

# Example of Analyses with **funrar**

*Matthias GreniÃ©*

*11 janvier, 2017*

With **funrar** you can compute **F**unctional **R**arity indices. We are using a North American Mammals datasets to show the kind of questions you can come up with.

For example, you can try to map the hotspots of Regional Functional Uniqueness, i.e. the communities with highest number of functionally unique species (relatively to the whole species pool).

## Hotspots

### Functional Uniqueness

```
library(dplyr)

# Compute Functional Uniqueness per site
site_rich = rowSums(subset_pres[, -c(1:3)])
names(site_rich) = subset_pres$GlobalID

site_coords = subset_pres[, 1:3]

make_index_df = function(mammal_funrar, index = "Ui") {
  species_index = mammal_funrar[[index]][[index]]

  names(species_index) = mammal_funrar[[index]]$species

  index_name = paste0("site_", index)

  site_index = data.frame(
    given_index = (as.matrix(subset_pres[, -c(1:3)]) %*%
                    species_index) / site_rich,
    GlobalID = site_coords$GlobalID)

  colnames(site_index)[1] = index_name

  site_index = site_index %>%
    filter_(paste0("!is.na(", site_index, ")")) %>%
    inner_join(site_coords, by = "GlobalID")

  return(site_index)
}

site_ui = make_index_df(mammal_funrar, "Ui")

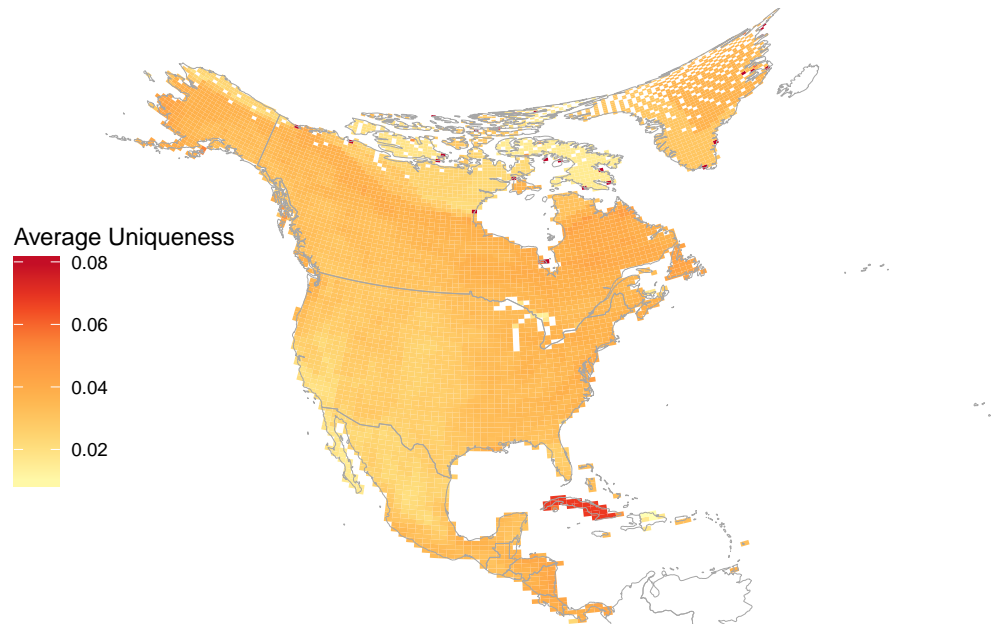
ui_map = ggplot(site_ui, aes(x = Longitude, y = Latitude, z = site_Ui)) +
  stat_summary_2d(bins = 150, color = NA) +
  coord_map("albers", at0 = 40, lat1 = 20, xlim = range(site_ui$Longitude),
            ylim = range(site_ui$Latitude)) +
  ggthemes::theme_map() +
```

```

theme(legend.position = c(0.1, 0.2)) +
labs(fill = "Average Uniqueness") +
scale_fill_distiller(palette = "YlOrRd", direction = 1) +
borders(xlim = c(-170, -23), ylim = c(9.5,83), size = 0.01,
        colour = "gray65")

```

ui\_map



## Geographical Restrictedness

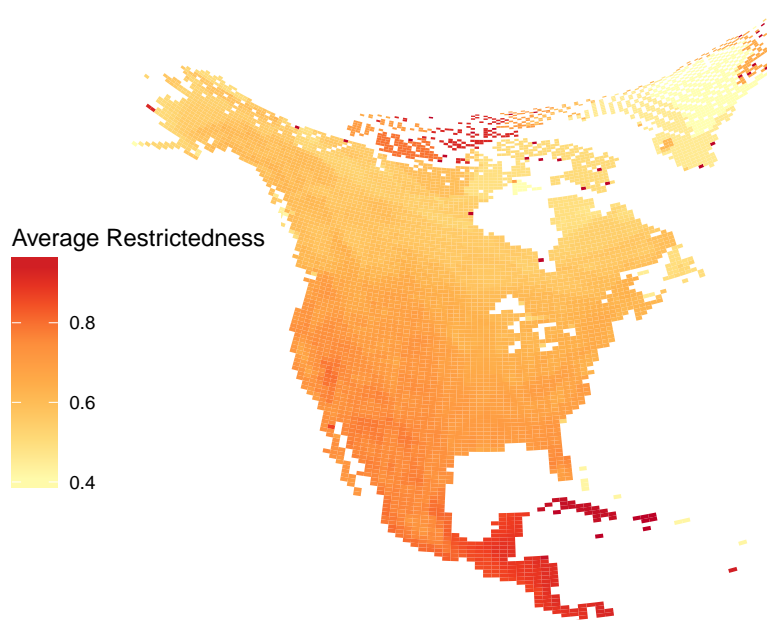
```

site_ri = make_index_df(mammal_funrar, "Ri")

ri_map = ggplot(site_ri, aes(x = Longitude, y = Latitude, z = site_Ri)) +
  stat_summary_2d(bins = 150, color = NA) +
  coord_map("albers", at0 = 40, lat1 = 20, xlim = range(site_ri$Longitude),
            ylim = range(site_ri$Latitude)) +
  ggthemes::theme_map() +
  theme(legend.position = c(0.1, 0.2)) +
  labs(fill = "Average Restrictedness") +
  scale_fill_distiller(palette = "YlOrRd", direction = 1)

