# Envr_Performance_Mantel_Tests

Kiran Bajaj

2025-01-23

This code analyzes the effect of pairwise environmental distance between conditions at the environments-of-origin and pairwise genetic distance between parent populations on pairwise differences in survival and shell length.

```
#setwd("~/Desktop/MVP-H2F-HatcheryField")
```

### Packages

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(grid)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(vegan)
```

```
## Loading required package: permute
## Loading required package: lattice
```

### Load data

```
# raw lengths of all oysters aged 15-78 days
hatchery_length <- read.csv(file.path("..", "..", "data", "performance_H2F", "latestage_length.csv"))

# survival rate data for larval oysters up to day 21 post hatching
hatchery_survival <-  read.csv(file.path("..", "..", "data", "performance_H2F", "CViMVP_larvae_survival

# lengths of all oysters averaged by bag for field monitoring events 1-3
field_length <- read.csv(file.path("..", "..", "data", "performance_H2F", "lengths_bags_022025.csv"))

# survival of all oysters averaged by bag for field monitoring events 1-3
field_survival <- read.csv(file.path("..", "..", "data", "performance_H2F", "mortality_bags_022025.csv"

# summary statistics - temperature and salinity quantiles for the 8 groups
envr_summary <-  read.csv(file.path("..", "..", "data", "envr_of_origin", "envr_quantiles_summary.csv")

# matrix of euclidean environmental distances using SCALED data
envr_dist <- readRDS(file.path("..", "..", "data", "envr_of_origin", "envr_quantiles_dist.rds"))

# matrix of genetic distances between each group
fst_mat <- readRDS(file.path("..", "..", "data", "pairwise_differences_analyses", "fst_results.rds"))
```

### Global variable

Population order

```
# save vector of population order
population_order <- c("W1-TX", "W2-LA", "W3-FL", "W4-VA", "S1-LOLA", "S2-DEBY", "W5-NH", "W6-ME")
```

### Data preparation

**Hatchery survival** For hatchery survival rate, we used survival from day 21, so filter for those values only. Days 15-21 were grouped for analysis because there were constant larval drops occurring during that time, but the number of drops varied by site. Also exclude LARMIX, since there is no environmental data for group, given that it represents oysters from multiple sites.

```
# filter for day 21, exclude LARMIX, and recode site names
survival_day21 <- hatchery_survival %>%
  filter(Day_analysis == "15-21") %>%
  filter(Tank_naming != "MVP-LARMIX") %>%
  mutate(Tank_naming = recode(Tank_naming,
                              "MVP-DEBY" = "S2-DEBY",
                              "MVP-FL" = "W3-FL",
                              "MVP-LA" = "W2-LA",
                              "MVP-LOLA" = "S1-LOLA",
                              "MVP-ME" = "W6-ME",
                              "MVP-NH" = "W5-NH",
                              "MVP-TX" = "W1-TX",
                              "MVP-JR" = "W4-VA"))
```

```r
# exclude any groups with "small" in their label, we only want to analyze eyed oysters
surv_day21 <- survival_day21 %>%
  filter(!grepl("small", Group_Day_Label, ignore.case = TRUE)) %>%
  select(Tank_naming, Survival_rate_perc) %>%
  arrange(factor(Tank_naming, levels = population_order)) # reorder

# rename columns
colnames(surv_day21) <- c("site_name", "surv_21")
```

```r
# exclude LARMIX
hatchery_length_filter <- hatchery_length %>%
  select(site_name, day_an, shell_length_mm)  %>%
  filter(site_name != "H1-LARMIX") # filter out LARMIX

# data frame for days 15-21
length_21 <- hatchery_length_filter %>%
  filter(day_an == "15-21")

# data frame for day 78
length_78 <- hatchery_length_filter %>%
  filter(day_an == "78")

# calc mean shell lengths of each group on each day
means_mm_21 <- aggregate(shell_length_mm ~ site_name, FUN = mean, data = length_21)
colnames(means_mm_21) <- c("site_name", "mean_length_mm_21")
means_mm_21 <- means_mm_21 %>% arrange(factor(site_name, levels = population_order)) # reorder

means_mm_78 <- aggregate(shell_length_mm ~ site_name, FUN = mean, data = length_78)
colnames(means_mm_78) <- c("site_name", "mean_length_mm_78")
means_mm_78 <- means_mm_78 %>% arrange(factor(site_name, levels = population_order)) # reorder
```

### Hatchery length

**Field length** The goal is to create 6 data frames: one for each monitoring event (3) and each field site (2) with the average length across all bags for each group.

```r
# separate bag labels, recode sites, and average by bag and field site
field_length_clean <- field_length %>%
  separate(bag, into = c("site_name", "bag_number"), sep = "-(?=\\d+$)") %>%
  filter(!site_name %in% c("MVP-SEEDMIX", "MVP-LARMIX")) %>% # remove LARMIX and SEEDMIX
  mutate(site_name = recode(site_name,
                           "MVP-DEBY" = "S2-DEBY",
                           "MVP-FL" = "W3-FL",
                           "MVP-LA" = "W2-LA",
                           "MVP-LOLA" = "S1-LOLA",
                           "MVP-ME" = "W6-ME",
                           "MVP-NH" = "W5-NH",
                           "MVP-TX" = "W1-TX",
                           "MVP-JR" = "W4-VA"))
```

```r
# average lengths across all bags per group
# within-bag lengths are are already averaged
# average separately for each monitoring event and field site
field_length_summary <- field_length_clean %>%
  group_by(monitoring_event, site, site_name) %>%
  summarise(mean_length = mean(length, na.rm = TRUE), .groups = "drop")

# split the data frame by monitoring event and site
split_list <- field_length_summary %>%
  group_split(monitoring_event, site)

# give each data frame a name according to monitoring event and field site
split_names <- field_length_summary %>%
  group_keys(monitoring_event, site) %>%
  mutate(name = paste0("length_event", monitoring_event, "_", gsub(" ", "", tolower(site)))) %>%
  pull(name)
```

```
## Warning: The ‘...‘ argument of ‘group_keys()‘ is deprecated as of dplyr 1.0.0.
## i Please ‘group_by()‘ first
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```r
split_list <- lapply(split_list, function(df) {
  df %>%
    arrange(factor(site_name, levels = population_order)) %>% # reorder
    select(site_name, mean_length) # remove columns except for population and length
  })

# assign each data frame to a variable
names(split_list) <- split_names
list2env(split_list, envir = .GlobalEnv)
```

```
## <environment: R_GlobalEnv>
```

**Field survival** The goal is to create 6 data frames: one for each monitoring event (3) and each field site (2) with the average survival across all bags for each group.

```r
# separate bag labels, recode sites, and average by bag and field site
field_survival_clean <- field_survival %>%
  separate(bags_label, into = c("site_name", "bag_number"), sep = "-(?=\\d+$)") %>%
  filter(!site_name %in% c("MVP-SEEDMIX", "MVP-LARMIX")) %>% # remove LARMIX and SEEDMIX
  mutate(site_name = recode(site_name,
                            "MVP-DEBY" = "S2-DEBY",
                            "MVP-FL" = "W3-FL",
                            "MVP-LA" = "W2-LA",
                            "MVP-LOLA" = "S1-LOLA",
                            "MVP-ME" = "W6-ME",
                            "MVP-NH" = "W5-NH",
                            "MVP-TX" = "W1-TX",
                            "MVP-JR" = "W4-VA")) %>%
  select(site_name, bag_number, bag_site, t1, t2, t3)
```

```
# pivot so there is one row per site name, bag, field site, and monitoring event
field_survival_long <- field_survival_clean %>%
  pivot_longer(
    cols = starts_with("t"),  # columns t1, t2, t3
    names_to = "monitoring_event",
    values_to = "survival"
  ) %>%
  mutate(
    monitoring_event = recode(monitoring_event,
                              t1 = "1",
                              t2 = "2",
                              t3 = "3")
  )

# now can use code from above to make one data frame per monitoring event per field site

# average lengths across all bags per group
# within-bag lengths are are already averaged
# average separately for each monitoring event and field site
field_survival_summary <- field_survival_long %>%
  group_by(monitoring_event, bag_site, site_name) %>%
  summarise(mean_survival = mean(survival, na.rm = TRUE), .groups = "drop")

# split the data frame by monitoring event and field site
split_list <- field_survival_summary %>%
  group_split(monitoring_event, bag_site)

# give each data frame a name according to monitoring event and field site
split_names <- field_survival_summary %>%
  group_keys(monitoring_event, bag_site) %>%
  mutate(name = paste0("survival_event", monitoring_event, "_", gsub(" ", "", tolower(bag_site)))) %>%
  pull(name)
```

```
## Warning: The ‘...‘ argument of ‘group_keys()‘ is deprecated as of dplyr 1.0.0.
## i Please ‘group_by()‘ first
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
split_list <- lapply(split_list, function(df) {
  df %>%
    arrange(factor(site_name, levels = population_order)) %>% # reorder
    select(site_name, mean_survival) # remove columns except for population and length
  })

# assign each data frame to a variable
names(split_list) <- split_names
list2env(split_list, envir = .GlobalEnv)
```

```
## <environment: R_GlobalEnv>
```

```r
# remove extra column from environmental data summary table
envr <- envr_summary %>% select(-X)

# add sampling site latitudes. from table 1 of Nicole thesis

envr$lat <- c(28.096000, 29.239925, 30.440030, 37.1501163, 37.98030, 37.249107, 43.053746, 44.01330)
```

### Environment-of-origin

```r
surv_length_envr <- left_join(envr, means_mm_21, by = "site_name") %>%
  left_join(means_mm_78, by = "site_name") %>%
  left_join(surv_day21, by = "site_name")

# save csv for further analyses
write.csv(surv_length_envr, file.path("..", "..", "data", "parent_effects_H2F", "Survival_Length_Envr_Da
```

### Data product: hatchery survival, hatchery length, environment-of-origin

```r
# reorder
fst_mat <- fst_mat[population_order, population_order]
```

### Fst

### Distance matrices

Make distance matrices for all survival and length variables

```r
# list of all the data frames that need to be made into distance matrices
df_list <- list(
  surv_day21 = surv_day21,
  len_21 = means_mm_21,
  len_78 = means_mm_78,
  len_1_lew = length_event1_lewisetta,
  len_1_york = length_event1_yorkriver,
  len_2_lew = length_event2_lewisetta,
  len_2_york = length_event2_yorkriver,
  len_3_lew = length_event3_lewisetta,
  len_3_york = length_event3_yorkriver,
  surv_1_lew = survival_event1_lewisetta,
  surv_1_york = survival_event1_yorkriver,
  surv_2_lew = survival_event2_lewisetta,
  surv_2_york = survival_event2_yorkriver,
  surv_3_lew = survival_event3_lewisetta,
  surv_3_york = survival_event3_yorkriver
)

# empty list to store distance matrices
dist_matrices <- list()

# loop through each data frame in the list
for (name in names(df_list)) {
  df <- df_list[[name]]

  # set the second column as values (first column is population names)
  values <- df[[2]]

  # create distance matrix
  dist_matrix <- as.matrix(dist(values))

  # set row and col names
  rownames(dist_matrix) <- population_order
  colnames(dist_matrix) <- population_order

  # save into the list
  dist_matrices[[name]] <- dist_matrix
}
```

```
check_matrix_order <- function(matrices, population_order) {
  all_good <- TRUE
  for (i in seq_along(matrices)) {
    if (!identical(rownames(matrices[[i]]), population_order)) {
      message(sprintf("Row names do not match in matrix %d", i))
      all_good <- FALSE
    }
    if (!identical(colnames(matrices[[i]]), population_order)) {
      message(sprintf("Column names do not match in matrix %d", i))
      all_good <- FALSE
    }
  }
  if (all_good) {
    message("All matrices have matching row and column names.")
  }
}

check_matrix_order(dist_matrices, population_order)
```

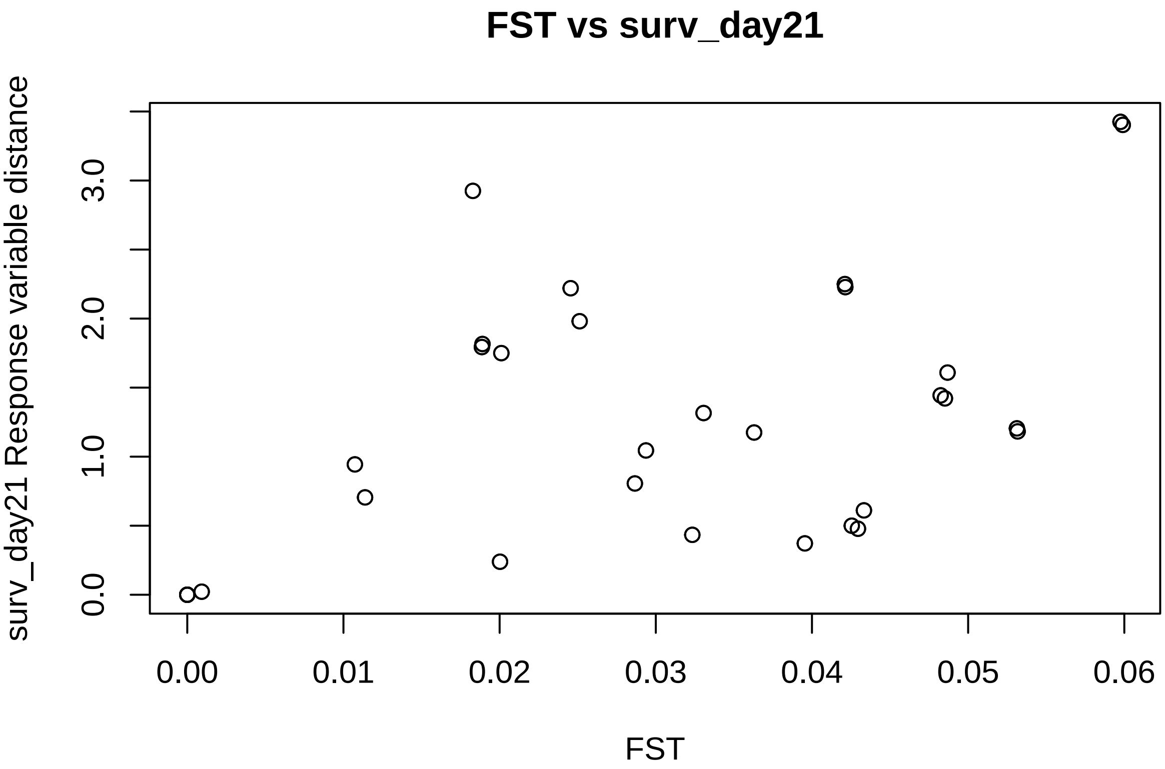
### Data check

