

# Internship Presentation

MD DIDER HOSSAIN  
Master in Environmental Science  
Matriculation Number: 218203108  
13<sup>th</sup> semester



2025 FEBRUARY

# Internship Details

## Institute

*Rheinland-Pfälzische Technische Universität  
Kaiserslautern-Landau (RPTU)*

## Working Group

Quantitative Landscape Ecology

## Supervisor

Dr. Jürg Spaak

## Location

Fortstraße 7  
76829 Landau, Germany

## Duration

4 months (20<sup>th</sup> September 2024 – 23<sup>rd</sup> January 2025)

## Reimbursement

No

## How did I find?

Contact Person: Dr. Sören Weiß

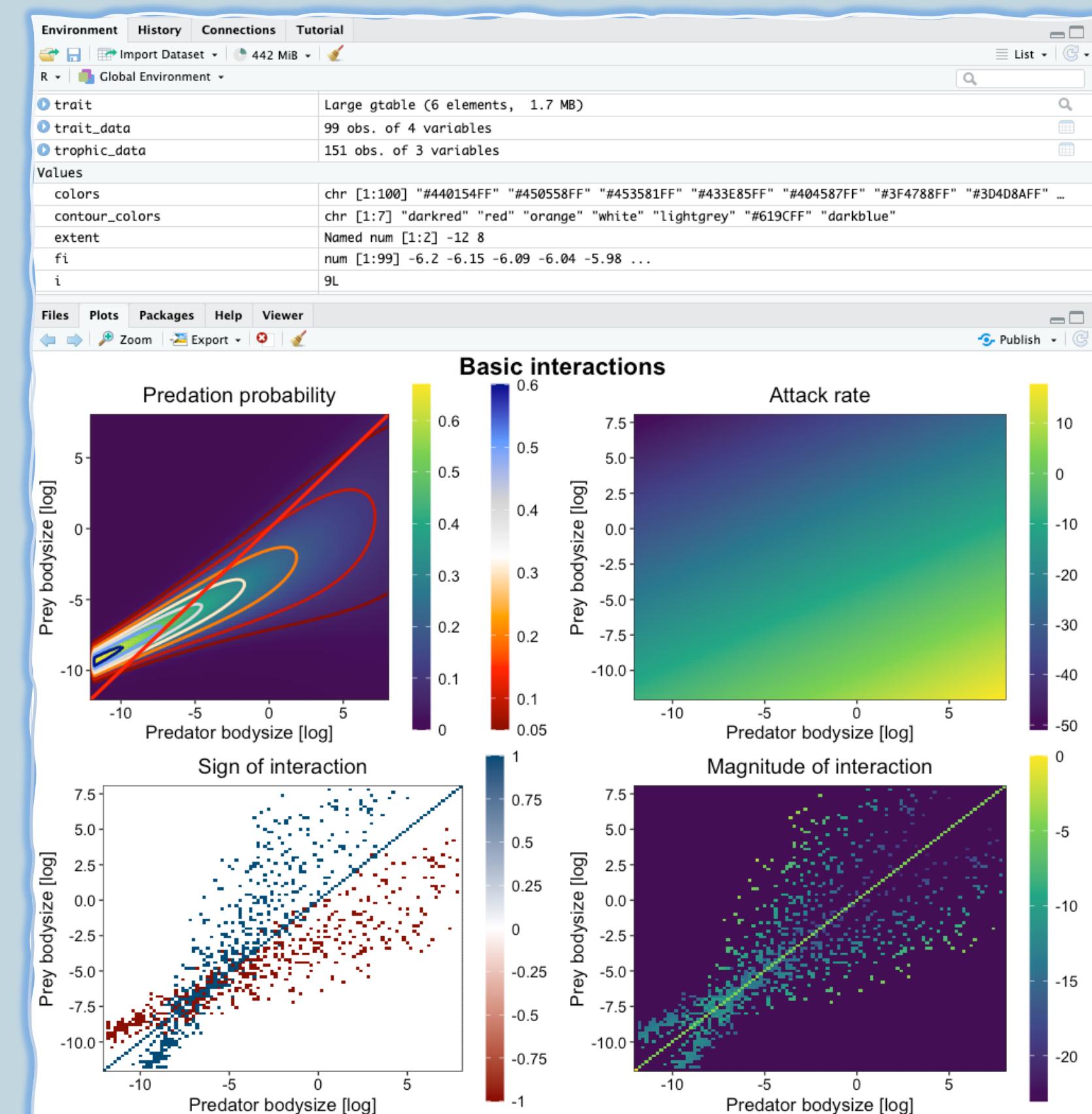
## Internship topics

1. Understanding food web model code in python and translate python code to RStudio code.
2. Machine learning to understand the effect of pesticide contamination.

# Task and Responsibilities

## 1. Food Web Model Project

- Developing Ecological model: Understand the python code to implement and tested Lotka-Volterra and predation model.
- Code Comprehension and Translation: Convert python code into R Studio code, ensuring equivalent functionality.
- Data Processing and Manipulation: Adapted data structures and transformations from python to R equivalents.
- Model Implementation in R.
- Validation of R code output with Python results.

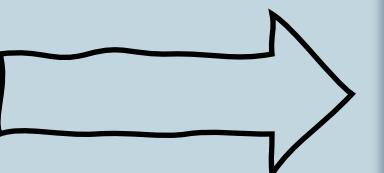


# Task and Responsibilities

## 1.1 Code Comparison

### Python code

```
10 # parameters taken for the Varying size ratio model with temperature for aquatic
11 # species
12 # \beta_x,0 \beta_x,1 \beta_x,temp
13 beta_base = np.array([[3.01503, 0.4600, -0.0738], # mu_i
14 [1.6645, 0.2172, -0.0912], # sigma_i
15 [-1.1012, -0.1533, 0.0309]]) # \theta_i
16
17 # this allows to still access beta_base for other programs
18 beta = beta_base.copy()
19
20 # change from f_i = m_i - mu_i
21 beta[0] = -beta[0]
22 beta[0,1] += 1
23 # change to natural log instead of log10
24 beta[:,1] /= np.log(10)
25
26 #issue is with uiterwaal, small species don't eav fast enough'
27 # what dimensions do we have in mind?
28
29 def generate_species(n_spec, e = 0.45, K = 1e7, c_intra = 0.01, random = True,
30 B0 = 1e-7):
31     species_id = {
32         "m_i": 10**(-np.random.normal(0,2,n_spec)),
33         "e": e, "K": K, "c_intra": c_intra}
34     species_id["m_i"] = np.sort(species_id["m_i"])
35     if not random:
36         species_id["m_i"] = 10**np.linspace(-6,4, n_spec)
37     if B0:
38         species_id["m_i"][0] = B0
39     species_id["random_state"] = np.random.randint(2**30-1)
40
41     return species_id
42
43 traits = ["f_i", "sig_i", "theta_i"]
44 def change_temperature(species_id, T = 20, beta = beta):
45
46     delta_T = T - 20 # Li et al used temperatures relative to 20°C
47     for i, trait in enumerate(traits):
48         species_id[trait] = beta[i, 0] + beta[i,1]*np.log(species_id["m_i"]) + beta[i, 2]*delta_T
49
50     # transformation of the traits
51     # adjust that body mass is given in natural log
52     species_id["f_i"] = np.exp(np.log(10)*species_id["f_i"])
53     # adjust that body mass is given in natural log
54     species_id["sig_i"] = np.log(10)*np.sqrt(np.exp(species_id["sig_i"]))
55     species_id["theta_i"] = np.exp(species_id["theta_i"])
56     species_id["theta_i"] = species_id["theta_i"]/(1+species_id["theta_i"])
57
58     species_id["f_i"][0] = 1e-50
59
60     return species_id
```