```

ri\_map



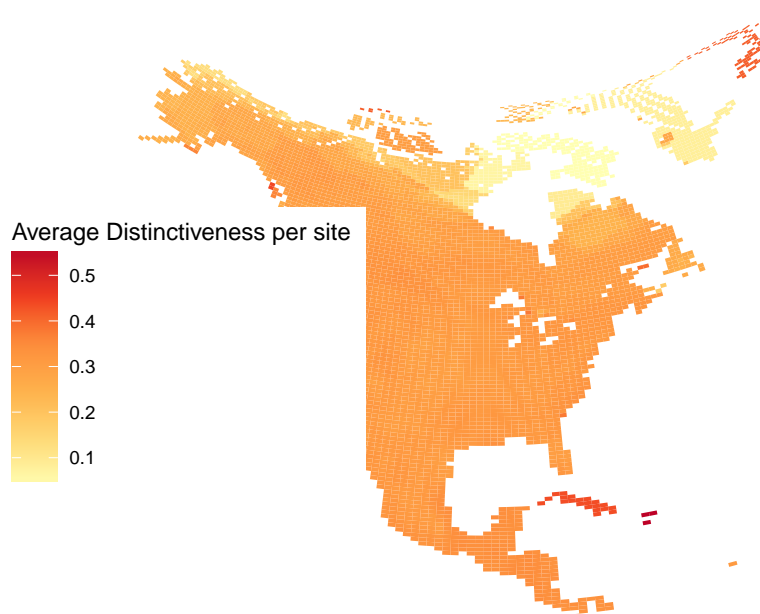
## Functional Distinctiveness

```
# Average Functional Distinctiveness
site_di = rowMeans(as.matrix(mammal_funrar$Di), na.rm = T) %>%
  as_data_frame() %>%
  mutate(GlobalID = rownames(.)) %>%
  rename(avg_Di = value) %>%
  inner_join(site_ri %>%
    dplyr::select(GlobalID, Longitude, Latitude) %>%
    mutate(GlobalID = as.character(GlobalID)),
    by = "GlobalID")

di_map = ggplot(site_di, aes(x = Longitude, y = Latitude, z = avg_Di)) +
  stat_summary_2d(bins = 150, color = NA) +
  coord_map("albers", at0 = 40, lat1 = 20, xlim = range(site_di$Longitude),
    ylim = range(site_di$Latitude)) +
  ggthemes::theme_map() +
  theme(legend.position = c(0.1, 0.2)) +
  labs(fill = "Average Distinctiveness per site") +
  scale_fill_distiller(palette = "YlOrRd", direction = 1)

di_map
```

```
## Warning: Removed 552 rows containing non-finite values (stat_summary2d).
```



## Relation between IUCN status and Functional Rarity Indices with Geographical Restrictedness

```
species_status = mammal_funrar$Ui %>%
  # Add IUCN status
  inner_join(subset_iucn_status %>%
    dplyr::select(TaxonName, IUCN_status) %>%
    mutate(species = gsub(" ", "_", TaxonName)) %>%
    dplyr::select(-TaxonName),
    by = "species") %>%
  # Add Geographical Restrictedness
  inner_join(mammal_funrar$Ri, by = "species") %>%
  mutate(IUCN_status = ifelse(is.na(IUCN_status), "NO",
    as.character(IUCN_status))) %>%
  mutate(std_Ui = (Ui - min(Ui))/(max(Ui) - min(Ui)))
```

```
## Warning in inner_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factor and character vector, coercing into character vector
```

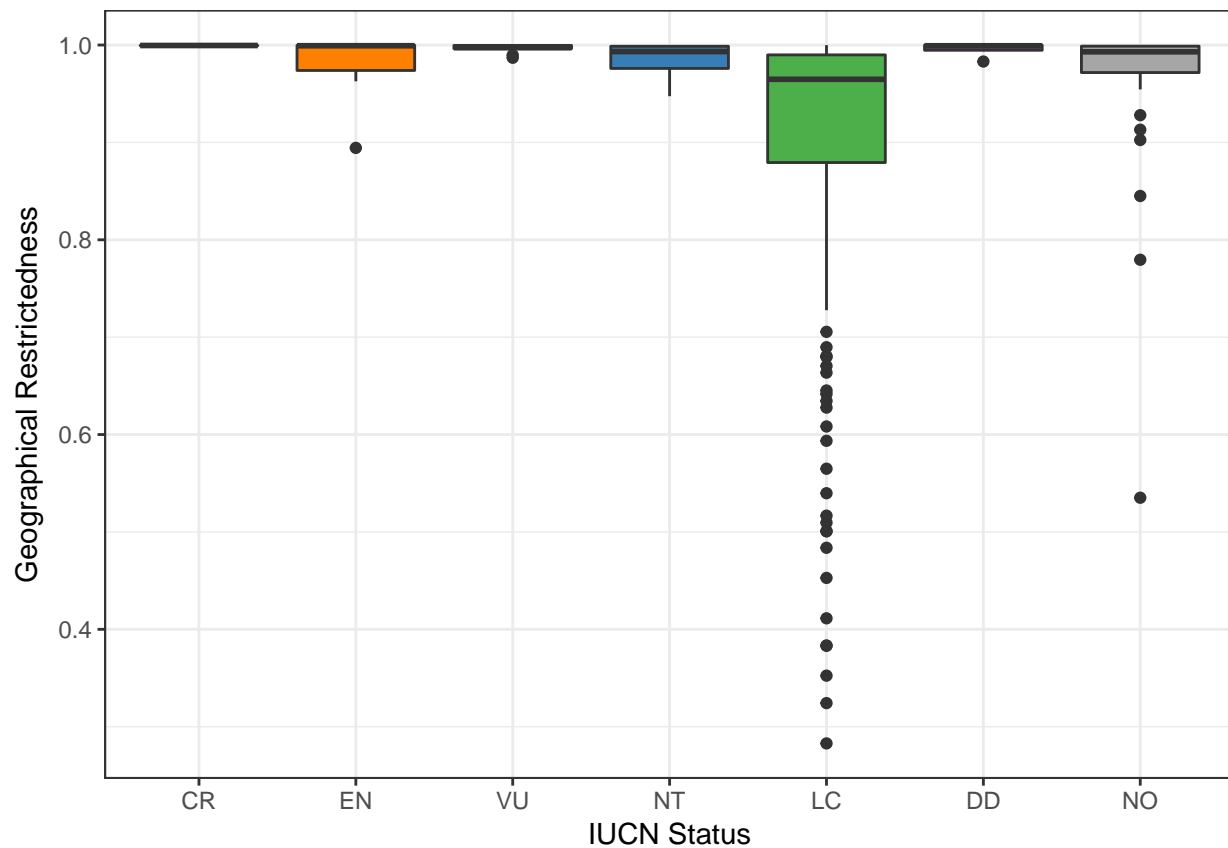
```
## Warning in inner_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## character vector and factor, coercing into character vector
```

```
species_status$IUCN_status = factor(species_status$IUCN_status,
  levels = names(status_values))
```

```
summary(species_status)
```