```
## All matrices have matching row and column names.
```

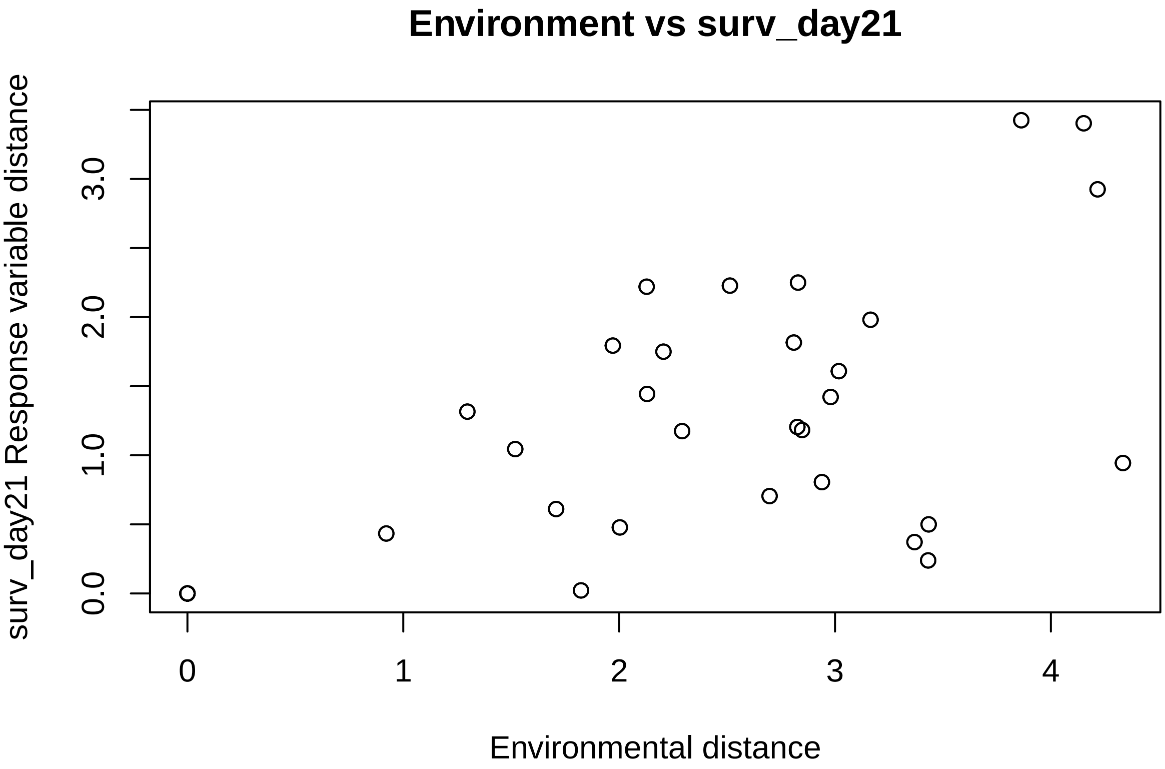
### Partial Mantel tests

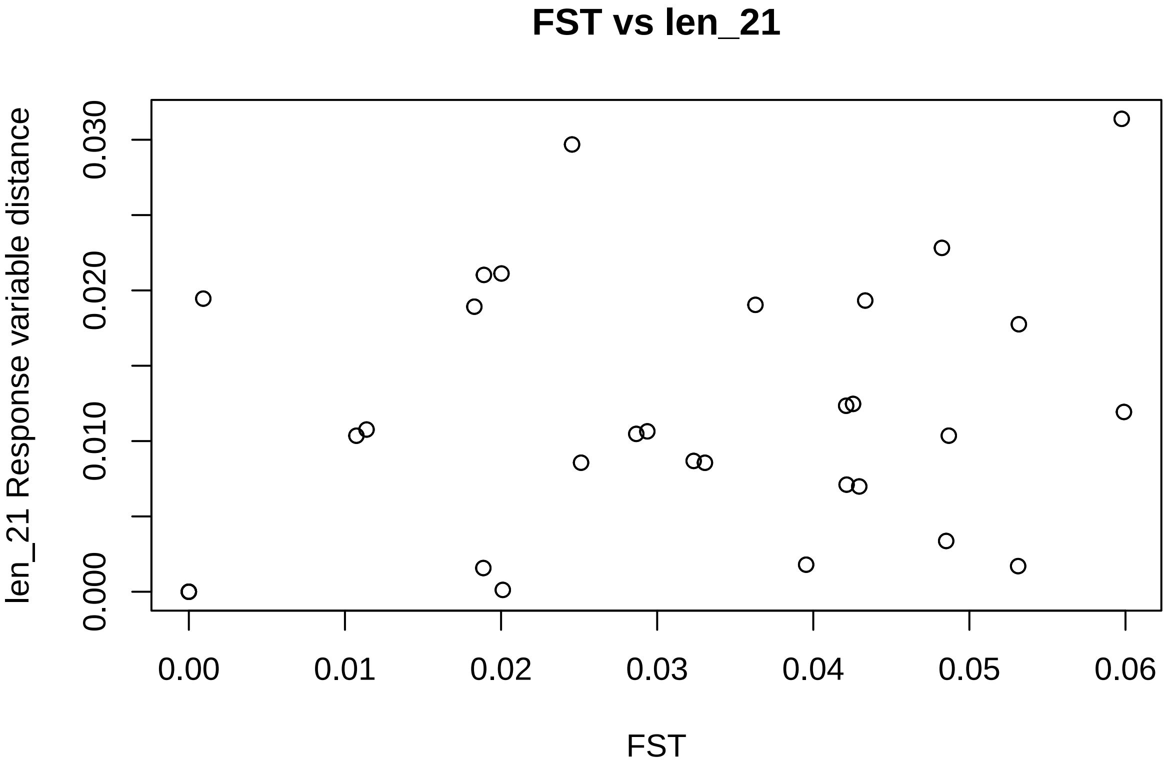
Effect of pairwise environmental distance when controlling for pairwise genetic difference, and vice certa.

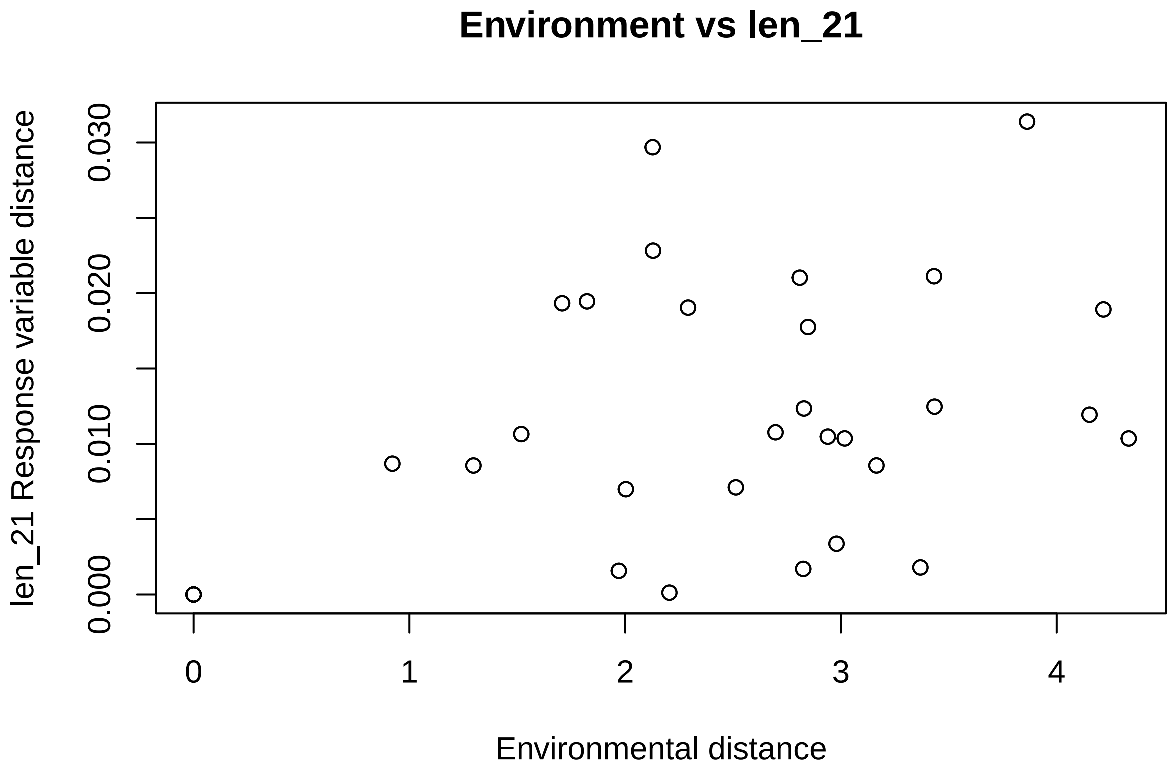
```r
# Linearity

for (name in names(dist_matrices)) {
  mat <- dist_matrices[[name]]

  # Plot against fst_mat
  plot(as.vector(as.matrix(fst_mat)), as.vector(as.matrix(mat)),
    main = paste("FST vs", name),
    xlab = "FST",
    ylab = paste(name, "Response variable distance")
  )

