Practical 13: Biodiversity and Productivity

APES2039A

Dr. Joseph White

03/06/2022

Contents

1 Background

In this practical we will explore the relationships between different types of biodiversity (species richness, functional diversity, phylogenetic diversity) and productivity (Net Primary Productivity).

We will make use of two different datasets.

1) Plants of South Africa (POSA) specimen and habit dataset
2) MODIS Terra satellite's derived annual Net Primary Productivity (NPP) at 500m pixel resolution.

The POSA specimen and habit dataset is an extensive collection of plant samples with locations from all of South Africa collected over more than 200 years. These include herbarium records and verified observations amongst others. Additionally, this dataset has information on the type of habit a plant has (e.g. tree, shrub, herb), which we will use as our functional type data.

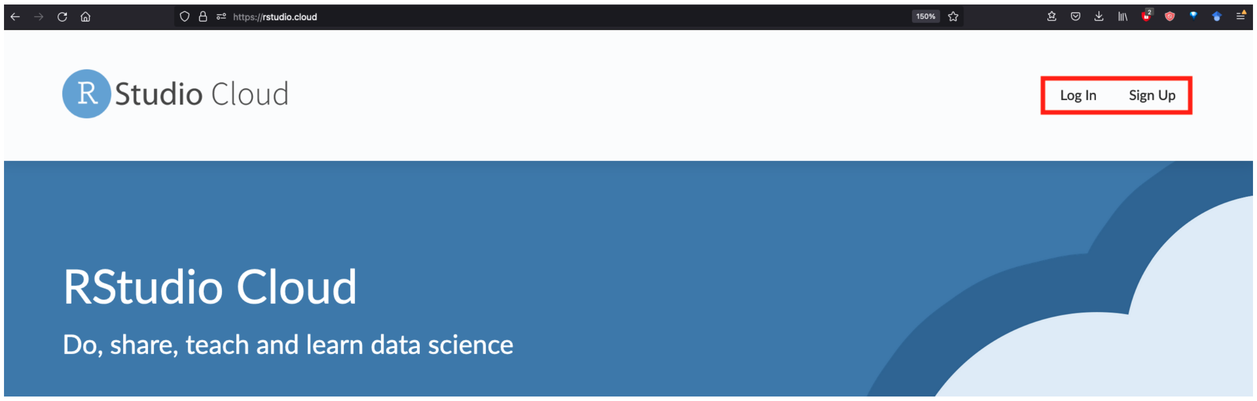
The MODIS NPP dataset provides annual derived productivity values across the globe. The dataset represents the average NPP values from 2001-2021.

The data for both POSA and NPP is subsetted to only include values for the grassland biome of South Africa. These are linked to a location across the country in the form of a grid system referred to as a Quarter Degree Square (QDS). Each grid cell is roughly 30 x 30 km in size.

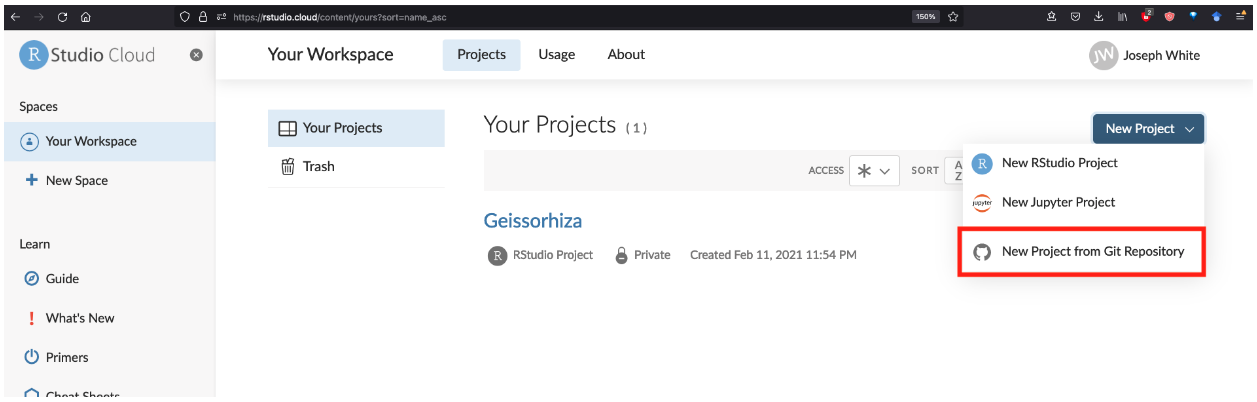
This document provides the code and output (summary values, maps, figures and tables) that you will work from to extract the relevant answers. All of the answers you require for your Practical 13 worksheet submission will be found in **this document**, your lecture notes from **Week 14 Lecture 3 Functional consequences of biodiversity** and **your own experience and understanding** of sampling field data.

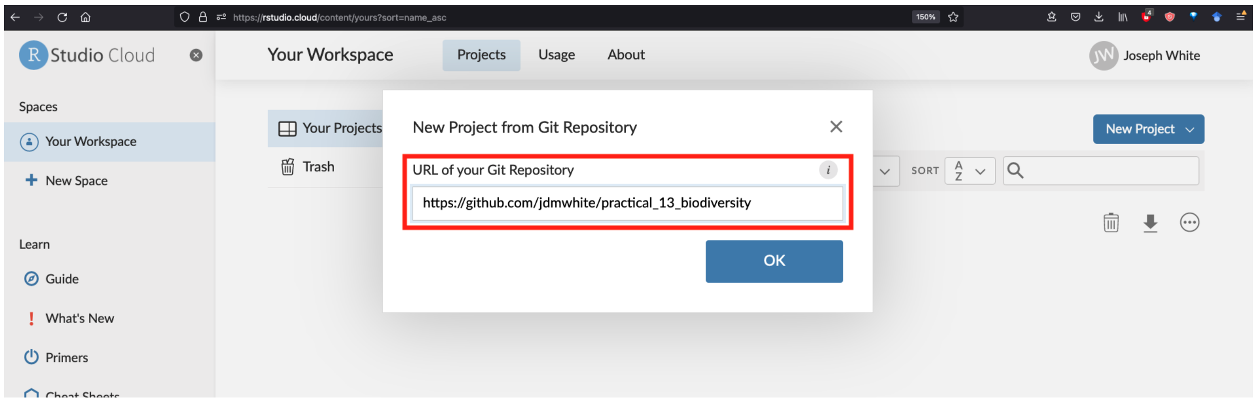
2 If you would like to follow along and run the code yourself

Go to R Studio Cloud. Either **Sign Up** or as you have already run a practical here before, **Log In** to your account.



Once you have logged in, click on **New Project** in the top right corner. Select **New Project from Git Repository**. Copy and paste this link into the open space: https://github.com/jdmwhite/practical_13_biodiversity



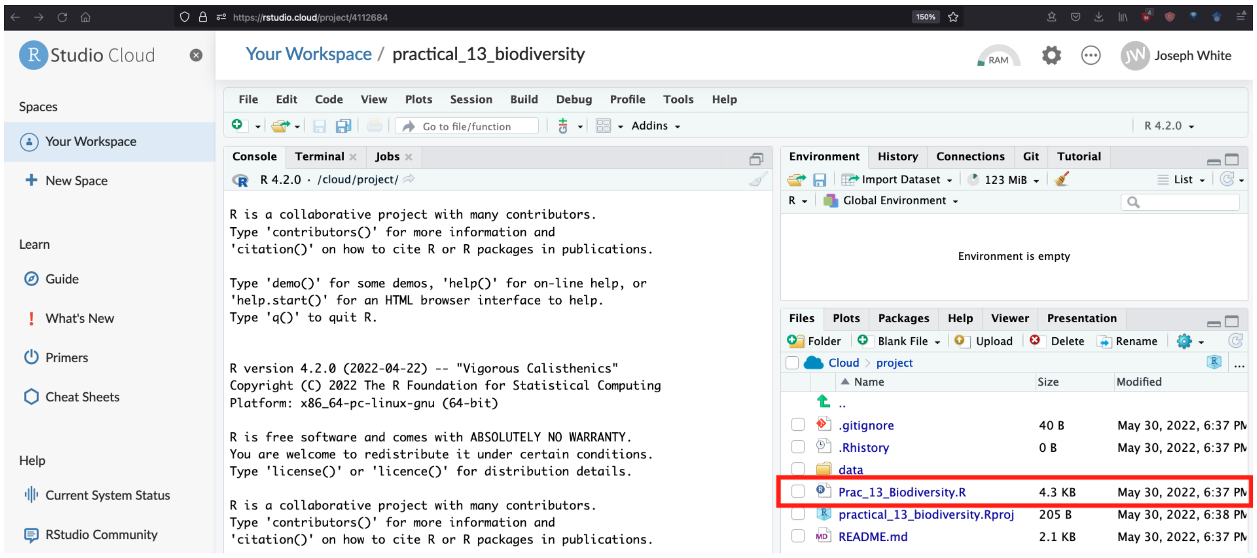


New Project from Git Repository

URL of your Git Repository: https://github.com/jdmwhite/practical_13_biodiversity