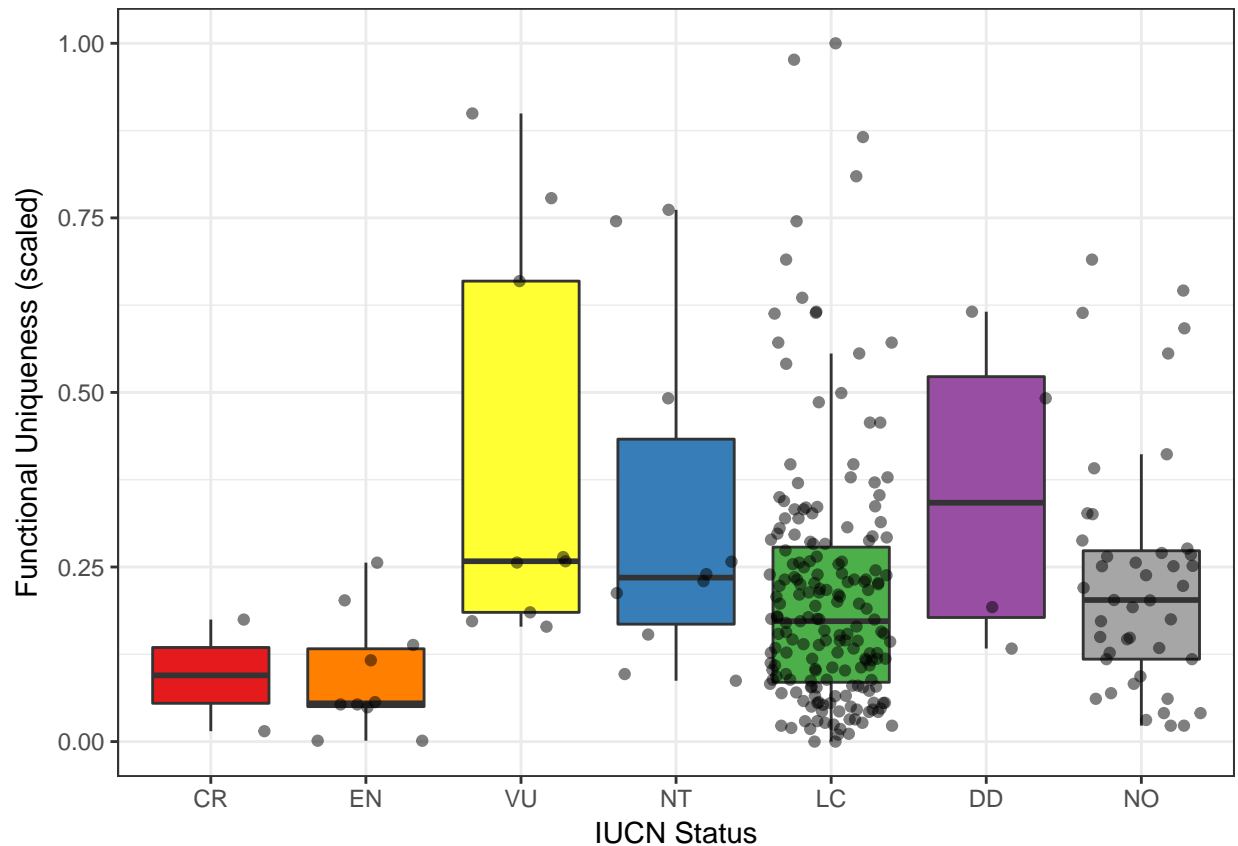
```
##      species              Ui      IUCN_status      Ri
## Length:265      Min.   :0.0006121  CR: 2      Min.   :0.2828
## Class :character 1st Qu.:0.0115721  EN: 10     1st Qu.:0.9242
## Mode  :character Median :0.0223622  VU: 9      Median :0.9787
##                      Mean  :0.0281722  NT: 10     Mean  :0.9174
##                      3rd Qu.:0.0344888  LC:187     3rd Qu.:0.9966
##                      Max.   :0.1243665  DD: 4      Max.   :0.9999
##                      NO: 43
##
##      std_Ui
## Min.   :0.00000
## 1st Qu.:0.08856
## Median :0.17575
## Mean   :0.22270
## 3rd Qu.:0.27374
## Max.   :1.00000
##
```

```
ggplot(species_status, aes(x = IUCN_status, y = Ri)) +
  geom_boxplot(aes(fill = IUCN_status)) +
  scale_fill_manual(values = status_values) +
  labs(x = "IUCN Status",
       y = "Geographical Restrictedness") +
  theme(legend.position = "none")
```



with Functional Uniqueness

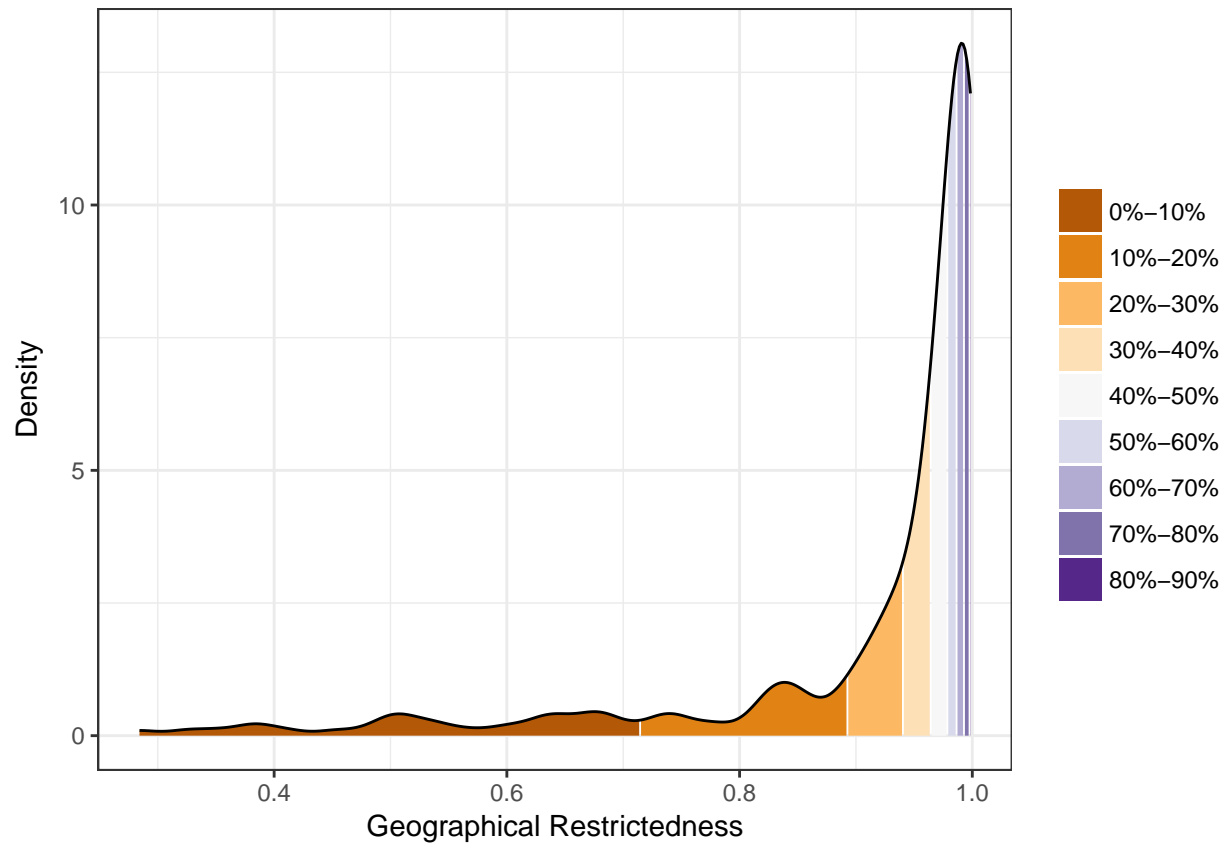
```
ggplot(species_status, aes(x = IUCN_status, y = std_Ui)) +  
  geom_boxplot(aes(fill = IUCN_status), outlier.colour = NA) +  
  geom_jitter(alpha = 0.5) +  
  scale_fill_manual(values = status_values) +  
  labs(x = "IUCN Status",  
       y = "Functional Uniqueness (scaled)") +  
  theme(legend.position = "none")
```