### R Studio code

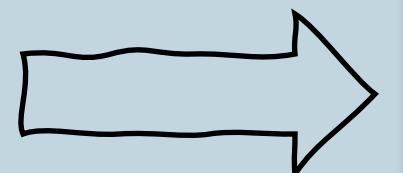
```
18 # parameters taken for the Varying size ratio model with temperature for aquatic
19 # species
20 # \beta_x,0 \beta_x,1 \beta_x,temp
21 beta_base <- matrix(c(3.01503, 0.4600, -0.0738, # mu_i
22 1.6645, 0.2172, -0.0912, # sigma_i
23 -1.1012, -0.1533, 0.0309), # theta_i
24 nrow=3, byrow=TRUE)
25
26
27 beta <- beta_base
28 # change from f_i = m_i - mu_i
29 beta[1,] <- -beta[1,]
30 beta[1, 2] <- beta[1, 2] + 1
31 # change to natural log instead of log10
32 beta[, 2] <- beta[, 2] / log(10)
33
34
35 #issue is with uiterwaal, small species don't eav fast enough'
36 # what dimensions do we have in mind?
37
38 generate_species <- function(n_spec, e = 0.45, K = 1e7, c_intra = 0.01, random = TRUE, B0 = 1e-7) {
39
40     species_id <- list(
41         m_i = 10^(-rnorm(n_spec, mean = 0, sd = 2)),
42         e = e,
43         K = K,
44         c_intra = c_intra
45     )
46
47     species_id$m_i <- sort(species_id$m_i)
48
49     if (!random) {
50         species_id$m_i <- 10^seq(-6, 4, length.out = n_spec)
51     }
52
53     if (!is.null(B0)) {
54         species_id$m_i[1] <- B0
55     }
56
57     species_id$random_state <- sample.int((2^30)-1, 1)
58
59     return(species_id)
60 }
61
62 traits <- c("f_i", "sig_i", "theta_i")
63
64 change_temperature <- function(species_id, T = 20, beta) {
65     delta_T <- T - 20 # Li et al used temperatures relative to 20°C
66
67     for (i in seq_along(traits)) {
68         trait <- traits[i]
69         species_id[[trait]] <- beta[i, 1] + beta[i, 2] * log(species_id$m_i) + beta[i, 3] * delta_T
70     }
71
72     # transformation of the traits
73     # adjust that body mass is given in natural log
74     species_id$f_i <- exp(log(10) * species_id$f_i)
75     # adjust that body mass is given in natural log
76     species_id$sig_i <- log(10) * sqrt(exp(species_id$sig_i))
77     species_id$theta_i <- exp(species_id$theta_i)
78     species_id$theta_i <- species_id$theta_i / (1 + species_id$theta_i)
79     species_id$f_i[1] <- 1e-50
80
81     return(species_id)
82 }
```

# Task and Responsibilities

## 1.1 Code Comparison

### Python code

```
62 # for plotting and testing
63 n_spec = 101
64 species_id_base = generate_species(n_spec)
65 species_id_base["m_i"] = 10**np.linspace(-12, 8, n_spec)
66 species_id_base = change_temperature(species_id_base)
67
68 def compute_predation_prob(species_id):
69     log_mi = np.log(species_id["m_i"])
70     log_fi = np.log(species_id["f_i"])
71
72     # preference matrix
73     s_ji = (species_id["theta_i"]
74             *np.exp(-(log_mi[:,np.newaxis] - log_fi)**2/(2*species_id["sig_i"]**2)))
75     return s_ji
76
77 def compute_links(species_id):
78     rng_interact = np.random.RandomState(species_id["random_state"])
79
80     s_ji = compute_predation_prob(species_id)
81     threshold = rng_interact.uniform(0.05, 1, s_ji.shape)
82
83     return 1.0*(s_ji>threshold)
84
85 if __name__ == "__main__":
86     s_ji = compute_predation_prob(species_id_base)
87     extent = np.percentile(np.log10(species_id_base["m_i"]), [0, 100, 0, 100])
88     fig, ax = plt.subplots(2,2, figsize = (9,9))
89     fig.suptitle("Basic interactions")
90     cmap = ax[0,0].imshow(s_ji, extent = extent, origin = "lower")
91     fig.colorbar(cmap, ax = ax[0,0])
92     cmap = ax[0,0].contour(s_ji, extent = extent, origin = "lower",
93                           cmap = "RdBu", levels = np.append([0.05],np.arange(0.1, 0.7, 0.1)))
94     ax[0,0].plot(extent, extent, 'r')
95     fig.colorbar(cmap, ax = ax[0,0])
96     ax[0,0].set_xlabel("Predator bodysize [log]")
97     ax[0,0].set_ylabel("Prey bodysize [log]")
98     ax[0,0].set_title("Predation probability")
99 #####
100 # empirical data from uiterwaal et al.
101
102 # 3D predation space
103 forage_attack = {
104     "intercept": -15.92, # intercept of regression
105     "temp_2": -0.01, # quadratic effect of temperature
106     "temp": 0.32, # linear effect of temperature
107     "log_pred": 0.54, # effect of predator mass
108     "log_prey": 0.05, # effect of prey mass
109     "T_ref": 26.7} # reference temperature for fit
110
111 def compute_attack_rate(species_id, T = 20):
112     # basic attack rate
113     log_mi = np.log(species_id["m_i"])
114
115     interm = {
116         "temp_2": (T-forage_attack["T_ref"])*2, "temp": T-forage_attack["T_ref"],
117         "log_pred": log_mi, "log_prey": log_mi[:,np.newaxis]}
118
119     attack_rate = forage_attack["intercept"]
120     for key in interm.keys():
121         attack_rate = attack_rate + interm[key]*forage_attack[key]
122
123     # convert from log space to normal space
124     attack_rate = np.exp(attack_rate)
125     # convert from attack rate/individuum to attack rate per kg
126     attack_rate /= species_id["m_i"][:,np.newaxis]
127
128     return attack_rate
129
```



