ICCB Species distribution modelling and validation

Scott Forrest and Charlotte Patterson

2025-06-06

In this script we are fitting species distribution models to the data of koalas (Phascolarctos cinereus) in the South-East Queensland (SEQ) region under current environmental conditions, which we will predict into future environmental conditions.

Table of contents

We wrote this script drawing on some of the following resources:

-Ecocommons Notebooks https://www.ecocommons.org.au/notebooks/

-Damaris Zurell's SDM Intro https://damariszurell.github.io/SDM-Intro/

https://damariszurell.github.io/EEC-MGC/b4_SDM_eval.html

Import packages

```
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(purrr)
library(ggplot2)
library(terra)
```

```
terra 1.8.50
```

```
library(sf)
```

```
Linking to GEOS 3.13.0, GDAL 3.8.5, PROJ 9.5.1; sf_use_s2() is TRUE
```

```
library(predicts)
library(blockCV)
```

```
blockCV 3.1.5
```

```
library(ecospat)
library(usdm)
library(randomForest)
```

```
randomForest 4.7-1.2

Type rfNews() to see new features/changes/bug fixes.
```

```
Attaching package: 'randomForest'

The following object is masked from 'package:ggplot2':

    margin

The following object is masked from 'package:dplyr':

    combine
```

```
library(precrec)
library(corrplot)
```

```
corrplot 0.95 loaded
```

```
# Install the mecofun package, used in the materials at https://damariszurell.github.io/SDM-
library(devtools)
```

```
Loading required package: usethis
```

```
devtools::install_git("https://gitup.uni-potsdam.de/macroecology/mecofun.git")
```

```
Skipping install of 'mecofun' from a xgit remote, the SHA1 (feb82c4a) has not changed since
  Use `force = TRUE` to force installation
```

```
# Load the mecofun package
library(mecofun)
```

Load koala presences and background points

They are loaded as spatvectors, but we also want them as dataframes for model input requirements.

```
koala_occ <- vect("Data/Biological_records/SEQ_koala_occurrences.shp")
background <- vect("Data/Biological_records/background_points_2.5k_random.shp")

# Make a dataframe of just x, y and presence
koala_occ_df <- koala_occ %>%
  as.data.frame(geom = "XY") %>%
```

```
  dplyr::select(x,y) %>%
  mutate(Presence = 1)

head(koala_occ_df)
```

```
        x        y Presence
1 1868673 -3265806        1
2 1851054 -3232106        1
3 1846917 -3231913        1
4 1874167 -3262365        1
5 1857487 -3259935        1
6 1874262 -3262395        1
```

```
background_df <- background %>%
  as.data.frame(geom = "XY") %>%
  dplyr::select(x,y) %>%
  mutate(Presence = 0)

head(background_df)
```

```
        x        y Presence
1 1854692 -3111330        0
2 1859816 -3111330        0
3 1864940 -3111330        0
4 1870064 -3111330        0
5 1875188 -3111330        0
6 1880312 -3111330        0
```

```
# Combine to one
pr_bg <- rbind(koala_occ_df, background_df)
```

Load environmental covariates

Loading current covariate rasters. We formatted these rasters in the same way as the Koala data, so that they are all in the same projection and extent. We did this in the script: 'ICCB_Environmental_data.qmd'

```
covs_current <- rast("Data/Environmental_variables/current_bioclim.tif")


# Define the BIOCLIM names for the raster layers
layer_names <- c(
```

```
  "BIO1_Annual_Mean_Temp",
  "BIO2_Mean_Diurnal_Temp_Range",
  "BIO3_Isothermality",
  "BIO4_Temperature_Seasonality",
  "BIO5_Max_Temp_Warmest_Month",
  "BIO6_Min_Temp_Coldest_Month",
  "BIO7_Temperature_Annual_Range",
  "BIO8_Mean_Temp_Wettest_Quarter",
  "BIO9_Mean_Temp_Driest_Quarter",
  "BIO10_Mean_Temp_Warmest_Quarter",
  "BIO11_Mean_Temp_Coldest_Quarter",
  "BIO12_Annual_Precipitation",
  "BIO13_Precip_Wettest_Month",
  "BIO14_Precip_Driest_Month",
  "BIO15_Precip_Seasonality",
  "BIO16_Precip_Wettest_Quarter",
  "BIO17_Precip_Driest_Quarter",
  "BIO18_Precip_Warmest_Quarter",
  "BIO19_Precip_Coldest_Quarter")

names(covs_current) <- layer_names
```