Total distribution of indices

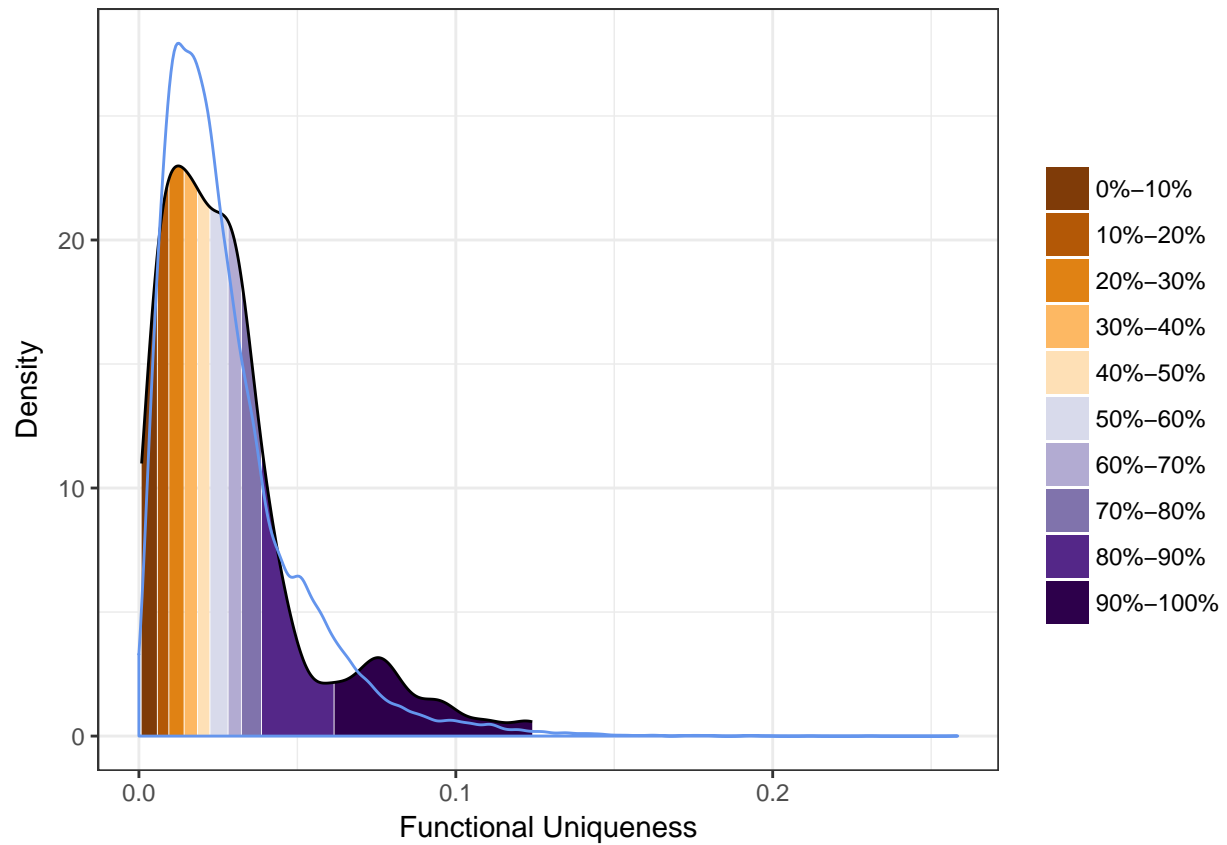
Geographical Restrictedness

```
divr::make_quantile_density(species_status$Ri) +  
  labs(x = "Geographical Restrictedness",  
       y = "Density",  
       fill = NULL)
```



### Functional Uniqueness

```
divr::make_quantile_density(species_status$Ui) +
  labs(x = "Functional Uniqueness",
       y = "Density",
       fill = NULL) +
  geom_density(data = null_flat_ui, aes(x = Ui, y = ..density..),
              color = "cornflowerblue", size = 0.5, show.legend = FALSE)
```

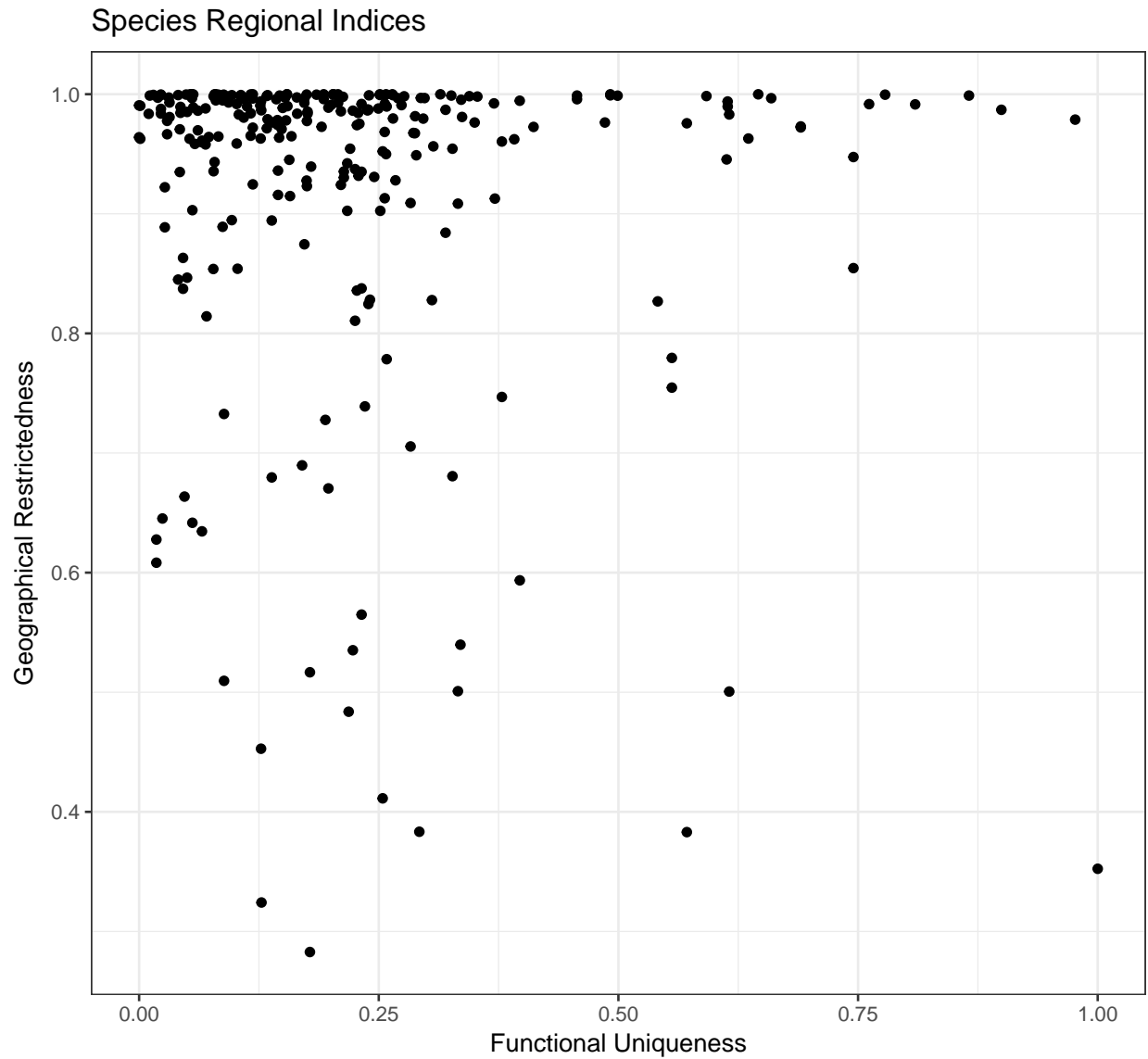


## Biplot

```
biplot_ui_ri = ggplot(species_status, aes(x = std_Ui, y = Ri)) +
  geom_point() +
  labs(x = "Functional Uniqueness",
       y = "Geographical Restrictedness",
       title = "Species Regional Indices")
```

```
biplot_ui_ri
```

