

# Exploring the Relationship Between Landscape Diversity and both Bee Diversity and Normalized Richess

Eliza Black and Emmie Sullivan

Dec. 2024

## Introduction:

### Relevant Background and References:

Wild and managed pollinators benefit society in various ways, including bolstering food security, supporting livelihoods, enriching cultural values, and sustaining biodiversity and ecosystem health (Potts et al., 2016). This has made the study of pollinators a prominent subject in ecology. Pollinator populations are experiencing significant declines due to habitat loss and land-use changes, such as deforestation and agricultural expansion, as well as wider “threats to global food supply and the stability of wider pollination systems in non-crop vegetation... [and] disruptions of numerous plant-pollinator systems” (Mader et al., 2011).

This study tests how landscape diversity impacts pollinator diversity and richness. Understanding how landscape diversity affects pollinator diversity provides insight into habitat conditions that support pollinator populations. Identifying the habitat diversity levels that best support pollinator diversity can inform land management strategies to protect these populations.

## Hypotheses:

### Hypothesis 1

- Null Hypothesis (H0): There is no statistically significant relationship between landscape diversity and bee species diversity (measured by the Shannon-Wiener index).
- Alternative Hypothesis (H1): There is a statistically significant positive relationship between landscape diversity and bee species diversity (measured by the Shannon-Wiener index).

### Hypothesis 2

- Null Hypothesis (H0): There is no statistically significant relationship between landscape diversity and normalized bee species richness.
- Alternative Hypothesis (H1): There is a statistically significant positive relationship between landscape diversity and normalized bee species richness.

### Hypothesis 3

- Null Hypothesis (H0): There is no difference in species richness and diversity between elevation bands (0-500m, 500-1000m, 1000-1500m, 1500m+).

- Alternative Hypothesis (H1): The relationship between landscape diversity and both normalized bee species richness and diversity will vary significantly across different elevational bands. Specifically, at higher elevations, landscape diversity may have a weaker effect on bee species richness and diversity due to environmental constraints such as reduced floral resources and harsher weather. Conversely, the effect of landscape diversity on bee diversity and richness will be more pronounced at lower elevations.

## Data Sources:

### 1. Oregon Bee Atlas (OBA)

- **Description:** Dataset compiled by citizen scientists and professional entomologists throughout Oregon. Includes bee species, geographic, and environmental metadata.
- **Attributes:**
  - Species: Taxonomic identification of each bee recorded.
  - Elevation: Elevation (in meters) at which each bee was collected.
  - Coordinates: Latitude and longitude in decimal degrees.

### 2. National Land Cover Database (NLCD) - Oregon

- **Description:** Comprehensive land cover classifications, clipped to Oregon's state boundary. Projected in Oregon Lambert (EPSG:2992).
- **Attributes:**
  - 16 land cover classifications including grassland, forested areas, and developed land.
- **Publication Details:**
  - Published: January 2019
  - Updated: November 2023
  - Publisher: U.S. Geological Survey (USGS)

### 3. Oregon State Boundary Dataset

- **Description:** Jurisdictional and cartographic perimeters of Oregon. Useful for resource management and mapping.

## Load in necessary libraries

## Load in Oregon basemap

```
or_boundary <- read_sf(dsn = "data/Oregon_State_Boundary_-2975945127261102476/Oregon_State_Boundary.shp"
# str(or_boundary)
```

## Load in OBA Data

```
oba <-
  read.csv("data/OBA_2018-2023_combined_dataset_working.csv")
```

## Segment OBA data

```

# - Metadata we need:
#   - `Species`: The taxonomic identification of each bee (group of closely related species)
#   - `Genus`: The taxonomic identification of each bee (specific type within Genus)
#   - `Elevation`: The elevation (in meters) at which each bee was collected.
#   - `Coordinates`: Latitude (`Dec.Lat.`) and longitude (`Dec.Long.`) in decimal degrees.

# Subset oba data to our needed metadata
oba <- dplyr::select(oba, Genus, Species, Dec..Lat., Dec..Long., Elevation..m., MonthJul)
colnames(oba) <- c("Genus", "Spec", "Lat", "Long", "Elev", "Month")

# Make elevation values numeric
oba$Elev <- as.numeric(oba$Elev)

## Warning: NAs introduced by coercion

```

## Clean OBA data

```

# Remove elevation rows with null values

for (col in colnames(oba)) {
  oba <- oba[oba[[col]] != "", ]
  oba <- oba[!is.na(oba[[col]]), ]
}
# Number of rows after cleaning: 41643
# nrow(oba)

```

## Combine genus & species columns in OBA data

```

# Combine genus & species
GenusSpec <- paste(oba$Genus, oba$Spec, sep=" ")

oba <- oba %>%
  mutate(GenusSpec)

```

## Load in NLCD - Oregon (National Land Cover Database)

### NLCD Manipulation

```

# Create a new template raster with the desired resolution
# The larger the resolution, the easier to load

# NOTE: need to reproject the raster before changing the crs to use meters as the unit
target_res <- 500 # Desired resolution in original map units (feet)
extent_nlcd <- ext(nlcd) # Get the extent of the original raster

```

```

# Create an empty raster template
template <- rast(extent = extent_nlcd, res = target_res, crs = crs(nlcd))

# Resample the original raster to match the new template
nlcd_resampled <- resample(nlcd, template, method = "near") # Use "bilinear" for continuous data or "n

# Convert raster to data frame for plotting
nlcd_df <- as.data.frame(nlcd_resampled, na.rm = TRUE, xy = TRUE)

# Add land cover type labels
nlcd_df <- nlcd_df %>%
  mutate(LandType = case_when(
    Layer_1 == 11 ~ "Open Water",
    Layer_1 == 12 ~ "Perennial Ice/Snow",
    Layer_1 == 21 ~ "Developed, Open Space",
    Layer_1 == 22 ~ "Developed, Low Intensity",
    Layer_1 == 23 ~ "Developed, Medium Intensity",
    Layer_1 == 24 ~ "Developed, High Intensity",
    Layer_1 == 31 ~ "Barren Land (Rock/Sand/Clay)",
    Layer_1 == 41 ~ "Deciduous Forest",
    Layer_1 == 42 ~ "Evergreen Forest",
    Layer_1 == 43 ~ "Mixed Forest",
    Layer_1 == 51 ~ "Dwarf Scrub",
    Layer_1 == 52 ~ "Shrub/Scrub",
    Layer_1 == 71 ~ "Grassland/Herbaceous",
    Layer_1 == 72 ~ "Sedge/Herbaceous",
    Layer_1 == 73 ~ "Lichens",
    Layer_1 == 74 ~ "Moss",
    Layer_1 == 81 ~ "Pasture/Hay",
    Layer_1 == 82 ~ "Cultivated Crops",
    Layer_1 == 90 ~ "Woody Wetlands",
    Layer_1 == 95 ~ "Emergent Herbaceous Wetlands",
    TRUE ~ NA_character_ # Exclude unknown values
  )) %>%
  filter(!is.na(LandType)) # Remove rows with NA values

# Observe dataframe columns
head(nlcd_df)

```

	x	y	Layer_1	LandType
## 1	420205.2	1674828	11	Open Water
## 2	420705.2	1674828	11	Open Water
## 3	421205.2	1674828	95	Emergent Herbaceous Wetlands
## 4	421705.2	1674828	11	Open Water
## 5	422205.2	1674828	11	Open Water
## 6	418205.2	1674328	11	Open Water

```

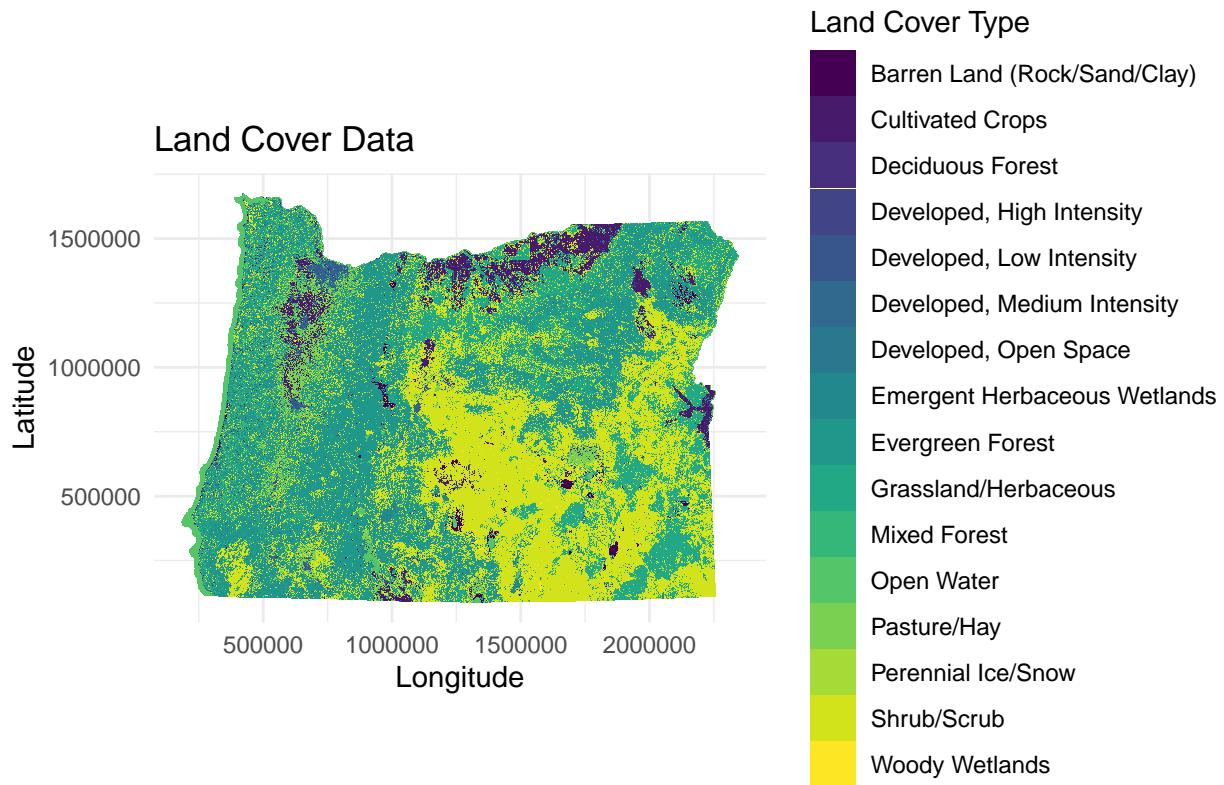
# Initial data exploration: ensuring the nlcd data loads and displays properly
ggplot(nlcd_df, aes(x = x, y = y, fill = LandType)) +
  geom_tile() + # Create a heatmap-like plot with tiles
  scale_fill_viridis_d() + # Use a discrete color scale (viridis is suitable for categorical data)
  theme_minimal() + # Clean theme
  labs(title = "Land Cover Data",