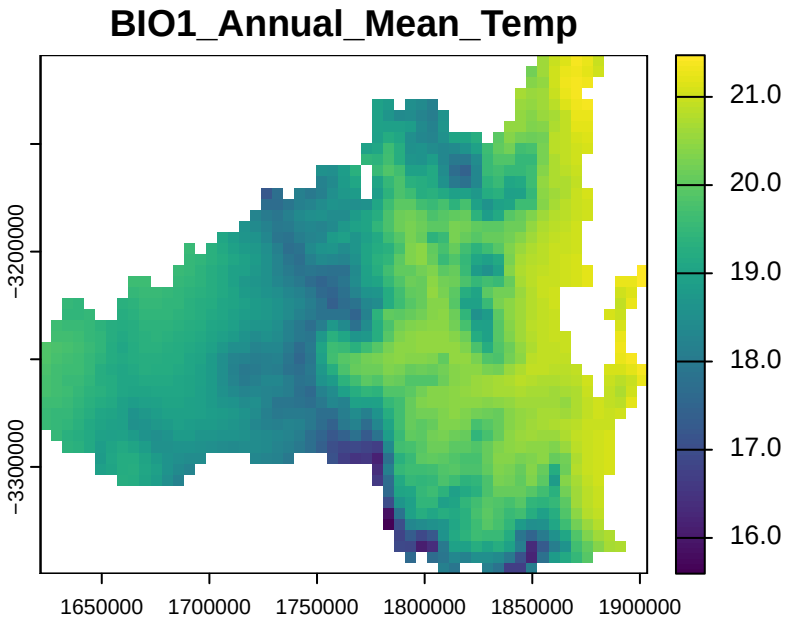
Covariate selection

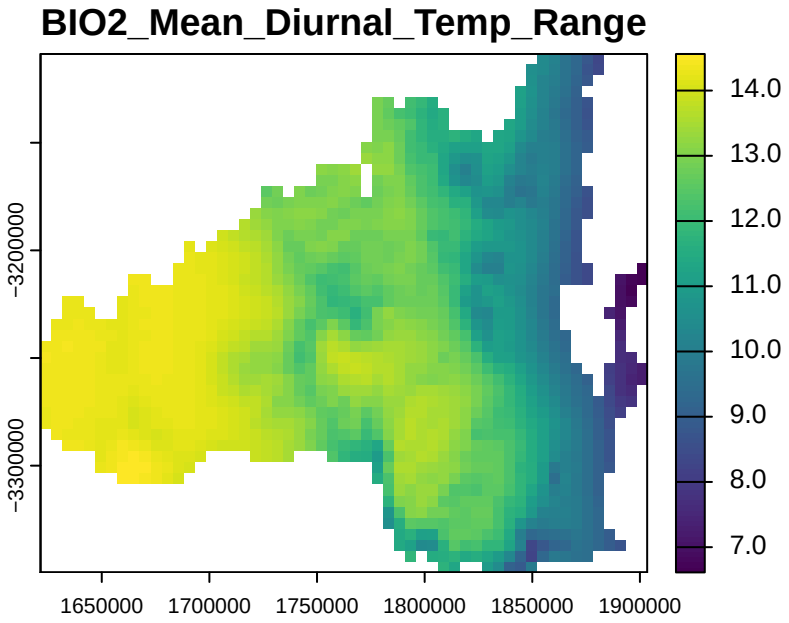
Option 1. Narrow down potential covariates based on ecological knowledge

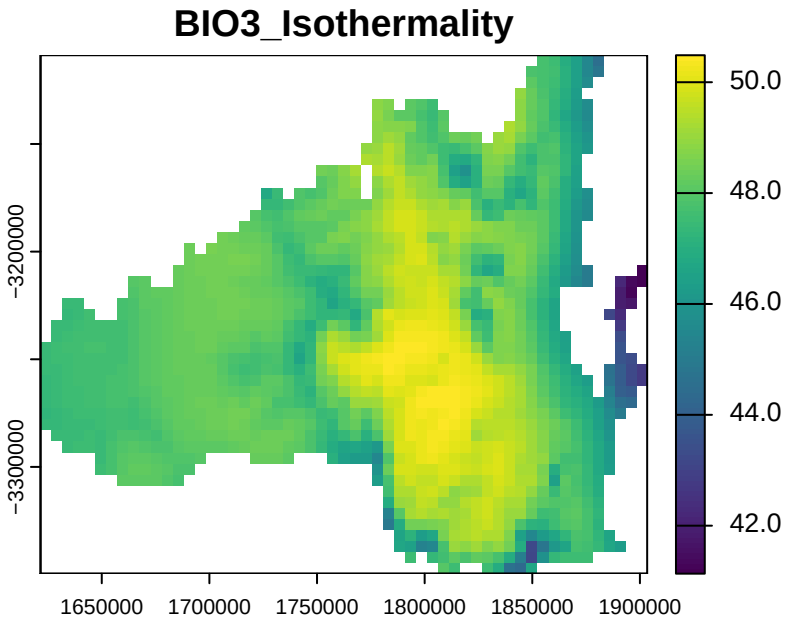
For this example, we had advice from CSIRO scientists who conducted an expert elicitation to gather a set of potential covariates that are likely to be important for koalas. We use this knowledge to filter out the key bioclim variables.

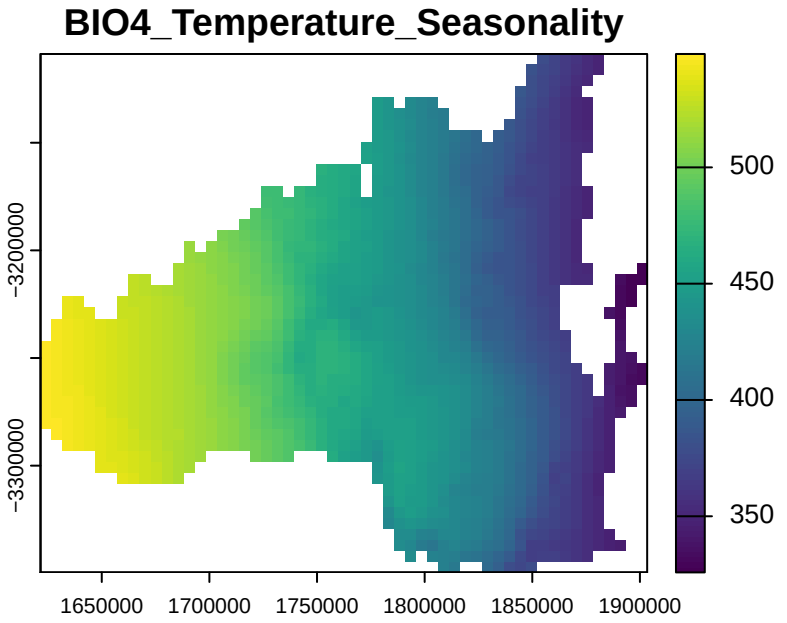
We select the following: Bio5 : Max temp of the warmest month (mainly for the northern populations) Bio6 : Min temp of the coldest month (mainly for southern populations, which essentially excludes alpine regions) Bio12 : Annual Precipitation Bio15 : Precipitation seasonality (coefficient of variation)