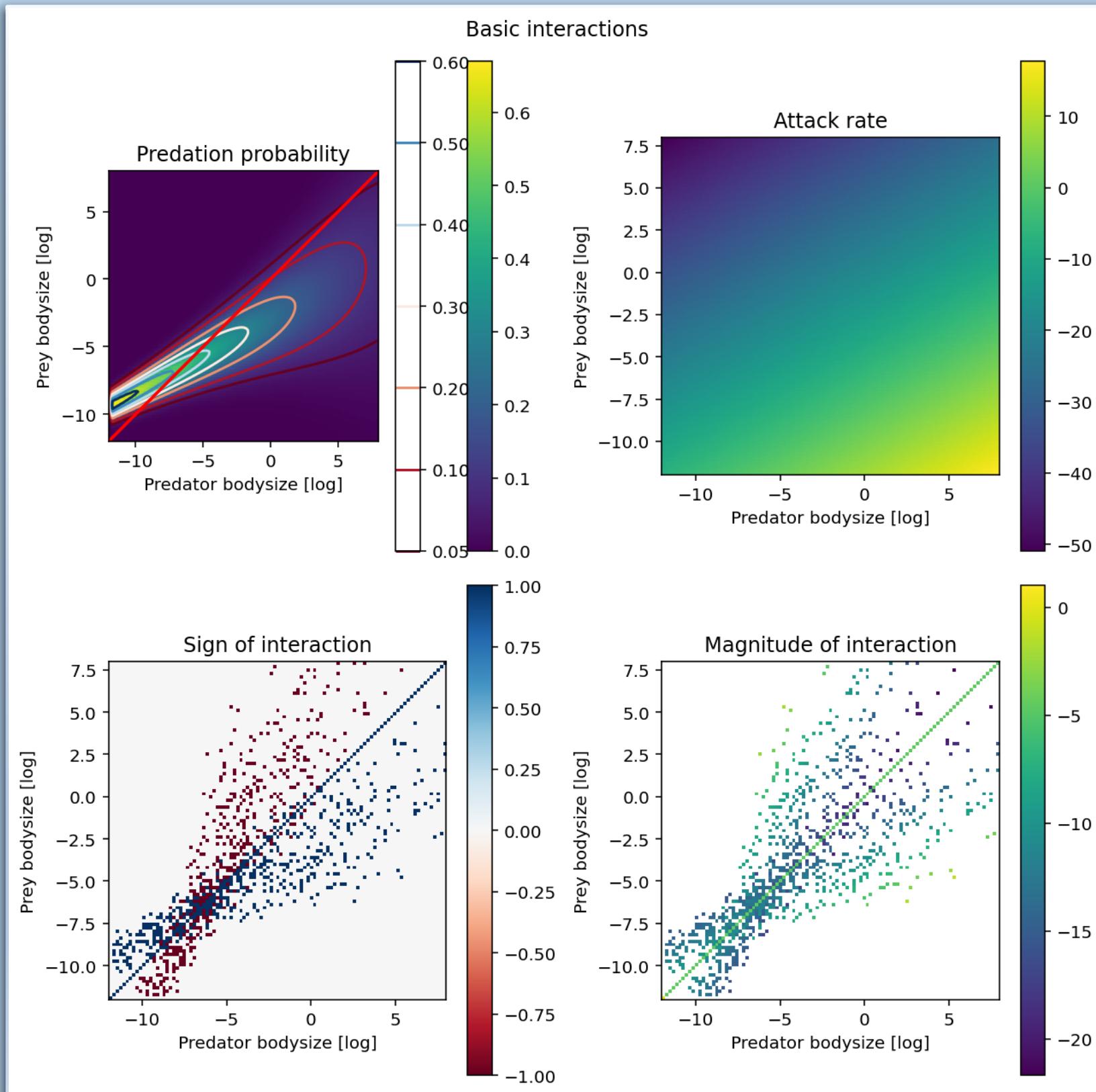
### R Studio code

```
85 # For plotting and testing
86 n_spec <- 101
87 species_id_base <- generate_species(n_spec)
88 species_id_base$m_i <- 10^seq(-12, 8, length.out = n_spec)
89 species_id_base <- change_temperature(species_id_base, beta = beta)
90
91 - compute_predation_prob <- function(species_id) {
92     log_mi <- log(species_id$m_i)
93     log_fi <- log(species_id$f_i)
94     log_mi_matrix <- matrix(log_mi, nrow = length(log_mi), ncol = length(log_mi), byrow = TRUE)
95
96     # Preference matrix
97     s_ji <- (species_id$theta_i
98             * exp(-(log_mi_matrix - log_fi)^2/(2 * species_id$sig_i^2)))
99
100    return(s_ji)
101 }
102
103 - compute_links <- function(species_id) {
104     set.seed(species_id$random_state)
105     s_ji <- compute_predation_prob(species_id)
106     threshold <- matrix(runif(length(s_ji), 0.05, 1), nrow = nrow(s_ji), ncol = ncol(s_ji))
107     return(1.0 * (s_ji > threshold))
108 }
109
110 - if (interactive()) {
111     s_ji <- compute_predation_prob(species_id_base)
112     extent <- quantile(log10(species_id_base$m_i), probs = c(0, 1))
113     df1 <- expand.grid(
114         Predator = seq(extent[1], extent[2], length.out = nrow(s_ji)),
115         Prey = seq(extent[1], extent[2], length.out = ncol(s_ji)))
116     )
117     df1$PredationProb <- as.vector(s_ji)
118 }
119
120 ## Predation probability
121 contour_colors <- c("darkred", "red", "orange", "white", "#619cff", "darkblue")
122
123 b11 <- ggplot(df1, aes(x = Predator, y = Prey, fill = PredationProb)) +
124     geom_tile() +
125     scale_fill_viridis_c(
126         breaks = seq(0.0, 0.6, by = 0.1),
127         labels = paste(seq(0.0, 0.6, by = 0.1)),
128         guide = guide_colorbar(barwidth = 0.9, barheight = 17)
129     ) +
130     scale_x_continuous(expand = c(0, 0)) +
131     scale_y_continuous(expand = c(0, 0)) +
132     geom_contour(aes(z = PredationProb, color = after_stat(level)),
133                 breaks = c(0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60),
134                 linewidth = 1) +
135     scale_color_gradient(
136         colors = contour_colors,
137         breaks = c(0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60),
138         labels = paste(c(0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60)),
139         guide = guide_colorbar(barwidth = 0.9, barheight = 17)) +
140     geom_abline(intercept = 0, slope = 1, color = "red", linewidth = 1.2) +
141     labs(
142         title = "Predation probability",
143         x = "Predator bodysize [log]",
144         y = "Prey bodysize [log]"
145     ) +
146     theme_bw() +
147     theme(
148         plot.title = element_text(hjust = 0.5, size = 15, color = "black"),
149         legend.title = element_blank(),
150         axis.text.x = element_text(size = 11, color = "black"),
151         axis.text.y = element_text(size = 11, color = "black"),
152         legend.text = element_text(size = 11, color = "black"),
153         axis.title.x = element_text(size = 13, color = "black"),
154         axis.title.y = element_text(size = 13, color = "black"),
155         legend.direction = "vertical",
156     )
```