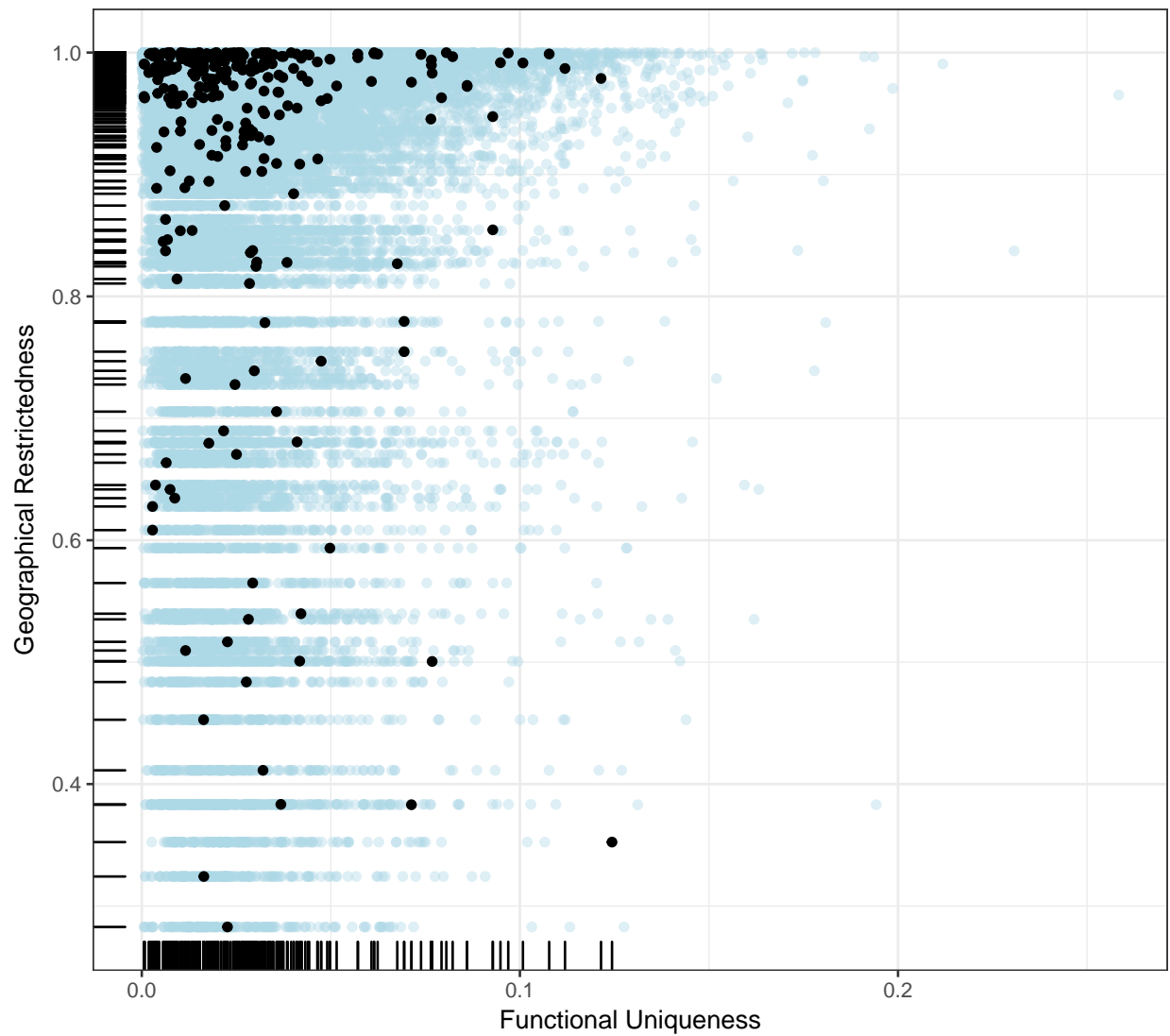




```
biplot_ui_ri_null = ggplot(species_status, aes(x = Ui, y = Ri)) +
  geom_point(data = null_flat_ui, colour = "lightblue", alpha = 0.4) +
  geom_point() +
  geom_rug() +
  labs(x = "Functional Uniqueness",
       y = "Geographical Restrictedness",
       title = "Species Regional Indices with null model (shuffled traits)")

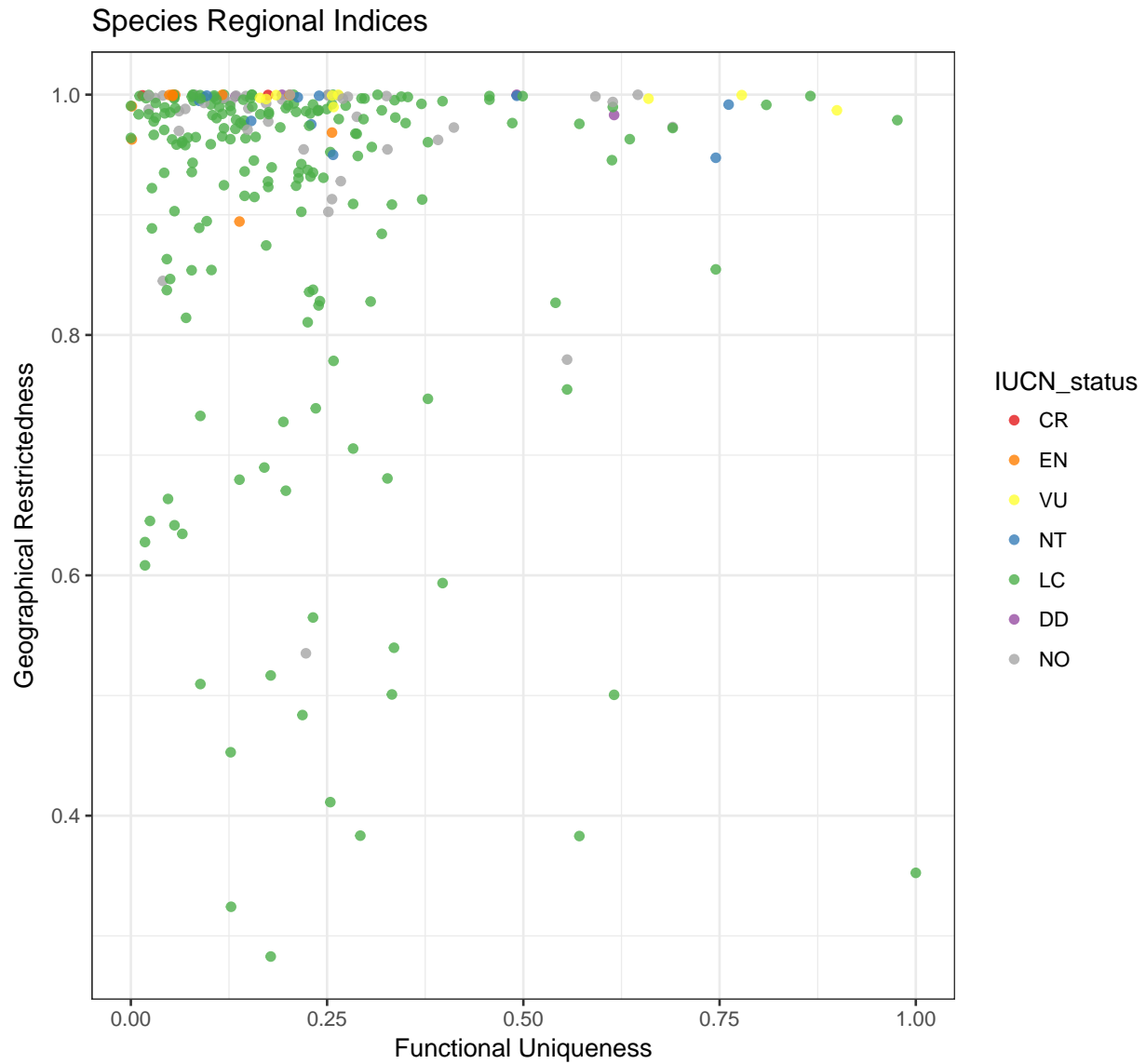
biplot_ui_ri_null
```