  # Plot against envr_dist
  plot(
    as.vector(as.matrix(envr_dist)), as.vector(as.matrix(mat)),
    main = paste("Environment vs", name),
    xlab = "Environmental distance",
    ylab = paste(name, "Response variable distance")
  )
}
```

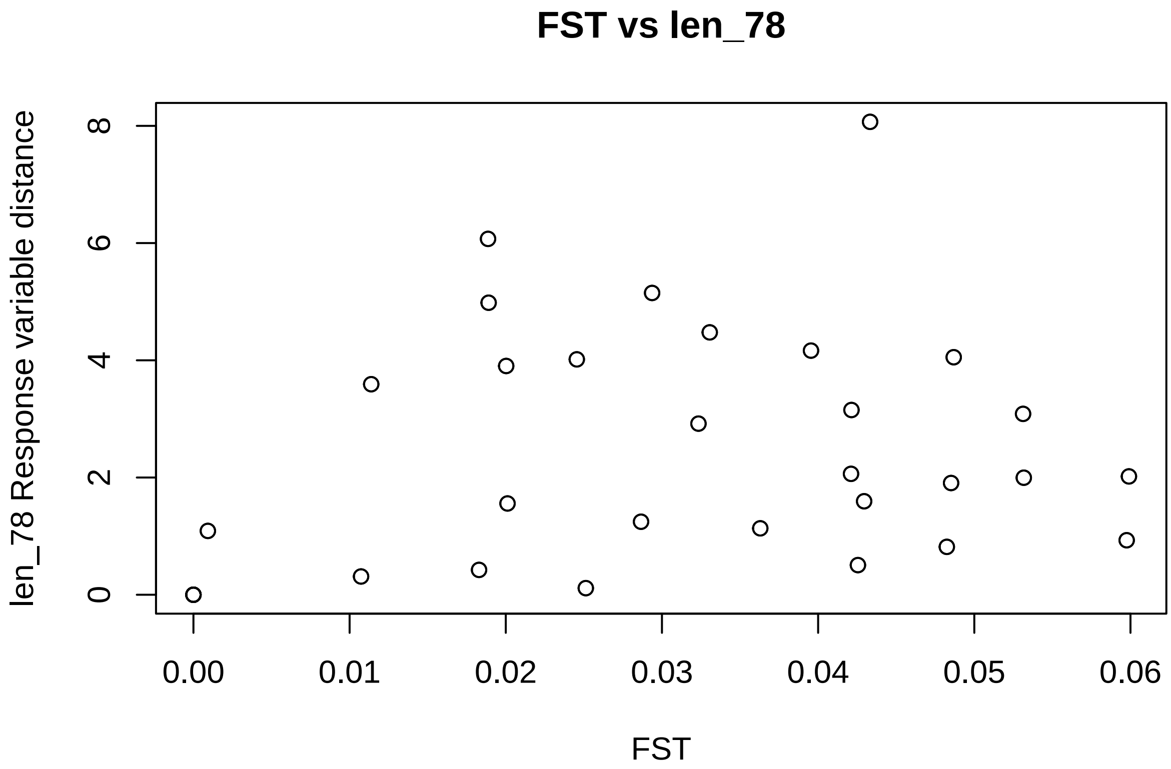


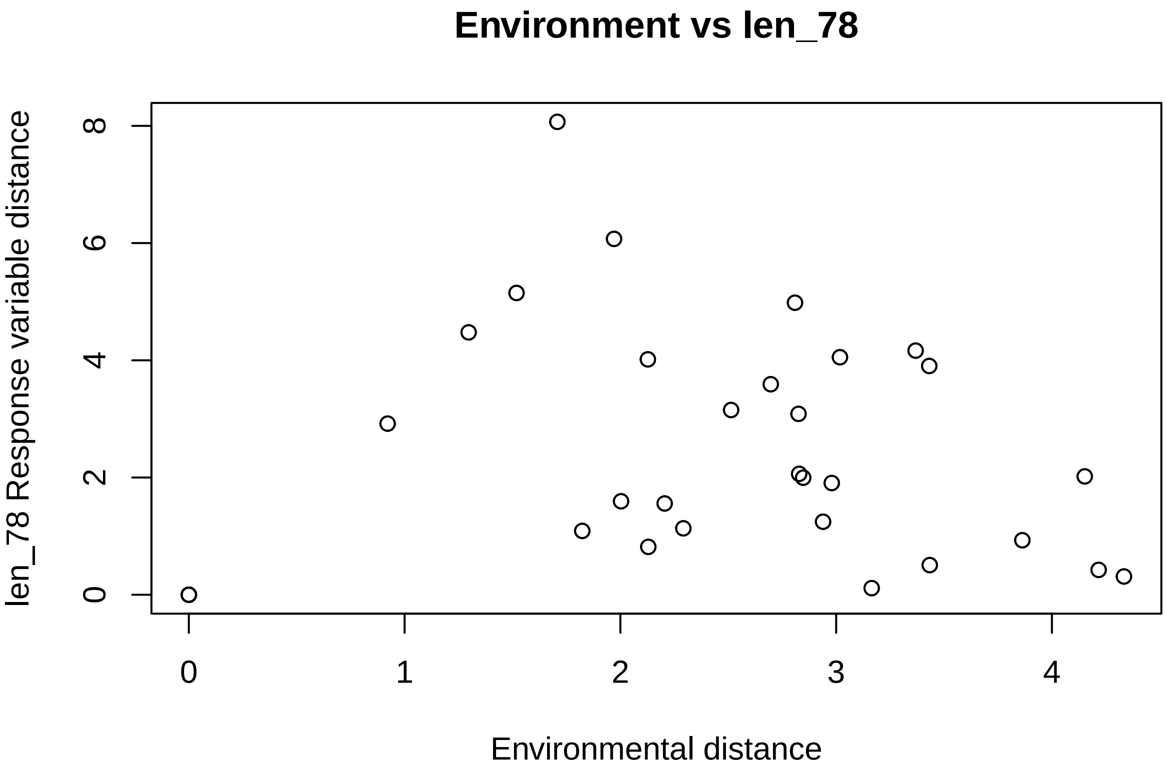
**Assumptions**

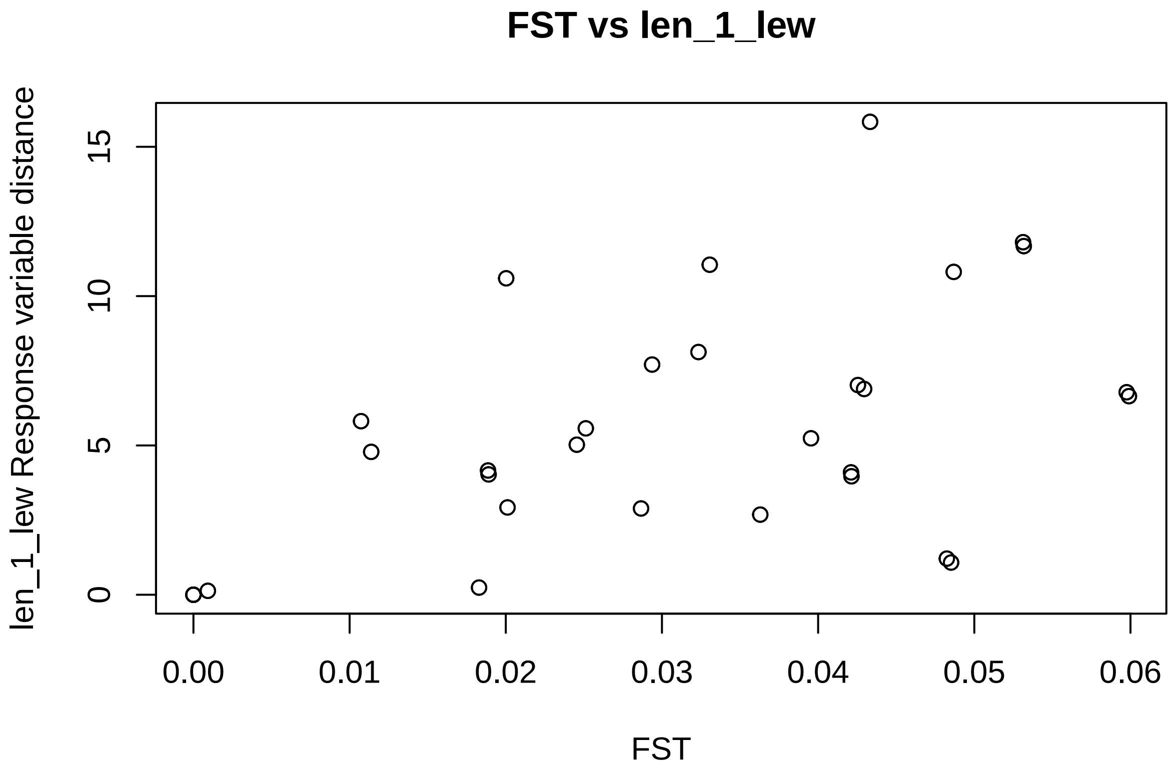


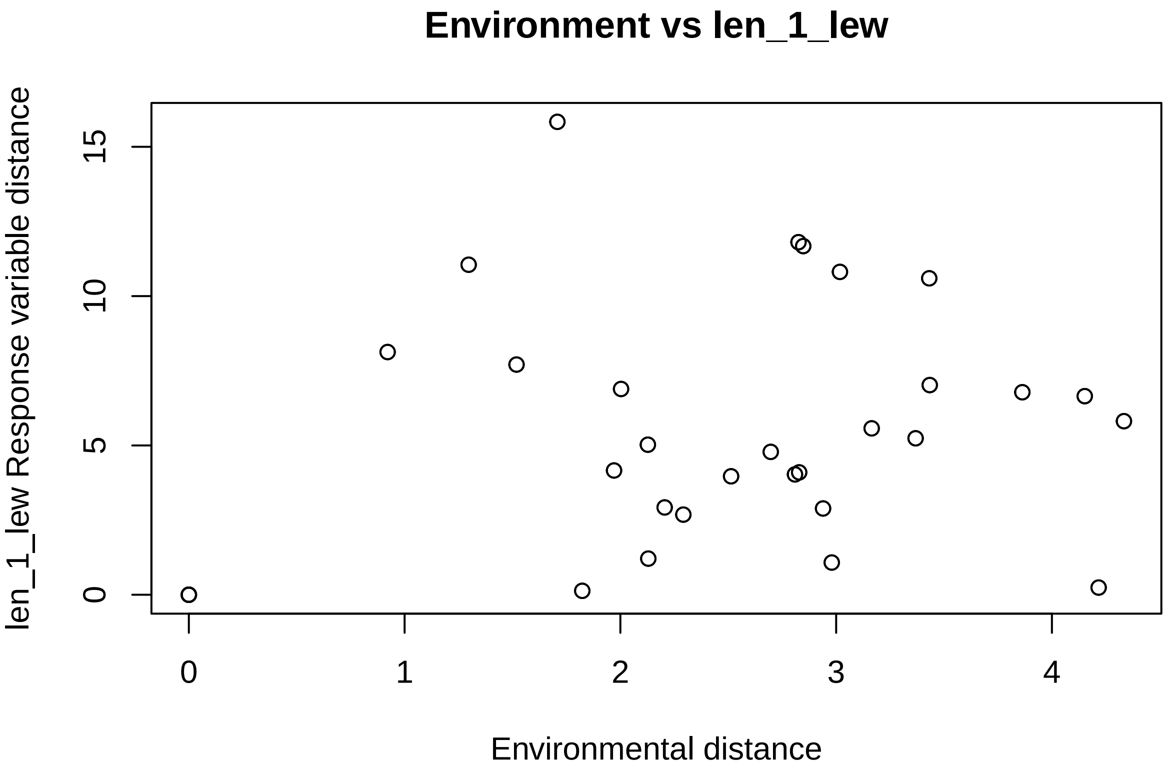

FST vs len_21
len_21 Response variable distance
0.000
0.010
0.020
0.030
0.00
0.01
0.02
0.03
0.04
0.05
0.06
FST

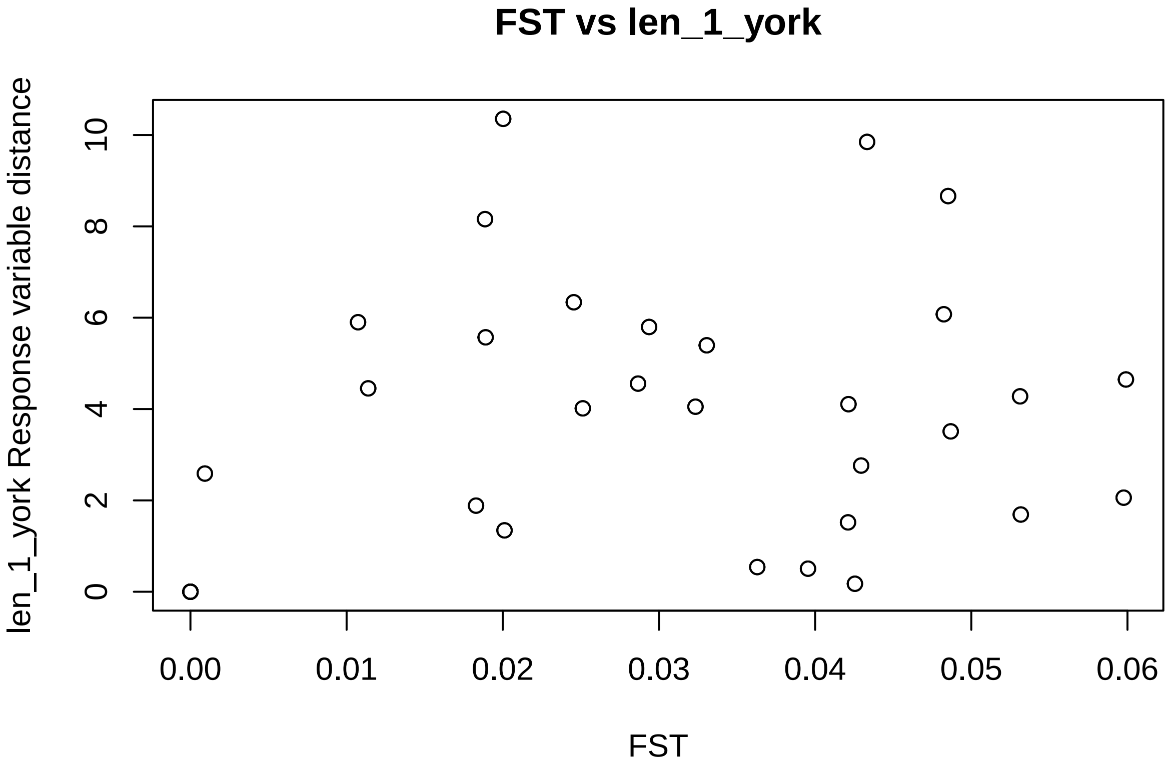

Environment vs len_21
len_21 Response variable distance
0.000
0.010
0.020
0.030
0
1
2
3
4
Environmental distance

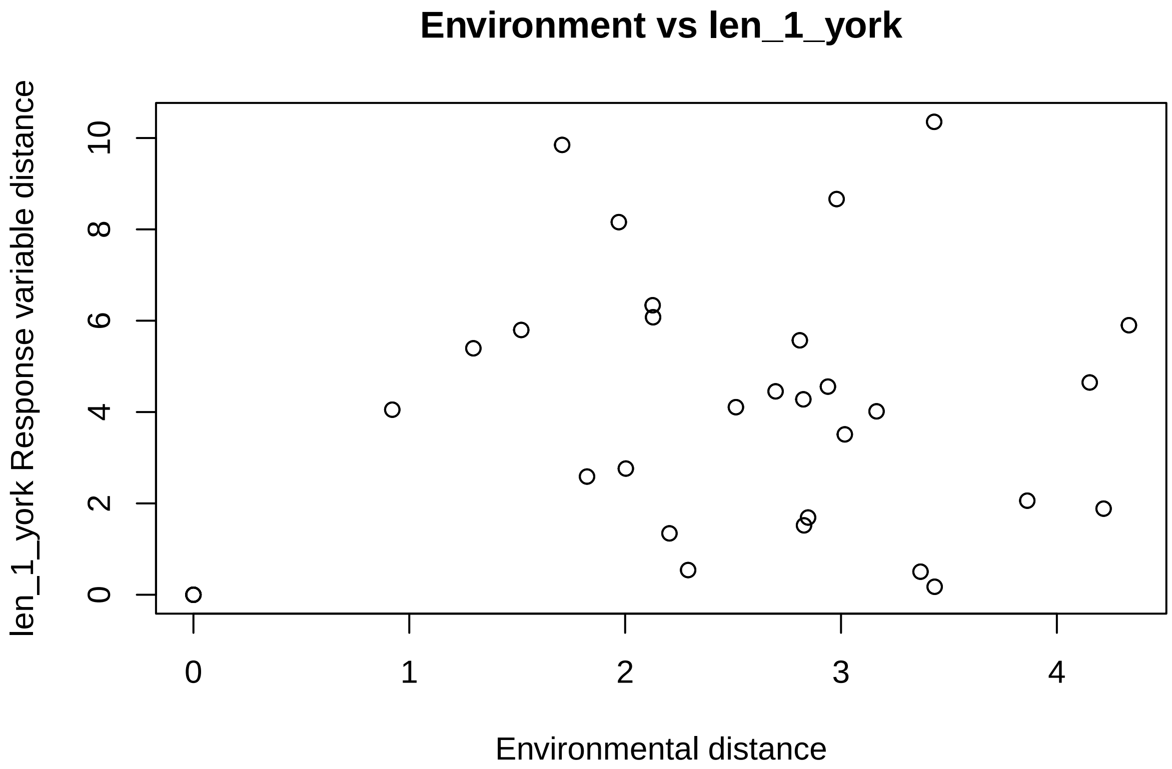

FST vs len_78
len_78 Response variable distance
0
2
4
6
8
0.00
0.01
0.02
0.03
0.04
0.05
0.06
FST

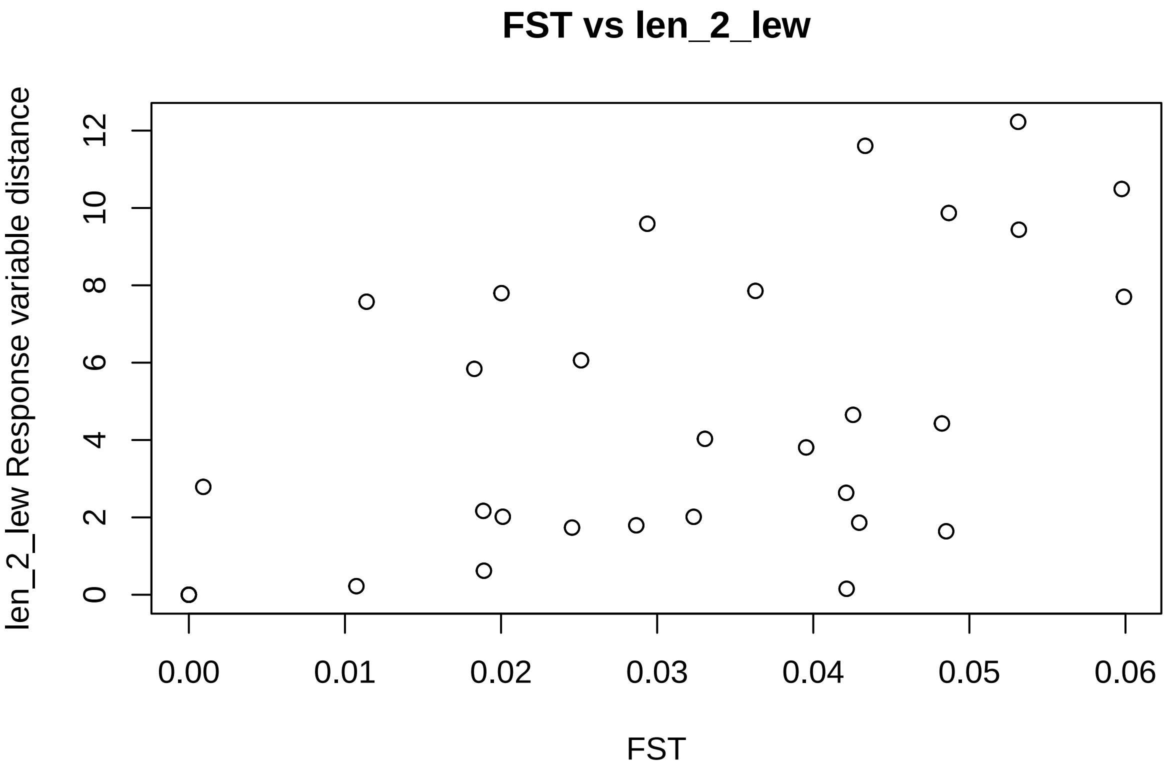

Environment vs len_78