OK

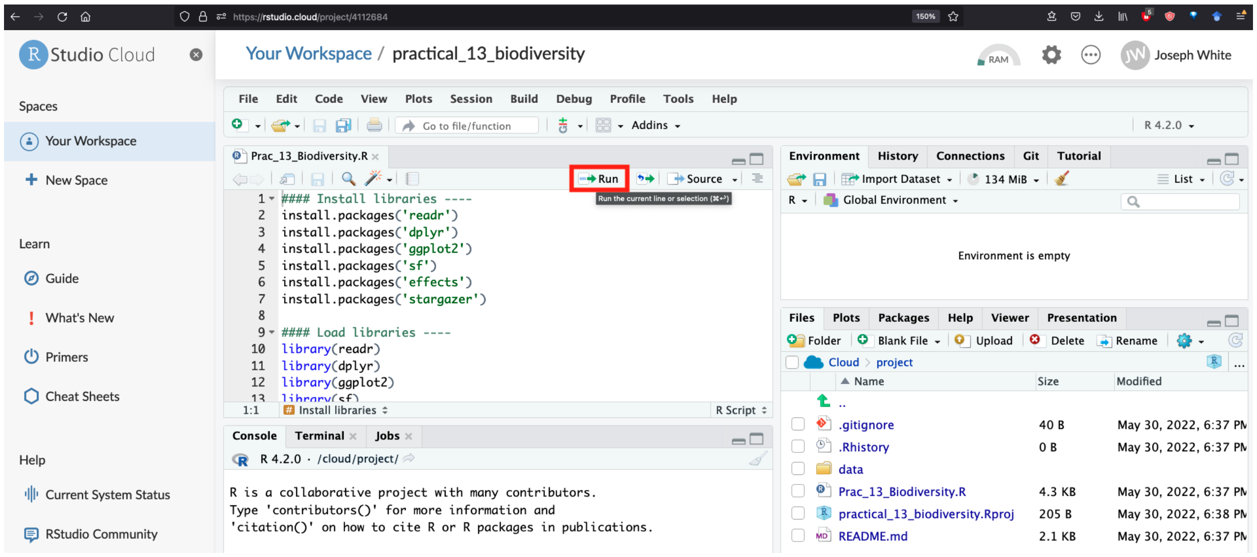
Wait for the project to deploy (this may take a couple of minutes). Once it has opened, click on the `Prac_13_Biodiversity.R` file.


https://rstudio.cloud/project/4112684
Studio Cloud
Your Workspace / practical_13_biodiversity
RAM
Joseph White
Spaces
Your Workspace
New Space
Learn
Guide
What's New
Primers
Cheat Sheets
Help
Current System Status
RStudio Community
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
R 4.2.0
Console Terminal Jobs
Environment History Connections Git Tutorial
Import Dataset
123 MiB
List
Global Environment
Environment is empty
Files Plots Packages Help Viewer Presentation
Folder Blank File Upload Delete Rename
Cloud > project

```
R 4.2.0 · /cloud/project/

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.


R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
```

Name	Size	Modified
..		
.gitignore	40 B	May 30, 2022, 6:37 PM
.Rhistory	0 B	May 30, 2022, 6:37 PM
data		
Prac_13_Biodiversity.R	4.3 KB	May 30, 2022, 6:37 PM
practical_13_biodiversity.Rproj	205 B	May 30, 2022, 6:38 PM
README.md	2.1 KB	May 30, 2022, 6:37 PM

You can now run the code, either using (Cntrl + ENTER) or click the **Run** button in the top middle of your screen.



```r
#### Install libraries ----
install.packages('readr')
install.packages('dplyr')
install.packages('ggplot2')
install.packages('sf')
install.packages('effects')
install.packages('stargazer')

#### Load libraries ----
library(readr)
library(dplyr)
library(ggplot2)
library(sf)
```

```
R 4.2.0 · /cloud/project/

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
```

Name	Size	Modified
..		
.gitignore	40 B	May 30, 2022, 6:37 PM
.Rhistory	0 B	May 30, 2022, 6:37 PM
data		
Prac_13_Biodiversity.R	4.3 KB	May 30, 2022, 6:37 PM
practical_13_biodiversity.Rproj	205 B	May 30, 2022, 6:38 PM
README.md	2.1 KB	May 30, 2022, 6:37 PM

Make sure to **LOG OUT** of your R Studio Cloud session when you are finished using the platform. You are granted 25 free hours per month. But these hours will be depleted quickly if you don't log out!

Let's now take a look at the code:

2.1 Install and load libraries

The first step in most coding languages is to install and load the libraries (normally called `packages`). These packages have special functions, which users created for specific tasks. The huge variety of R packages are one of the major reasons R is so popular with the ecology community.

```
#### Load libraries ----
library(readr)
library(dplyr)
library(ggplot2)
library(sf)
library(effects)
library(stargazer)
```

Now we will load in the data. The POSA data is stored in a csv file. This file type is similar to an excel file, but is more efficient in its use of space.

We will also load in the QDS and covariate dataset. This is a **shapefile**, which is a *spatial* file format. It is a table, with each row representing one grid cell (QDS). The columns include variables associated with each QDS. Then there is a special column called **geometry**. This represents the *spatial* component of the table, and tells R where each polygon or point is found in space.

We will also load the South African border **shapefile** for our maps.

```
#### Load in data ----
# POSA richness
posa_rich <- read_csv('data/posa_richness.csv')

# QDS and covariates
qds_cov <- read_sf('data/grass_qds_cov.shp')

# Map of SA
sa <- read_sf('data/SA_border/SA_border.shp')
```

2.2 Explore the data

2.2.1 POSA data

Now that the data is loaded into our **working environment**, let's take a look at it:

```
#### Explore the data ----
#### Plants of SA data ----

# Take a look at the data
posa_rich
```

```
## # A tibble: 86,859 x 7
##    QDS    Taxon        Family func_type spp_rich func_diversity phylog_diversity
##    <chr>  <chr>         <chr>  <chr>         <dbl>          <dbl>            <dbl>
##  1 3030CA Melica race~ Poace~ graminoid       731             24              139
##  2 2528CC Themeda tri~ Poace~ graminoid       972             24              139
##  3 3226DB Indigofera ~ Fabac~ herb           1110             23              165
##  4 2731CB Pavetta ede~ Rubia~ shrub           449             18              100
##  5 2831CD Miscanthus ~ Poace~ graminoid       990             24              182
##  6 2930CB Asterella b~ Ayton~ bryophyte      1578             26              219
##  7 3029CB Eragrostis ~ Poace~ graminoid       574             20               98
##  8 2926AA Paspalum di~ Poace~ graminoid       605             18               96
##  9 3128AA Gladiolus l~ Irida~ geophyte        631             19              101
## 10 2930CB Schoenoplec~ Cyper~ cyperoid       1578             26              219
## # ... with 86,849 more rows
```

As we can see it is a big dataset! 86859 rows and 7 columns. By default R only shows us the first 10 rows. We have information about the QDS (the grid cell the species is found in), the taxon name, the plant family, the functional type, and then summary information for each QDS on the species richness, functional diversity and phylogenetic diversity.

Next, let's count what the most common species across all of the QDSs is. We will use the `count()` function and `arrange()` function. `count()` tells us how many times each species name is found. `arrange()` then sorts this for us. We use a negative sign to tell R we would like the result in *descending* order.

```
# What is the species found in most Quarter Degree Square (QDS)?
posa_rich %>% count(Taxon) %>% arrange(-n)
```

```
## # A tibble: 7,278 x 2
##    Taxon                       n
##    <chr>                   <int>
##  1 Eragrostis curvula        281
##  2 Helichrysum nudifolium    155
##  3 Diospyros lycioides       147
##  4 Searsia pyroides          147
##  5 Themeda triandra          147
##  6 Commelina africana        134
##  7 Setaria sphacelata        134
##  8 Scabiosa columbaria       132
##  9 Digitaria eriantha        131
## 10 Helichrysum rugulosum     124
## # ... with 7,268 more rows
```

We will now continue to look at different summary metrics for our POSA dataset.