Species Regional Indices with null model (shuffled traits)



With IUCN status

```
ggplot(species_status, aes(x = std_Ui, y = Ri, color = IUCN_status)) +
  geom_point(size = 1.5, alpha = 0.8) +
  scale_color_manual(values = status_values) +
  labs(x = "Functional Uniqueness",
       y = "Geographical Restrictedness",
       title = "Species Regional Indices")
```



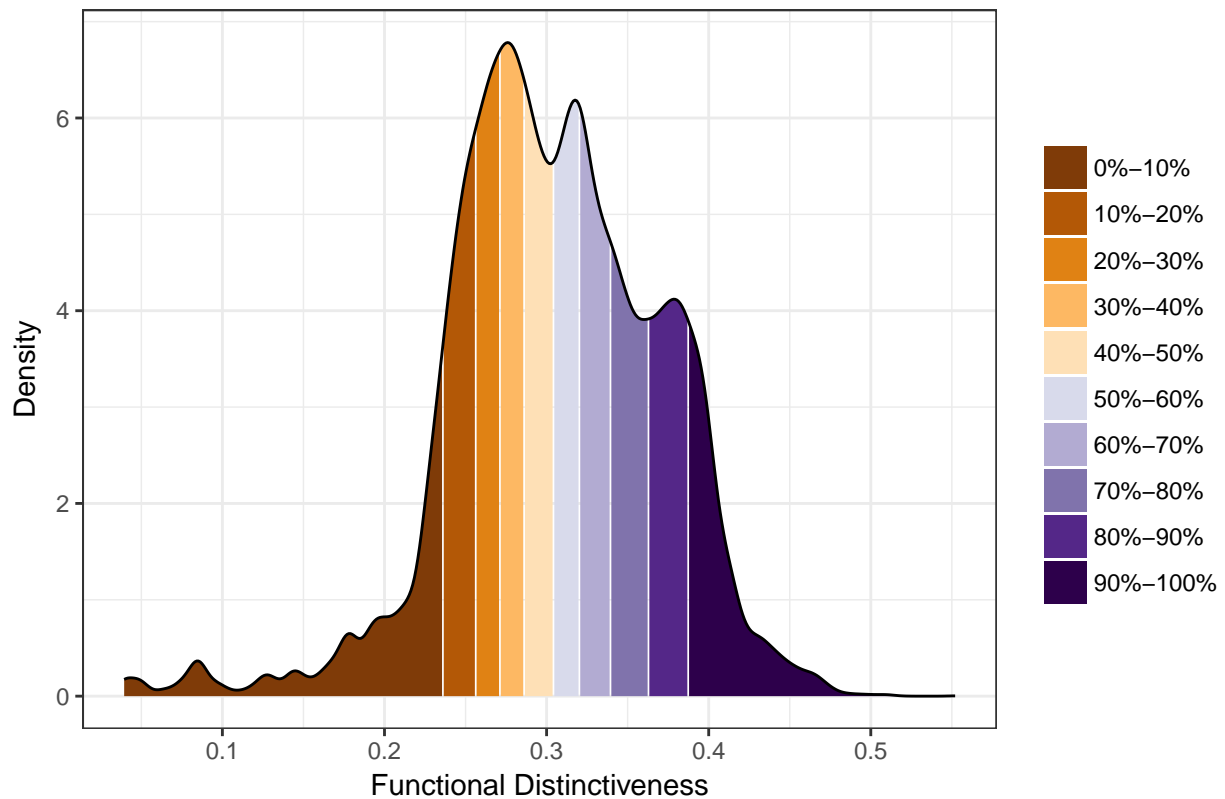
### Functional Distinctiveness

```
mammal_di = as.matrix(mammal_funrar$Di)

only_di = mammal_di[!is.na(mammal_di)]

dens_di = divr::make_quantile_density(only_di) +
  labs(x = "Functional Distinctiveness",
       y = "Density",
       fill = NULL,
       title = "Density of each species-site Distinctiveness value")
dens_di
```

