xvar = "PC1", by = "region", overlay = TRUE)
# !!!!!

tidy(m4)
glance(m4)

# (c) Combined-regions models with combinations of variables -----

```

```

# TODO

m_all <- lm(
  HDS_richness ~
    Elevation_mean_value + Elevation_roughness +
    MAP_mean_value       + MAP_roughness +
    NDVI_mean_value      + NDVI_roughness +
    PDQ_mean_value       + PDQ_roughness +
    pH_mean_value        + pH_roughness +
    Soil.C_mean_value    + Soil.C_roughness +
    Surface.T_mean_value + Surface.T_roughness,
  HDS
)
summary(m_all)
m_all_step <- step(m_all)
summary(m_all_step)

m_all_int <- lm(
  HDS_richness ~
    Elevation_mean_value      + Elevation_roughness +
    MAP_mean_value            + MAP_roughness +
    NDVI_mean_value           + NDVI_roughness +
    PDQ_mean_value            + PDQ_roughness +
    pH_mean_value             + pH_roughness +
    Soil.C_mean_value         + Soil.C_roughness +
    Surface.T_mean_value      + Surface.T_roughness +
    Elevation_mean_value : region + Elevation_roughness : region +
    MAP_mean_value       : region + MAP_roughness       : region +
    NDVI_mean_value      : region + NDVI_roughness      : region +
    PDQ_mean_value       : region + PDQ_roughness       : region +
    pH_mean_value        : region + pH_roughness        : region +
    Soil.C_mean_value    : region + Soil.C_roughness    : region +
    Surface.T_mean_value : region + Surface.T_roughness : region,
  HDS
)
m_all_int2 <- lm(
  QDS_richness ~
    Elevation_mean_value      + Elevation_roughness +
    MAP_mean_value            + MAP_roughness +
    NDVI_mean_value           + NDVI_roughness +
    PDQ_mean_value            + PDQ_roughness +
    pH_mean_value             + pH_roughness +
    Soil.C_mean_value         + Soil.C_roughness +
    Surface.T_mean_value      + Surface.T_roughness +
    Elevation_mean_value : region + Elevation_roughness : region +
    MAP_mean_value       : region + MAP_roughness       : region +
    NDVI_mean_value      : region + NDVI_roughness      : region +
    PDQ_mean_value       : region + PDQ_roughness       : region +
    pH_mean_value        : region + pH_roughness        : region +
    Soil.C_mean_value    : region + Soil.C_roughness    : region +
    Surface.T_mean_value : region + Surface.T_roughness : region,
  QDS
)

```

```

summary(m_all_int)
summary(m_all_int2)
m_all_int_step <- step(m_all_int)
m_all_int_step2 <- step(m_all_int2)
summary(m_all_int_step)
summary(m_all_int_step2)
non_sigs <- m_all_int_step %>%
  tidy() %>%
  filter(p.value > 0.05, term != "(Intercept)") %>%
  pull(term) %>%
  paste(collapse = " - ") %>%
  {paste("-", .)}

m_all_int_step_manual <- update(
  m_all_int_step,
  as.formula(paste("~ .", non_sigs))
)
summary(m_all_int_step_manual)
non_sigs <- m_all_int_step_manual %>%
  tidy() %>%
  filter(p.value > 0.05, term != "(Intercept)") %>%
  pull(term) %>%
  paste(collapse = " - ") %>%
  {paste("-", .)}
m_all_int_step_manual2 <- update(
  m_all_int_step_manual,
  as.formula(paste("~ .", non_sigs))
)
summary(m_all_int_step_manual2)

AIC(m_all, m_all_step, m_all_int, m_all_int_step)

foo <- tibble(
  fit = m_all_int_step$fitted.values,
  obs = HDS$HDS_richness,
  region = HDS$region
)
ggplot(foo, aes(obs, fit)) +
  geom_smooth(method = lm, colour = "black") +
  geom_point(aes(colour = region))
#plot(m_all_int_step)

foo2 <- tibble(
  fit = m_all_int_step2$fitted.values,
  obs = QDS$QDS_richness,
  region = QDS$region
)
ggplot(foo2, aes(obs, fit)) +
  geom_smooth(method = lm, colour = "black") +
  geom_point(aes(colour = region), alpha = 0.25)
#plot(m_all_int_step2)