How many different plant families are there?

```
# How many different plant families are there?
n_distinct(posa_rich$Family)
```

```
## [1] 319
```

What are the different functional types? To make R print out more rows, we specify `n = 29` within the `print` function.

```r
# What different functional types are there?
posa_rich %>% distinct(func_type) -> functional_types
print(functional_types, n = 29)
```

```
## # A tibble: 29 x 1
##    func_type
##    <chr>
##  1 graminoid
##  2 herb
##  3 shrub
##  4 bryophyte
##  5 geophyte
##  6 cyperoid
##  7 tree
##  8 helophyte
##  9 scrambler
## 10 succulent
## 11 dwarf shrub
## 12 emergent hydrophyte
## 13 climber
## 14 mesophyte
## 15 suffrutex
## 16 hydrophyte
## 17 parasite
## 18 epiphyte
## 19 lithophyte
## 20 carnivore
## 21 tenagophyte
## 22 epihydate
## 23 pleustophyte
## 24 hyperhydate
## 25 restioid
## 26 sudd hydrophyte
## 27 creeper
## 28 haptophyte
## 29 lichen
```

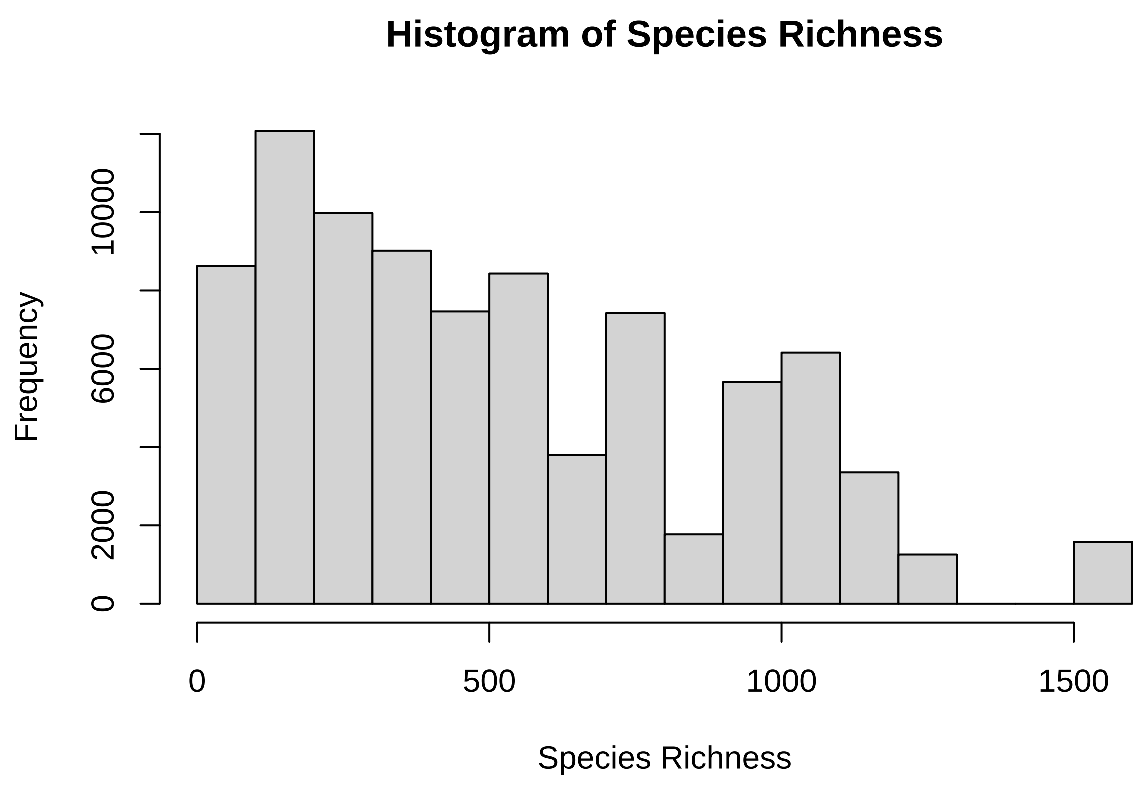
What is the most common functional type? Again, we use the `count()` and `arrange()` functions.

```r
# What is the most common functional type?
posa_rich %>% count(func_type) %>% arrange(-n)
```

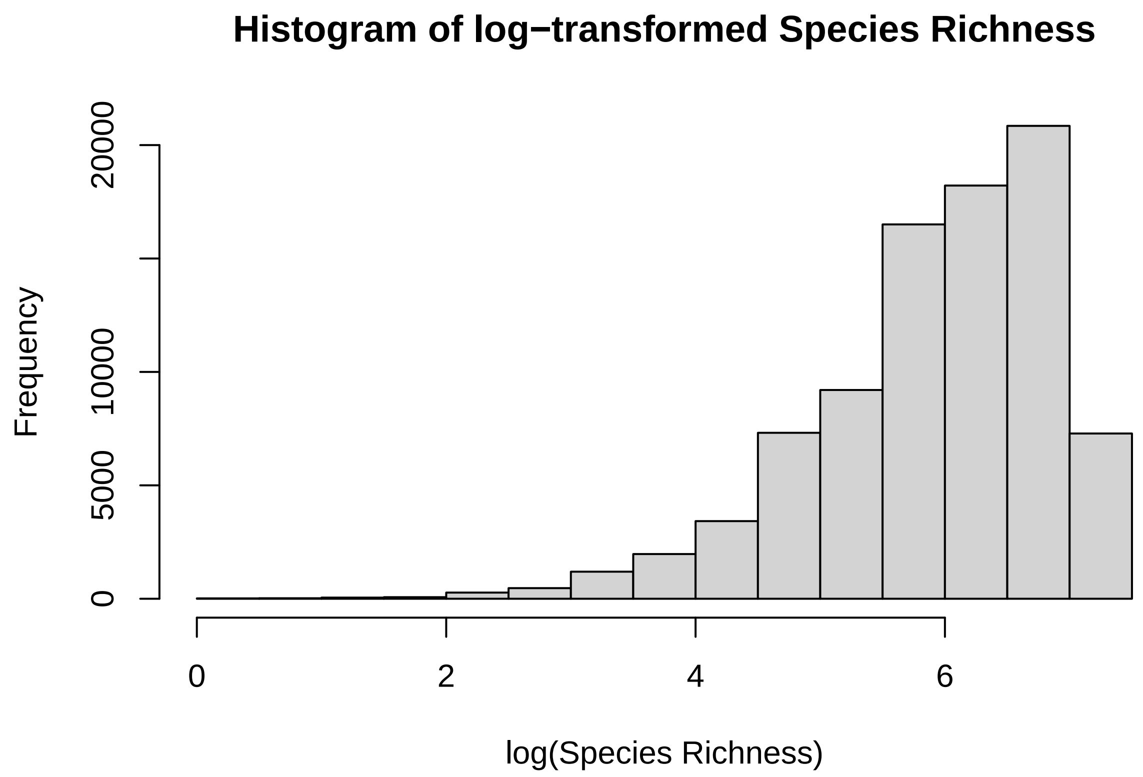
```
## # A tibble: 29 x 2
##    func_type       n
##    <chr>       <int>
##  1 herb        29910
##  2 shrub       10638
##  3 graminoid    9766
##  4 geophyte     7609
##  5 succulent    7006
##  6 dwarf shrub  4369
##  7 bryophyte    2768
##  8 tree          2611
##  9 climber       2572
## 10 lithophyte    2171
## # ... with 19 more rows
```

Before we do any analysis of our data, it is important to consider whether our data is **normally distributed** or not. This is due to the *assumptions* that some models, such as linear regression make. Your input data does not necessarily need to be normally distributed, but rather your prediction error (or residuals). If you want to learn more on this, look HERE. We will log-transform our species richness data to use later in our regression analysis.

```r
# Is the richness data normally distributed?
hist(posa_rich$spp_rich, xlab = 'Species Richness',
     main = 'Histogram of Species Richness')
```



```r
hist(log(posa_rich$spp_rich), xlab = 'log(Species Richness)',
     main = 'Histogram of log-transformed Species Richness')
```



2.2.2 NPP data

Now we have an idea of what's inside our POSA data. Let's look at our Net Primary Productivity data, which is stored in the QDS shapefile:

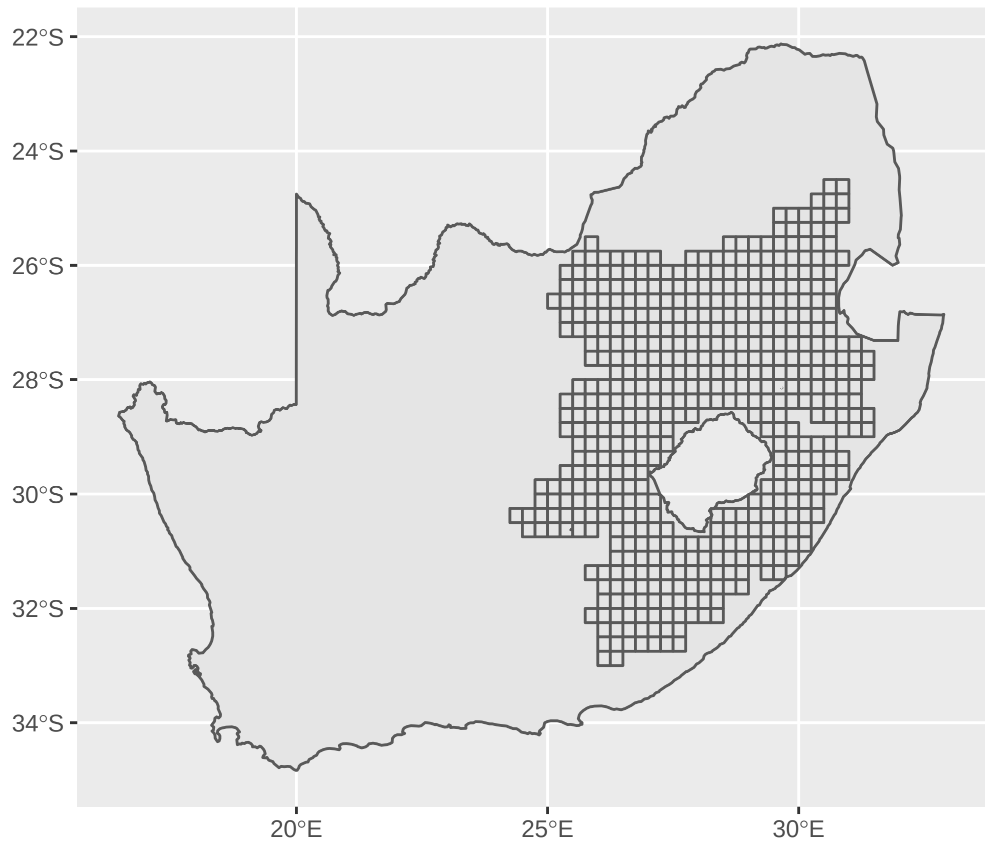
```
#### Spatial data (QDS) ----
qds_cov
```

```
## Simple feature collection with 510 features and 5 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 24.25 ymin: -33 xmax: 31.5 ymax: -24.5
## Geodetic CRS:  WGS 84
## # A tibble: 510 x 6
##    QDS      MAP Elev_STD   NPP NPP_var                                 geometry
##    <chr>  <dbl>    <dbl> <dbl>   <dbl>                            <POLYGON [°]>
##  1 2430CD  816.    212.  0.878  0.0974 ((30.5 -25, 30.41667 -25, 30.33333 -25, ~
##  2 2430DA  933.    179.  0.806  0.106  ((30.75 -24.75, 30.66667 -24.75, 30.6347~
##  3 2430DB  928.    315.  1.07   0.117  ((31 -24.75, 30.91667 -24.75, 30.83333 -~
##  4 2430DC  967.    210.  0.983  0.0926 ((30.75 -25, 30.66667 -25, 30.58333 -25,~
##  5 2430DD 1035.    268.  1.49   0.113  ((31 -25, 30.91667 -25, 30.83333 -25, 30~
##  6 2525DB  568.     31.3 0.402  0.0762 ((26 -25.75, 25.91667 -25.75, 25.83333 -~
##  7 2525DC  542.     35.8 0.330  0.0749 ((25.75 -26, 25.66667 -26, 25.58333 -26,~
##  8 2525DD  579.     36.4 0.412  0.0741 ((26 -26, 25.91667 -26, 25.83333 -26, 25~
##  9 2526CC  618.     28.2 0.435  0.0636 ((26.25 -26, 26.16667 -26, 26.08333 -26,~
## 10 2526CD  647.     44.2 0.445  0.0667 ((26.5 -26, 26.41667 -26, 26.33333 -26, ~
## # ... with 500 more rows
```

There are 510 different grid cells, each with its own associated covariate data (i.e. other columns with environmental values). You should also notice the `geometry` column, which includes the coordinates of each grid cell.

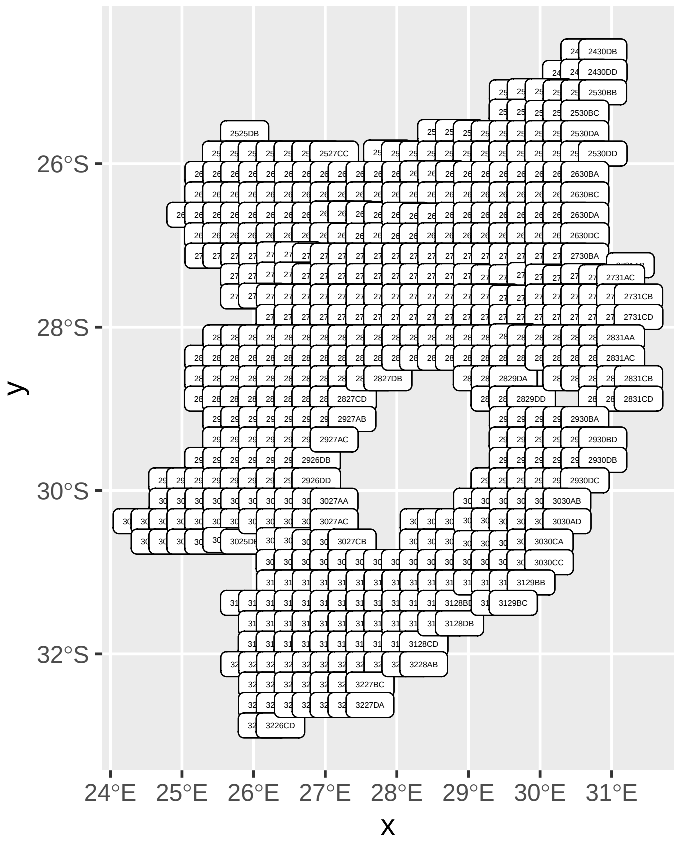
Using the flexible `ggplot2` package, we will now make some figures First, plot the QDSs on top of the South Africa border to see where our data is found within the country. We simply specify `geom_sf` (a geometry feature, plotted in `ggplot2` using the `sf` package extension). We tell each **`geom_sf` layer** which data to plot (**sa** and then **qds_cov**). The order is important. Each layer plots on top of the next. So in this case `geom_sf(data = qds_cov)` will plot on top of the `geom_sf(data = sa)` layer.

```r
# Where are our grids inside SA?
ggplot() +
  geom_sf(data = sa) +
  geom_sf(data = qds_cov)
```



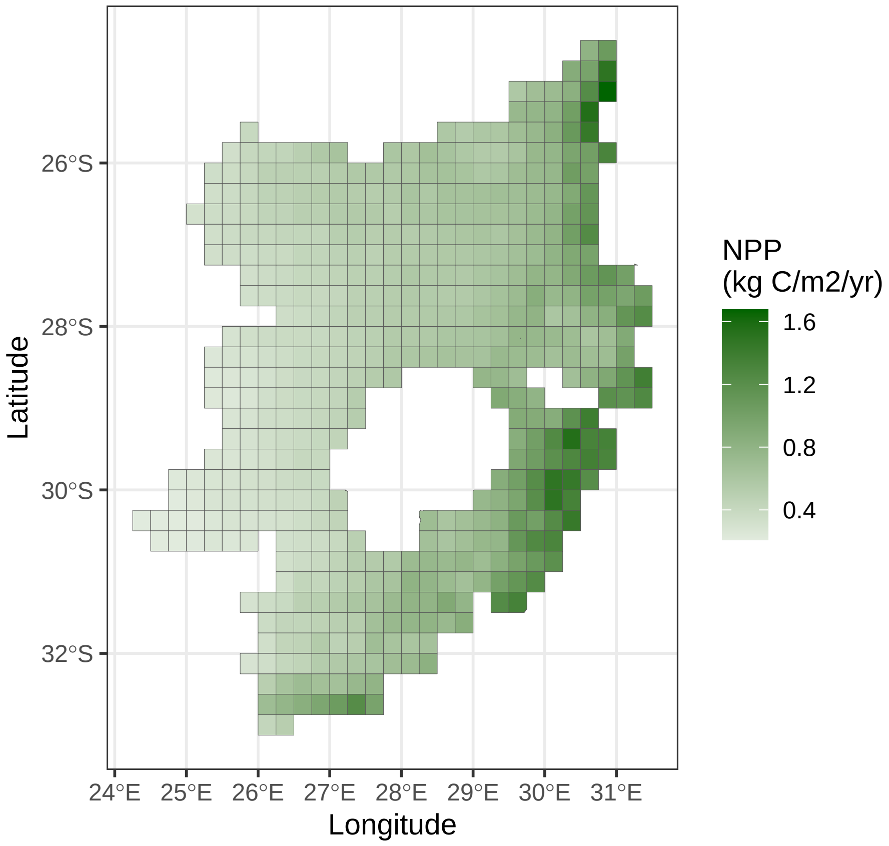
Each QDS is associated with a **unique identifying code**. The code represents the latitude and longitude of the grid and then which block this falls in to. There are 16 blocks within each combination of one degree of latitude and longitude, respectively. These are represented by letters. The top left block within each one degree is A. The smaller block within is AA. The bottom right block is DD and so on. Let's plot this to try and visualise it by labeling each QDS:

```r
# What are the labels of our grids?
ggplot() +
  geom_sf(data = qds_cov) +
  geom_sf_label(data = qds_cov, aes(label = QDS), size = 1)
```