Density of each species–site Distinctiveness value



```

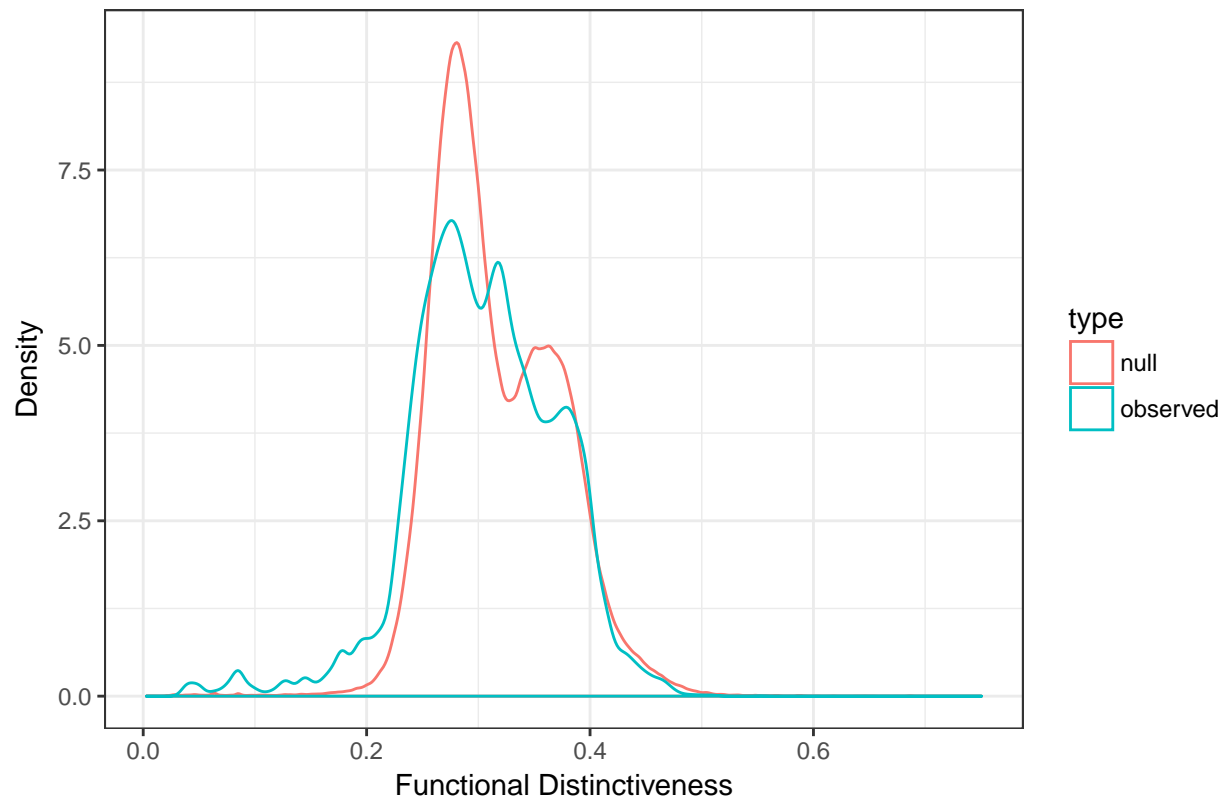
null_di = null_shuffled_matrix %>%
  lapply(as.matrix) %>%
  purrr::map(~.x[!is.na(.x)]) %>%
  purrr::map(~data.frame(Di = .x) %>%
    mutate(type = "null"))

all_di = data.frame(Di = only_di) %>%
  mutate(type = "observed") %>%
  bind_rows(null_di)

ggplot(all_di, aes(x = Di, colour = type)) +
  geom_density() +
  labs(x = "Functional Distinctiveness",
       y = "Density",
       title = "Site-Species Di Density with null model")

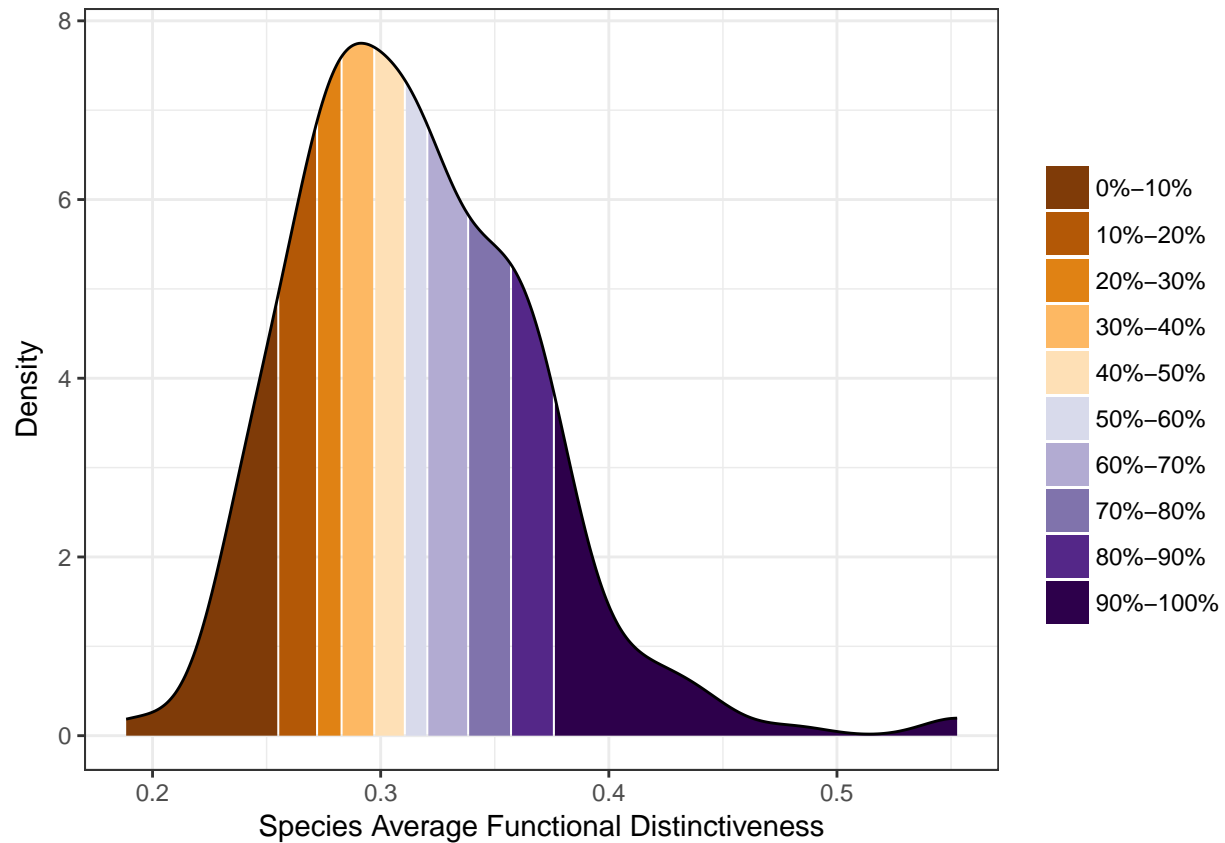
```

Site–Species Di Density with null model



```
dens_avg_di = divr::make_quantile_density(colMeans(mammal_di, na.rm = TRUE)) +  
  labs(x = "Species Average Functional Distinctiveness",  
       y = "Density",  
       fill = NULL)
```

```
dens_avg_di
```