# (*) Plots -----

```

```

ggplot(HDS, aes(PC1, HDS_richness)) +
  geom_point(aes(colour = region)) +
  geom_smooth(method = lm, colour = "black")
ggplot(QDS, aes(PC1, QDS_richness)) +
  geom_point(aes(colour = region)) +
  geom_smooth(method = lm, colour = "black")
m <- lm(HDS_richness ~ PC1, HDS)
#plot(m)
ggplot(HDS, aes(Elevation_mean_value, HDS_richness)) +
  geom_point(aes(colour = region)) +
  geom_smooth(method = lm, colour = "black")
ggplot(HDS, aes(Elevation_roughness, HDS_richness)) +
  geom_point(aes(colour = region)) +
  geom_smooth(method = lm, colour = "black")

ggplot(HDS, aes(PC1, Elevation_roughness, colour = region)) +
  geom_point()
ggplot(HDS, aes(PC2, Elevation_roughness, colour = region)) +
  geom_point()

ggplot(QDS, aes(lon, lat, colour = PC1)) +
  geom_point(size = 3) +
  facet_grid(~region, scales = "free_x") +
  scale_colour_viridis_c()

ggplot(HDS, aes(lon, lat, colour = PC1)) +
  geom_point(size = 3) +
  facet_wrap(~region, scales = "free") +
  scale_colour_viridis_c()
ggplot(HDS, aes(lon, lat, colour = PC2)) +
  geom_point(size = 3) +
  facet_wrap(~region, scales = "free")

# (...) PCA biplots again -----

HDS_PCA      <- read_rds(here("outputs/QDS_roughness_cells_PCA.RDS"))
HDS_PCA_data <- read_csv(here("outputs/QDS_roughness_cells_prepped.csv"))
QDS_PCA      <- read_rds(here("outputs/EDS_roughness_cells_PCA.RDS"))
QDS_PCA_data <- read_csv(here("outputs/EDS_roughness_cells_prepped.csv"))

# Log to match PCA
HDS_PCA_data[, -c(1, 2)] %<>% log()
QDS_PCA_data[, -c(1, 2)] %<>% log()

autoplot(HDS_PCA,
  data      = HDS_PCA_data,
  colour    = "region",
  loadings  = TRUE,
  loadings.colour = "blue",
  loadings.label = TRUE
)
autoplot(QDS_PCA,
  data      = QDS_PCA_data,

```

```

colour      = "region",
loadings    = TRUE,
loadings.colour = "blue",
loadings.label = TRUE
)

HDS %>%
  filter(n_QDS == 4) %>%
  dplyr::select(region, PC1, PC2) %>%
  group_by(region) %>%
  summarise_all(.funs = list(mean = mean, sd = sd)) %>%
  mutate(
    PC1_upp = PC1_mean + PC1_sd,
    PC1_low = PC1_mean - PC1_sd,
    PC2_upp = PC2_mean + PC2_sd,
    PC2_low = PC2_mean - PC2_sd
  ) %>%
  ggplot(aes(PC1_mean, PC2_mean, colour = region)) +
  geom_hline(yintercept = 0, lty = "dashed", colour = "grey25") +
  geom_vline(xintercept = 0, lty = "dashed", colour = "grey25") +
  geom_point() +
  geom_errorbar(aes(ymin = PC2_low, ymax = PC2_upp), width = 0) +
  geom_errorbarh(aes(xmin = PC1_low, xmax = PC1_upp), height = 0) +
  geom_point(
    data = filter(HDS, n_QDS == 4),
    mapping = aes(PC1, PC2, colour = region),
    alpha = 0.5
  ) +
  theme_minimal()

QDS %>%
  filter(n_EDS == 4) %>%
  dplyr::select(region, PC1, PC2) %>%
  group_by(region) %>%
  summarise_all(.funs = list(mean = mean, sd = sd)) %>%
  mutate(
    PC1_upp = PC1_mean + PC1_sd,
    PC1_low = PC1_mean - PC1_sd,
    PC2_upp = PC2_mean + PC2_sd,
    PC2_low = PC2_mean - PC2_sd
  ) %>%
  ggplot(aes(PC1_mean, PC2_mean, colour = region)) +
  geom_hline(yintercept = 0, lty = "dashed", colour = "grey25") +
  geom_vline(xintercept = 0, lty = "dashed", colour = "grey25") +
  geom_point() +
  geom_errorbar(aes(ymin = PC2_low, ymax = PC2_upp), width = 0) +
  geom_errorbarh(aes(xmin = PC1_low, xmax = PC1_upp), height = 0) +
  geom_point(
    data = filter(QDS, n_EDS == 4),
    mapping = aes(PC1, PC2, colour = region),
    alpha = 0.25
  ) +
  theme_minimal()

```



```

ggplot(HDS, aes(PC1, PC2, colour = PC2 > 0)) +
  geom_point()

HDS %>%
  split(.$region) %>%
  map(~.x %$% table(PC2 > 0, PC1 > 0))

my_PCA_plot <- function(data) {
  get_lim <- function(x) {
    lim <- ceiling(max(x))
    c(-lim, lim)
  }
  plot_xlim <- get_lim(data$PC1)
  plot_ylim <- get_lim(data$PC2)
  no_legend_no_grid <- theme(
    legend.position = "none",
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  )
  white_rect <- grid::grid.rect(gp = grid::gpar(col = "white"))