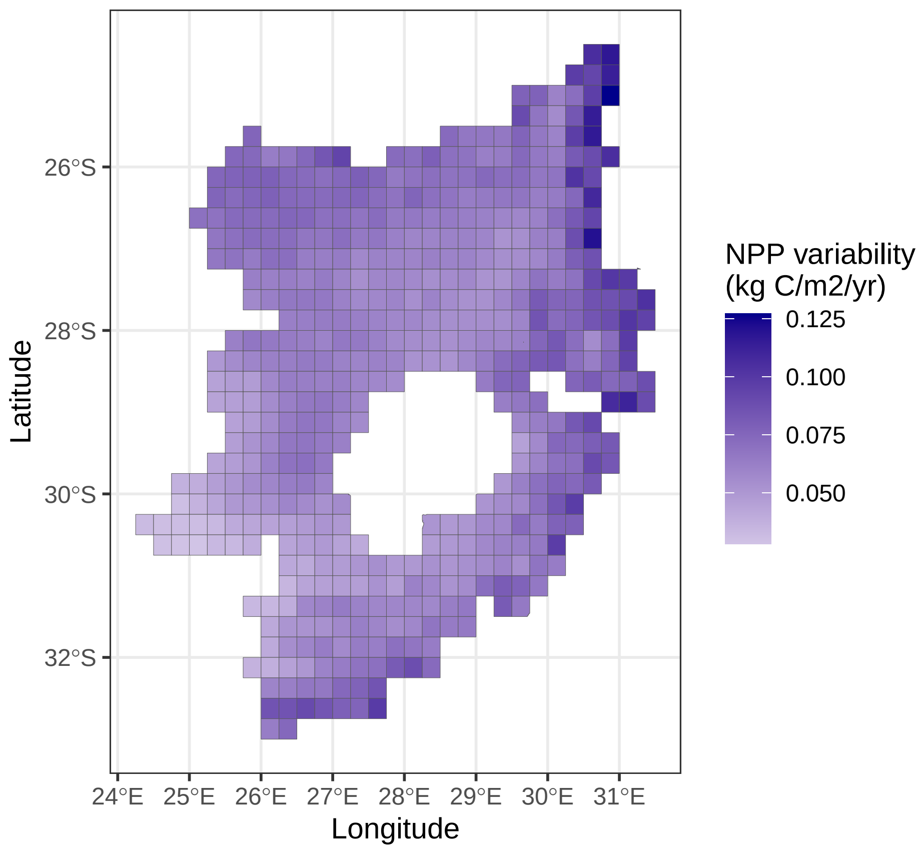
We can now visualise our Net Primary Productivity (NPP) layer. We have the core of our code. But now we will add in a mapping aesthetic (or `aes()`). In this case we want each cell to be filled by the corresponding value of NPP. Then we can chose a colour gradient to fill this in with (white to darkgreen) and give our legend a good label (NPP (kg C/m2/yr)). We can also change the x and y labels using `labs()` and lastly edit the overall theme of the plot/map using a present function called `theme_bw()`.

```
# NPP
ggplot() +
  geom_sf(data = qds_cov, aes(fill = NPP), lwd = 0) +
  scale_fill_gradient2(low = 'white', high = 'darkgreen',
                       name = 'NPP\n(kg C/m2/yr)') +
  labs(x = 'Longitude', y = 'Latitude') +
  theme_bw()
```



What spatial trends do you see in the NPP data? Let's now plot the NPP variability layer. This uses the same copy and pasted code from above. We have just changed the fill variable in `aes()`, the colour gradient (white to darkblue) and the legend label. What spatial trends do you see in the NPP variability data?

```
ggplot() +
  geom_sf(data = qds_cov, aes(fill = NPP_var), lwd = 0) +
  scale_fill_gradient2(low = 'white', high = 'darkblue',
                       name = 'NPP variability\n(kg C/m2/yr)') +
  labs(x = 'Longitude', y = 'Latitude') +
  theme_bw()
```



2.3 Clean the data

The next step is to clean the data a bit further. In the POSA data, we only want one row per QDS. To do this we run the `distinct()` function. Next we `select()` only the columns we want. Lastly, take a look at each dataset.

```
#### Clean the data and join it together ----
# only select one row for each QDS
posa_rich %>% distinct(QDS, .keep_all = TRUE) -> posa_rich

# select the columns we want
posa_rich %>% select(QDS, spp_rich, func_diversity, phylog_diversity) -> posa_rich

# join the two datasets together
# reminder of each dataset
head(posa_rich)
```

```
## # A tibble: 6 x 4
##   QDS    spp_rich func_diversity phylog_diversity
##   <chr>     <dbl>          <dbl>            <dbl>
## 1 3030CA      731             24              139
## 2 2528CC      972             24              139
## 3 3226DB     1110             23              165
## 4 2731CB      449             18              100
## 5 2831CD      990             24              182
## 6 2930CB     1578              26               219
```

```
head(qds_cov)
```

```
## Simple feature collection with 6 features and 5 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 25.75 ymin: -25.75 xmax: 31 ymax: -24.5
## Geodetic CRS:  WGS 84
## # A tibble: 6 x 6
##   QDS      MAP Elev_STD   NPP NPP_var                                  geometry
##   <chr>  <dbl>    <dbl> <dbl>   <dbl>                             <POLYGON [°]>
## 1 2430CD  816.    212.  0.878  0.0974 ((30.5 -25, 30.41667 -25, 30.33333 -25, 3~
## 2 2430DA  933.    179.  0.806  0.106  ((30.75 -24.75, 30.66667 -24.75, 30.63474~
## 3 2430DB  928.    315.  1.07   0.117  ((31 -24.75, 30.91667 -24.75, 30.83333 -2~
## 4 2430DC  967.    210.  0.983  0.0926 ((30.75 -25, 30.66667 -25, 30.58333 -25, ~
## 5 2430DD 1035.    268.  1.49   0.113  ((31 -25, 30.91667 -25, 30.83333 -25, 30.~
## 6 2525DB  568.     31.3 0.402  0.0762 ((26 -25.75, 25.91667 -25.75, 25.83333 -2~
```

2.3.1 Join the datasets

Both datasets have a column with the unique QDS identifier. We will use this to column to join them together with the `left_join()` function. Lastly, `select()` the columns we want to use and `filter()` out any columns that have no species richness data.

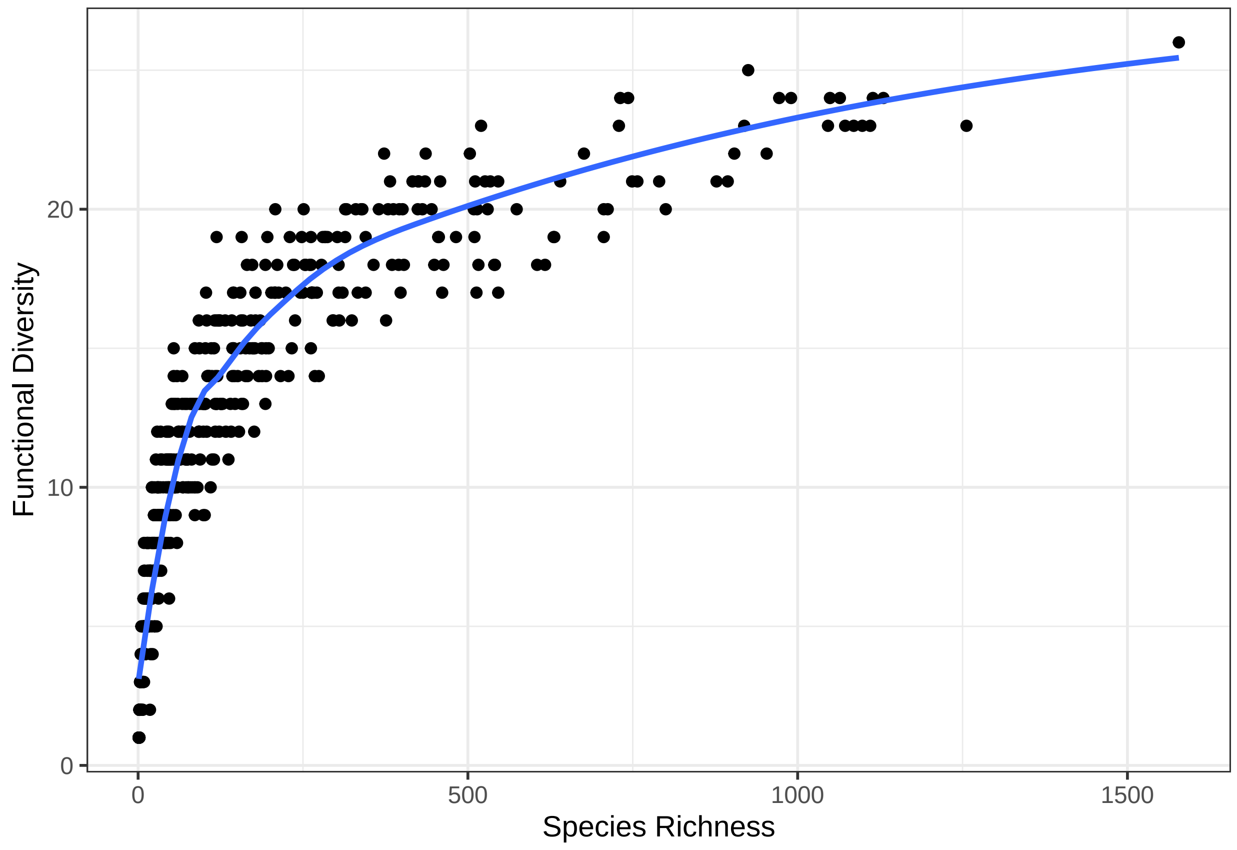
```
# join the two datasets based on the shared column 'QDS'
qds_cov %>% left_join(posa_rich, by = 'QDS') %>%
  select(QDS:NPP_var, spp_rich:phylog_diversity, geometry) %>%
  filter(!is.na(spp_rich)) -> joined_data
```

2.4 Analyse the data

After loading, exploring and cleaning, we can now begin to analyse the trends in the data. Let's make some figures to explore the relationships between the variables.