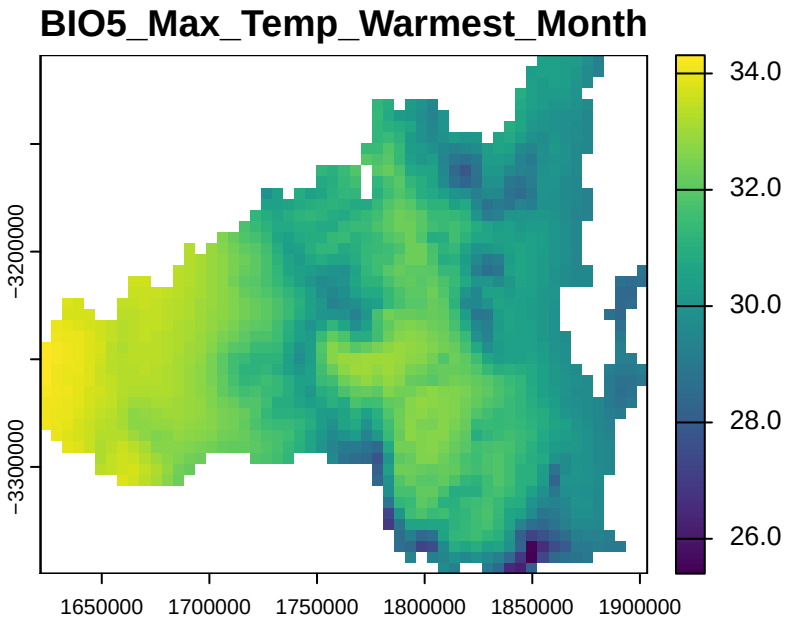
```
for(i in 1:nlyr(covs_current)) {
  terra::plot(covs_current[[i]], main = names(covs_current)[[i]])
}
```

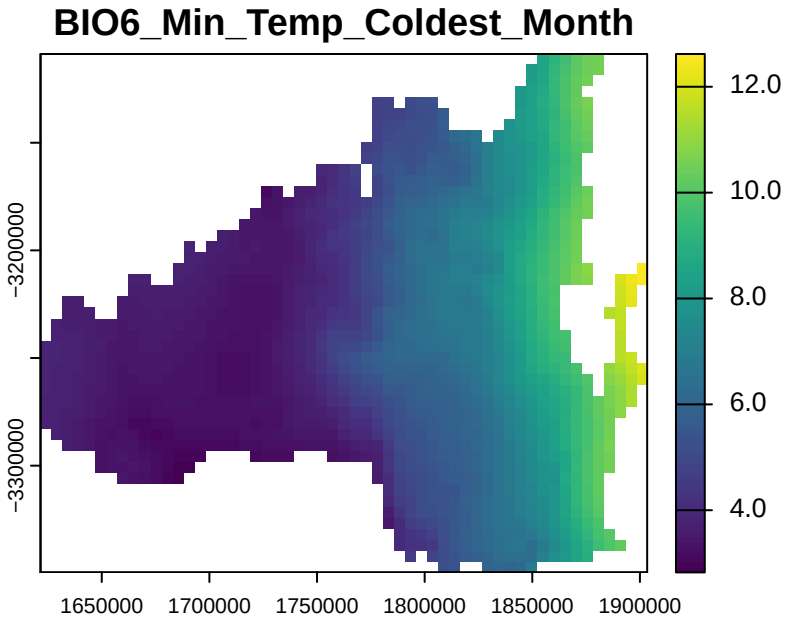

BIO1_Annual_Mean_Temp
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
21.0
20.0
19.0
18.0
17.0
16.0

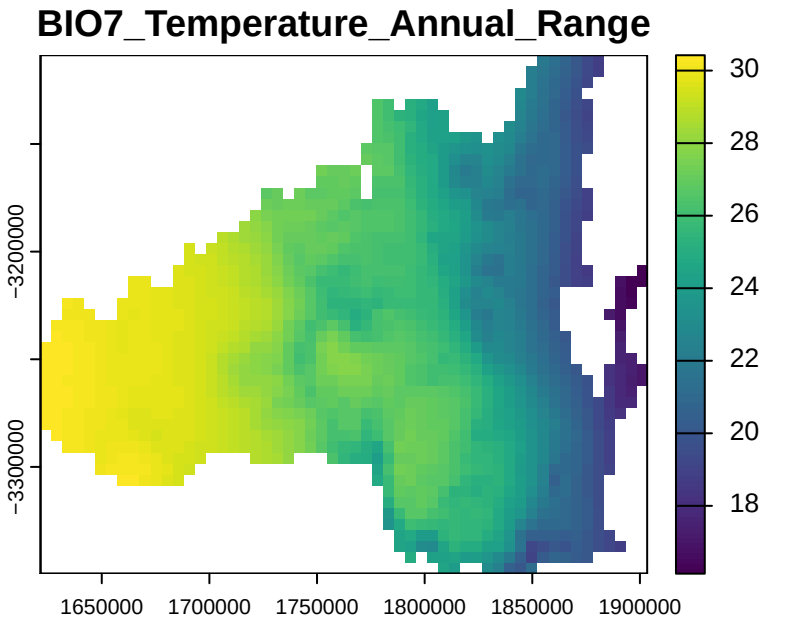

BIO2_Mean_Diurnal_Temp_Range
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
14.0
13.0
12.0
11.0
10.0
9.0
8.0
7.0