## Latitudinal gradient in average index

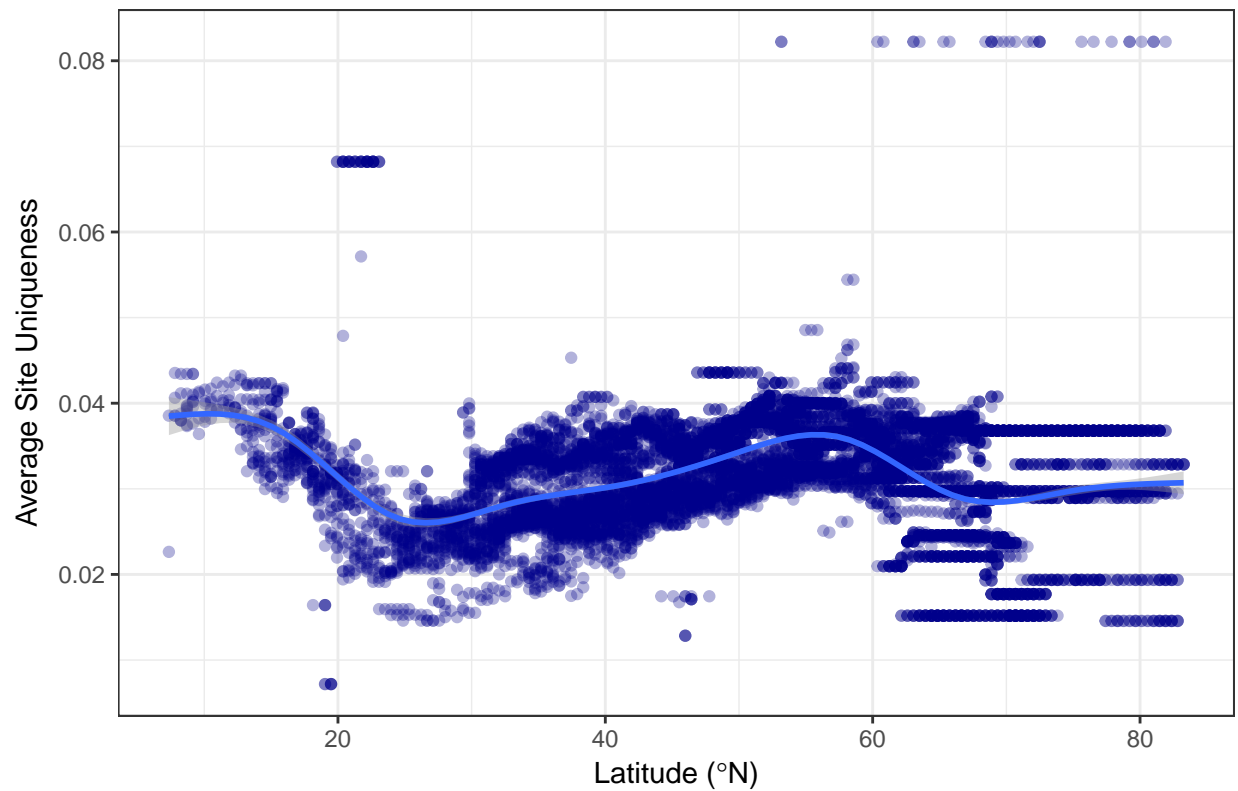
### Functional Uniqueness

```
latitudinal_ui = ggplot(site_ui, aes(x = Latitude, y = site_Ui)) +
  geom_point(alpha = 0.3, color = "darkblue") +
  geom_smooth() +
  labs(x = expression('Latitude (*degree*N)'),
       y = "Average Site Uniqueness",
       title = "Latitudinal Gradient of Uniqueness (n=9497)")
```

```
latitudinal_ui
```

```
## `geom_smooth()` using method = 'gam'
```

### Latitudinal Gradient of Uniqueness (n=9497)



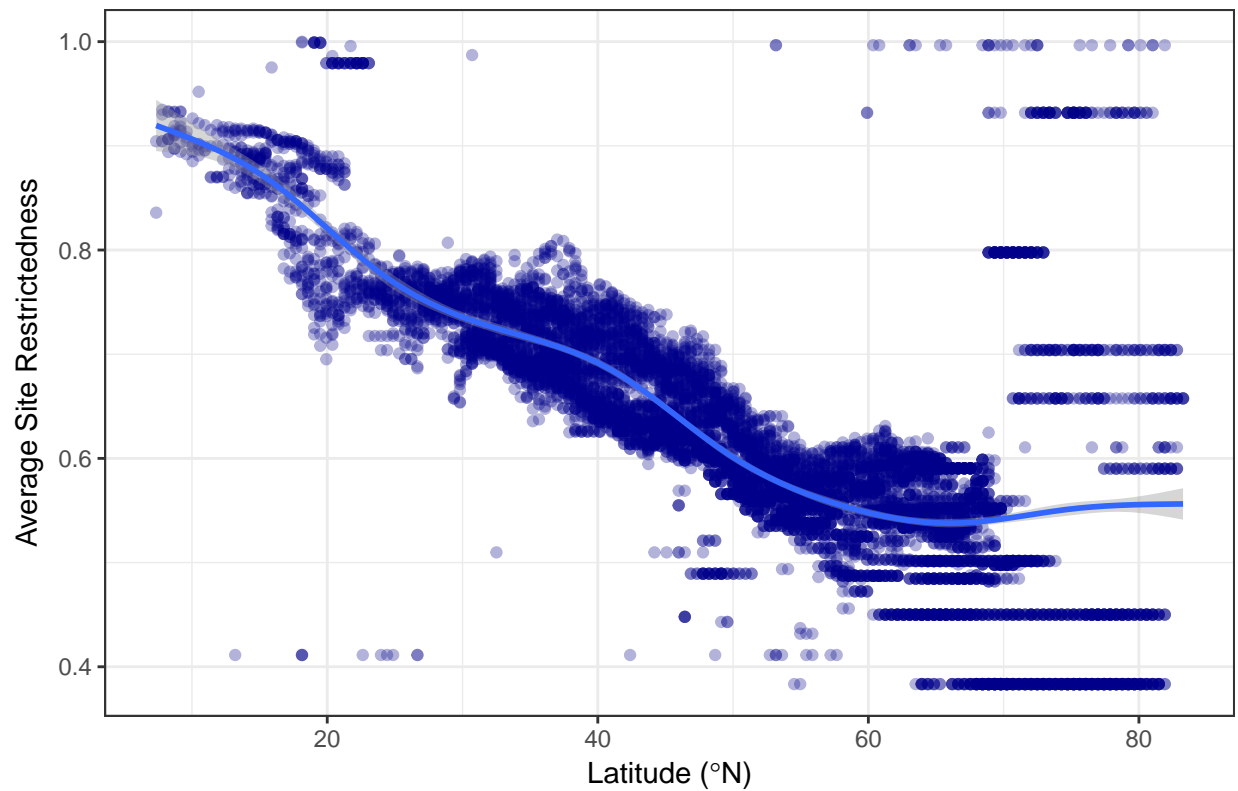
### Geographical Restrictedness

```
latitudinal_ri = ggplot(site_ri, aes(x = Latitude, y = site_Ri)) +
  geom_point(alpha = 0.3, color = "darkblue") +
  geom_smooth() +
  labs(x = expression('Latitude (*degree*N)'),
       y = "Average Site Restrictedness",
       title = "Latitudinal Gradient of Restrictedness (n=9497)")
```

```
latitudinal_ri
```

```
## `geom_smooth()` using method = 'gam'
```

### Latitudinal Gradient of Restrictedness (n=9497)



### Functional Distinctiveness

```
latitudinal_di = ggplot(site_di, aes(x = Latitude, y = avg_Di)) +
  geom_point(alpha = 0.3, color = "darkblue") +
  geom_smooth() +
  labs(x = expression('Latitude (*degree*N)'),
       y = "Average Site Distinctiveness",
       title = "Latitudinal Gradient of Distinctiveness (n=9055)")
```

```
latitudinal_di
```

```
## `geom_smooth()` using method = 'gam'
```

```
## Warning: Removed 552 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 552 rows containing missing values (geom_point).
```



Latitudinal Gradient of Distinctiveness (n=9055)