```

```

x = "Longitude",
y = "Latitude",
fill = "Land Cover Type") +
coord_fixed() # Fix aspect ratio to prevent distortion

```



### 3. Spatial Integration

```

# Spatial Integration
# Convert OBA bee observations to spatial points
# Convert Oregon boundary to spatial object

# Convert OBA data to spatial points
oba_sf <- st_as_sf(oba, coords = c("Long", "Lat"), crs = 4326)

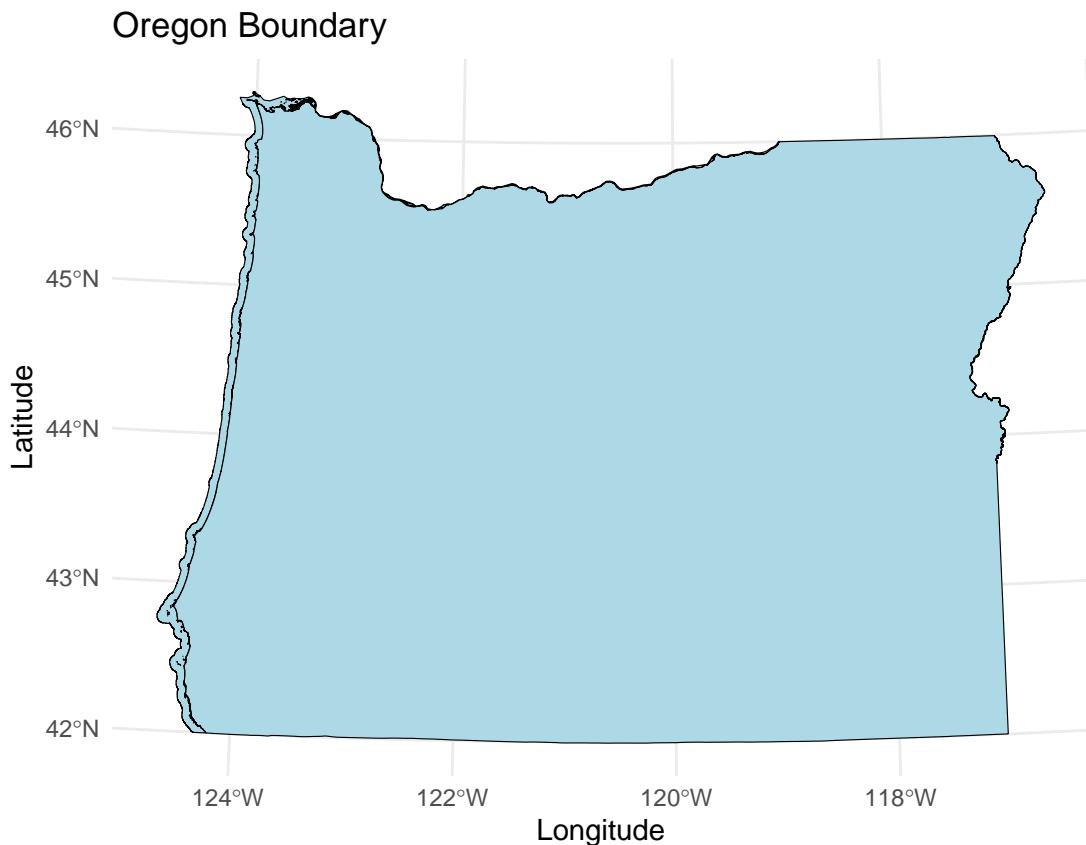
# Transform to the same CRS as the NLCD raster (assuming the raster is in UTM)
oba_sf <- st_transform(oba_sf, crs = crs(nlcd_resampled))

# Convert Oregon boundary to spatial object
or_boundary <- st_as_sf(or_boundary, coords = c("lon", "lat"), crs = crs(oba_sf))

# Plot the Oregon boundary

```

```
# Ensure that the Oregon boundary plot loads properly
ggplot() +
  geom_sf(data = or_boundary, fill = "lightblue", color = "black") +
  theme_minimal() +
  labs(title = "Oregon Boundary",
       x = "Longitude",
       y = "Latitude")
```



## Crop oba and nlcd to or

```
# Clip OBA data to Oregon boundary
oba_sf <- st_crop(oba_sf, or_boundary)

## Warning: attribute variables are assumed to be spatially constant throughout
## all geometries

str(oba_sf)

## Classes 'sf' and 'data.frame': 15170 obs. of 6 variables:
## $ Genus      : chr  "Lasioglossum" "Ceratina" "Agapostemon" "Lasioglossum" ...
## $ Spec       : chr  "zephyrus" "acantha" "texanus" "titusi" ...
```

```

## $ Elevation : num 48 79 79 79 79 79 80 80 80 80 123 ...
## $ Month    : chr "April" "April" "April" "April" ...
## $ GenusSpec: chr "Lasioglossum zephyrus" "Ceratina acantha" "Agapostemon texanus" "Lasioglossum titus"
## $ geometry :sfc_POINT of length 15170; first list element: 'XY' num 656326 1171801
## - attr(*, "sf_column")= chr "geometry"
## - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",...: NA NA NA NA NA
## ..- attr(*, "names")= chr [1:5] "Genus" "Spec" "Elev" "Month" ...

```

## Elevational Band Categorization

Define elevation bands (Low, Middle, High).

Assign OBA observations to corresponding elevation bands.

```

# Divide data into elevational bands
elevation_bands <- cut(oba_sf$Elev, breaks = c(0, 500, 1000, 1500, Inf),
                      labels = c("0-500m", "500-1000m", "1000-1500m", "1500m+"))

# Add to spatial data
oba_sf <- oba_sf %>%
  mutate(ElevationBand = elevation_bands) %>%
  filter(!is.na(ElevationBand))

head(oba_sf)

## Simple feature collection with 6 features and 6 fields
## Geometry type: POINT
## Dimension: XY
## Bounding box: xmin: 597641.2 ymin: 1171801 xmax: 656326.4 ymax: 1183516
## Projected CRS: NAD83 / Oregon GIC Lambert (ft)
##           Genus      Spec Elev Month          GenusSpec
## 19365 Lasioglossum zephyrus 48 April Lasioglossum zephyrus
## 19386 Ceratina acantha 79 April Ceratina acantha
## 19387 Agapostemon texanus 79 April Agapostemon texanus
## 19392 Lasioglossum titus 79 April Lasioglossum titus
## 19393 Halictus tripartitus 79 April Halictus tripartitus
## 19394 Lasioglossum titus 79 April Lasioglossum titus
##           geometry ElevationBand
## 19365 POINT (656326.4 1171801) 0-500m
## 19386 POINT (598675.8 1183143) 0-500m
## 19387 POINT (598429.4 1183516) 0-500m
## 19392 POINT (597641.2 1183178) 0-500m
## 19393 POINT (597641.2 1183178) 0-500m
## 19394 POINT (597899.8 1183169) 0-500m

```

## Create initial visualizations and stats

```

# Manipulation and summarizing data in relevant ways (initial exploration of data)

# Print summary stats for oba_sf
summary(oba_sf)

##      Genus           Spec          Elev        Month
##  Length:15158    Length:15158    Min.   : 1  Length:15158
##  Class :character  Class :character  1st Qu.: 97  Class :character
##  Mode  :character  Mode  :character  Median : 406  Mode  :character
##                                         Mean   : 721
##                                         3rd Qu.:1277
##                                         Max.  :7278
##      GenusSpec            geometry     ElevationBand
##  Length:15158    POINT       :15158  0-500m   :8009
##  Class :character  epsg:NA     :     0  500-1000m:1737
##  Mode  :character  +proj=lcc ...:     0  1000-1500m:3447
##                                         1500m+  :1965
##  ##

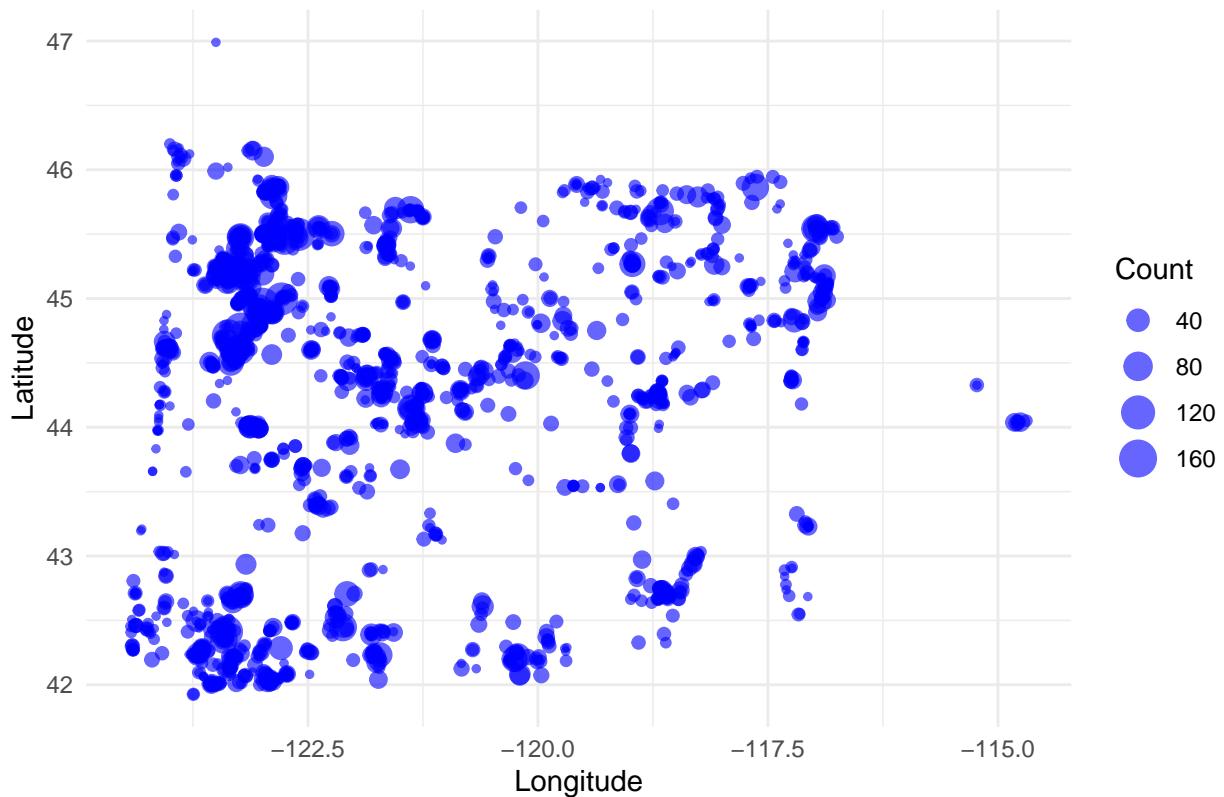
## Add counts for species
species_counts_location <- oba %>%
  group_by(Long, Lat) %>%
  summarise(Count = n())

## `summarise()` has grouped output by 'Long'. You can override using the
## `.`groups` argument.

# Scatter plot with point size by count
ggplot(species_counts_location, aes(x = Long, y = Lat, size = Count)) +
  geom_point(color = "blue", alpha = 0.6) +
  theme_minimal() +
  labs(title = "Species Count by Location",
       x = "Longitude",
       y = "Latitude",
       size = "Count")

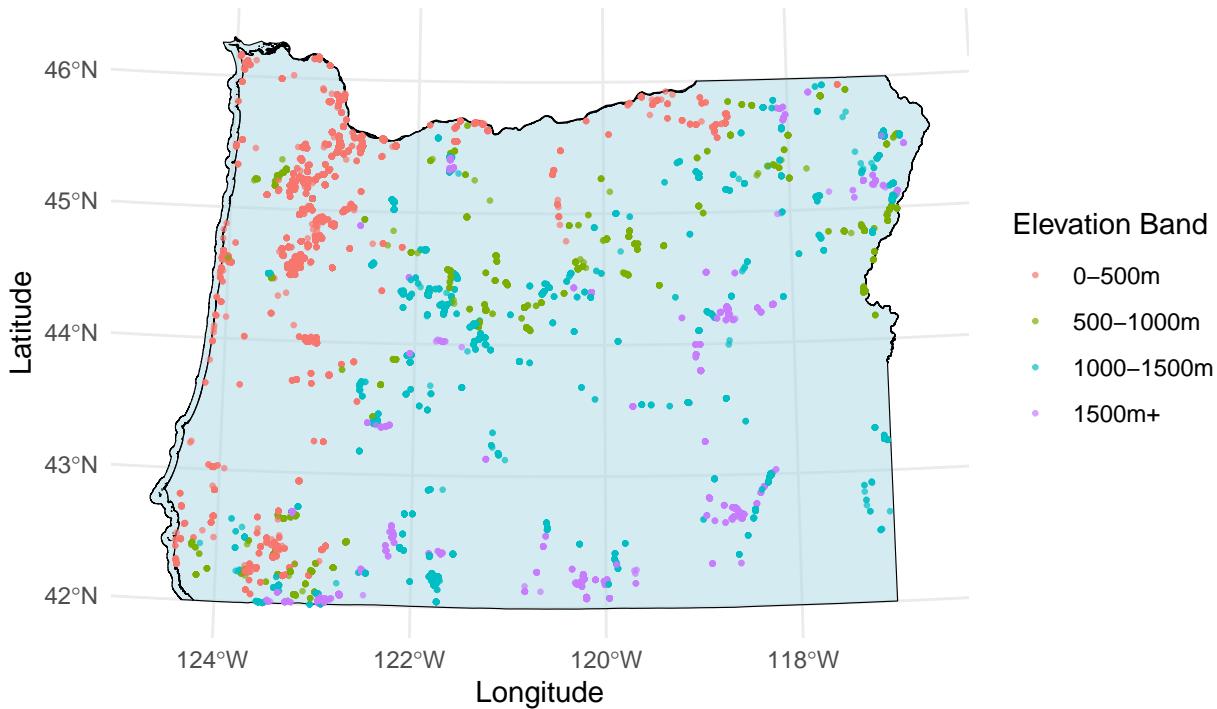
```