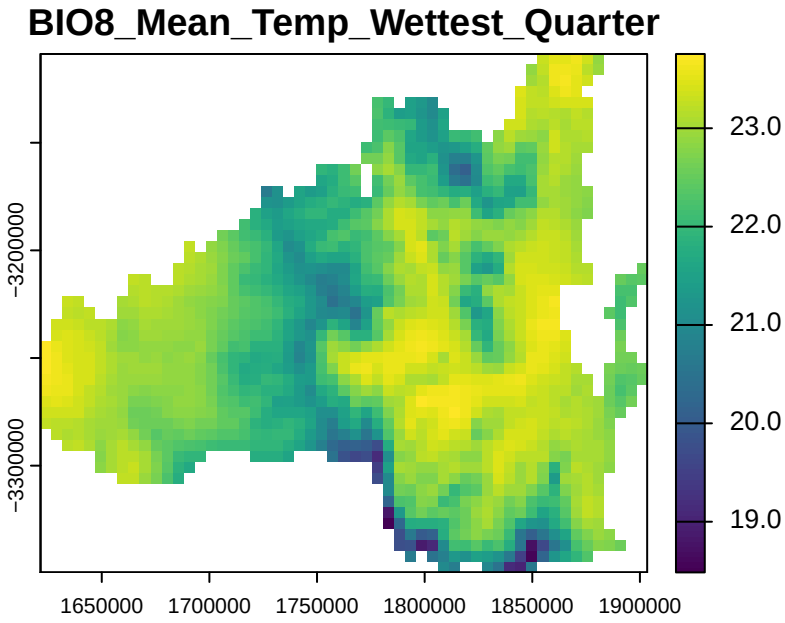

BIO3_Isothermality
50.0
48.0
46.0
44.0
42.0
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000

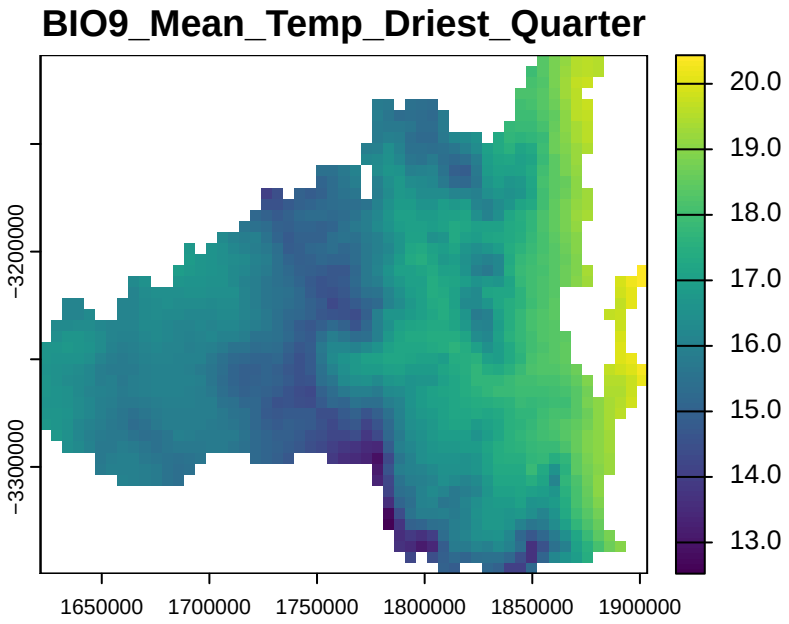

BIO4_Temperature_Seasonality
500
450
400
350
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000

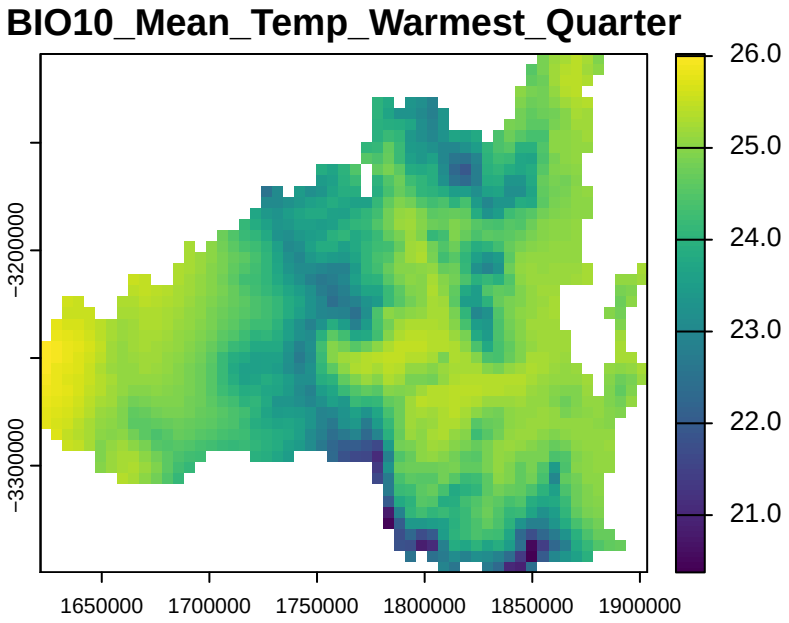

BIO5_Max_Temp_Warmest_Month
34.0
32.0
30.0
28.0
26.0
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000

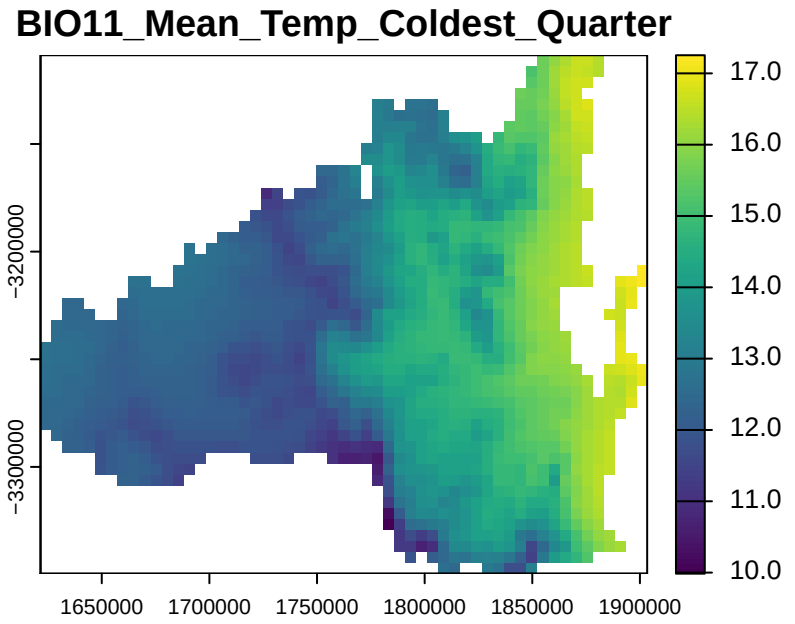

BIO6_Min_Temp_Coldest_Month
12.0
10.0
8.0
6.0
4.0
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000