2.4.1 Species richness vs. functional diversity

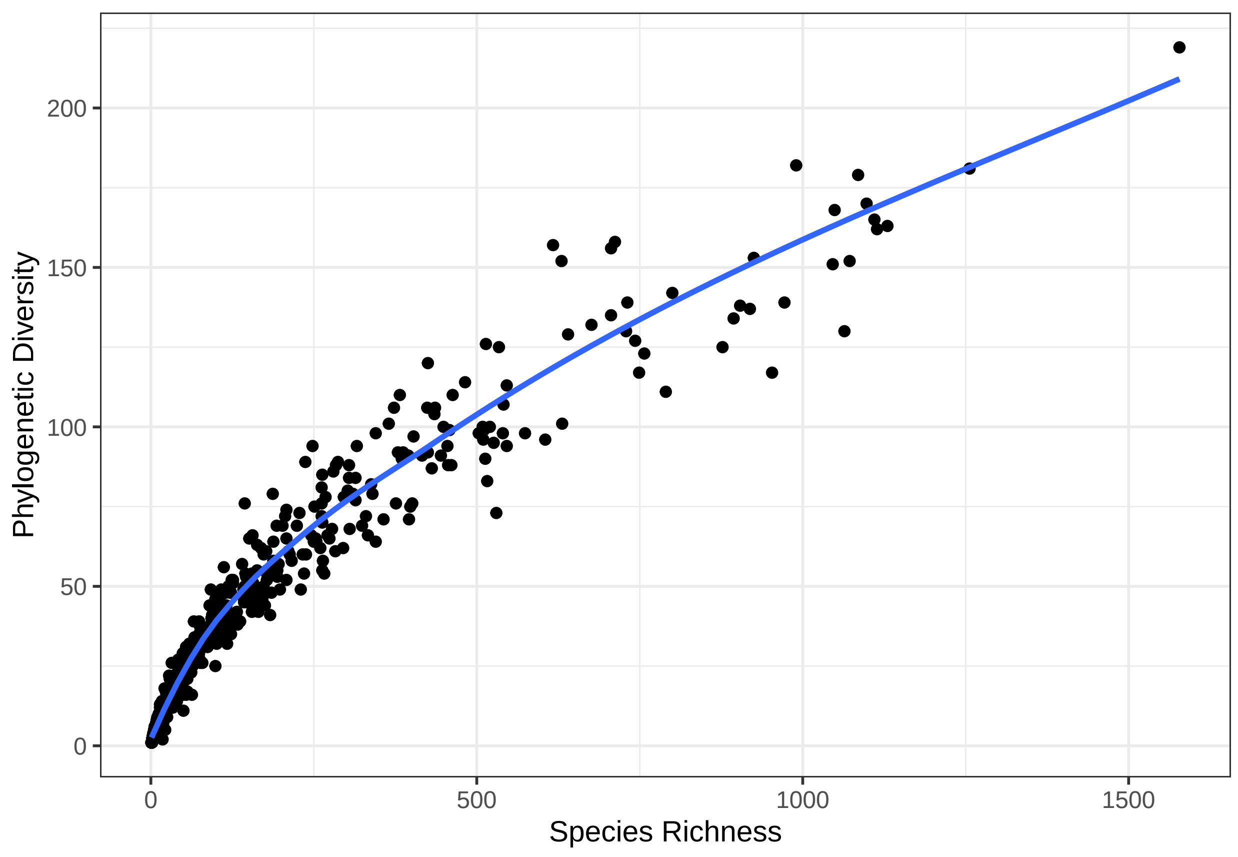
```
#### Analyse the data ----

# 1) Species richness vs. functional diversity
ggplot(data = joined_data, aes(x = spp_rich, y = func_diversity)) +
  geom_point() +
  geom_smooth(se = F) +
  labs(x = 'Species Richness', y = 'Functional Diversity') +
  theme_bw()
```



What relationship do you see between species richness and functional diversity?

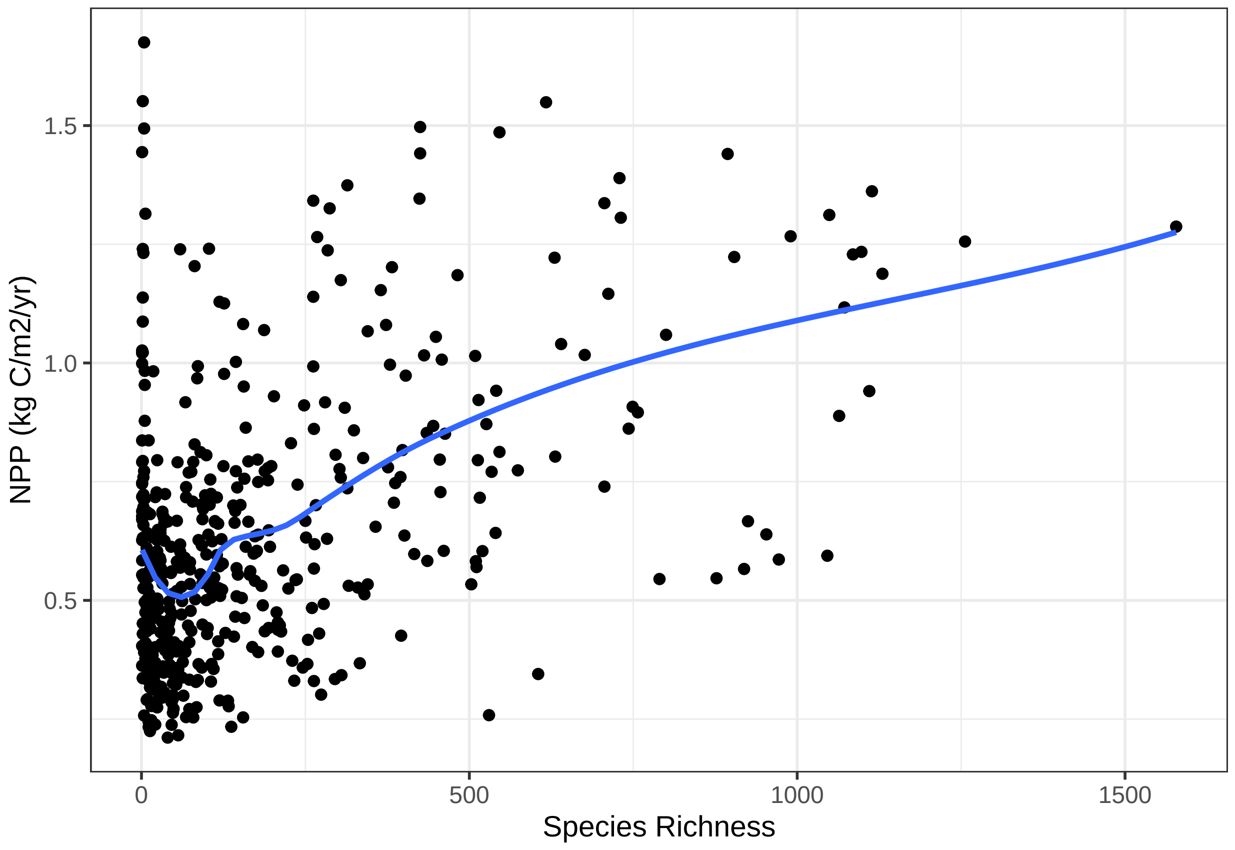
2.4.2 Species richness vs. phylogenetic diversity

```
# 2) Species richness vs. Phylogenetic Diversity
ggplot(data = joined_data, aes(x = spp_rich, y = phylog_diversity)) +
  geom_point() +
  geom_smooth(se = F) +
  labs(x = 'Species Richness', y = 'Phylogenetic Diversity') +
  theme_bw()
```