## Species Count by Location



```
# Plot colored by ElevationBand
ggplot() +
  # Add Oregon boundary layer
  geom_sf(data = or_boundary, fill = "lightblue", color = "black", alpha = 0.5) +
  
  # OBA points colored by ElevationBand
  geom_sf(data = oba_sf, aes(color = ElevationBand), size = 0.5, alpha = 0.7) +
  
  theme_minimal() +
  labs(title = "OBA Occurrences Locations Colored by Elevation Bands",
       x = "Longitude",
       y = "Latitude",
       color = "Elevation Band") +
  theme(legend.position = "right")
```

## OBA Occurrences Locations Colored by Elevation Bands



## Creating buffers

```
# Change to crs that uses meters, not feet
oba_sf <- st_transform(oba_sf, crs = 32610)

# Generate 500m buffers (one around each oba data point)
buffers_500m <- st_buffer(oba_sf, dist = 500)

# add buffer_id column to buffer
buffers_500m$buffer_id <- seq_len(nrow(buffers_500m))

#perform spatial join: this assigns each species point to a buffer by buffer_id
species_in_buffers <- st_join(oba_sf, buffers_500m, join = st_within)
```

## Calculating and organizing species richness for each buffer

```
community_matrix <- species_in_buffers %>%
  st_drop_geometry() %>%
  group_by(buffer_id, GenusSpec.x) %>%
  summarize(count = n()) %>%
  # Drop spatial data for tabular manipulation
  # Replace `buffer_id` with your buffer identifier column
  # Count occurrences
```

```

pivot_wider(names_from = GenusSpec.x, values_from = count, values_fill = 0) %>%
  column_to_rownames("buffer_id")           # Set buffer IDs as row names

## `summarise()` has grouped output by 'buffer_id'. You can override using the
## `.groups` argument.

richness <- specnumber(community_matrix)
# richness$buffer_id <- as.integer(richness$buffer_id) # Make buffer ids ints
head(richness)

##  1  2  3  4  5  6
## 24 13 13 13 13 13

# Calculate total observations (sum of individuals per buffer)
total_observations <- rowSums(community_matrix) # Sum across species for each buffer

# Calculate normalized richness (richness / total observations)
normalized_richness <- richness / total_observations

# Create a dataframe with buffer ID and normalized richness
richness_df <- tibble(
  buffer_id = as.integer(names(richness)),      # Convert names (buffer IDs) to integer
  normalized_richness = normalized_richness       # Add the normalized richness values
)

# Check result
head(richness_df)

## # A tibble: 6 x 2
##   buffer_id normalized_richness
##       <int>          <dbl>
## 1 1            0.24
## 2 2            0.289
## 3 3            0.289
## 4 4            0.289
## 5 5            0.289
## 6 6            0.289

# Join richness data back to the buffers
buffers_500m <- buffers_500m %>%
  left_join(richness_df, by = "buffer_id")

head(buffers_500m)

## Simple feature collection with 6 features and 8 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: 478837.1 ymin: 4975452 xmax: 497817.3 ymax: 4979484
## Projected CRS: WGS 84 / UTM zone 10N
##           Genus      Spec Elev Month           GenusSpec ElevationBand
## 1 Lasioglossum    zephyrus     48 April Lasioglossum zephyrus        0-500m

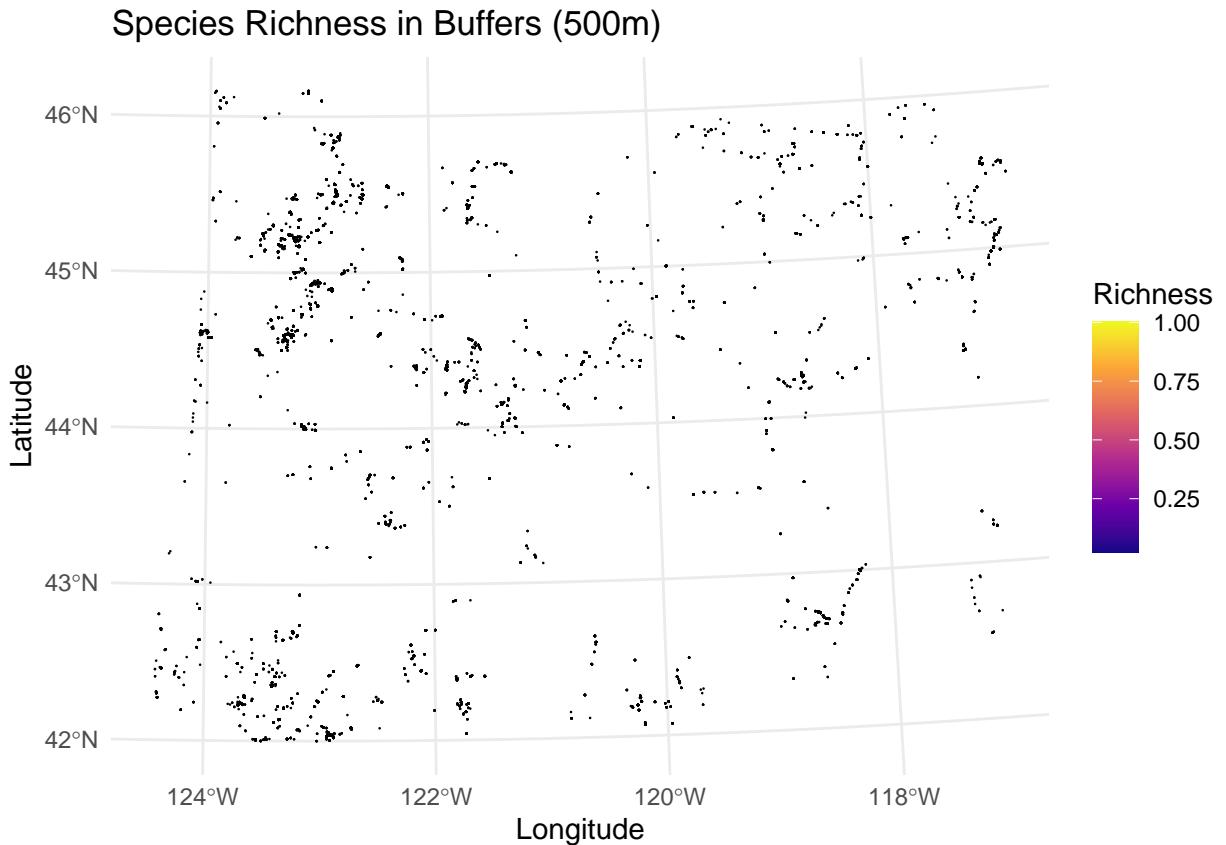
```

```

## 2      Ceratina    acantha   79 April     Ceratina acantha   0-500m
## 3  Agapostemon texanus   79 April  Agapostemon texanus   0-500m
## 4 Lasioglossum    titusi   79 April  Lasioglossum titusi   0-500m
## 5  Halictus tripartitus 79 April  Halictus tripartitus 0-500m
## 6 Lasioglossum    titusi   79 April  Lasioglossum titusi   0-500m
##   buffer_id normalized_richness           geometry
## 1          1        0.2400000 POLYGON ((497817.3 4975952, ...
## 2          2        0.2888889 POLYGON ((480152.6 4978873, ...
## 3          3        0.2888889 POLYGON ((480074.1 4978984, ...
## 4          4        0.2888889 POLYGON ((479837.1 4978874, ...
## 5          5        0.2888889 POLYGON ((479837.1 4978874, ...
## 6          6        0.2888889 POLYGON ((479916 4978873, 4 ...

# Plot buffers with richness levels
ggplot() +
  geom_sf(data = buffers_500m, aes(fill = normalized_richness), color = "black", size = 0.1) +
  scale_fill_viridis_c(option = "plasma", na.value = "white") +
  labs(
    title = "Species Richness in Buffers (500m)",
    fill = "Richness",
    x = "Longitude",
    y = "Latitude"
  ) +
  theme_minimal()