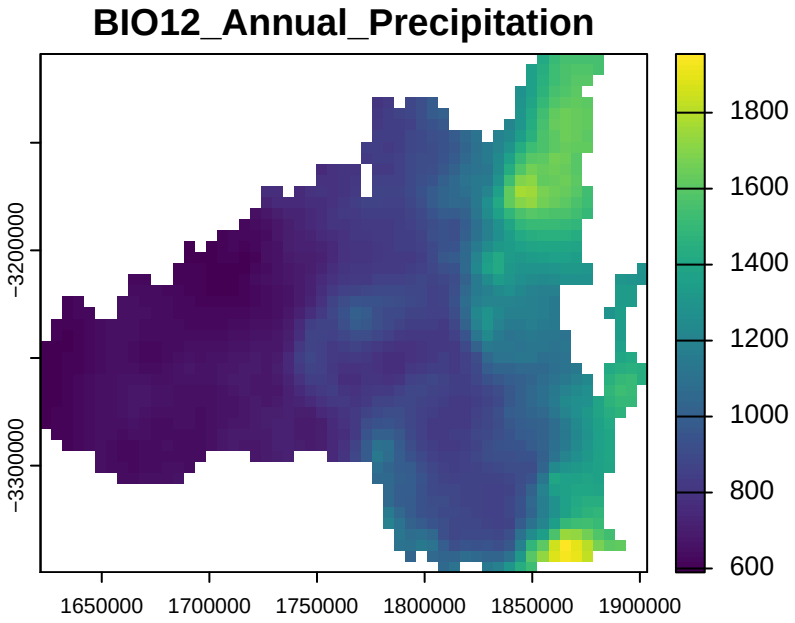

BIO7_Temperature_Annual_Range
30
28
26
24
22
20
18
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000

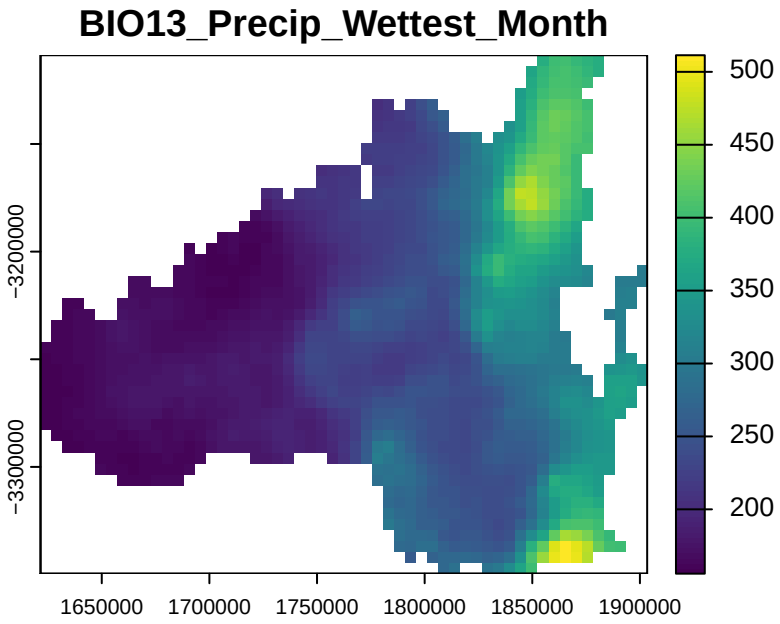

BIO8_Mean_Temp_Wettest_Quarter
23.0
22.0
21.0
20.0
19.0
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000

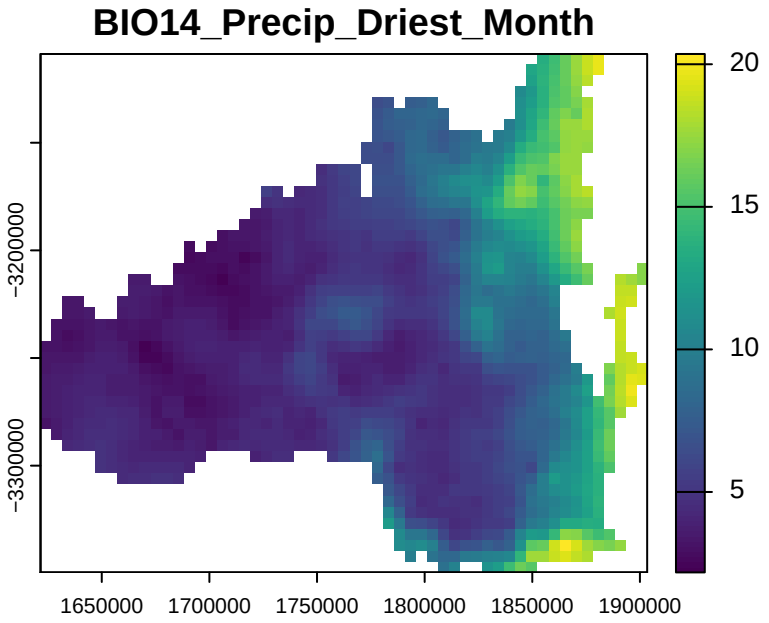

BIO9_Mean_Temp_Driest_Quarter
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
20.0
19.0
18.0
17.0
16.0
15.0
14.0
13.0