len_78 Response variable distance
0
2
4
6
8
0
1
2
3
4
Environmental distance

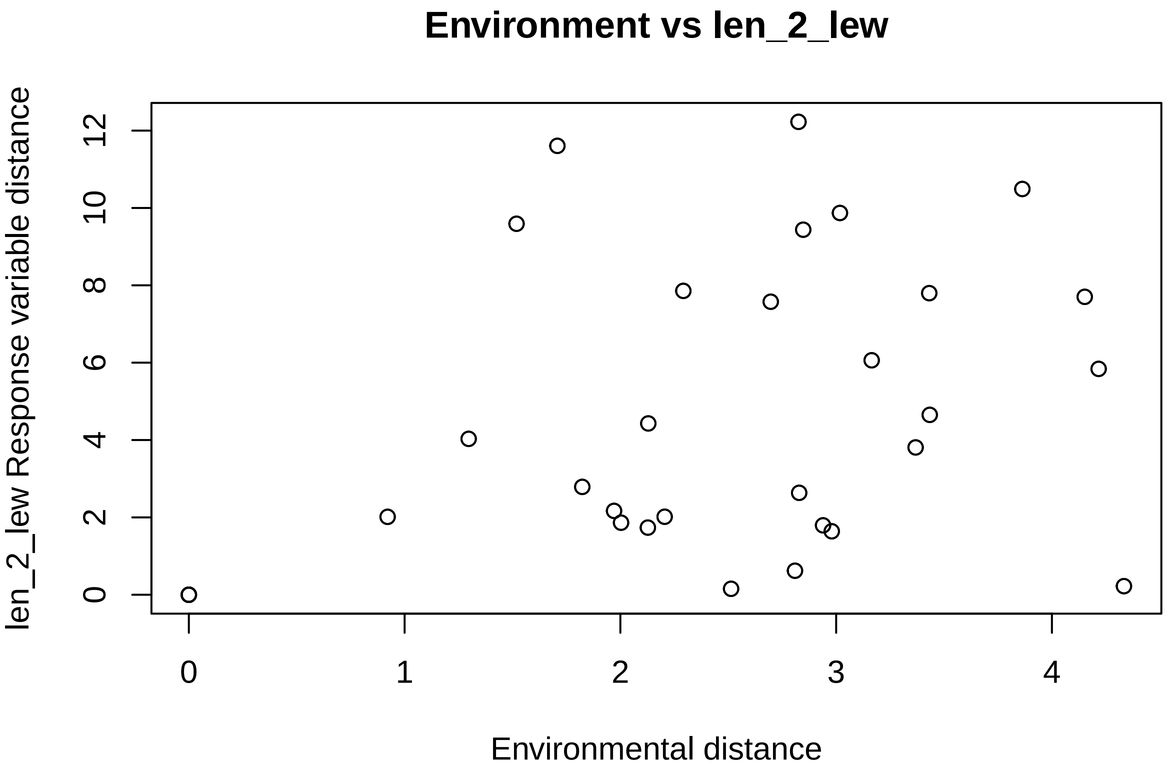

FST vs len_1_lew
len_1_lew Response variable distance
0
5
10
15
0.00
0.01
0.02
0.03
0.04
0.05
0.06
FST

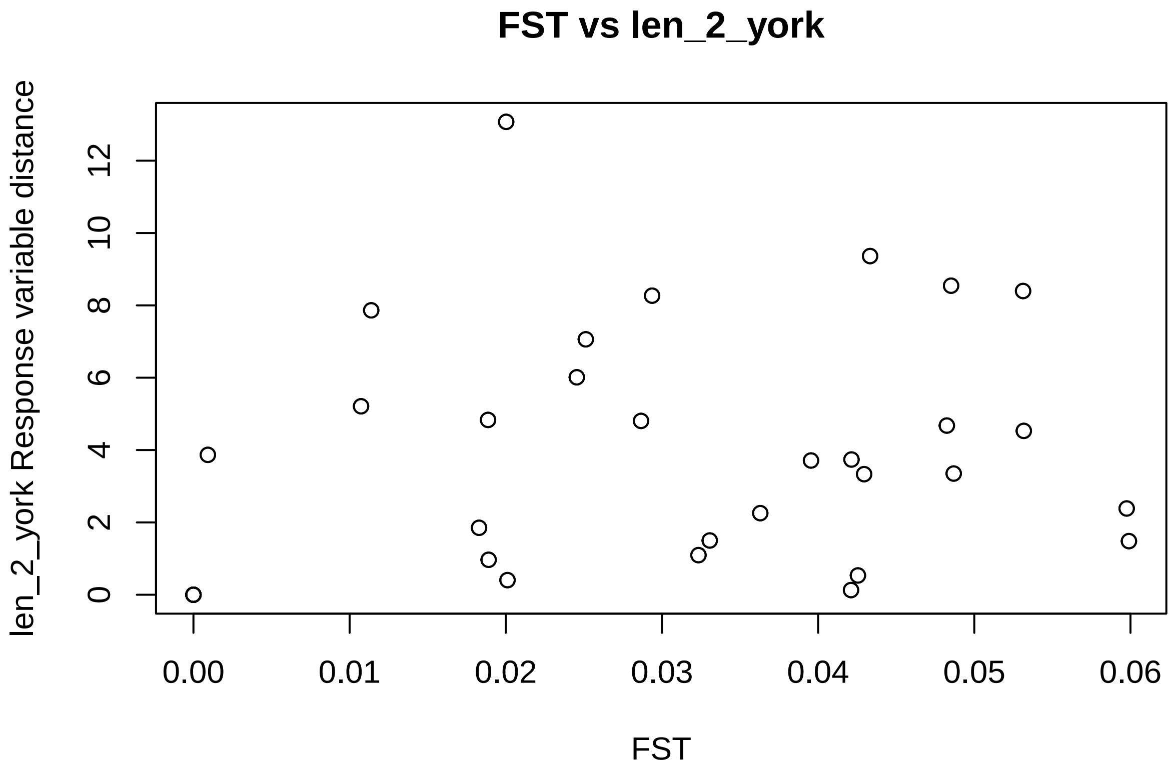

Environment vs len_1_lew
len_1_lew Response variable distance
0
5
10
15
0
1
2
3
4
Environmental distance

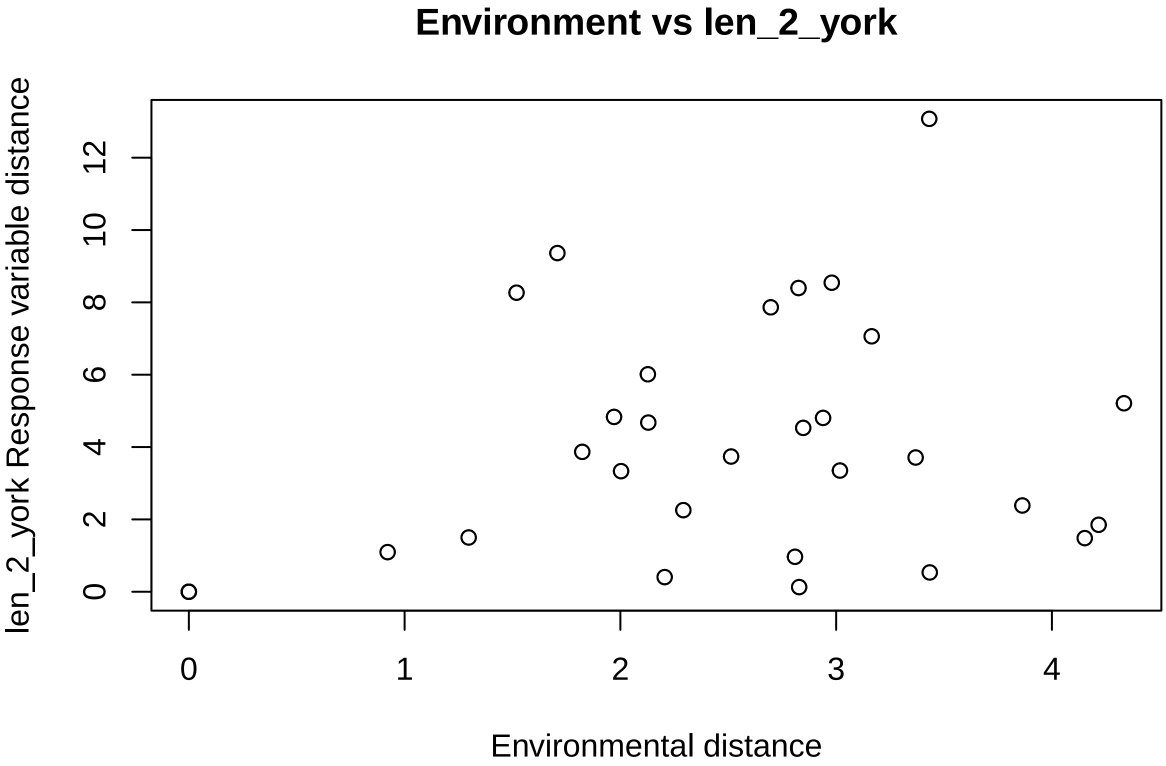

FST vs len_1_york
len_1_york Response variable distance
0
2
4
6
8
10
0.00
0.01
0.02
0.03
0.04
0.05
0.06
FST

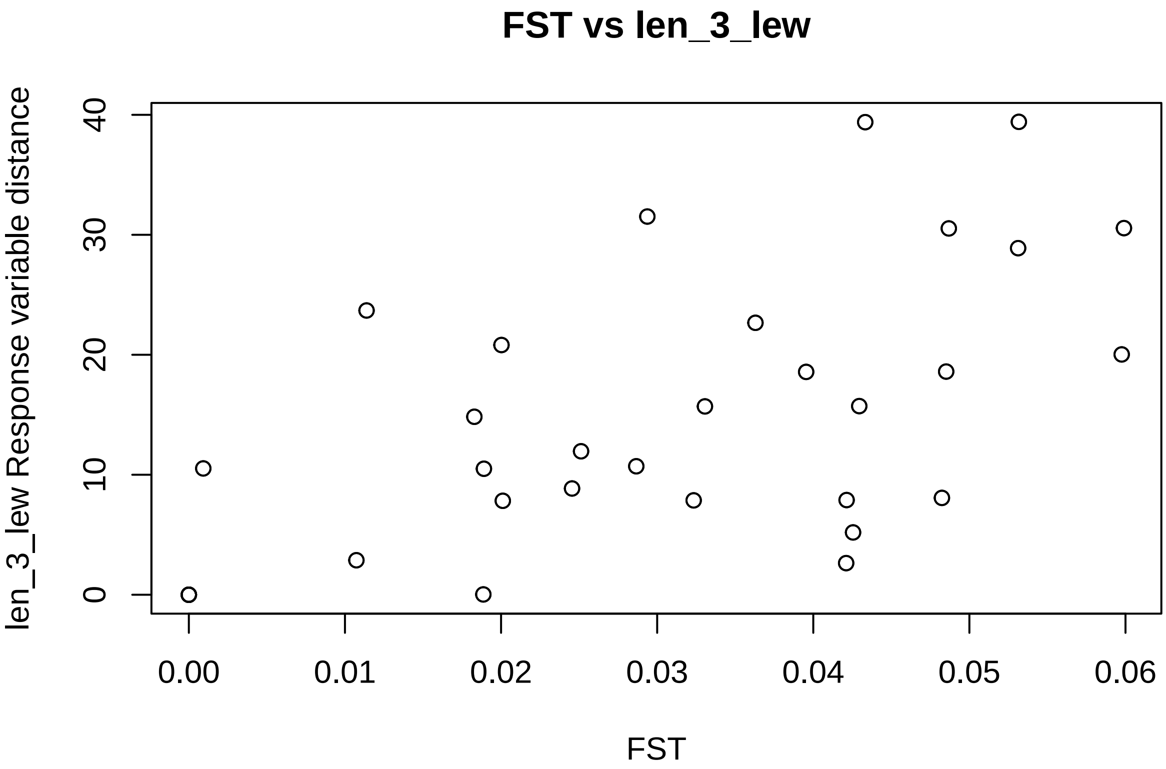

Environment vs len_1_york
len_1_york Response variable distance
0
2
4
6
8
10
0
1
2
3
4
Environmental distance

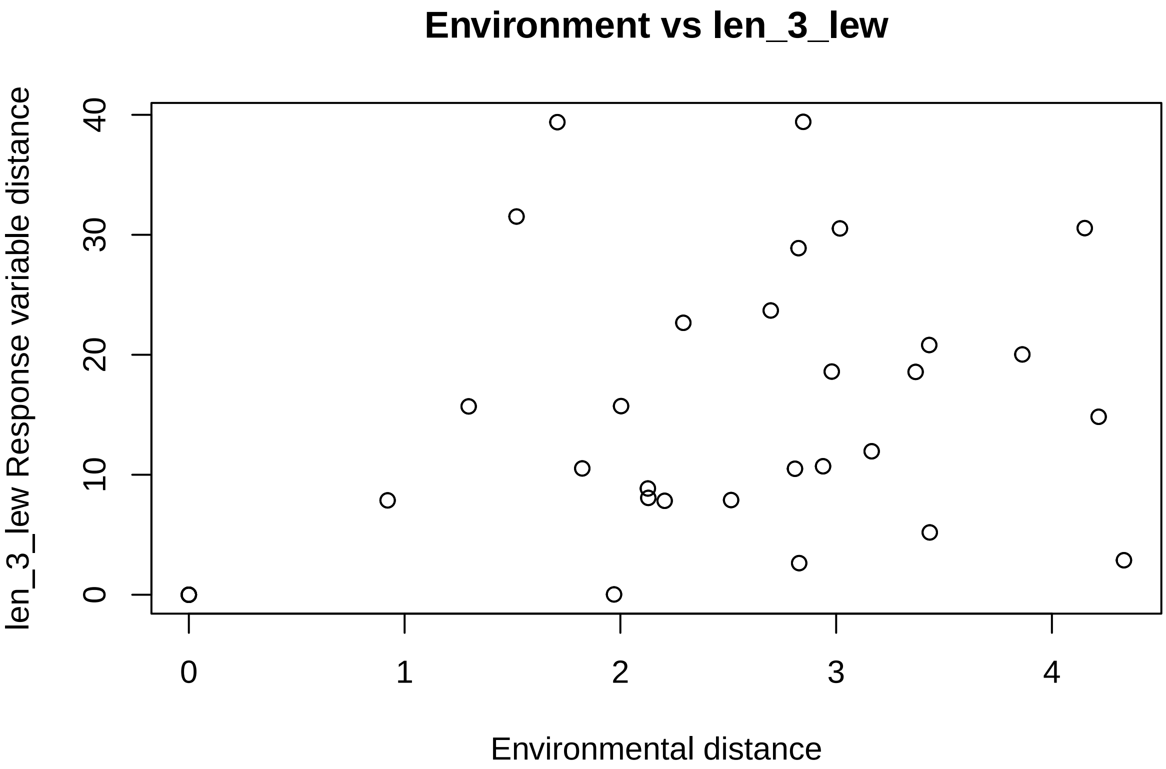

FST vs len_2_lew
len_2_lew Response variable distance
0
2
4
6
8
10
12
0.00
0.01
0.02
0.03
0.04
0.05
0.06
FST

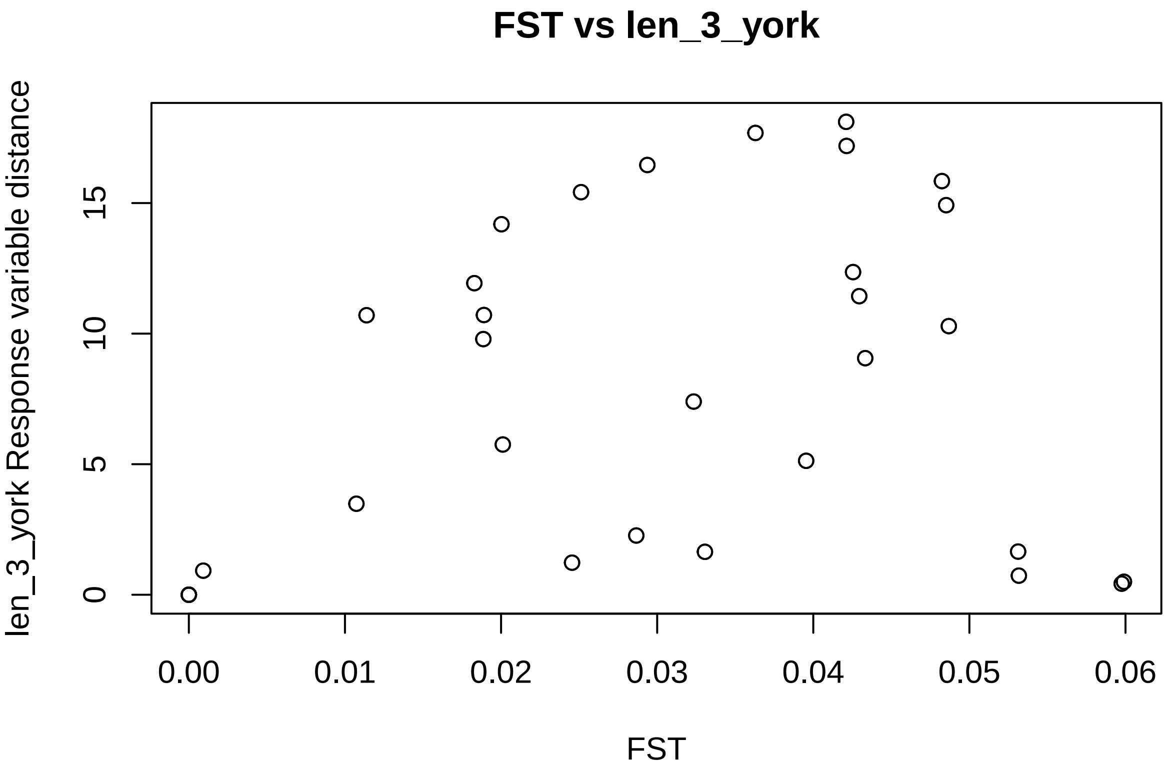

Environment vs len_2_lew