```



## Calculating species diversity (using Shannon's)

```
## Calculating and organizing diversity (Shannon's index) for each buffer
# Using the vegan package to compute Shannon's diversity index

# Calculate Shannon's diversity index for each buffer
bee_diversity <- vegan::diversity(community_matrix, index = "shannon")

# Create a dataframe with buffer ID and Shannon's diversity
diversity_df <- tibble(
  buffer_id = as.integer(rownames(community_matrix)), # Extract buffer IDs as integers
  bee_diversity = bee_diversity # Add diversity values
)

# Join diversity data back to the buffers
buffers_500m <- buffers_500m %>%
  left_join(diversity_df, by = "buffer_id")

# Check result
head(buffers_500m)
```

```
## Simple feature collection with 6 features and 9 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: 478837.1 ymin: 4975452 xmax: 497817.3 ymax: 4979484
## Projected CRS: WGS 84 / UTM zone 10N
##           Genus      Spec Elev Month          GenusSpec ElevationBand
## 1 Lasioglossum zephyrus 48 April Lasioglossum zephyrus 0-500m
## 2 Ceratina acantha 79 April Ceratina acantha 0-500m
## 3 Agapostemon texanus 79 April Agapostemon texanus 0-500m
## 4 Lasioglossum titusi 79 April Lasioglossum titusi 0-500m
## 5 Halictus tripartitus 79 April Halictus tripartitus 0-500m
## 6 Lasioglossum titusi 79 April Lasioglossum titusi 0-500m
##   buffer_id normalized_richness bee_diversity geometry
## 1         1            0.2400000    2.577413 POLYGON ((497817.3 4975952, ...
## 2         2            0.2888889    1.966617 POLYGON ((480152.6 4978873, ...
## 3         3            0.2888889    1.966617 POLYGON ((480074.1 4978984, ...
## 4         4            0.2888889    1.966617 POLYGON ((479837.1 4978874, ...
## 5         5            0.2888889    1.966617 POLYGON ((479837.1 4978874, ...
## 6         6            0.2888889    1.966617 POLYGON ((479916 4978873, 4...
```

## Find landscape diversity for each buffer; append to buffers\_500m

```
buffers_500m <- st_transform(buffers_500m, crs = st_crs(nlcd_resampled))
# Spatial join to extract land cover types from NLCD raster within each buffer
land_cover_in_buffers <- extract(nlcd_resampled, buffers_500m, fun = function(x) {
  # Get the most frequent land cover type in the buffer area
  if (length(x) > 0) {
    # Count occurrences of each land cover type
    land_cover_count <- table(x)
```

```

# Use vegan diversity() function to calculate Shannon's Div Index
shannon_index <- vegan::diversity(as.numeric(land_cover_count), index = "shannon")

return(shannon_index)
} else {
  return(NULL) # If no data within the buffer, return NULL
}
})

# Rename cols to match buffers_500m
land_cover_in_buffers <- land_cover_in_buffers %>%
  rename(
    landscape_div = Layer_1,
    buffer_id = ID
  )

# Join diversity vals with buffers df
# NOTE for testing: only run once
buffers_500m <- buffers_500m %>%
  left_join(land_cover_in_buffers, by = "buffer_id")

head(buffers_500m)

## Simple feature collection with 6 features and 10 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: 596000.8 ymin: 1170161 xmax: 657966.8 ymax: 1185156
## Projected CRS: NAD83 / Oregon GIC Lambert (ft)
##           Genus      Spec Elev Month          GenusSpec ElevationBand
## 1 Lasioglossum zephyrus  48 April Lasioglossum zephyrus      0-500m
## 2 Ceratina acantha   79 April     Ceratina acantha      0-500m
## 3 Agapostemon texanus 79 April     Agapostemon texanus      0-500m
## 4 Lasioglossum titusi 79 April Lasioglossum titusi      0-500m
## 5 Halictus tripartitus 79 April     Halictus tripartitus      0-500m
## 6 Lasioglossum titusi 79 April Lasioglossum titusi      0-500m
##   buffer_id normalized_richness bee_diversity landscape_div
## 1           1            0.2400000      2.577413      1.1771512
## 2           2            0.2888889      1.966617      0.8002288
## 3           3            0.2888889      1.966617      0.5670578
## 4           4            0.2888889      1.966617      0.7297747
## 5           5            0.2888889      1.966617      0.7297747
## 6           6            0.2888889      1.966617      0.6747075
##   geometry
## 1 POLYGON ((657966.4 1171751, ...
## 2 POLYGON ((600315.9 1183093, ...
## 3 POLYGON ((600069.5 1183466, ...
## 4 POLYGON ((599281.2 1183128, ...
## 5 POLYGON ((599281.2 1183128, ...
## 6 POLYGON ((599539.9 1183119, ...

bee_div <- buffers_500m$bee_diversity
names(bee_div) <- NULL
land_div <- buffers_500m$landscape_div

```

```

dats_div <- data.frame(bee_div = bee_div,
                       land_div=land_div)

mod <- lm(bee_div ~ scale(land_div), data=dats_div)
summary(mod)

## 
## Call:
## lm(formula = bee_div ~ scale(land_div), data = dats_div)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.14139 -0.51584  0.09407  0.58285  1.48276 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.910737  0.006271 304.71 <2e-16 ***
## scale(land_div) 0.096932  0.006271   15.46 <2e-16 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.772 on 15156 degrees of freedom
## Multiple R-squared:  0.01552,    Adjusted R-squared:  0.01546 
## F-statistic: 238.9 on 1 and 15156 DF,  p-value: < 2.2e-16

bee_rich <- buffers_500m$normalized_richness
names(bee_rich) <- NULL

dats_rich <- data.frame(bee_rich = bee_rich,
                        land_div=land_div)

mod2 <- lm(bee_rich ~ scale(land_div), data=dats_rich)
summary(mod2)

## 
## Call:
## lm(formula = bee_rich ~ scale(land_div), data = dats_rich)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.38426 -0.14892 -0.05456  0.10336  0.67828 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 0.402468  0.001742 231.00 <2e-16 ***
## scale(land_div) -0.033934  0.001742  -19.48 <2e-16 *** 
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2145 on 15156 degrees of freedom
## Multiple R-squared:  0.02442,    Adjusted R-squared:  0.02435 
## F-statistic: 379.3 on 1 and 15156 DF,  p-value: < 2.2e-16

```

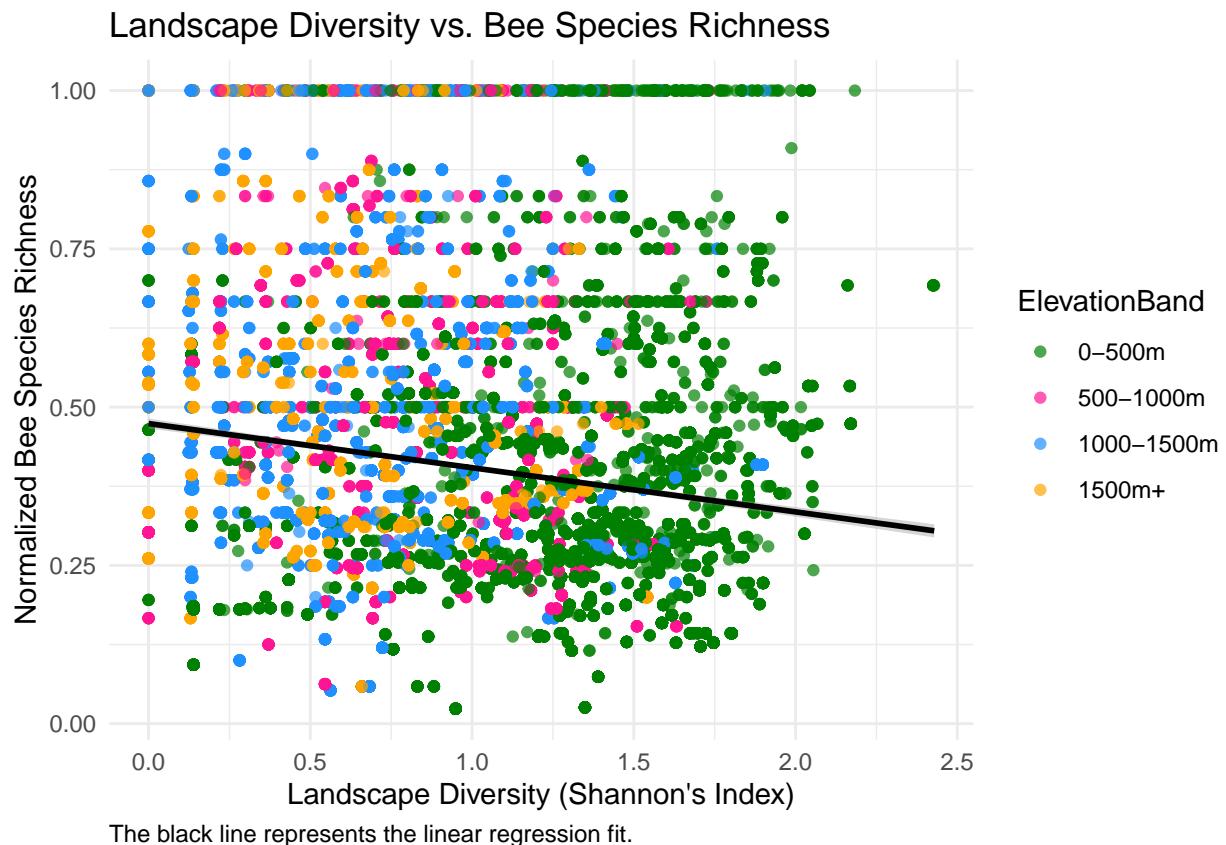
```

# Set up plot colors
main_colors <- c("#008000", "#FF1493", "#1E90FF", "#FFA500")

# Scatter plot of landscape diversity vs. bee species richness
ggplot(buffers_500m, aes(x = landscape_div, y = normalized_richness)) +
  geom_point(aes(color = ElevationBand), alpha = 0.7) + # Color by ElevationBand
  scale_color_manual(values = main_colors) +
  labs(
    title = "Landscape Diversity vs. Bee Species Richness",
    x = "Landscape Diversity (Shannon's Index)",
    y = "Normalized Bee Species Richness",
    caption = "The black line represents the linear regression fit."
  ) +
  geom_smooth(method = "lm", color = "black") +
  theme_minimal() +
  theme(
    plot.caption = element_text(hjust = 0) # Left-align caption
  )