2.4.3 Species richness vs. NPP

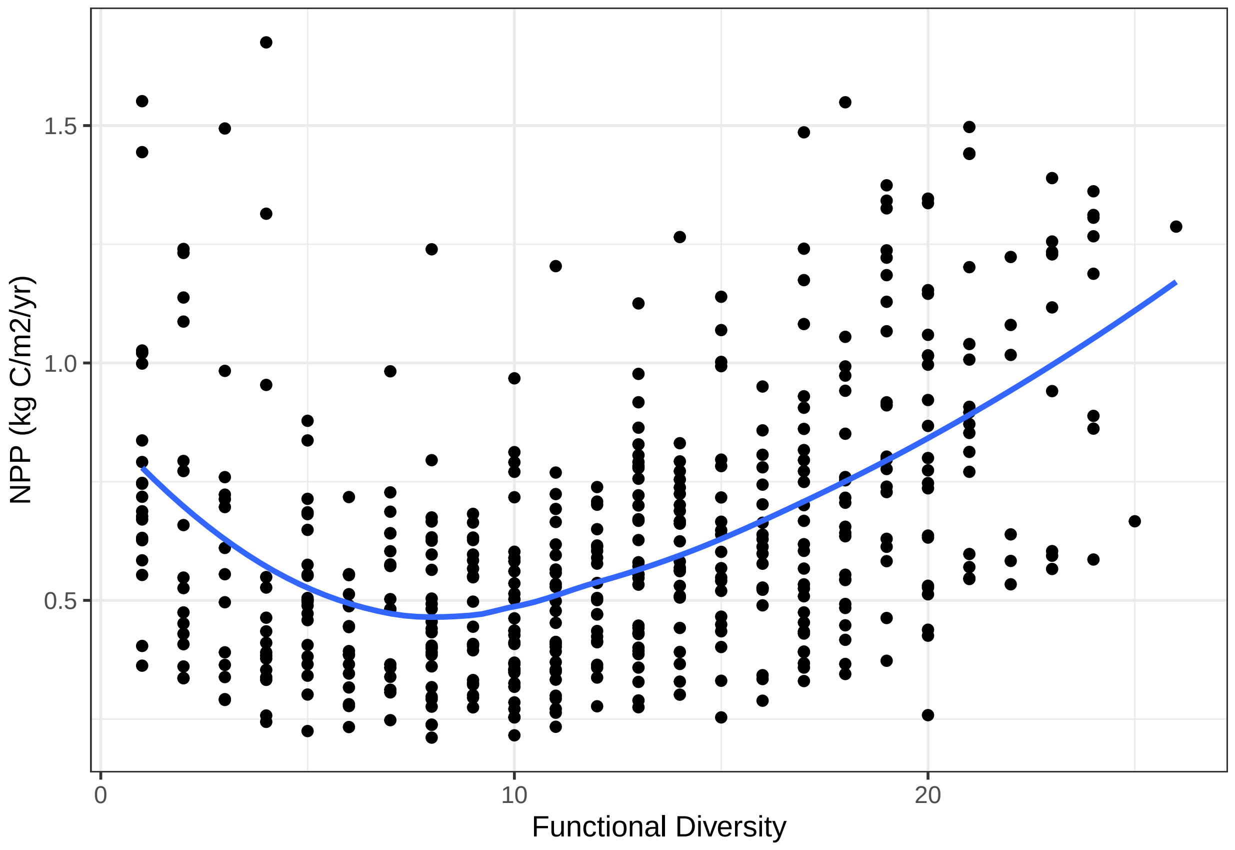
```
# 3) Species richness vs. NPP
ggplot(data = joined_data, aes(x = spp_rich, y = NPP)) +
  geom_point() +
  geom_smooth(se = F) +
  labs(x = 'Species Richness', y = 'NPP (kg C/m2/yr)') +
  theme_bw()
```



The raw data looks a bit messy. The modelled predictions in Section 2.5 will make these patterns clearer.

2.4.4 Functional diversity vs. NPP

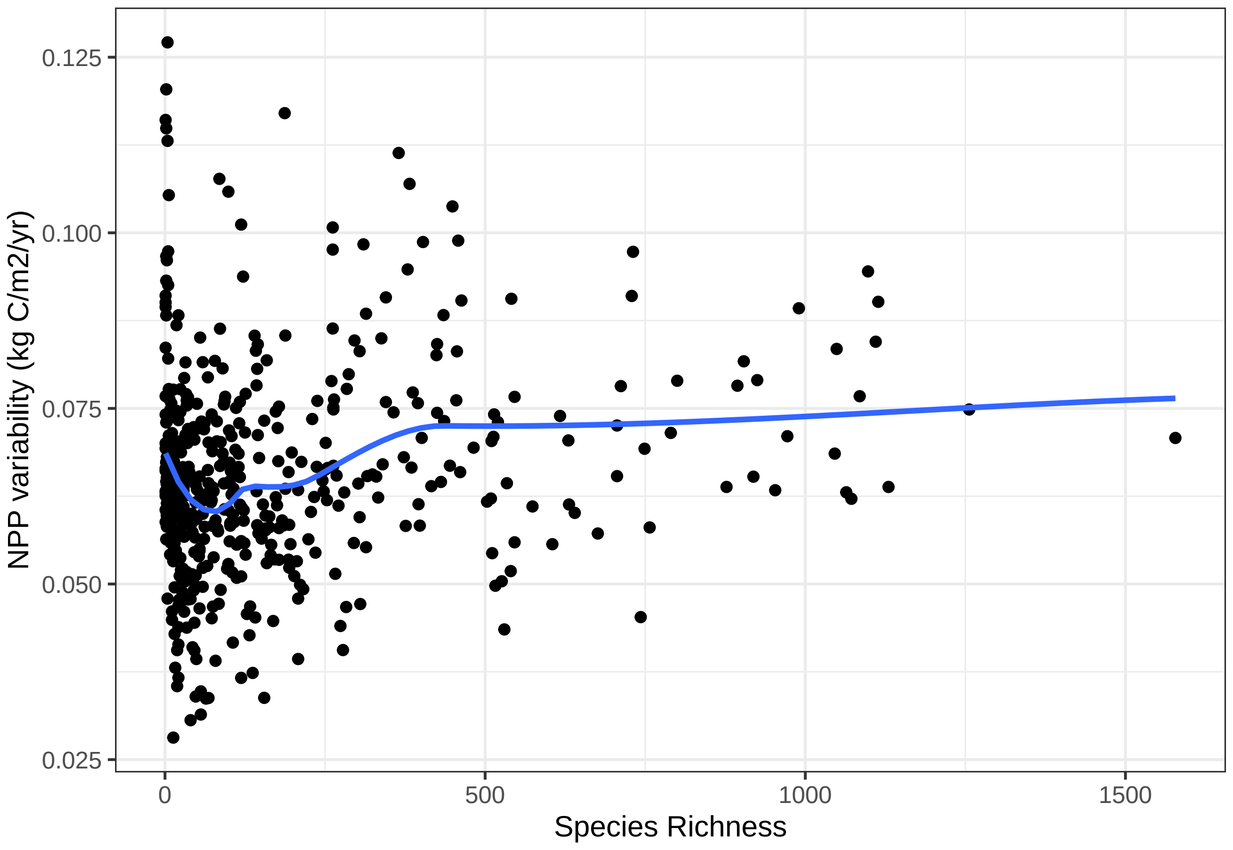
```
# 4) Functional Diversity vs. NPP
ggplot(data = joined_data, aes(x = func_diversity, y = NPP)) +
  geom_point() +
  geom_smooth(se = F) +
  labs(x = 'Functional Diversity', y = 'NPP (kg C/m2/yr)') +
  theme_bw()
```



Again, the raw data looks a bit messy. The modelled predictions in Section 2.5 will make these patterns clearer.

2.4.5 Species richness vs. NPP variability

```r
# 5) Species richness vs. NPP variability
ggplot(data = joined_data, aes(x = spp_rich, y = NPP_var)) +
  geom_point() +
  geom_smooth(se = F) +
  labs(x = 'Species Richness', y = 'NPP variability (kg C/m2/yr)') +
  theme_bw()
```



There does not seem to be a clear trend in the species richness versus NPP variability plot. So we will stick to only analysing the relationships between NPP and 1) species richness and 2) functional diversity, for our linear models.

2.5 Run a linear model

To statistically analyse the relationships between our variables we will use linear models/regressions. These models will explain the relationships between our dependent (response) variables and our independent (predictor) variables. Think of these as a straight line equation:

$$y = mx + c$$

In this equation,

$$y$$

is your response variable and

$$x$$

is predictor variable.

$$m$$

represents your slope or gradient. In R, it will be stored under the **coefficient** output using the `summary` function.

$$c$$

represents your intercept.