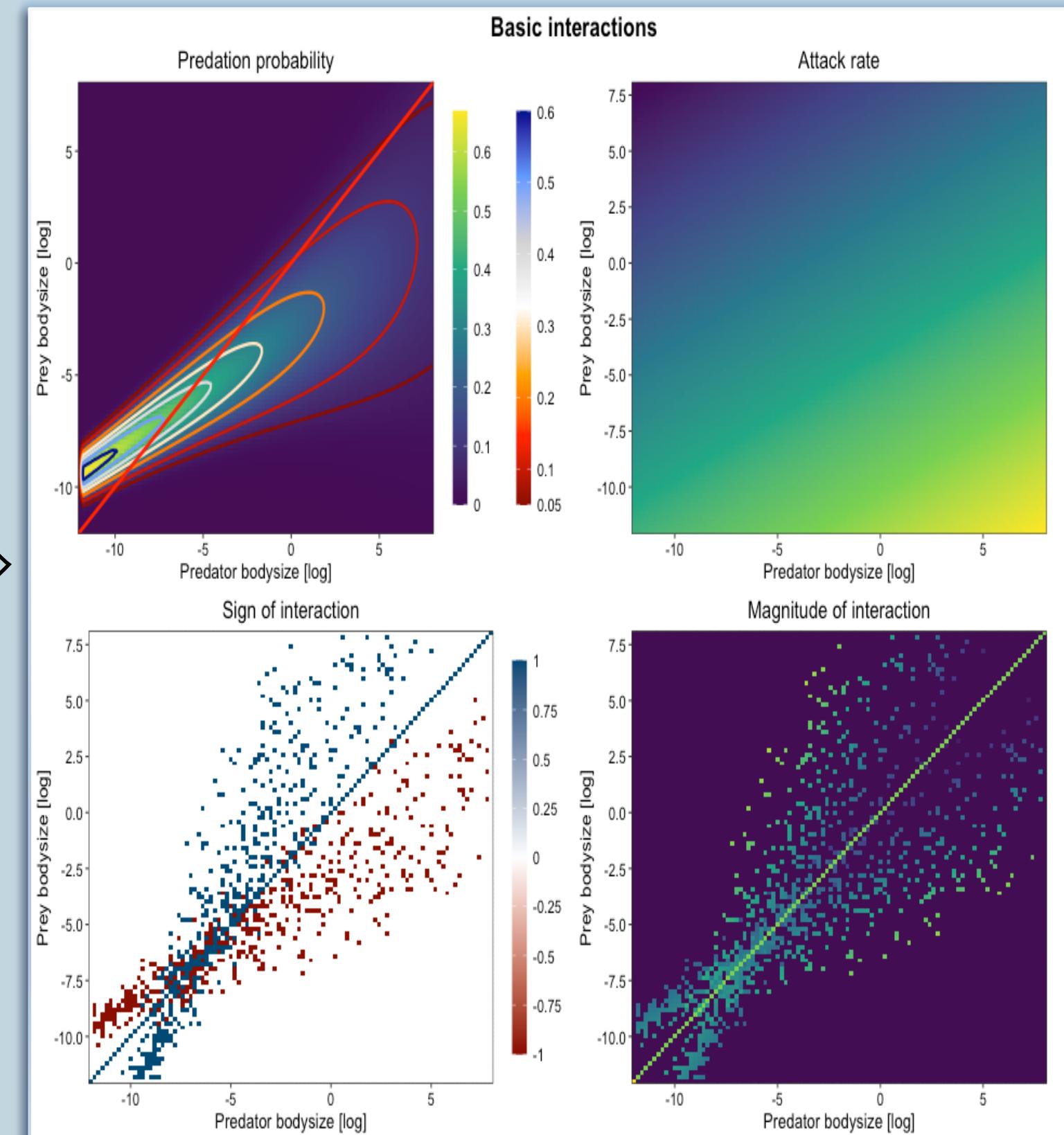
# Task and Responsibilities

## 1.1 Visualization Comparison

Graph created by Python



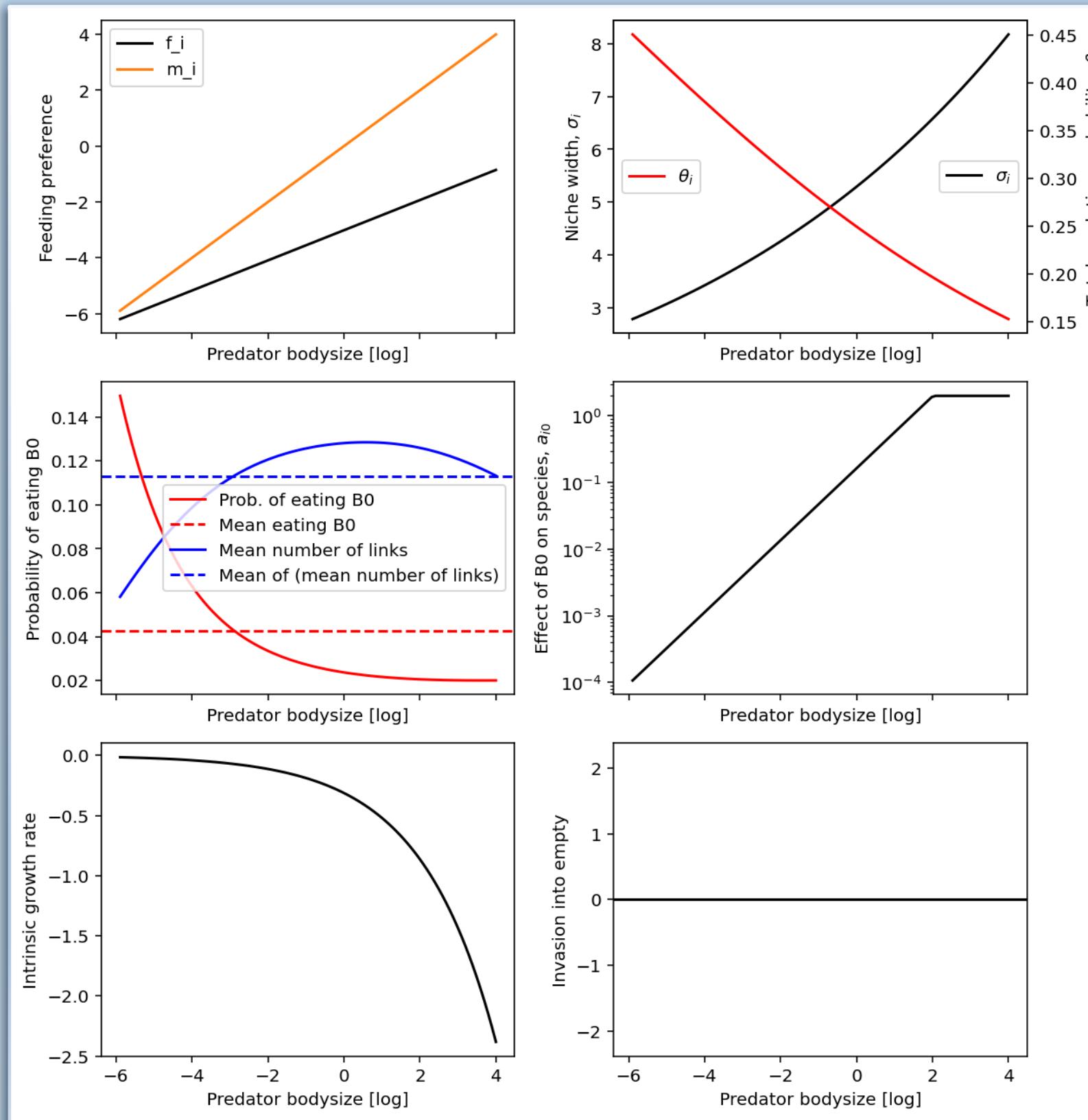
Graph created by R Studio



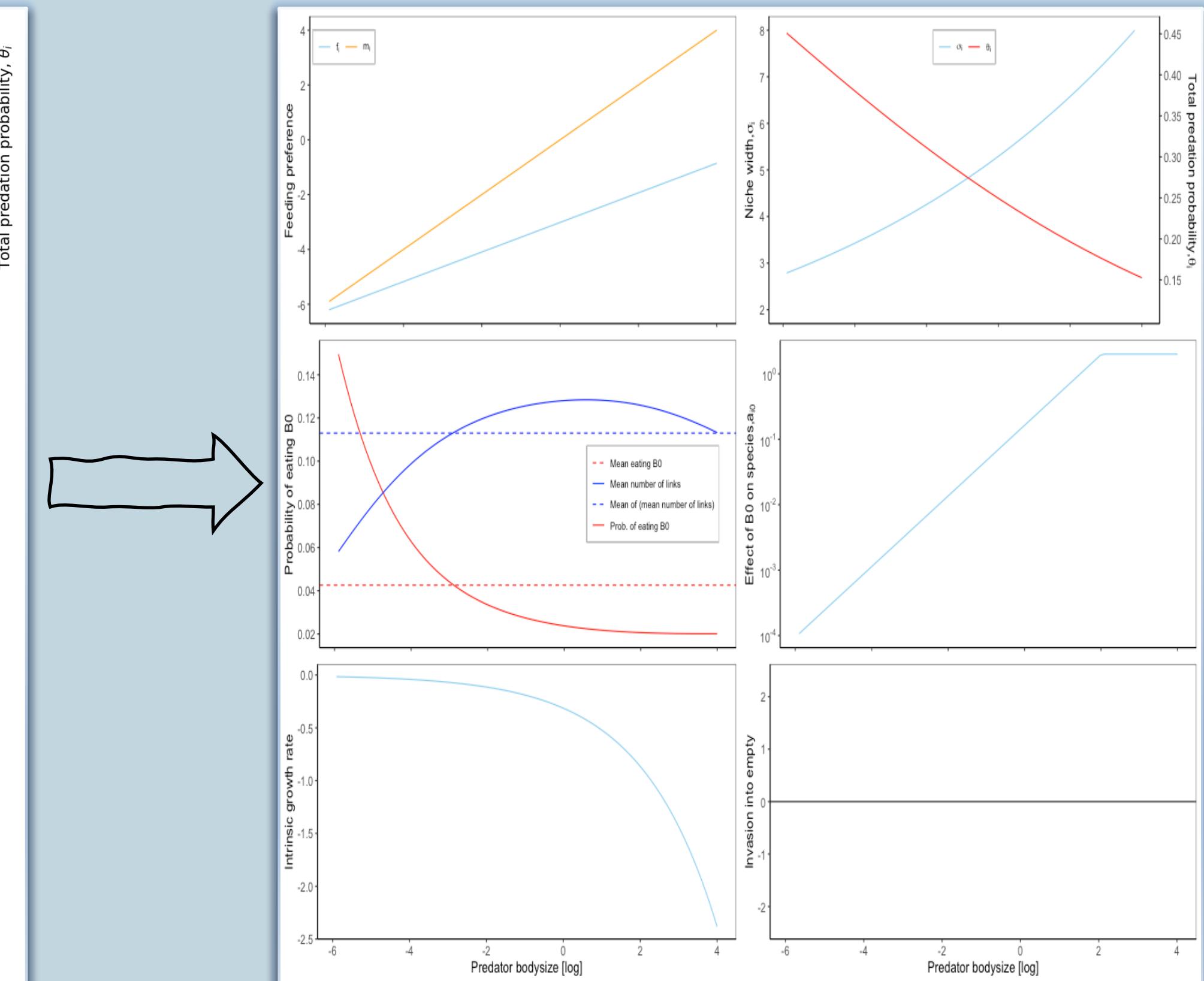
# Task and Responsibilities

## 1.1 Visualization Comparison

Graph created by Python