len_2_lew Response variable distance
0
2
4
6
8
10
12
0
1
2
3
4
Environmental distance

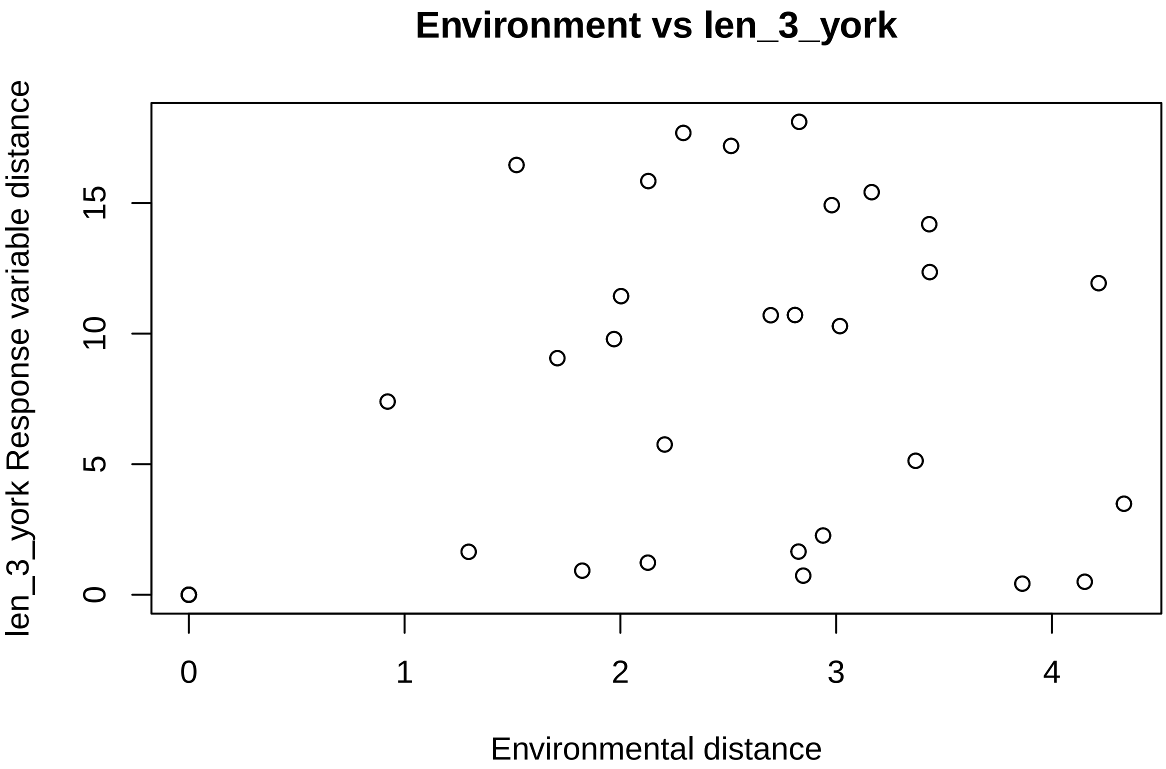

FST vs len_2_york
len_2_york Response variable distance
0
2
4
6
8
10
12
0.00
0.01
0.02
0.03
0.04
0.05
0.06
FST

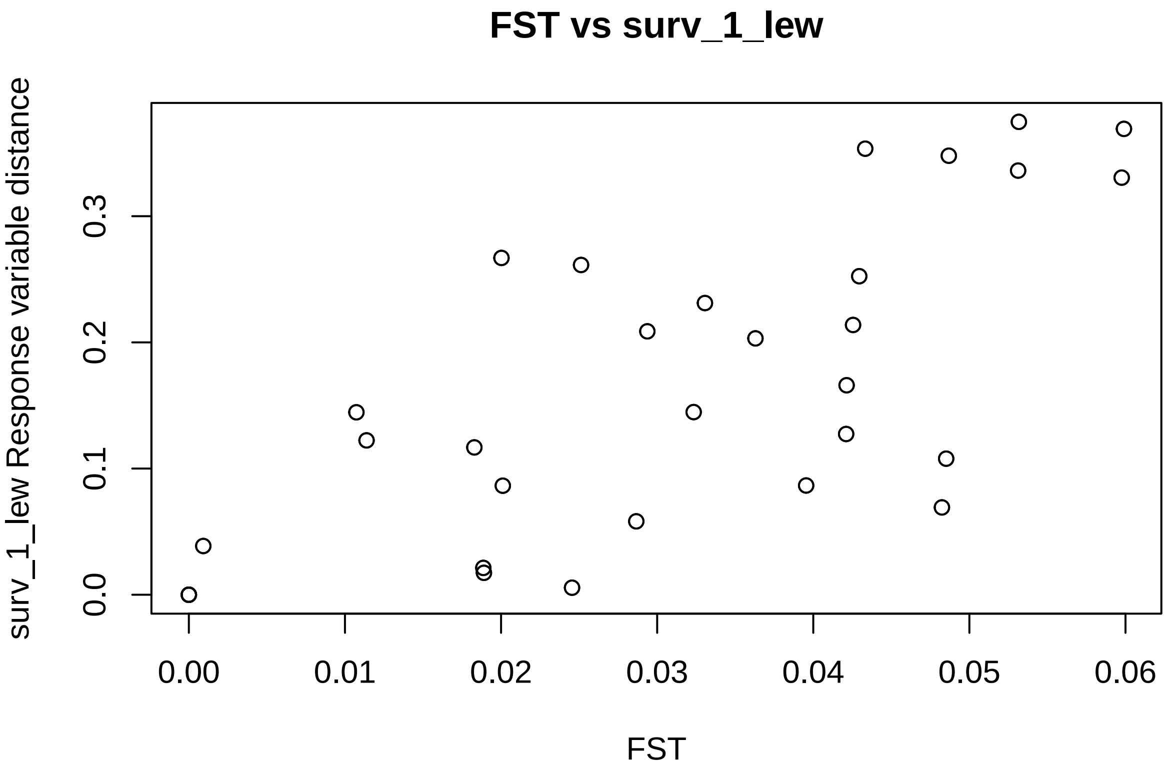

Environment vs len_2_york
len_2_york Response variable distance
0
2
4
6
8
10
12
0
1
2
3
4
Environmental distance

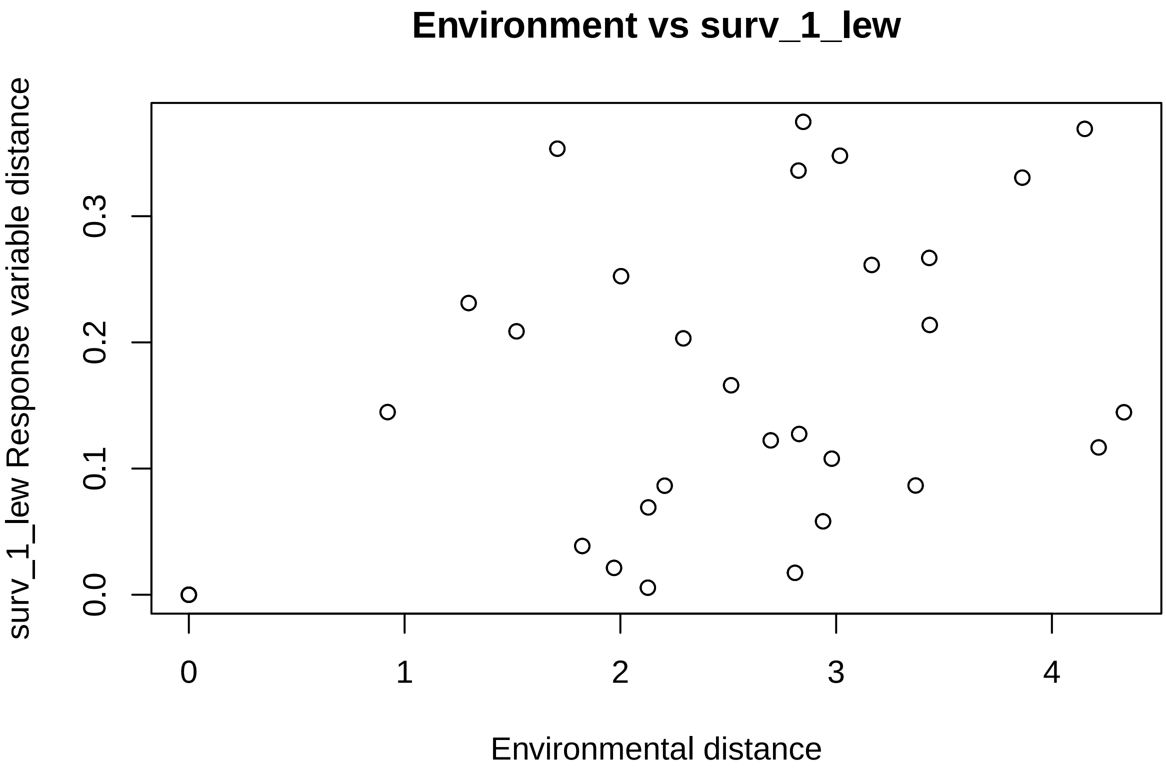

FST vs len_3_lew
len_3_lew Response variable distance
0
10
20
30
40
0.00
0.01
0.02
0.03
0.04
0.05
0.06
FST

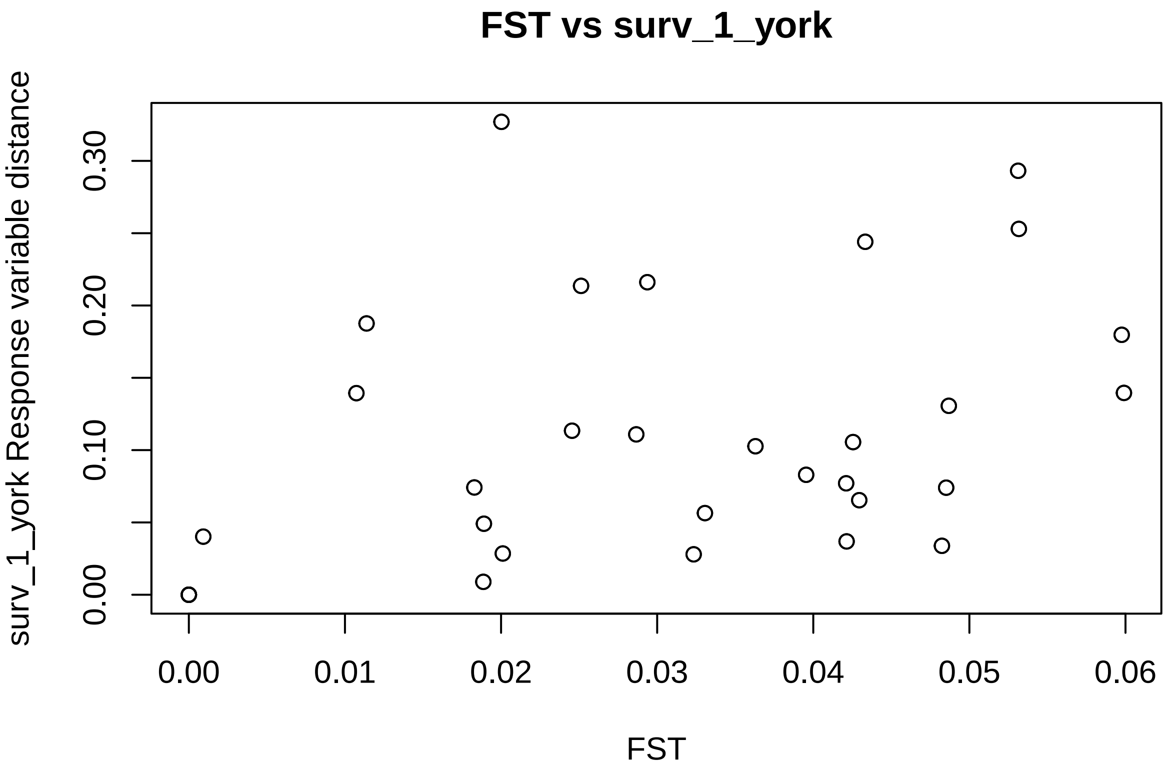

Environment vs len_3_lew
len_3_lew Response variable distance
0
10
20
30
40
0
1
2
3
4
Environmental distance

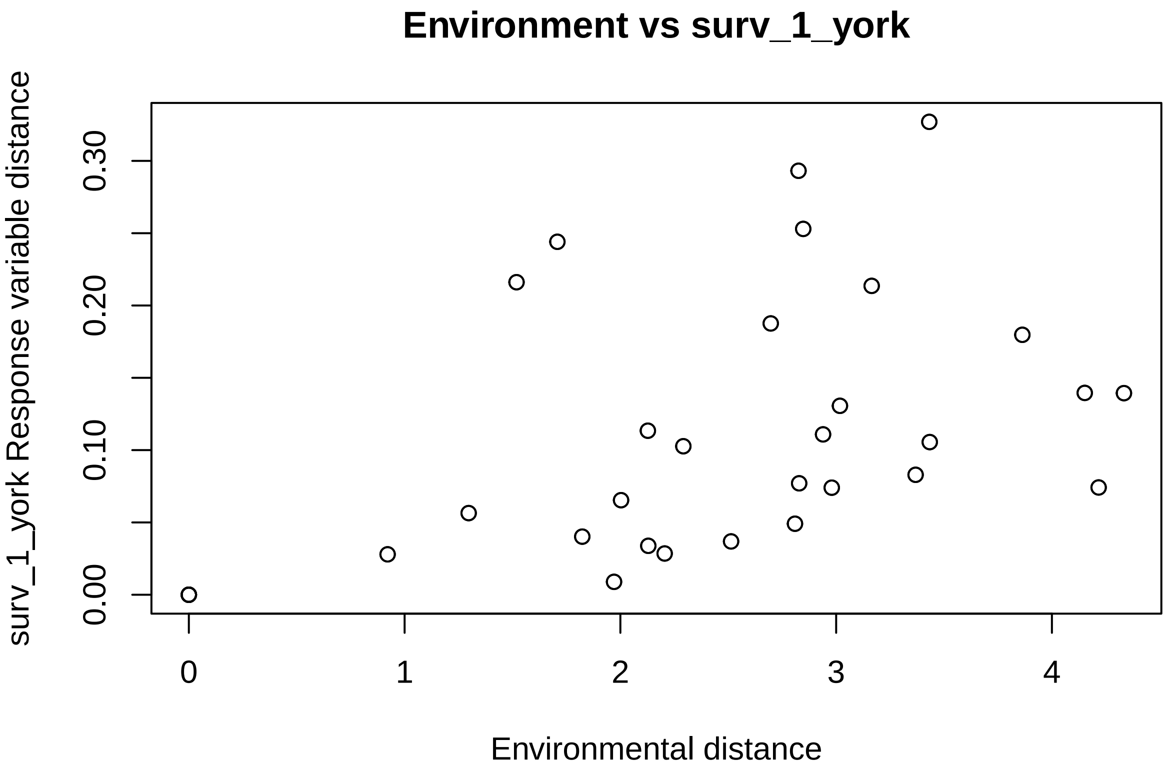

FST vs len_3_york
len_3_york Response variable distance
0
5
10
15
0.00
0.01
0.02
0.03
0.04
0.05
0.06
FST

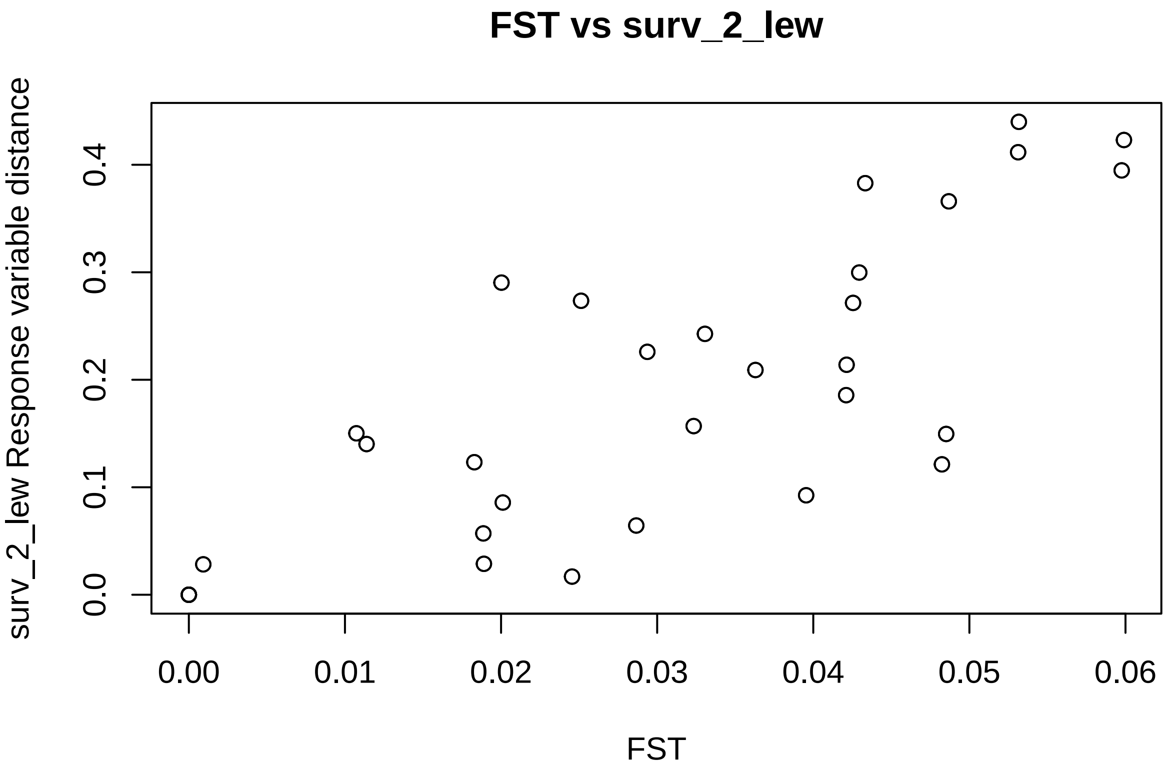