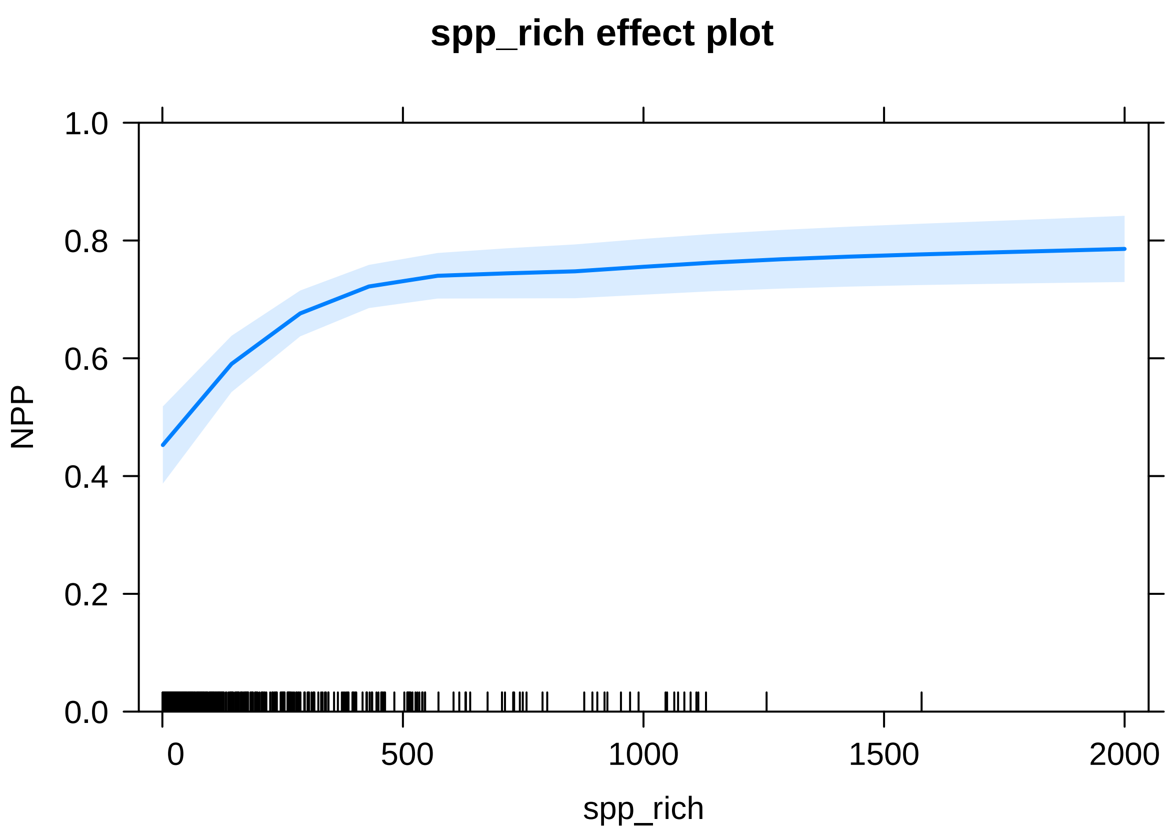
To run a linear model in R, we simply call the `lm()` function. Inside it, we need to specify our formula, which is `NPP ~ log(spp_rich)`, and the dataset we are using: `joined_data`.

2.5.1 NPP versus species richness

```r
#### run a linear model ----
# 1) What is the statistical relationship between NPP and species richness?
model1 <- lm(NPP ~ log(spp_rich), data = joined_data)
# look at regression coefficients
summary(model1)
```

```
##
## Call:
## lm(formula = NPP ~ log(spp_rich), data = joined_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46929 -0.19989 -0.06139  0.12888  1.16177
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.452698   0.033349  13.575  < 2e-16 ***
## log(spp_rich) 0.043815   0.007428   5.899 6.93e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2795 on 480 degrees of freedom
## Multiple R-squared:  0.06759,    Adjusted R-squared:  0.06564
## F-statistic: 34.79 on 1 and 480 DF,  p-value: 6.929e-09
```

```r
# Plot the predicted relationship between NPP and species richness
plot(allEffects(model1), ylim = c(0,1))
```



Take a careful look at the `summary()` output and identify the important values, such as the **coefficient estimates** and the **R^2 (R-squared)**.

Lastly, we use the `plot(allEffects())` nested functions to produce the predicted slope from our model.

What is the **predicted** relationship between NPP and species richness?

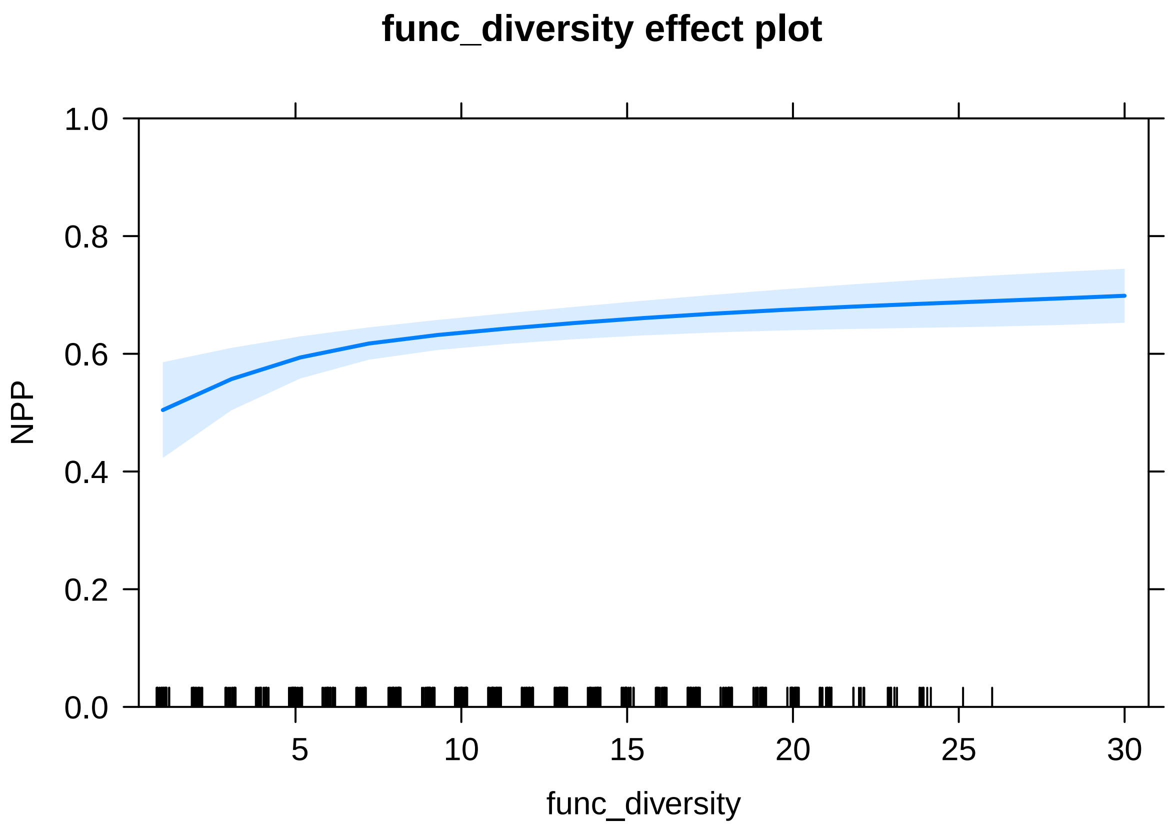
2.5.2 NPP versus functional diversity

Now do the same thing for NPP and functional diversity.

```r
# 2) What is the statistical relationship between NPP and functional diversity?
model2 <- lm(NPP ~ log(func_diversity), data = joined_data)
# look at regression coefficients
summary(model2)
```

```
##
## Call:
## lm(formula = NPP ~ log(func_diversity), data = joined_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42006 -0.21343 -0.06659  0.12929  1.09163
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.50444    0.04143  12.177  < 2e-16 ***
## log(func_diversity)  0.05709    0.01726   3.308  0.00101 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2862 on 480 degrees of freedom
## Multiple R-squared:  0.02229,    Adjusted R-squared:  0.02025
## F-statistic: 10.94 on 1 and 480 DF,  p-value: 0.001011
```

```
# Plot the predicted relationship between NPP and functional diversity
plot(allEffects(model2), ylim = c(0,1))
```



What is the **predicted** relationship between NPP and functional diversity?

2.6 Linear model summaries

The outputs from `summary()` can look a bit messy. To tidy this up and provide a neater output, we will use the `stargazer()` function.

```
#### Summaries of linear models ----
# This function provides a neat summary of the regressions for both models
stargazer(model1, model2, type = 'text')
```

```
##
```

```
## ============================================================
##                                   Dependent variable:
##                               ----------------------------
##                                           NPP
##                                    (1)            (2)
## ------------------------------------------------------------
## log(spp_rich)                    0.044***
##                                  (0.007)
##
## log(func_diversity)                            0.057***
##                                                 (0.017)
##
## Constant                         0.453***      0.504***
##                                  (0.033)        (0.041)
##
## ------------------------------------------------------------
## Observations                       482            482
## R2                                0.068          0.022
## Adjusted R2                       0.066          0.020
## Residual Std. Error (df = 480)    0.279          0.286
## F Statistic (df = 1; 480)       34.793***      10.942***
## ============================================================
## Note:                          *p<0.1; **p<0.05; ***p<0.01
```

These are the same values from our earlier outputs using `summary()`, but they are organised in a much better way for reading.

2.7 Submission

Your submission for this practical needs to be:

- Answer the 9 questions in the Practical 13 worksheet.
- Submit a word document (.docx) format.
- Submit by 10 June 17h00.