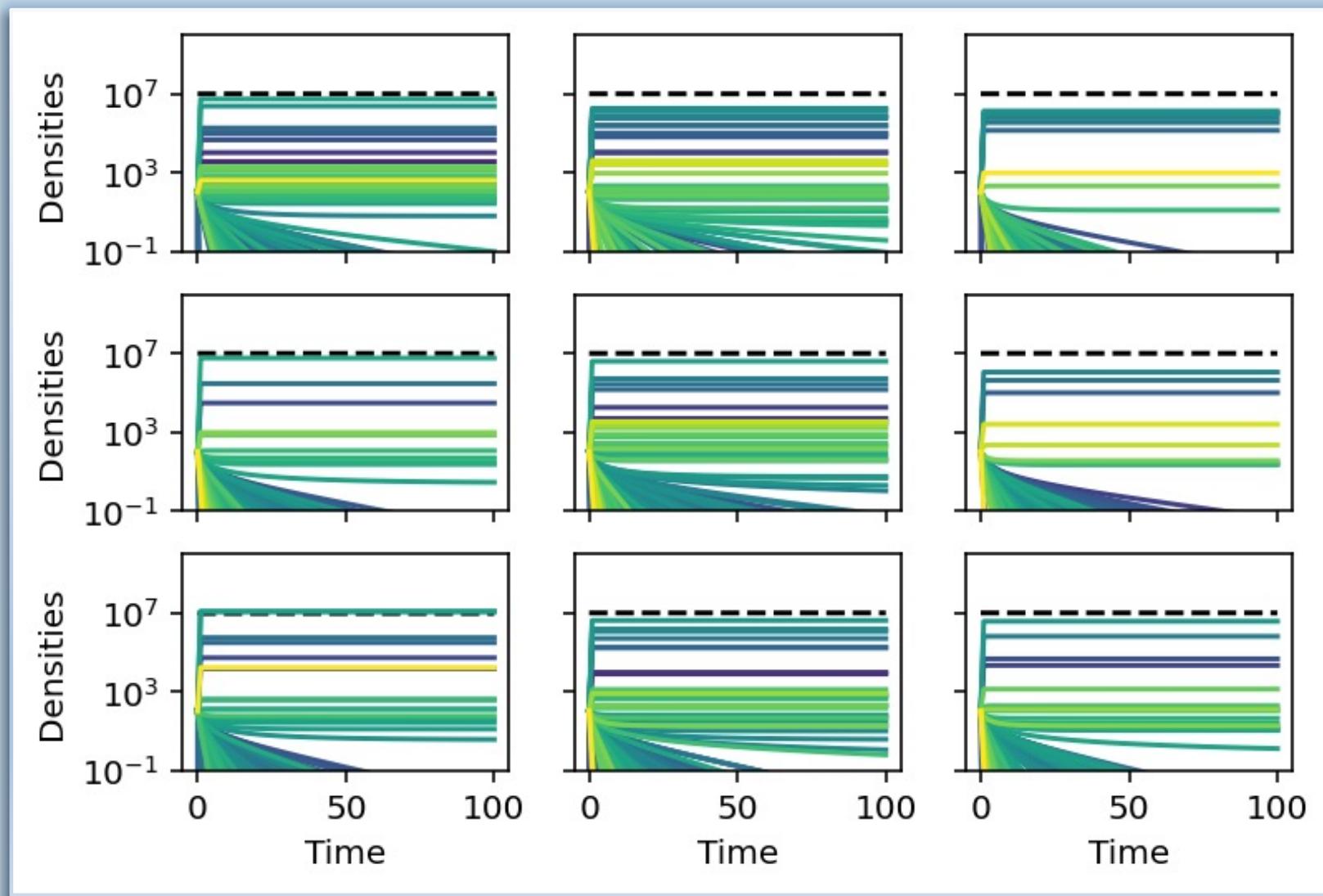
Graph created by R Studio



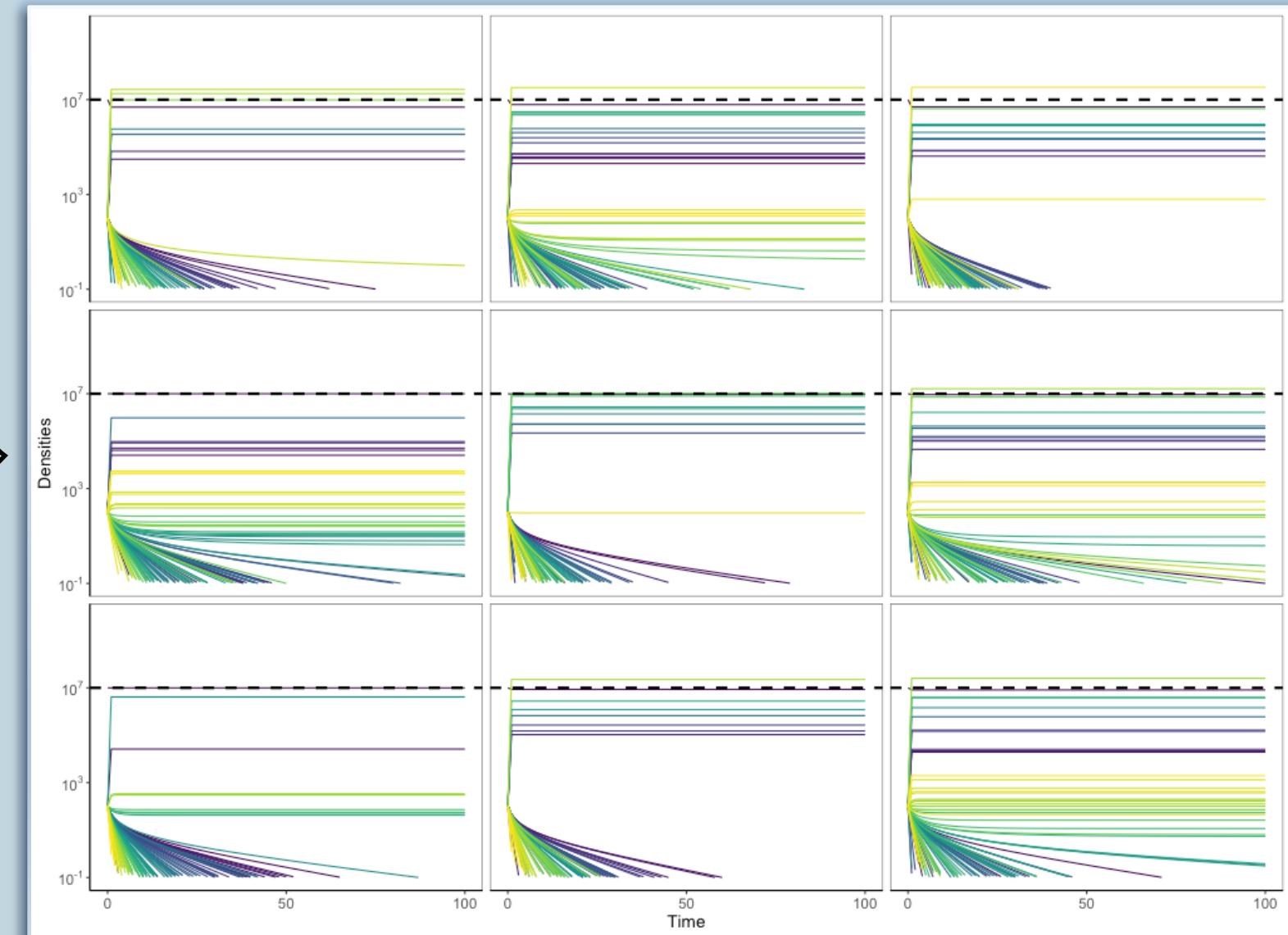
# Task and Responsibilities

## 1.1 Visualization Comparison

**Graph created by Python**



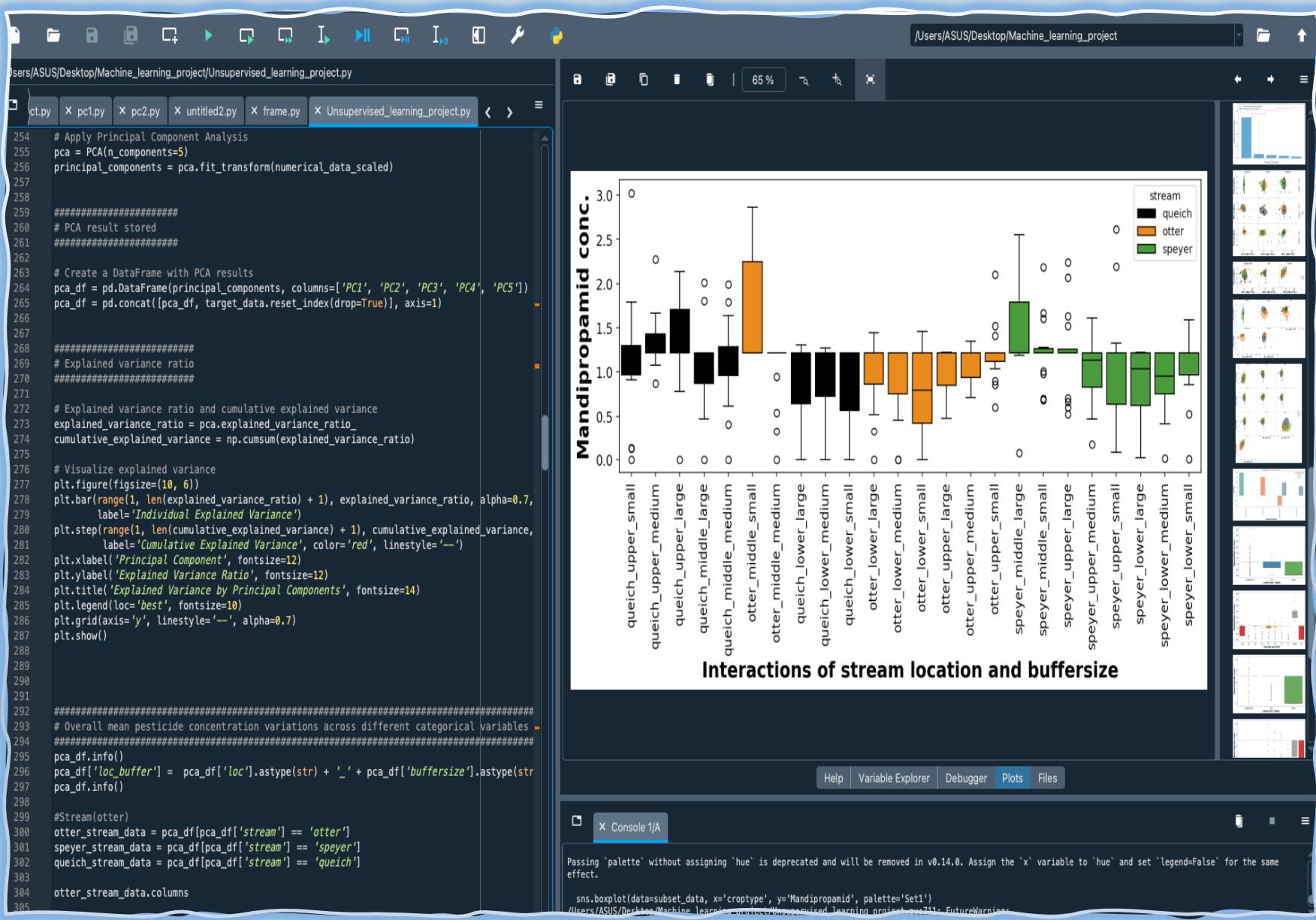
**Graph created by R Studio**



# Task and Responsibilities

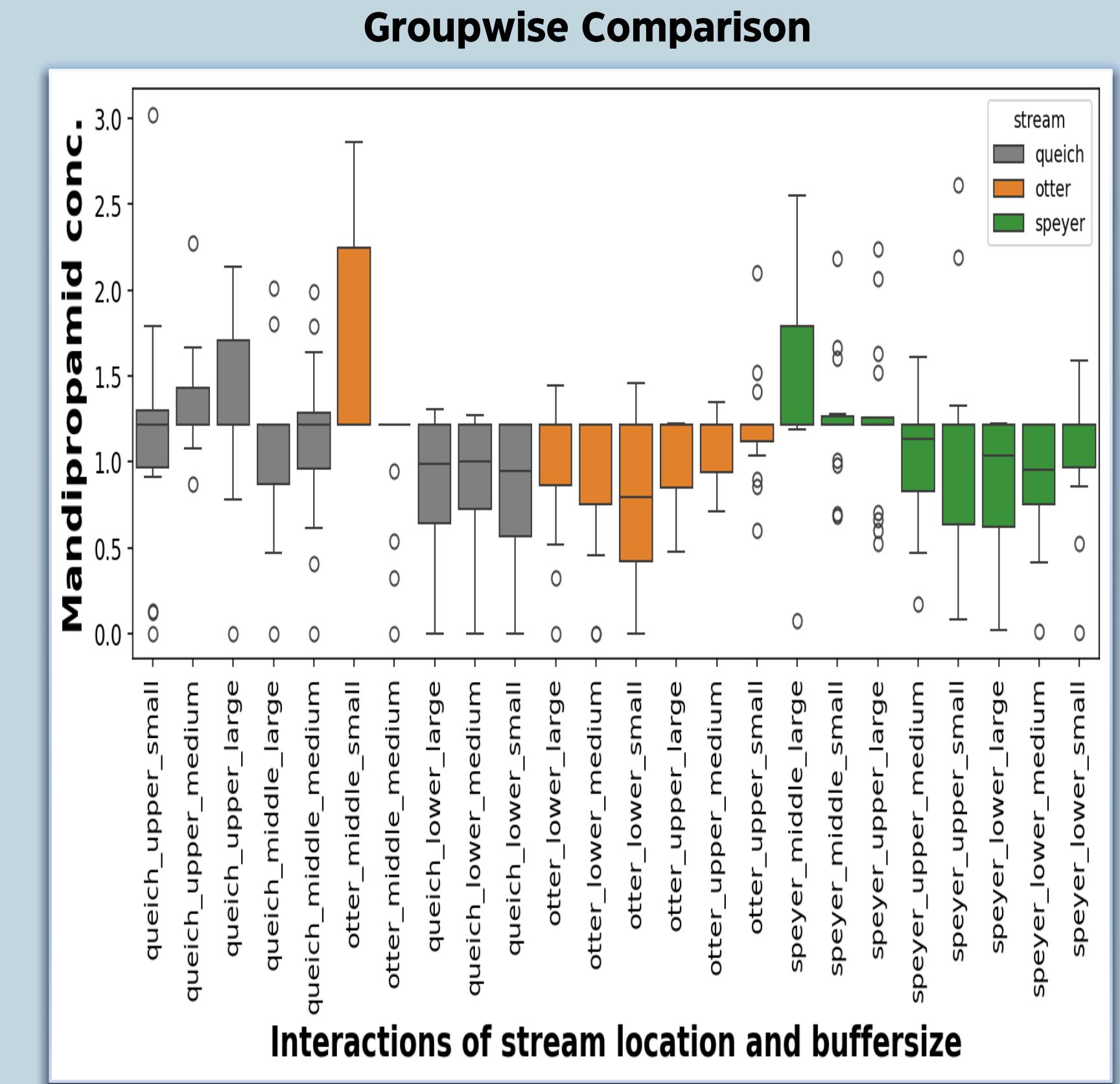