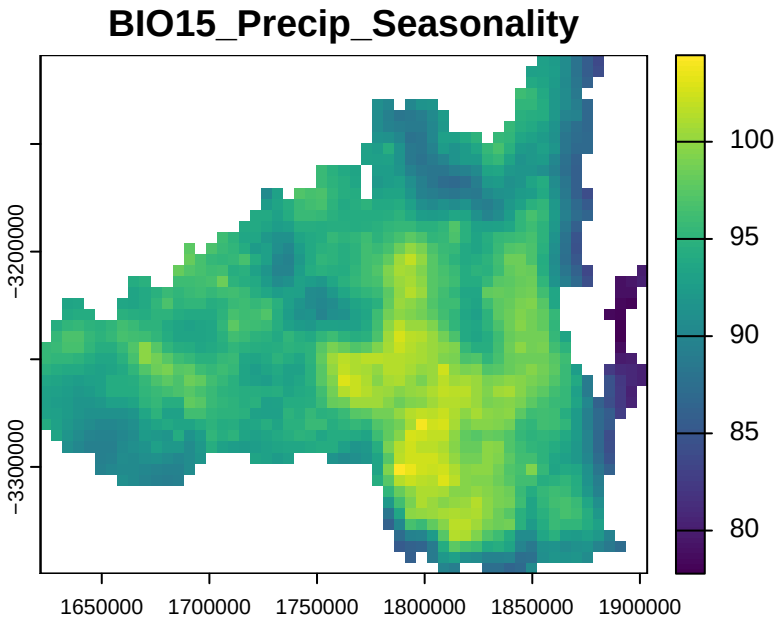

BIO10_Mean_Temp_Warmest_Quarter
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
26.0
25.0
24.0
23.0
22.0
21.0

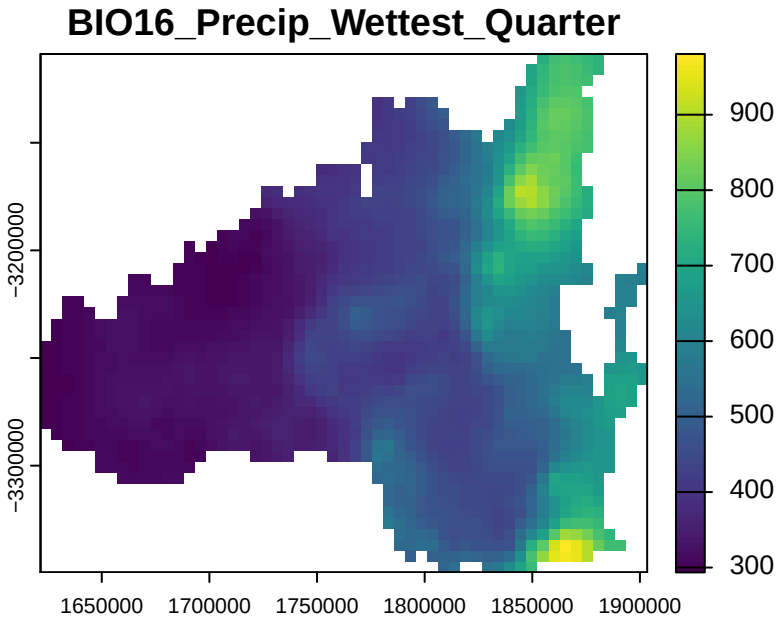

BIO11_Mean_Temp_Coldest_Quarter
17.0
16.0
15.0
14.0
13.0
12.0
11.0
10.0
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000

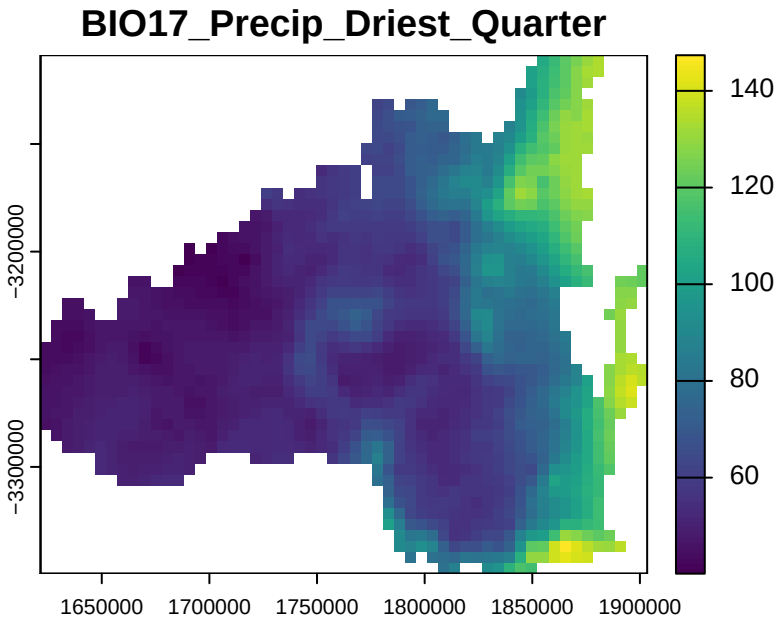

BIO12_Annual_Precipitation
1800
1600
1400
1200
1000
800
600
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000