```

## `geom\_smooth()` using formula = 'y ~ x'



```

# Scatter plot of landscape diversity vs. bee species richness
ggplot(buffers_500m, aes(x = landscape_div, y = bee_diversity)) +
  geom_point(aes(color = ElevationBand), alpha = 0.7) + # Color by ElevationBand
  scale_color_manual(values = main_colors) +

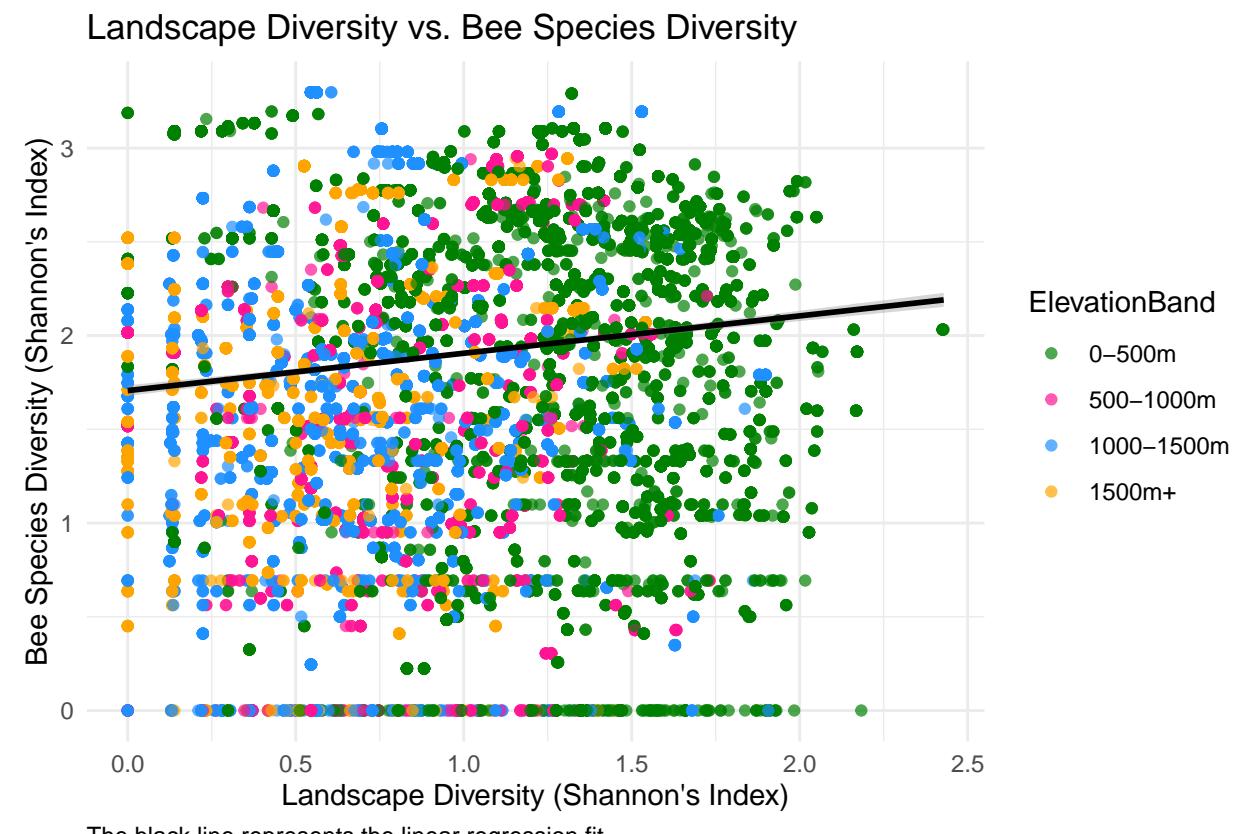
```

```

  labs(
    title = "Landscape Diversity vs. Bee Species Diversity",
    x = "Landscape Diversity (Shannon's Index)",
    y = "Bee Species Diversity (Shannon's Index)",
    caption = "The black line represents the linear regression fit."
  ) +
  geom_smooth(method = "lm", color = "black") +
  theme_minimal() +
  theme(
    plot.caption = element_text(hjust = 0) # Left-align caption
  )

```

## `geom\_smooth()` using formula = 'y ~ x'



```

# Plot landcover, oba, buffers

# Import colors
library(RColorBrewer)
colors <- c(brewer.pal(9, "Dark2"), brewer.pal(9, "Paired"))

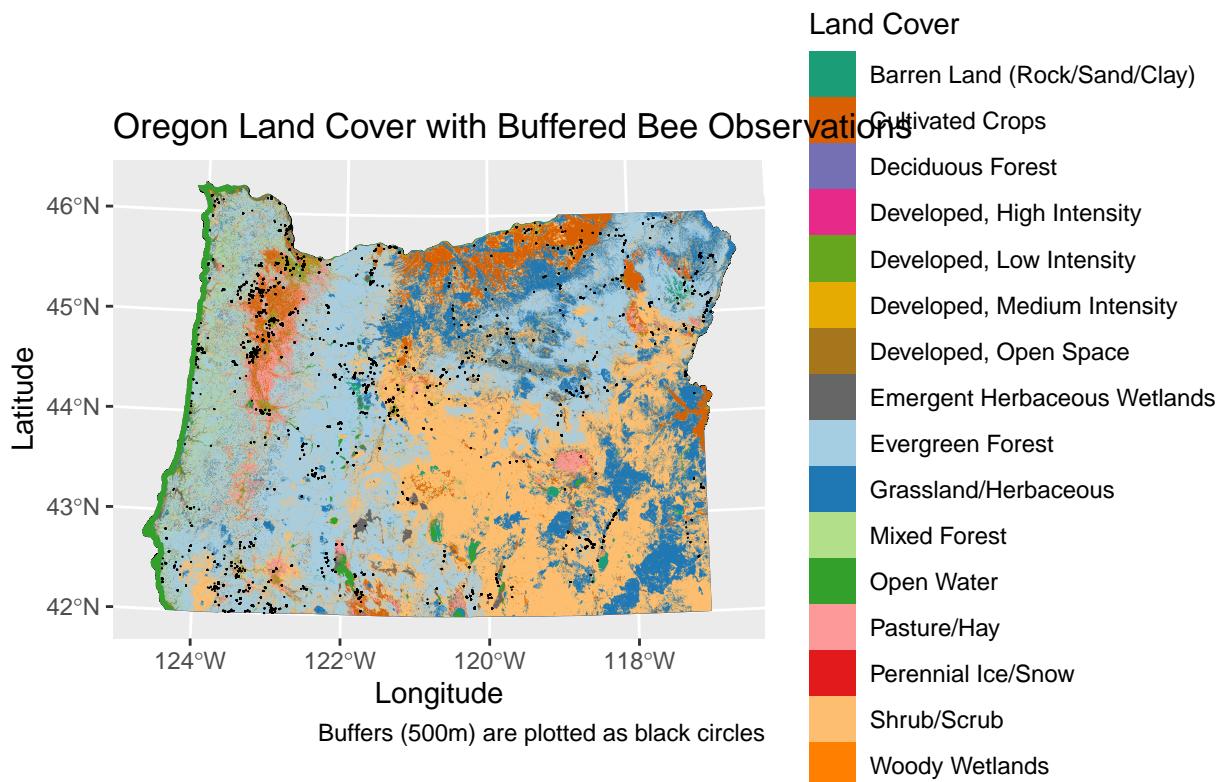
## Warning in brewer.pal(9, "Dark2"): n too large, allowed maximum for palette Dark2 is 8
## Returning the palette you asked for with that many colors

```

```

# Plot buffers around the bee observations
ggplot() +
  # Plot the Oregon boundary
  geom_sf(data = or_boundary, fill = NA, color = "black") +
  # Plot the land cover data (NLCD)
  geom_raster(data = nlcd_df, aes(x = x, y = y, fill = LandType)) +
  scale_fill_manual(values = colors, name = "Land Cover") +
  # Plot the 500m buffers around OBA data
  geom_sf(data = buffers_500m, fill = "black", alpha = 1, color = "black") +
  # Customize the plot appearance
  labs(title = "Oregon Land Cover with Buffered Bee Observations",
       x = "Longitude",
       y = "Latitude",
       caption = "Buffers (500m) are plotted as black circles")

```



```
theme_minimal()
```

```

## List of 136
## $ line                               :List of 6
##   ..$ colour      : chr "black"
##   ..$ linewidth   : num 0.5
##   ..$ linetype    : num 1

```

```

## ..$ lineend      : chr "butt"
## ..$ arrow        : logi FALSE
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_line" "element"
## $ rect           :List of 5
##   ..$ fill        : chr "white"
##   ..$ colour      : chr "black"
##   ..$ linewidth   : num 0.5
##   ..$ linetype    : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ text            :List of 11
##   ..$ family      : chr ""
##   ..$ face        : chr "plain"
##   ..$ colour      : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 0points
##   ... .- attr(*, "unit")= int 8
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ title           : NULL
## $ aspect.ratio    : NULL
## $ axis.title      : NULL
## $ axis.title.x    :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.75points 0points 0points 0points
##   ... .- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.title.x.top :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.75points 0points
##   ... .- attr(*, "unit")= int 8
##   ..$ debug        : NULL

```

```

##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.title.x.bottom           : NULL
## $ axis.title.y                  :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : NULL
##   ..$ size        : NULL
##   ..$ hjust       : NULL
##   ..$ vjust       : num 1
##   ..$ angle       : num 90
##   ..$ lineheight  : NULL
##   ..$ margin      : 'margin' num [1:4] 0points 2.75points 0points 0points
##   ..- attr(*, "unit")= int 8
##   ..$ debug       : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.title.y.left            : NULL
## $ axis.title.y.right           :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : NULL
##   ..$ size        : NULL
##   ..$ hjust       : NULL
##   ..$ vjust       : num 1
##   ..$ angle       : num -90
##   ..$ lineheight  : NULL
##   ..$ margin      : 'margin' num [1:4] 0points 0points 0points 2.75points
##   ..- attr(*, "unit")= int 8
##   ..$ debug       : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text                 :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : chr "grey30"
##   ..$ size        : 'rel' num 0.8
##   ..$ hjust       : NULL
##   ..$ vjust       : NULL
##   ..$ angle       : NULL
##   ..$ lineheight  : NULL
##   ..$ margin      : NULL
##   ..$ debug       : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.x                :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : NULL
##   ..$ size        : NULL
##   ..$ hjust       : NULL
##   ..$ vjust       : num 1
##   ..$ angle       : NULL
##   ..$ lineheight  : NULL