Environment vs len_3_york
len_3_york Response variable distance
0
5
10
15
0
1
2
3
4
Environmental distance

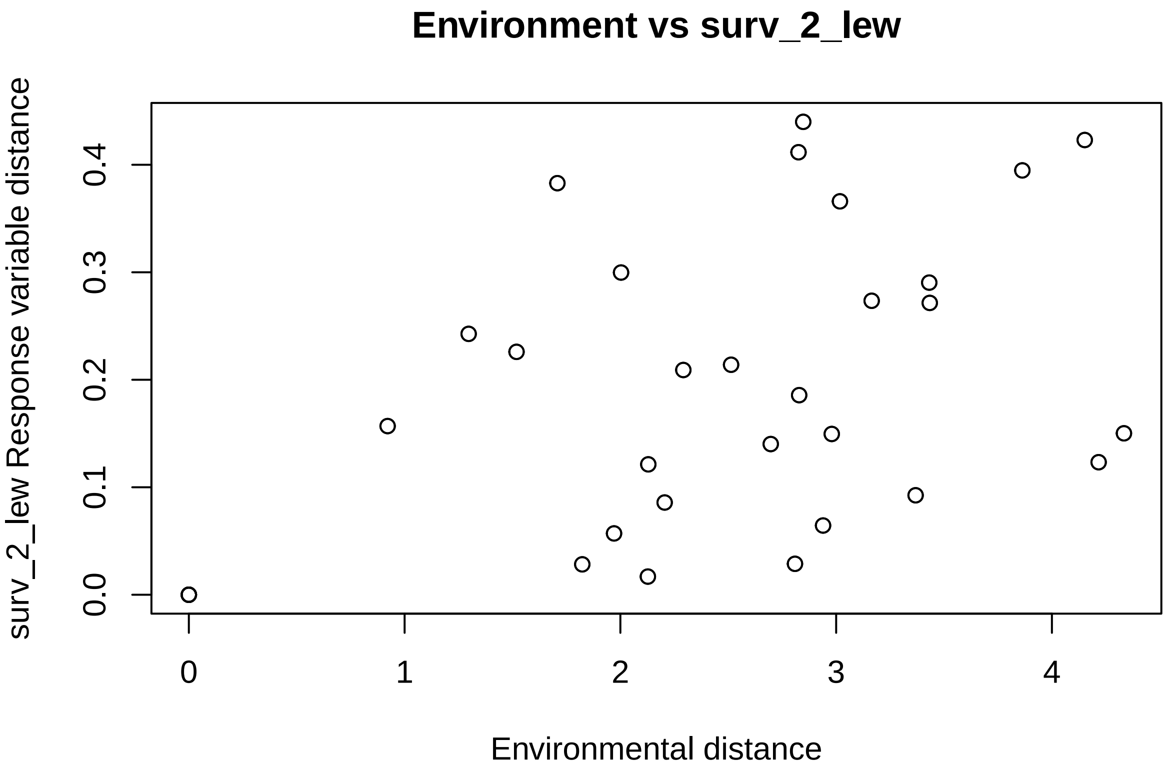

FST vs surv_1_lew
surv_1_lew Response variable distance
0.0
0.1
0.2
0.3
0.00
0.01
0.02
0.03
0.04
0.05
0.06
FST

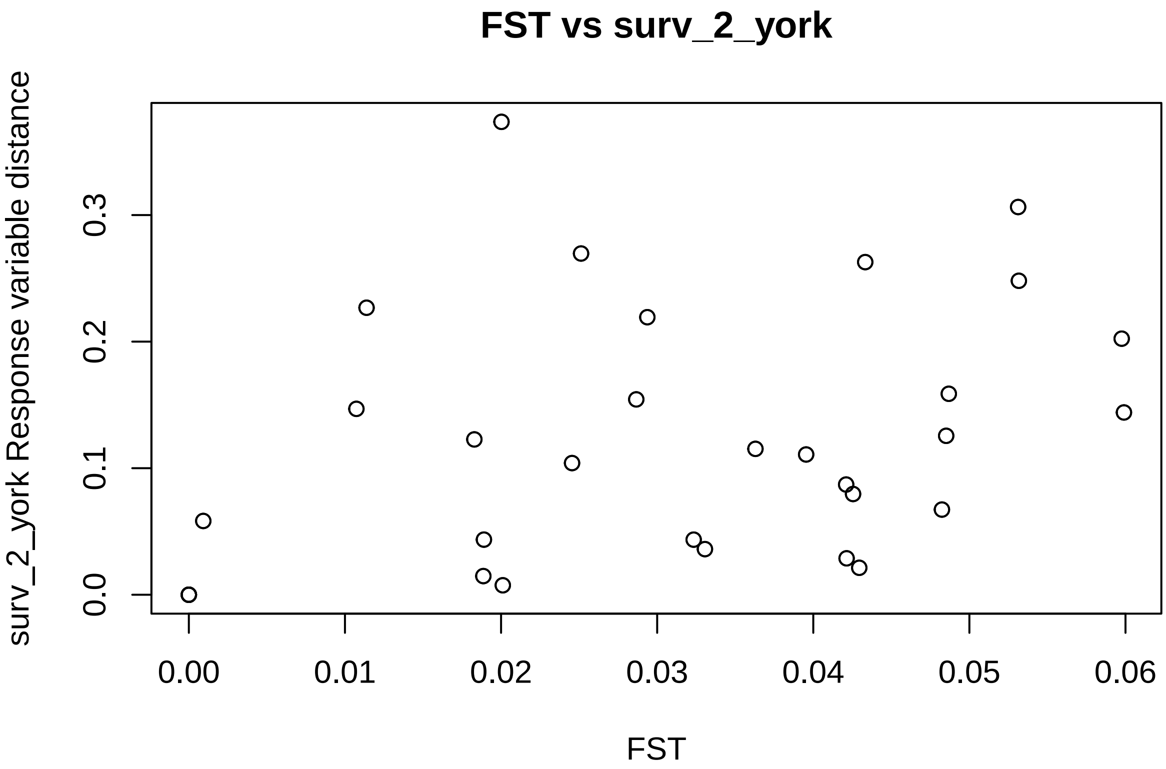

Environment vs surv_1_lew
surv_1_lew Response variable distance
0.0
0.1
0.2
0.3
0
1
2
3
4
Environmental distance

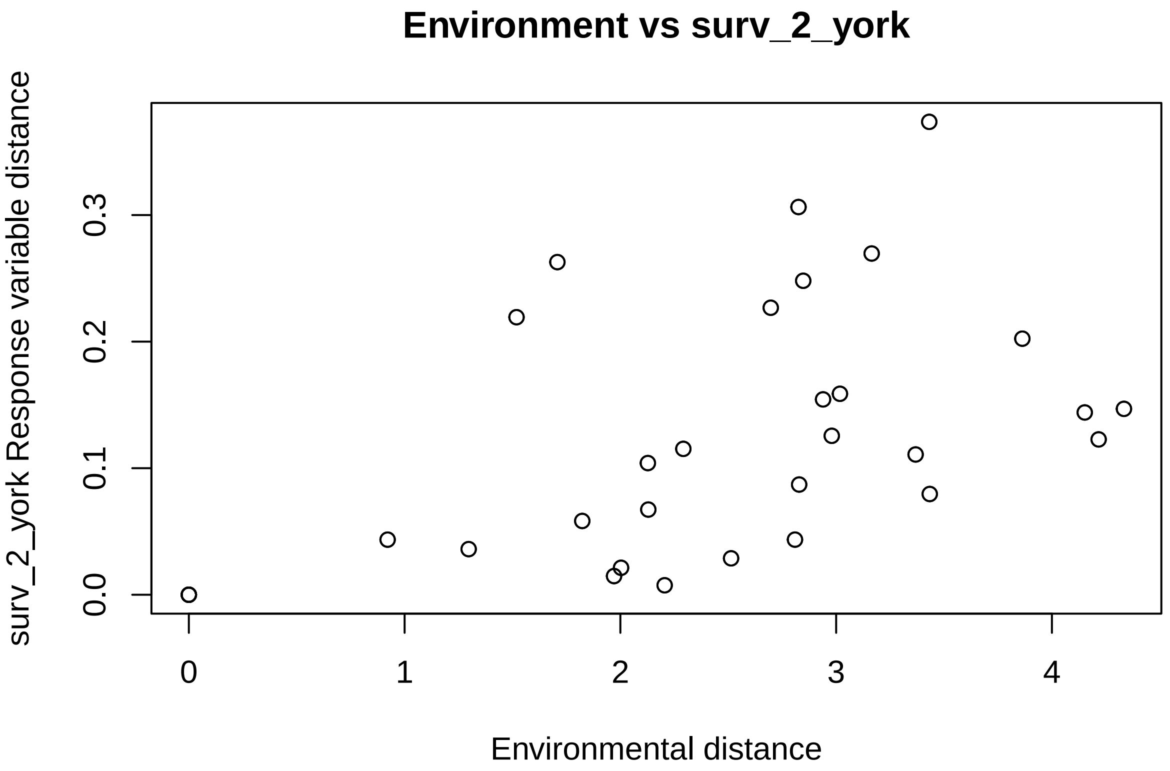

FST vs surv_1_york
surv_1_york Response variable distance
0.00
0.10
0.20
0.30
0.00
0.01
0.02
0.03
0.04
0.05
0.06
FST

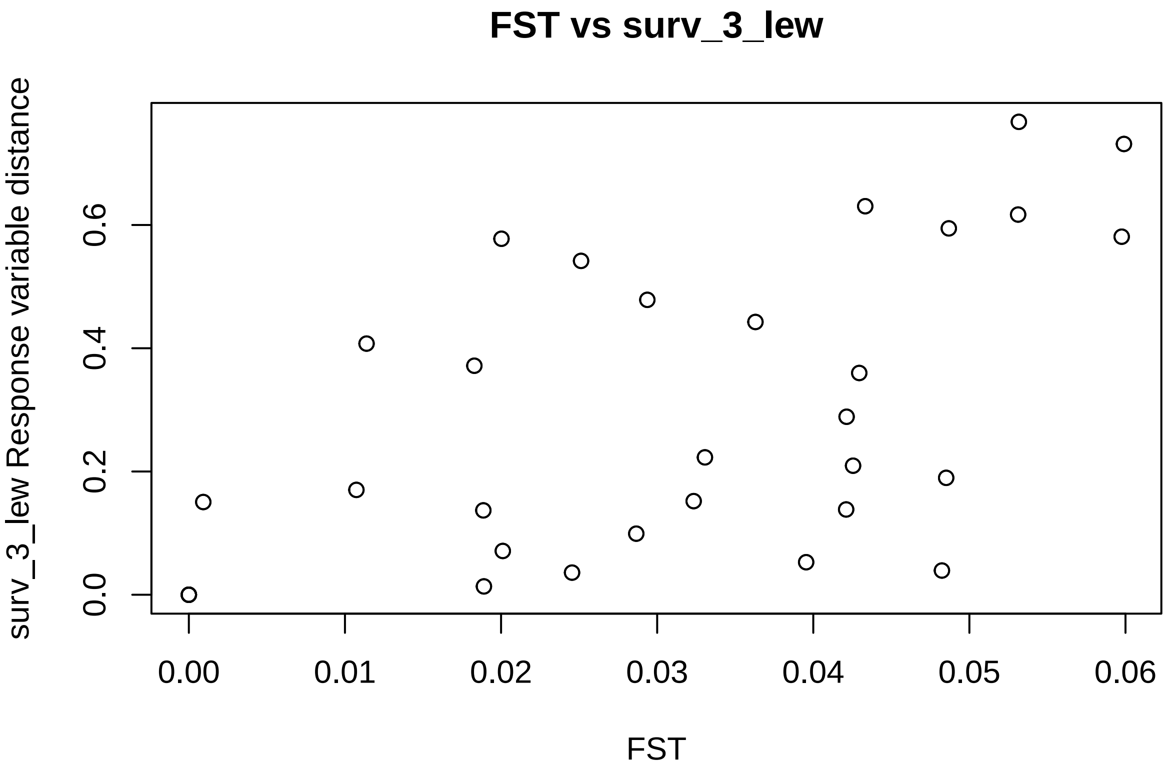

Environment vs surv_1_york
surv_1_york Response variable distance
0.00
0.10
0.20
0.30
0
1
2
3
4
Environmental distance

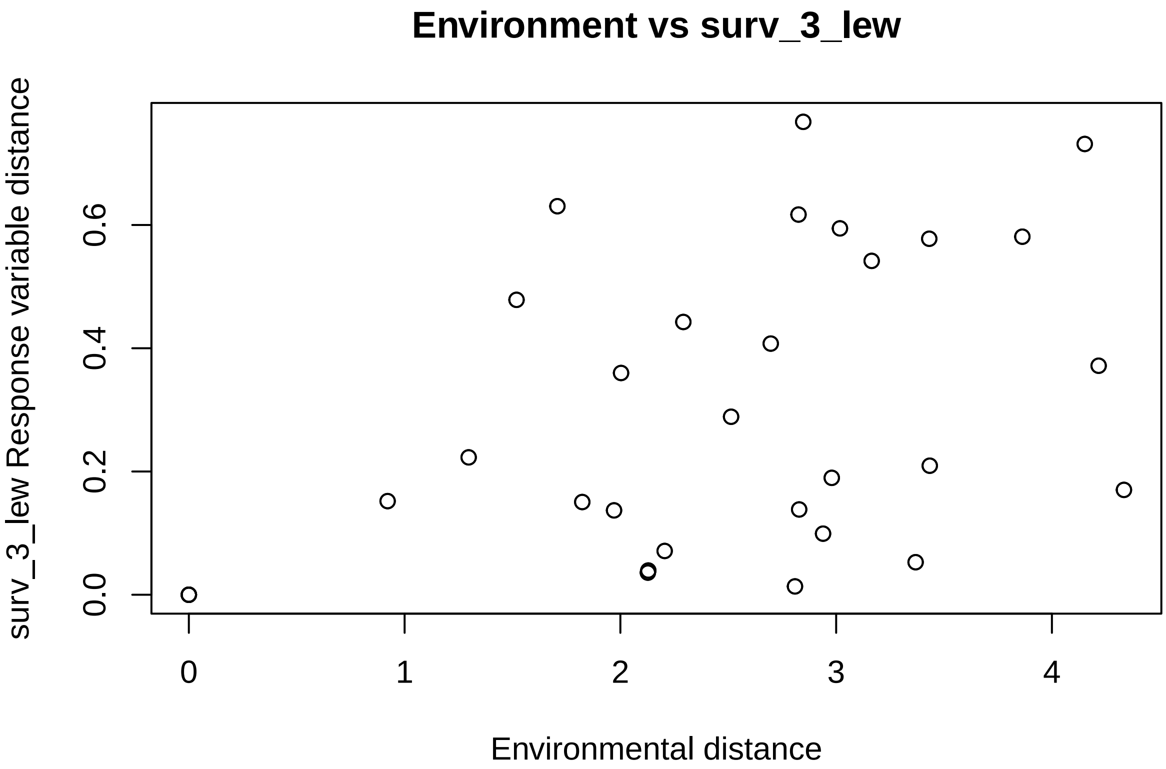

FST vs surv_2_lew
surv_2_lew Response variable distance
0.0
0.1
0.2
0.3
0.4
0.00
0.01
0.02
0.03
0.04
0.05
0.06