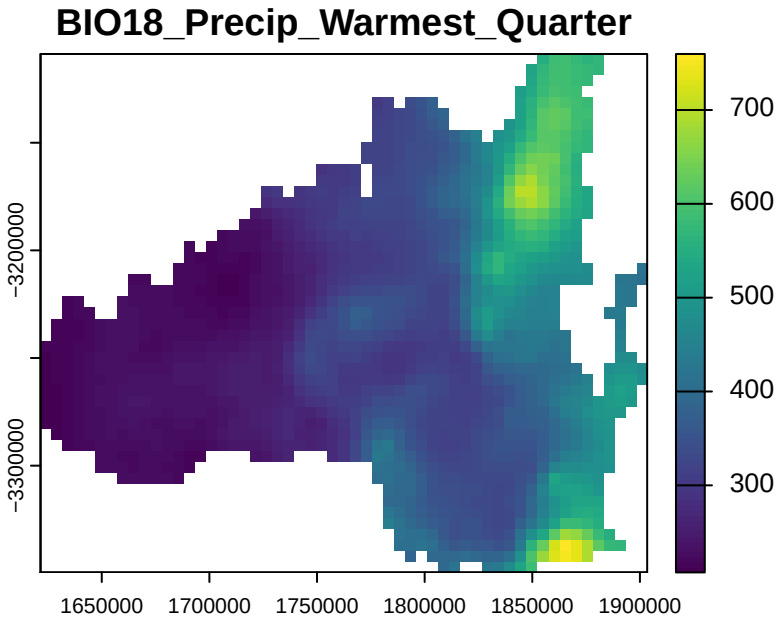

BIO13_Precip_Wettest_Month
500
450
400
350
300
250
200
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000

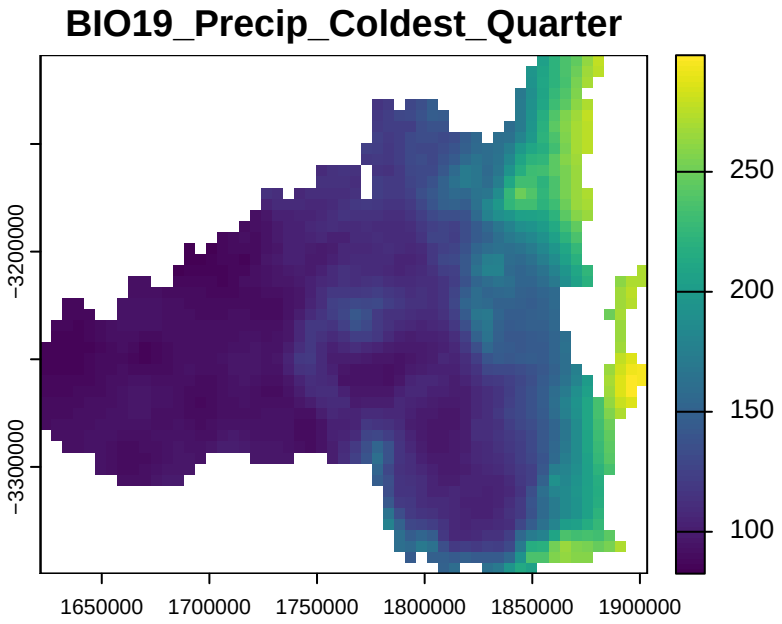

BIO14_Precip_Driest_Month
20
15
10
5
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000


BIO15_Precip_Seasonality
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
100
95
90
85
80


BIO16_Precip_Wettest_Quarter
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
900
800
700
600
500
400
300

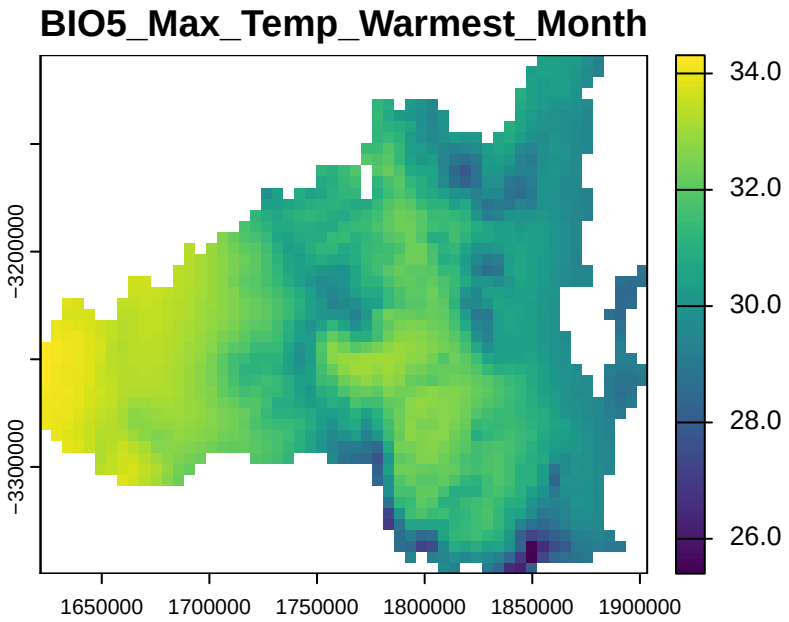

BIO17_Precip_Driest_Quarter
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
140
120
100
80
60

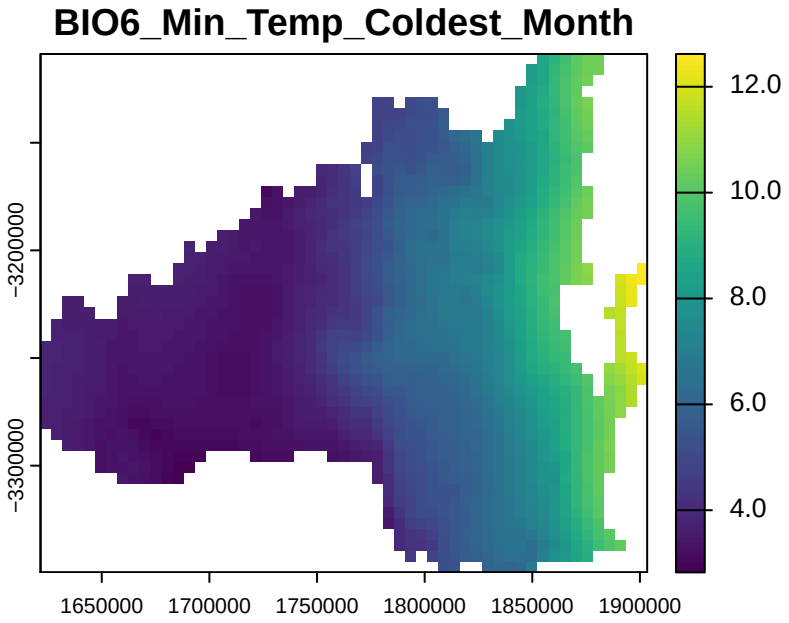

BIO18_Precip_Warmest_Quarter
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
700
600
500
400
300