```

```

## ..$ margin      : 'margin' num [1:4] 2.2points 0points 0points 0points
## ... - attr(*, "unit")= int 8
## ..$ debug       : NULL
## ..$ inherit.blank: logi TRUE
## ... - attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.x.top           :List of 11
##   ..$ family     : NULL
##   ..$ face       : NULL
##   ..$ colour     : NULL
##   ..$ size       : NULL
##   ..$ hjust      : NULL
##   ..$ vjust      : num 0
##   ..$ angle      : NULL
##   ..$ lineheight : NULL
##   ..$ margin      : 'margin' num [1:4] 0points 0points 2.2points 0points
##   ... - attr(*, "unit")= int 8
##   ..$ debug       : NULL
##   ..$ inherit.blank: logi TRUE
##   ... - attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.x.bottom        : NULL
## $ axis.text.y               :List of 11
##   ..$ family     : NULL
##   ..$ face       : NULL
##   ..$ colour     : NULL
##   ..$ size       : NULL
##   ..$ hjust      : num 1
##   ..$ vjust      : NULL
##   ..$ angle      : NULL
##   ..$ lineheight : NULL
##   ..$ margin      : 'margin' num [1:4] 0points 2.2points 0points 0points
##   ... - attr(*, "unit")= int 8
##   ..$ debug       : NULL
##   ..$ inherit.blank: logi TRUE
##   ... - attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.y.left          : NULL
## $ axis.text.y.right         :List of 11
##   ..$ family     : NULL
##   ..$ face       : NULL
##   ..$ colour     : NULL
##   ..$ size       : NULL
##   ..$ hjust      : num 0
##   ..$ vjust      : NULL
##   ..$ angle      : NULL
##   ..$ lineheight : NULL
##   ..$ margin      : 'margin' num [1:4] 0points 0points 0points 2.2points
##   ... - attr(*, "unit")= int 8
##   ..$ debug       : NULL
##   ..$ inherit.blank: logi TRUE
##   ... - attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.theta           : NULL
## $ axis.text.r               :List of 11
##   ..$ family     : NULL
##   ..$ face       : NULL
##   ..$ colour     : NULL

```

```

## ..$ size      : NULL
## ..$ hjust     : num 0.5
## ..$ vjust     : NULL
## ..$ angle     : NULL
## ..$ lineheight : NULL
## ..$ margin     : 'margin' num [1:4] 0points 2.2points 0points 2.2points
## ...- attr(*, "unit")= int 8
## ..$ debug     : NULL
## ..$ inherit.blank: logi TRUE
## ...- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.ticks      : list()
## ..- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ axis.ticks.x      : NULL
## $ axis.ticks.x.top    : NULL
## $ axis.ticks.x.bottom  : NULL
## $ axis.ticks.y      : NULL
## $ axis.ticks.y.left   : NULL
## $ axis.ticks.y.right  : NULL
## $ axis.ticks.theta   : NULL
## $ axis.ticks.r       : NULL
## $ axis.minor.ticks.x.top  : NULL
## $ axis.minor.ticks.x.bottom : NULL
## $ axis.minor.ticks.y.left  : NULL
## $ axis.minor.ticks.y.right : NULL
## $ axis.minor.ticks.theta  : NULL
## $ axis.minor.ticks.r       : NULL
## $ axis.ticks.length    : 'simpleUnit' num 2.75points
## ...- attr(*, "unit")= int 8
## $ axis.ticks.length.x   : NULL
## $ axis.ticks.length.x.top : NULL
## $ axis.ticks.length.x.bottom : NULL
## $ axis.ticks.length.y   : NULL
## $ axis.ticks.length.y.left  : NULL
## $ axis.ticks.length.y.right : NULL
## $ axis.ticks.length.theta : NULL
## $ axis.ticks.length.r       : NULL
## $ axis.minor.ticks.length : 'rel' num 0.75
## $ axis.minor.ticks.length.x : NULL
## $ axis.minor.ticks.length.x.top : NULL
## $ axis.minor.ticks.length.x.bottom : NULL
## $ axis.minor.ticks.length.y   : NULL
## $ axis.minor.ticks.length.y.left  : NULL
## $ axis.minor.ticks.length.y.right : NULL
## $ axis.minor.ticks.length.theta : NULL
## $ axis.minor.ticks.length.r       : NULL
## $ axis.line      : list()
## ...- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ axis.line.x      : NULL
## $ axis.line.x.top    : NULL
## $ axis.line.x.bottom  : NULL
## $ axis.line.y      : NULL
## $ axis.line.y.left   : NULL
## $ axis.line.y.right  : NULL
## $ axis.line.theta   : NULL

```

```

## $ axis.line.r : NULL
## $ legend.background : list()
## ..- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ legend.margin : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
## ..- attr(*, "unit")= int 8
## $ legend.spacing : 'simpleUnit' num 11points
## ..- attr(*, "unit")= int 8
## $ legend.spacing.x : NULL
## $ legend.spacing.y : NULL
## $ legend.key : list()
## ..- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ legend.key.size : 'simpleUnit' num 1.2lines
## ..- attr(*, "unit")= int 3
## $ legend.key.height : NULL
## $ legend.key.width : NULL
## $ legend.key.spacing : 'simpleUnit' num 5.5points
## ..- attr(*, "unit")= int 8
## $ legend.key.spacing.x : NULL
## $ legend.key.spacing.y : NULL
## $ legend.frame : NULL
## $ legend.ticks : NULL
## $ legend.ticks.length : 'rel' num 0.2
## $ legend.axis.line : NULL
## $ legend.text :List of 11
##   ..$ family : NULL
##   ..$ face : NULL
##   ..$ colour : NULL
##   ..$ size : 'rel' num 0.8
##   ..$ hjust : NULL
##   ..$ vjust : NULL
##   ..$ angle : NULL
##   ..$ lineheight : NULL
##   ..$ margin : NULL
##   ..$ debug : NULL
##   ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ legend.text.position : NULL
## $ legend.title :List of 11
##   ..$ family : NULL
##   ..$ face : NULL
##   ..$ colour : NULL
##   ..$ size : NULL
##   ..$ hjust : num 0
##   ..$ vjust : NULL
##   ..$ angle : NULL
##   ..$ lineheight : NULL
##   ..$ margin : NULL
##   ..$ debug : NULL
##   ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ legend.title.position : NULL
## $ legend.position : chr "right"
## $ legend.position.inside : NULL
## $ legend.direction : NULL

```

```

## $ legend.byrow           : NULL
## $ legend.justification   : chr "center"
## $ legend.justification.top: NULL
## $ legend.justification.bottom: NULL
## $ legend.justification.left: NULL
## $ legend.justification.right: NULL
## $ legend.justification.inside: NULL
## $ legend.location         : NULL
## $ legend.box               : NULL
## $ legend.box.just          : NULL
## $ legend.box.margin        : 'margin' num [1:4] 0cm 0cm 0cm 0cm
## ..- attr(*, "unit")= int 1
## $ legend.box.background     : list()
## ..- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ legend.box.spacing        : 'simpleUnit' num 11points
## ..- attr(*, "unit")= int 8
## [list output truncated]
## - attr(*, "class")= chr [1:2] "theme" "gg"
## - attr(*, "complete")= logi TRUE
## - attr(*, "validate")= logi TRUE

```

```
# Create boxplots for diversity:
```

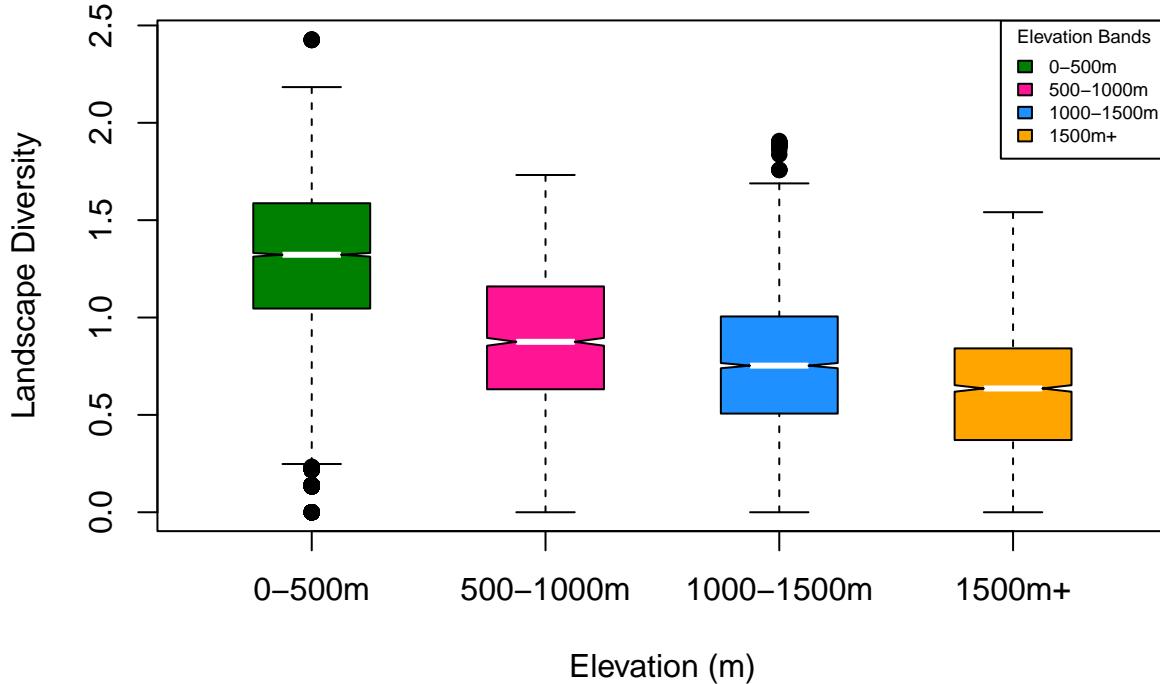
```

# Boxplot comparing landscape diversity for elevation bands
boxplot(landscape_div~ElevationBand, data=buffers_500m, main="Landscape Diversity at Elevation Bands",
        xlab="Elevation (m)", ylab="Landscape Diversity",
        col = main_colors, border = "black", notch = TRUE, notchwidth = 0.5,
        medcol = "white", whiskcol = "black", boxwex = 0.5, outpch = 19,
        outcol = "black")

# Legend
legend("topright", legend = unique(buffers_500m$ElevationBand),
       fill = main_colors, border = "black", title = "Elevation Bands",
       cex = 0.6) # cex to control font size