FST

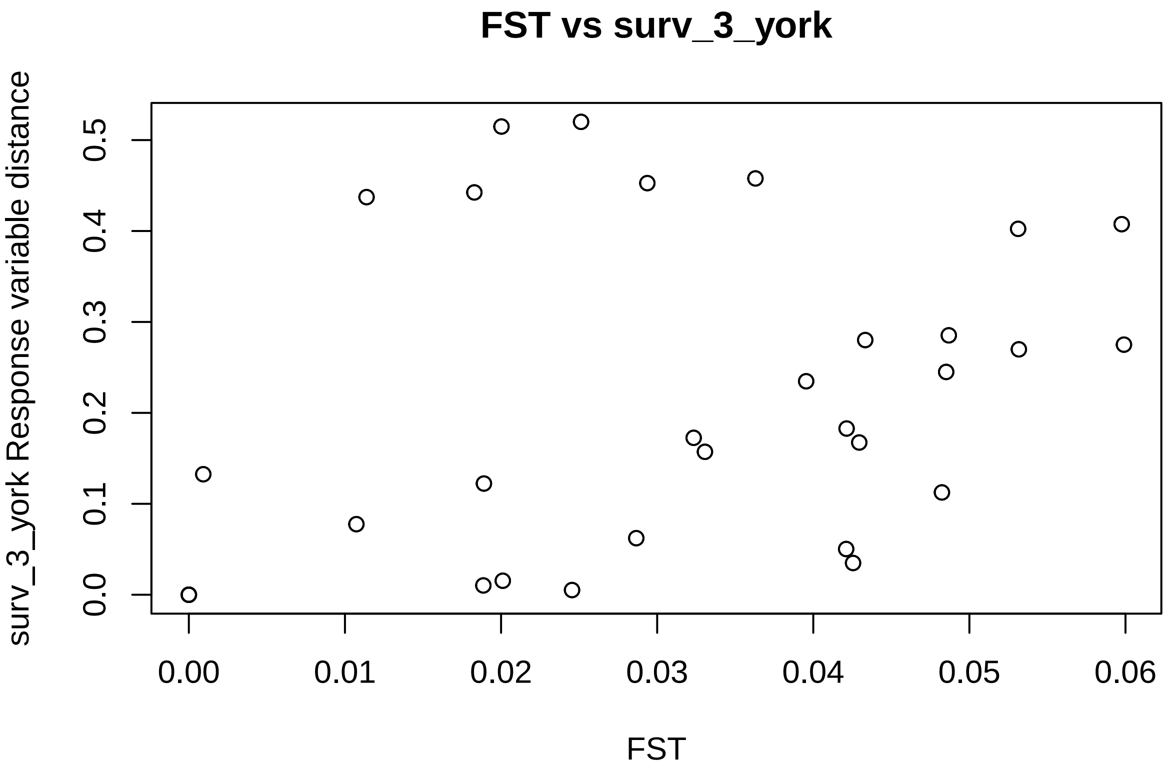

Environment vs surv_2_lew
surv_2_lew Response variable distance
0.0
0.1
0.2
0.3
0.4
0
1
2
3
4
Environmental distance

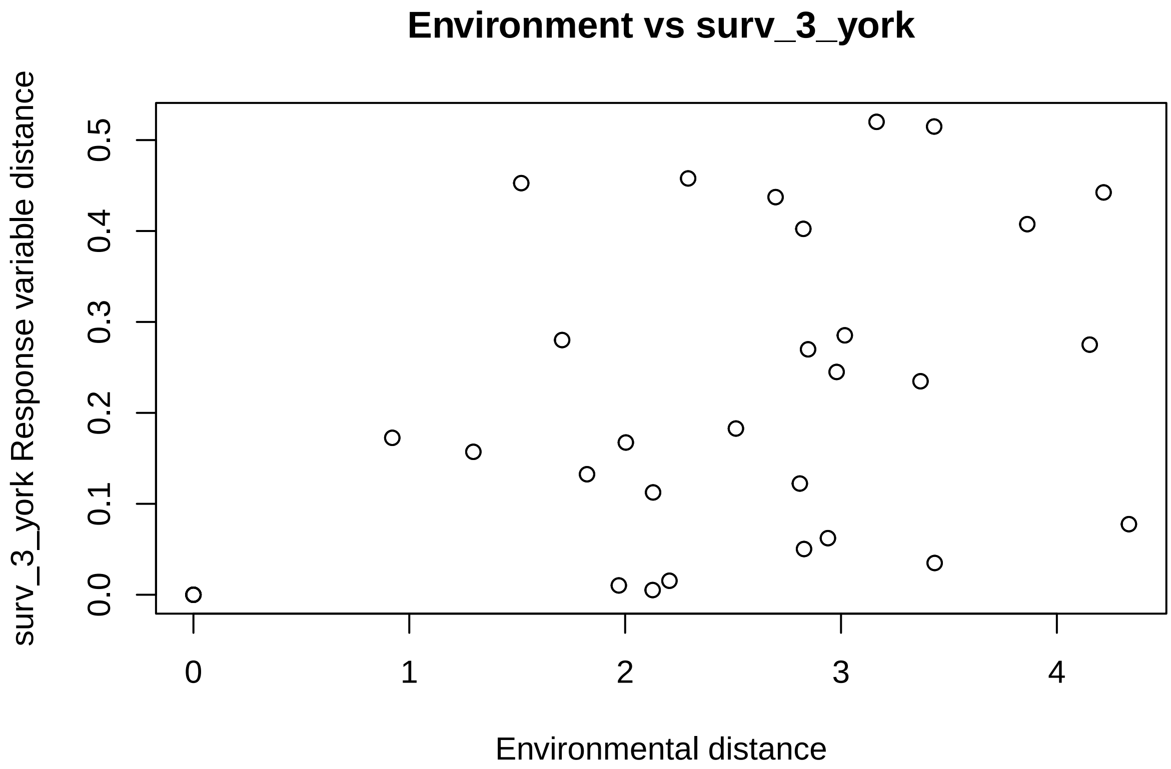

FST vs surv_2_york
surv_2_york Response variable distance
0.0
0.1
0.2
0.3
0.00
0.01
0.02
0.03
0.04
0.05
0.06
FST


Environment vs surv_2_york
surv_2_york Response variable distance
0.0
0.1
0.2
0.3
0
1
2
3
4
Environmental distance


FST vs surv_3_lew
surv_3_lew Response variable distance
0.0
0.2
0.4
0.6
0.00
0.01
0.02
0.03
0.04
0.05
0.06
FST


Environment vs surv_3_lew
surv_3_lew Response variable distance
0.0
0.2
0.4
0.6
0
1
2
3
4
Environmental distance


FST vs surv_3_york
surv_3_york Response variable distance
0.0
0.1
0.2
0.3
0.4
0.5
0.00
0.01
0.02
0.03
0.04
0.05
0.06
FST



**Partial tests** To test the effect of genetic distance while controlling for environmental distance.