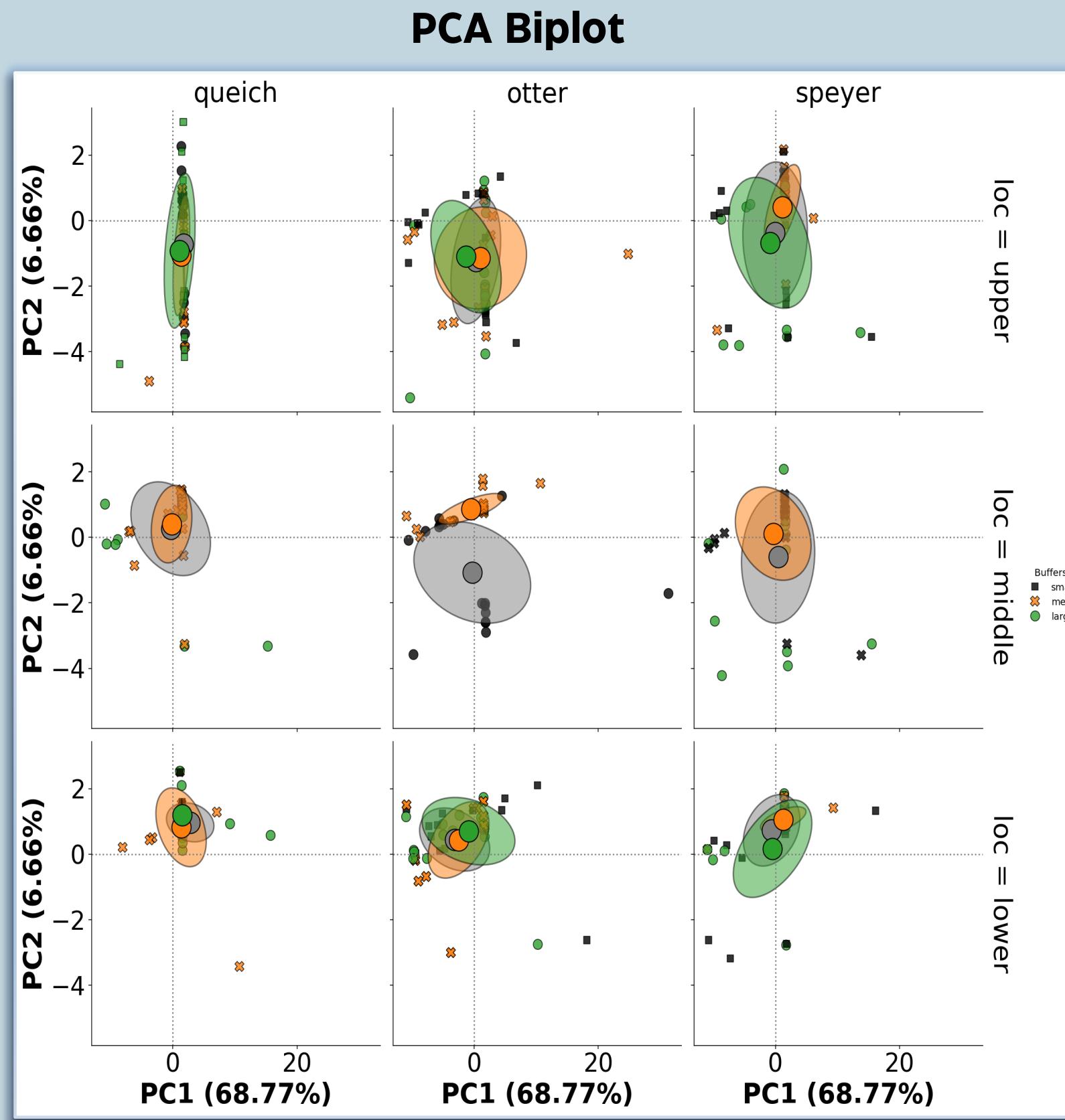
## 2. Analysis of Pesticide Contamination

- Data Preprocessing and Cleaning: Handled missing values and transformed datasets for analysis
- Principal Component Analysis (PCA): Reduced dimensionality and identify key chemical components.
- ANOVA Tests: Assessed the significance of categorical variables on pesticide concentrations.
- Post-hoc Analyses: Identify group differences.



# Task and Responsibilities

## 2.1 Pesticide Contamination across Stream Locations



# Skills Development Through Internships

## Technical Skills

Ecological modeling,  
data analysis & machine  
learning in Python and  
R Studio

## Soft Skills

Problem-solving,  
communication,  
collaboration, and  
adaptability

## Professional Practices

Project Management,  
research & analysis, and  
professional conduct

- Suitability for other international students
- Can I imagine working in the same field in the future?

# The End

MD DIDER HOSSAIN

2025 FEBRUARY