```

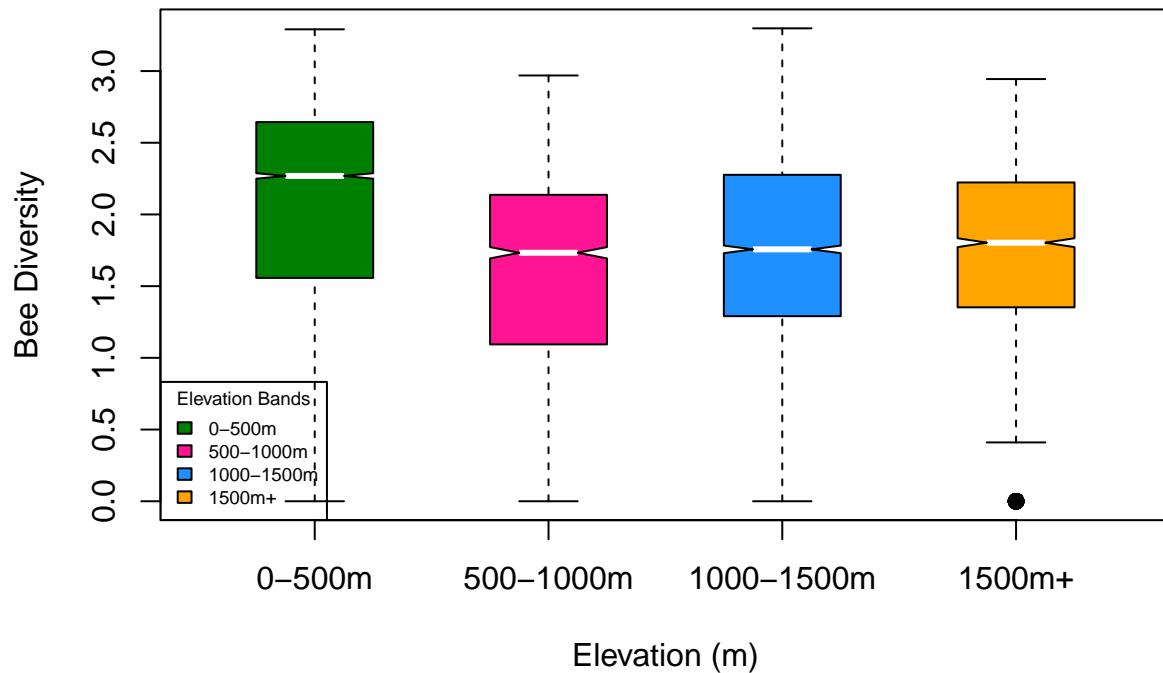
## Landscape Diversity at Elevation Bands



```
# Boxplot comparing bee diversity for elevation bands
boxplot(bee_diversity~ElevationBand, data=buffers_500m, main="Bee Diversity at Elevation Bands",
        xlab="Elevation (m)", ylab="Bee Diversity",
        col = main_colors, border = "black", notch = TRUE, notchwidth = 0.5,
        medcol = "white", whiskcol = "black", boxwex = 0.5, outpch = 19,
        outcol = "black")

# Legend
legend("bottomleft", legend = unique(buffers_500m$ElevationBand),
       fill = main_colors, border = "black", title = "Elevation Bands", cex = 0.6)
```

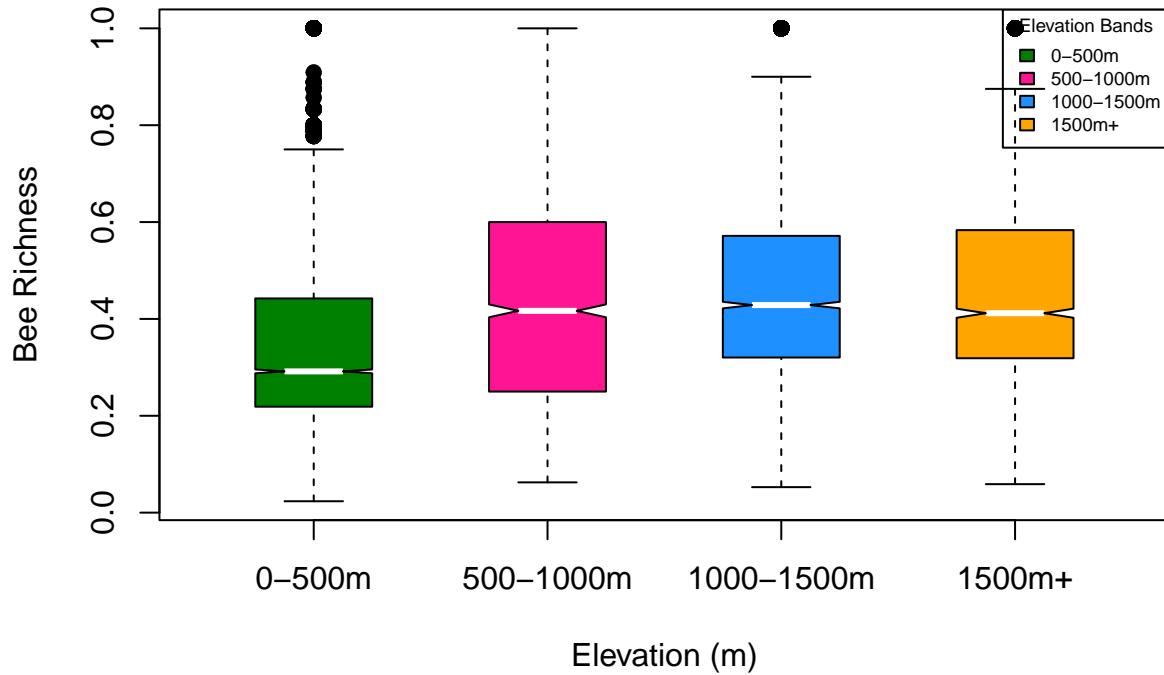
## Bee Diversity at Elevation Bands



```
# Boxplot comparing bee richness for elevation bands
boxplot(bee_rich~ElevationBand, data=buffers_500m, main="Bee Richness at Elevation Bands",
        xlab="Elevation (m)", ylab="Bee Richness",
        col = main_colors, border = "black", notch = TRUE, notchwidth = 0.5,
        medcol = "white", whiskcol = "black", boxwex = 0.5, outpch = 19,
        outcol = "black")

# Legend
legend("topright", legend = unique(buffers_500m$ElevationBand),
       fill = main_colors, border = "black", title = "Elevation Bands", cex = 0.6)
```

## Bee Richness at Elevation Bands



## Summary stats for boxplots

```

# Summary stats for Landscape Diversity by Elevation Band
landscape_stats <- aggregate(landscape_div ~ ElevationBand, data = buffers_500m,
                                FUN = function(x) c(mean = mean(x), median = median(x),
                                min = min(x), max = max(x),
                                IQR = IQR(x)))

# Unnest the results for better readability
landscape_stats <- do.call(data.frame, landscape_stats)

# Summary stats for Bee Diversity by Elevation Band
bee_diversity_stats <- aggregate(bee_diversity ~ ElevationBand, data = buffers_500m,
                                  FUN = function(x) c(mean = mean(x), median = median(x),
                                  min = min(x), max = max(x),
                                  IQR = IQR(x)))

bee_diversity_stats <- do.call(data.frame, bee_diversity_stats)

# Summary stats for Bee Richness by Elevation Band
bee_richness_stats <- aggregate(bee_rich ~ ElevationBand, data = buffers_500m,
                                 FUN = function(x) c(mean = mean(x), median = median(x),
                                 min = min(x), max = max(x),
                                 IQR = IQR(x)))

bee_richness_stats <- do.call(data.frame, bee_richness_stats)

```

```
# Print the results  
landscape_stats
```

```
##   ElevationBand landscape_div.mean landscape_div.median landscape_div.min  
## 1      0-500m        1.2699788        1.3222984          0  
## 2     500-1000m       0.8536870        0.8752772          0  
## 3    1000-1500m       0.7625931        0.7529747          0  
## 4    1500m+          0.6441310        0.6356987          0  
##   landscape_div.max landscape_div.IQR  
## 1        2.428611        0.5407765  
## 2        1.732021        0.5281419  
## 3        1.906668        0.4986522  
## 4        1.540910        0.4707665
```

```
bee_diversity_stats
```

```
##   ElevationBand bee_diversity.mean bee_diversity.median bee_diversity.min  
## 1      0-500m        2.055283        2.268511          0  
## 2     500-1000m       1.638324        1.732868          0  
## 3    1000-1500m       1.779727        1.756537          0  
## 4    1500m+          1.792219        1.803033          0  
##   bee_diversity.max bee_diversity.IQR  
## 1        3.291037        1.0866470  
## 2        2.969123        1.0430794  
## 3        3.297823        0.9856601  
## 4        2.944265        0.8709497
```

```
bee_richness_stats
```

```
##   ElevationBand bee_rich.mean bee_rich.median bee_rich.min bee_rich.max  
## 1      0-500m        0.3466653        0.2916667        0.02352941        1  
## 2     500-1000m       0.4535867        0.4166667        0.06250000        1  
## 3    1000-1500m       0.4716089        0.4285714        0.05263158        1  
## 4    1500m+          0.4634356        0.4117647        0.05882353        1  
##   bee_rich.IQR  
## 1        0.2235577  
## 2        0.3500000  
## 3        0.2511161  
## 4        0.2644928
```