```
# run partial Mantel tests
partial_mantel_results <- lapply(dist_matrices, function(dist_mat) {
  mantel.partial(dist_mat, fst_mat, envr_dist, method = "spearman",
                 permutations = 10000, na.rm = TRUE)
})

# extract p-values
partial_mantel_p <- sapply(partial_mantel_results, function(res) res$signif)

# adjust p-values (Benjamini-Hochberg)
partial_mantel_p_adj <- p.adjust(partial_mantel_p, method = "BH")

# output results
results_df <- data.frame(
  test = names(partial_mantel_results),
  p = partial_mantel_p,
  p_adj = partial_mantel_p_adj
)

print(results_df)
```

```
##                    test          p     p_adj
## surv_day21   surv_day21 0.14838516 0.2696730
## len_21           len_21 0.51714829 0.6464354
## len_78           len_78 0.28537146 0.3891429
## len_1_lew     len_1_lew 0.03439656 0.1031897
## len_1_york   len_1_york 0.69793021 0.6979302
## len_2_lew     len_2_lew 0.04699530 0.1174883
## len_2_york   len_2_york 0.64953505 0.6959304
## len_3_lew     len_3_lew 0.02449755 0.1031897
## len_3_york   len_3_york 0.63373663 0.6959304
## surv_1_lew   surv_1_lew 0.01469853 0.1031897
## surv_1_york surv_1_york 0.12348765 0.2646164
## surv_2_lew   surv_2_lew 0.00669933 0.1004900
## surv_2_york surv_2_york 0.17158284 0.2696730
## surv_3_lew   surv_3_lew 0.03289671 0.1031897
## surv_3_york surv_3_york 0.17978202 0.2696730
```

To test the effect of environmental distance while controlling for genetic distance.

```
# run partial Mantel tests
partial_mantel_results <- lapply(dist_matrices, function(dist_mat) {
  mantel.partial(dist_mat, envr_dist, fst_mat, method = "spearman",
                 permutations = 10000, na.rm = TRUE)
})

# extract p-values
partial_mantel_p <- sapply(partial_mantel_results, function(res) res$signif)

# adjust p-values (Benjamini-Hochberg)
partial_mantel_p_adj <- p.adjust(partial_mantel_p, method = "BH")

# output results
results_df <- data.frame(
  test = names(partial_mantel_results),
  p = partial_mantel_p,
  p_adj = partial_mantel_p_adj
)

print(results_df)
```

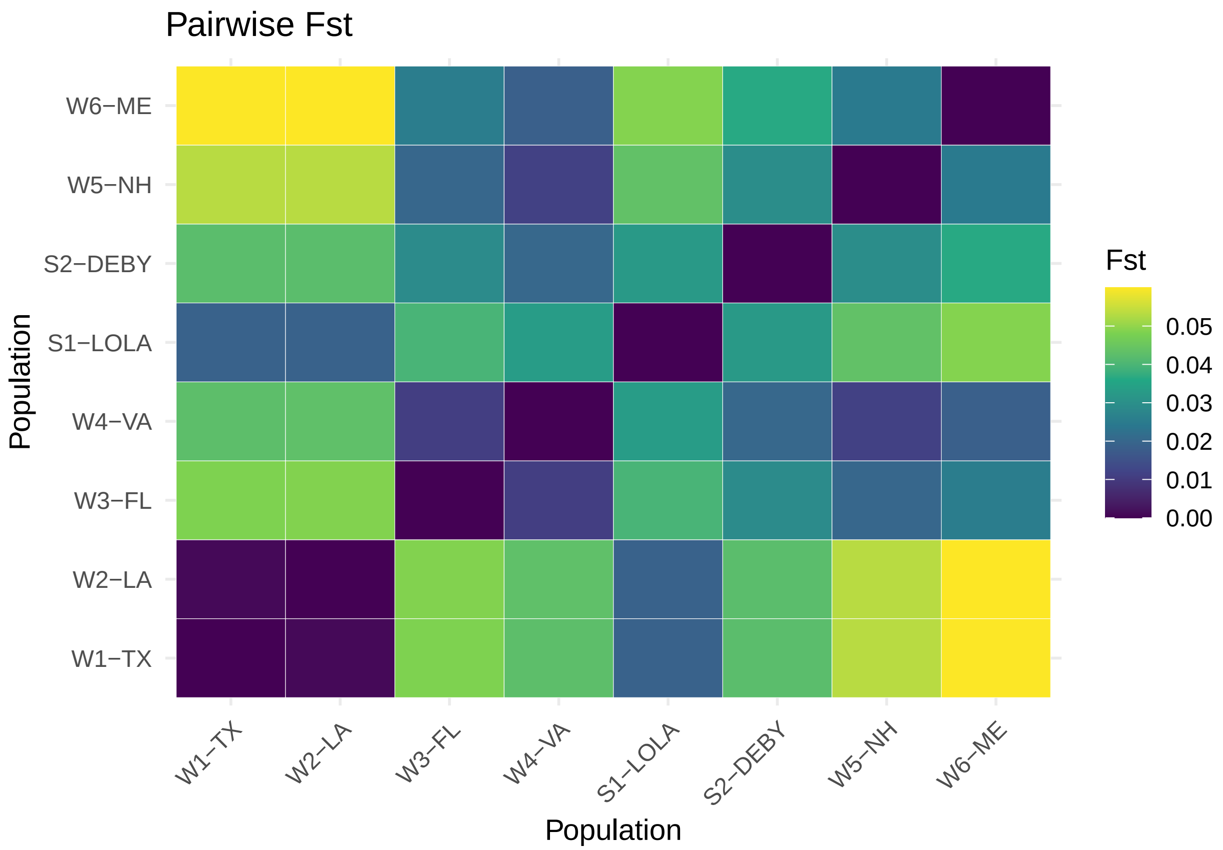
```
##                    test          p     p_adj
## surv_day21   surv_day21 0.14658534 0.3801048
## len_21           len_21 0.28087191 0.5266348
## len_78           len_78 0.98980102 0.9898010
## len_1_lew     len_1_lew 0.69923008 0.8068039
## len_1_york   len_1_york 0.83391661 0.8934821
## len_2_lew     len_2_lew 0.37816218 0.6302703
## len_2_york   len_2_york 0.54954505 0.7121788
## len_3_lew     len_3_lew 0.47435256 0.7115288
## len_3_york   len_3_york 0.56974303 0.7121788
## surv_1_lew   surv_1_lew 0.17738226 0.3801048
## surv_1_york surv_1_york 0.06429357 0.3801048
## surv_2_lew   surv_2_lew 0.15408459 0.3801048
## surv_2_york surv_2_york 0.05519448 0.3801048
## surv_3_lew   surv_3_lew 0.16928307 0.3801048
## surv_3_york surv_3_york 0.14488551 0.3801048
```

### Heatmaps

```
fst_long <- as.data.frame(as.table(fst_mat))

heatmap_fst <- ggplot(fst_long, aes(Var1, Var2, fill = Freq)) +
  geom_tile(color = "white") +
  #geom_text(aes(label = round(value, 3)), size = 2, color = "black") +
  scale_fill_viridis_c(option = "D", na.value = "grey50") +  # Use viridis color scale
  labs(title = "Pairwise Fst", x = "Population", y = "Population", fill = "Fst") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

ggsave(file.path("..", "..", "figures", "pairwise_differences_analyses", "fig_fst_heatmap.png"), heatmap
```

```
## Saving 6.5 x 4.5 in image
```

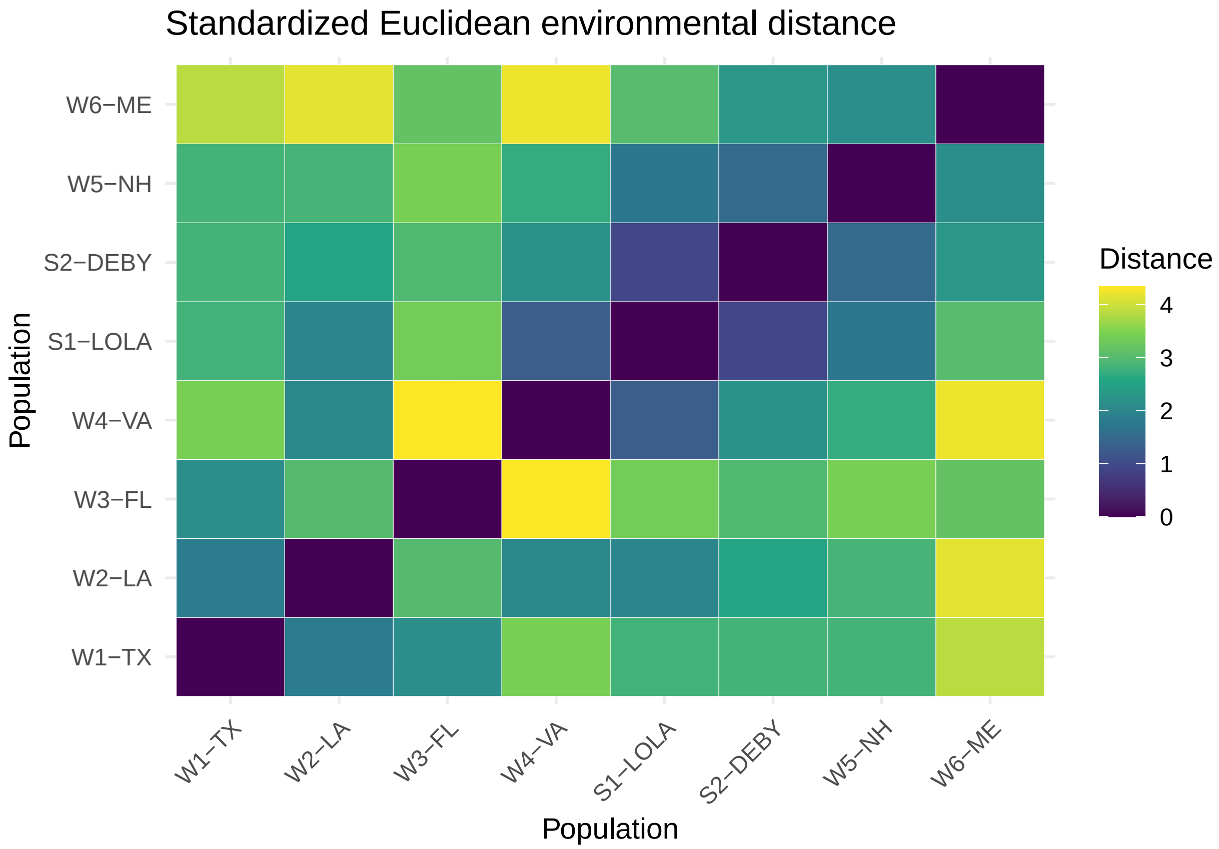
```
heatmap_fst
```



```
envr_long <- as.data.frame(as.table(envr_dist))

heatmap_envr <- ggplot(envr_long, aes(Var1, Var2, fill = Freq)) +
  geom_tile(color = "white") +
  #geom_text(aes(label = round(value, 3)), size = 2, color = "black") +
  scale_fill_viridis_c(option = "D", na.value = "grey50") +  # Use viridis color scale
  labs(title = "Standardized Euclidean environmental distance",
       x = "Population", y = "Population", fill = "Distance") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

ggsave(file.path("..", "..", "figures", "pairwise_differences_analyses", "fig_envr_dist_heatmap.png"), 
```

```
## Saving 6.5 x 4.5 in image
```

```
heatmap_envr
```


Standardized Euclidean environmental distance
W6−ME
W5−NH
S2−DEBY
S1−LOLA
W4−VA
W3−FL
W2−LA
W1−TX
Population
W1−TX
W2−LA
W3−FL
W4−VA
S1−LOLA
S2−DEBY
W5−NH
W6−ME
Population
Distance