  PC1_histograms <- ggplot(data, aes(PC1, fill = region)) +
    geom_histogram(bins = 20, position = "dodge") +
    xlim(plot_xlim) +
    theme_minimal() +
    theme(
      axis.title.x = element_blank(),
      axis.text.x = element_blank(),
      axis.text.y = element_text(angle = 90),
      axis.line.x.bottom = element_blank(),
      axis.line.x.top = element_blank(),
      axis.line.y.right = element_blank()
    ) +
    no_legend_no_grid
  PC2_histograms <- ggplot(data, aes(PC2, fill = region)) +
    geom_histogram(bins = 20, position = "dodge") +
    xlim(plot_ylim) +
    coord_flip() +
    theme_minimal() +
    theme(
      axis.title.y = element_blank(),
      axis.text.y = element_blank(),
      axis.line.y.right = element_blank(),
      axis.line.y.left = element_blank(),
      axis.line.x.top = element_blank()
    ) +
    no_legend_no_grid

  PCA_biplot <- ggplot(data, aes(PC1, PC2, colour = region)) +
    geom_hline(yintercept = 0, lty = "dashed", colour = "grey25") +
    geom_vline(xintercept = 0, lty = "dashed", colour = "grey25") +
    geom_point() +
    lims(x = plot_xlim, y = plot_ylim) +

```

```

    theme_bw() +
    theme(axis.text.y = element_text(angle = 90)) +
    no_legend_no_grid

cowplot::plot_grid(
  PC1_histograms, white_rect,
  PCA_biplot,      PC2_histograms,
  ncol = 2,
  rel_widths = c(4, 1), rel_heights = c(1, 4)
)
}
foo <- my_PCA_plot(HDS)
foo
PC1_histograms <- ggplot(QDS, aes(PC1, fill = region)) +
  xlim(-5, 5) +
  geom_histogram(bins = 20, position = "dodge") +
  theme_minimal() +
  theme(legend.position = "none", axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_text(angle = 90),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
PC2_histograms <- ggplot(QDS, aes(PC2, fill = region)) +
  xlim(-5, 5) +
  geom_histogram(bins = 20, position = "dodge") +
  coord_flip() +
  theme_minimal() +
  theme(legend.position = "none", axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
PC_biplot <- ggplot(QDS, aes(PC1, PC2, colour = region)) +
  lims(x = c(-5, 5), y = c(-5, 5)) +
  geom_hline(yintercept = 0, lty = "dashed", colour = "grey25") +
  geom_vline(xintercept = 0, lty = "dashed", colour = "grey25") +
  geom_point() +
  theme_bw() +
  theme(legend.position = "none", axis.text.y = element_text(angle = 90),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

white_rect <- grid::grid.rect(gp = grid::gpar(col = "white"))
cowplot::plot_grid(
  PC1_histograms, white_rect,
  PC_biplot,      PC2_histograms,
  ncol = 2,
  rel_widths = c(4, 1), rel_heights = c(1, 4)
)

PC1_histograms <- ggplot(HDS, aes(PC1, fill = region)) +
  #xlim(-5, 5) +
  geom_histogram(bins = 20, position = "dodge") +
  theme_minimal() +

```

```

    theme(legend.position = "none", axis.title.x = element_blank(),
          axis.text.x = element_blank(),
          axis.title.y = element_blank(),
          axis.text.y = element_blank(),
          #axis.text.y = element_text(angle = 90),

          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank())
PC2_histograms <- ggplot(HDS, aes(PC2, fill = region)) +
  #xlim(-5, 5) +
  geom_histogram(bins = 20, position = "dodge") +
  coord_flip() +
  theme_minimal() +
  theme(legend.position = "none", axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.title.x = element_blank(),
        axis.text.x = element_blank(),

        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
PC_biplot <- ggplot(HDS, aes(PC1, PC2, colour = region)) +
  #lims(x = c(-5, 5), y = c(-5, 5)) +
  geom_hline(yintercept = 0, lty = "dashed", colour = "grey25") +
  geom_vline(xintercept = 0, lty = "dashed", colour = "grey25") +
  geom_point() +
  theme_bw() +
  theme(legend.position = "none", axis.text.y = element_text(angle = 90),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

white_rect <- grid::grid.rect(gp = grid::gpar(col = "white"))
cowplot::plot_grid(
  PC1_histograms, white_rect,
  PC_biplot,      PC2_histograms,
  ncol = 2,
  rel_widths = c(4, 1), rel_heights = c(1, 4)
)

all_PCA_data <- rbind(
  cbind(scale = "HDS", HDS_PCA_data[, -2]),
  cbind(scale = "QDS", QDS_PCA_data[, -2])
)

foo <- prcomp(all_PCA_data[, -c(1, 2)], scale. = TRUE)
if (all(foo$rotation[, 1] <= 0)) {
  foo$rotation[, 1] %<>% multiply_by(-1)
  foo$x[, 1] %<>% multiply_by(-1)
}
autoplot(foo, data = unite(all_PCA_data, region_scale, region, scale), colour = "region_scale", alpha =

```

- 3.1. Separate-regions models with combinations of variables
- 3.2. Combined-regions models with individual variables
- 3.3. Combined-regions models with combinations of variables