Test the hypothesis that the relationship between landscape diversity and bee species richness and diversity varies significantly across different elevational bands

```
# Summarize species richness and diversity by elevation band  
summarized_by_elevation <- buffers_500m %>%  
  group_by(ElevationBand) %>%
```

```

summarise(
  mean_richness = mean(normalized_richness, na.rm = TRUE), # Mean of normalized richness
  median_richness = median(normalized_richness, na.rm = TRUE), # Median of normalized richness
  mean_diversity = mean(bee_diversity, na.rm = TRUE), # Mean of bee diversity
  median_diversity = median(bee_diversity, na.rm = TRUE), # Median of bee diversity
  # Optionally add other summary statistics like SD, count, etc.
  sd_richness = sd(normalized_richness, na.rm = TRUE), # Standard deviation of richness
  sd_diversity = sd(bee_diversity, na.rm = TRUE) # Standard deviation of diversity
)

# View the summarized data
print(summarized_by_elevation)

## Simple feature collection with 4 features and 7 fields
## Geometry type: MULTIPOLYGON
## Dimension: XY
## Bounding box: xmin: 251817.9 ymin: 100065.8 xmax: 2274776 ymax: 1627662
## Projected CRS: NAD83 / Oregon GIC Lambert (ft)
## # A tibble: 4 x 8
##   ElevationBand mean_richness median_richness mean_diversity median_diversity
##   <fct>          <dbl>        <dbl>        <dbl>        <dbl>
## 1 0-500m         0.347       0.292       2.06        2.27
## 2 500-1000m     0.454       0.417       1.64        1.73
## 3 1000-1500m    0.472       0.429       1.78        1.76
## 4 1500m+         0.463       0.412       1.79        1.80
## # i 3 more variables: sd_richness <dbl>, sd_diversity <dbl>,
## #   geometry <MULTIPOLYGON [foot]>

# ANOVA for normalized richness across elevation bands
anova_richness <- aov(normalized_richness ~ landscape_div * ElevationBand, data = buffers_500m)
summary(anova_richness)

##                                     Df Sum Sq Mean Sq F value Pr(>F)
## landscape_div                      1   17.5  17.453  406.77 <2e-16 ***
## ElevationBand                       3   35.9  11.975  279.10 <2e-16 ***
## landscape_div:ElevationBand        3   11.4   3.806   88.69 <2e-16 ***
## Residuals                           15150  650.0   0.043
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# ANOVA for bee diversity across elevation bands
anova_diversity <- aov(bee_diversity ~ landscape_div * ElevationBand, data = buffers_500m)
summary(anova_diversity)

##                                     Df Sum Sq Mean Sq F value Pr(>F)
## landscape_div                      1   142   142.41  249.32 <2e-16 ***
## ElevationBand                       3   249    82.88  145.10 <2e-16 ***
## landscape_div:ElevationBand        3   131    43.61   76.34 <2e-16 ***
## Residuals                           15150   8654    0.57
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

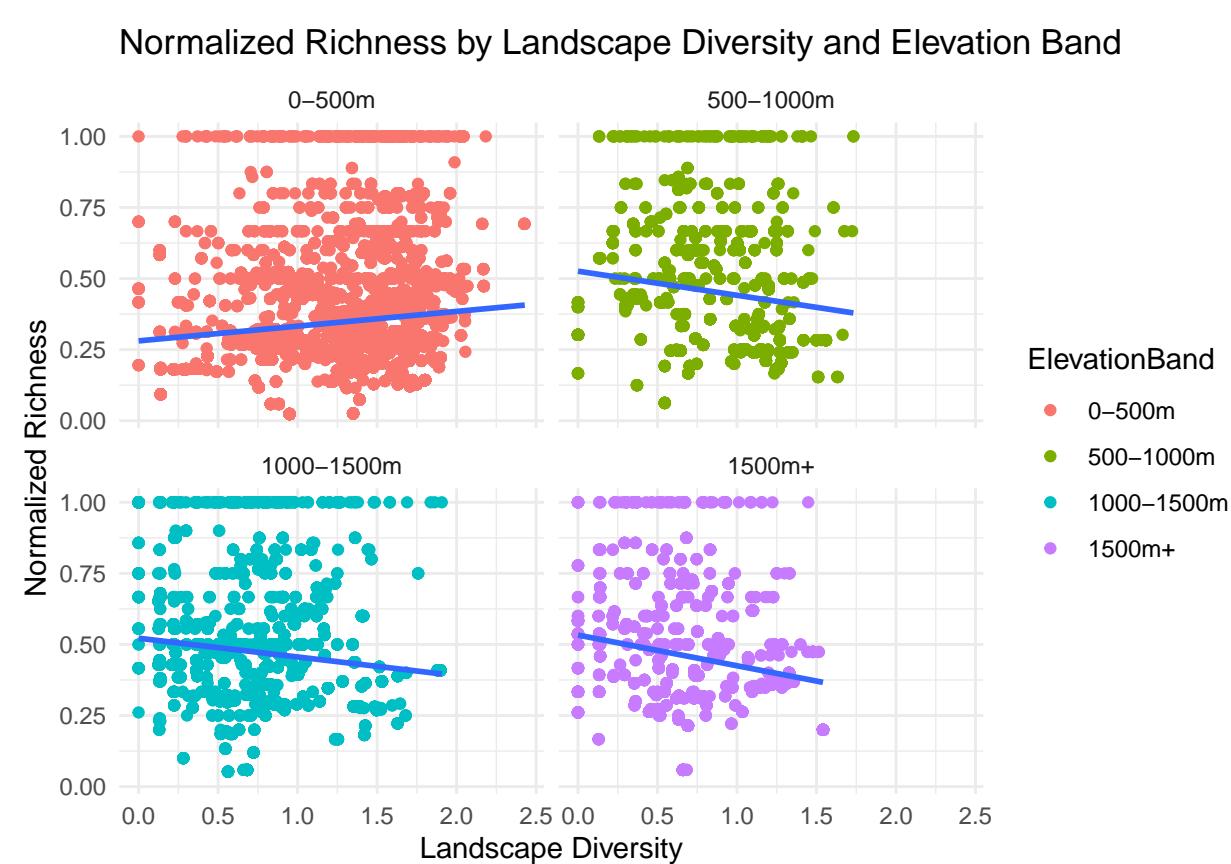
```

# The ANOVA results indicate that the relationship between landscape diversity and bee richness as well

# Facet grid plot for normalized richness
ggplot(buffers_500m, aes(x = landscape_div, y = normalized_richness)) +
  geom_point(aes(color = ElevationBand)) +
  geom_smooth(method = "lm", aes(group = ElevationBand), se = FALSE) +
  facet_wrap(~ ElevationBand) +
  labs(title = "Normalized Richness by Landscape Diversity and Elevation Band",
       x = "Landscape Diversity", y = "Normalized Richness") +
  theme_minimal()

## `geom_smooth()` using formula = 'y ~ x'

```



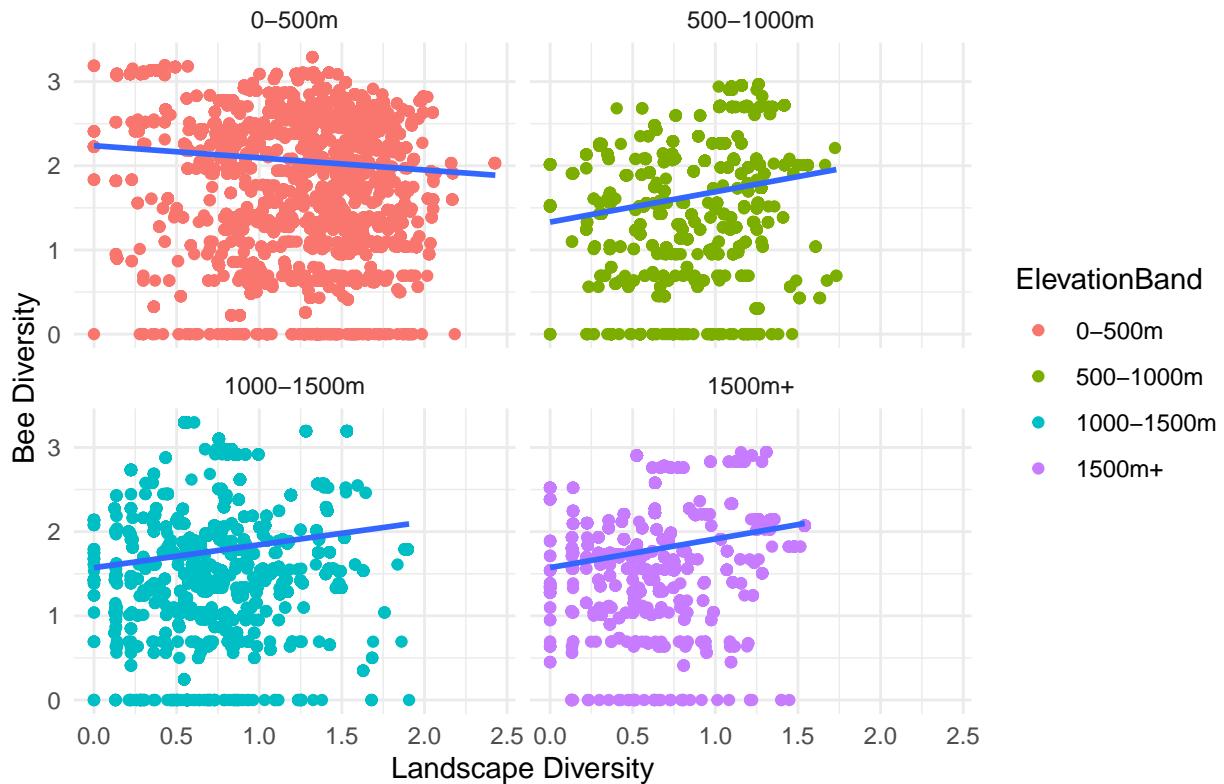
```

# Facet grid plot for bee diversity
ggplot(buffers_500m, aes(x = landscape_div, y = bee_diversity)) +
  geom_point(aes(color = ElevationBand)) +
  geom_smooth(method = "lm", aes(group = ElevationBand), se = FALSE) +
  facet_wrap(~ ElevationBand) +
  labs(title = "Bee Diversity by Landscape Diversity and Elevation Band",
       x = "Landscape Diversity", y = "Bee Diversity") +
  theme_minimal()

## `geom_smooth()` using formula = 'y ~ x'

```

## Bee Diversity by Landscape Diversity and Elevation Band



**Conclusion:** summarize findings and address original hypotheses

### Landscape Diversity in Relation to Bee Diversity

There is a statistically significant positive relationship between bee diversity and landscape diversity. However, the effect size is small (low  $R^2$ ), meaning landscape diversity explains very little of the variation in bee diversity.

### Landscape Diversity in Relation to Bee Richness

There is a statistically significant negative relationship between bee richness and landscape diversity. However, the effect size is small (low  $R^2$ ), meaning landscape diversity explains very little of the variation in bee richness.

### Comparisons

- **Effect Size:** The slope is stronger for bee diversity than bee richness (absolute value of coefficients: 0.0969 vs 0.0339).
- **Model Fit:** The bee richness model explains slightly more variability ( $R^2 = 0.02442$ ) than the bee diversity model ( $R^2 = 0.01552$ ).
- **Residual Standard Error:** The errors are smaller in the bee richness model (0.2145) compared to bee diversity (0.772).

## Variance by Elevation of Landscape Diversity's Relation to Bee Diversity and Bee Richness

Landscape diversity and elevation band both significantly predict the response variable (bee diversity or richness), as indicated by the low p-values ( $< 2 \times 10^{-16}$ ) and large F-values.

The significant interaction between these two predictors (landscape diversity and elevation band) suggests that the relationship between landscape diversity and the response variables (bee richness and diversity) varies across elevation bands.

The high residual variance, indicated by the residual sum of squares, illustrates that a large portion of the variance is not explained by the model and that other variables are influencing the response variables.

## Secondary Relationships

- **Mean Landscape Diversity:** Mean landscape diversity decreases as elevation increases, with the highest values in the 0 – 500m band (mean = 1.27) and the lowest in the 1500m+ band (mean = 0.64).
- **Bee Diversity:** Higher bee diversity is observed at low and high elevations (mean = 2.06 at 0 – 500m and 1.79 at 1500m+).
- **Bee Richness:** Bee richness peaks at mid-elevations, with the highest mean at 1000 – 1500m (mean = 0.47), rising from 0.35 at 0 – 500m and slightly declining to 0.46 at 1500m+.

## Limits and Considerations

Data collected by citizen scientists introduces potential sampling biases. For example, the large number of buffers housing a single data point could reflect instances where scientists only recorded specialized bees. The study compares bee diversity and richness to clarify these potential biases and observe how they manifest differently.

## Recommendations for Future Studies

A future modification could involve grouping land cover types differently, such as combining developed land categories. This could reduce sampling bias and enhance the model's explanatory power.