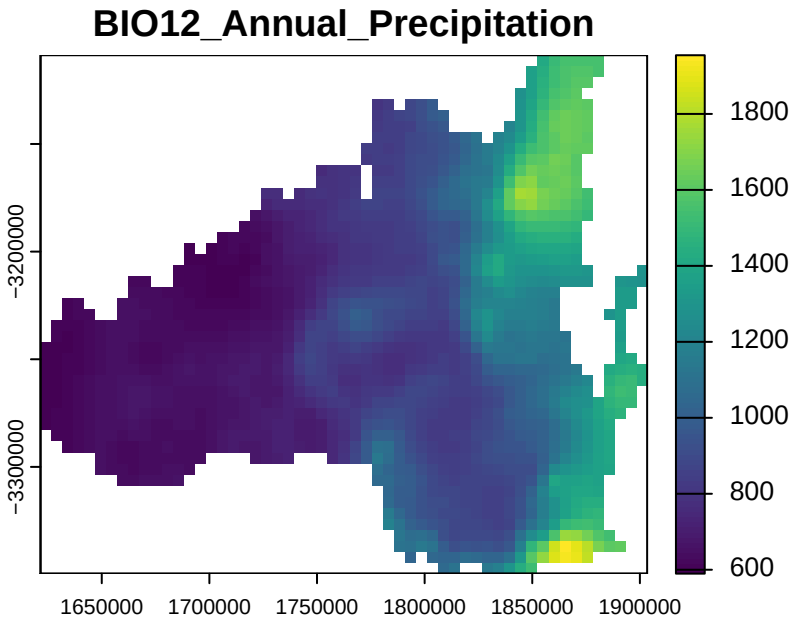


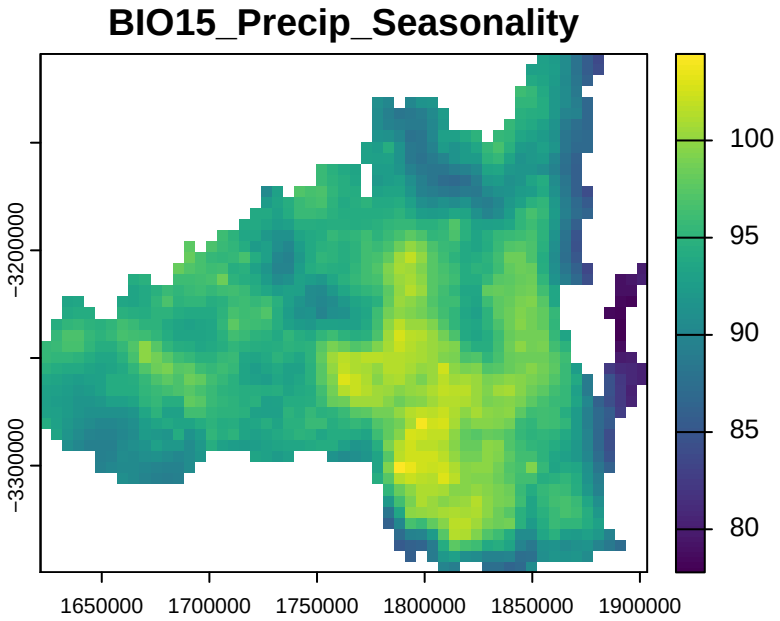
Show the four from expert elicitation the layers

```
covs_current_expert <- subset(covs_current, names(covs_current) %in% c("BIO5_Max_Temp_Warmes
                                                                       "BIO6_Min_Temp_Coldes
                                                                       "BIO12_Annual_Precipi
                                                                       "BIO15_Precip_Seasona


for(i in 1:nlyr(covs_current_expert)) {
  terra::plot(covs_current_expert[[i]], main = names(covs_current_expert)[[i]])
}
```


BIO5_Max_Temp_Warmest_Month
34.0
32.0
30.0
28.0
26.0
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000


BIO6_Min_Temp_Coldest_Month
12.0
10.0
8.0
6.0
4.0
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000





Extract environmental covariate values from presence and background locations (training locations)

```
train_PB_covs <- terra::extract(covs_current, pr_bg[,c("x", "y")], xy = T)
train_PB_covs <- cbind(train_PB_covs, pr_bg["Presence"])

# Remove rows where there's values missing from at least one covariate
print(paste0("RECORDS FROM ", nrow(train_PB_covs) - sum(complete.cases(train_PB_covs)), " RO
```

```
[1] "RECORDS FROM 68 ROWS IN TRAINING DATA REMOVED DUE TO MISSING COVARIATE VALUES"
```

```r
train_PB_covs <- train_PB_covs[complete.cases(train_PB_covs), ]
train_PB_covs <- dplyr::select(train_PB_covs, -ID)

head(train_PB_covs)
```

```
  BIO1_Annual_Mean_Temp BIO2_Mean_Diurnal_Temp_Range BIO3_Isothermality
1              20.73219                    9.515615           46.85551
2              20.85025                   10.291359           47.88751
3              20.61474                   10.648253           48.39870
4              21.00405                    9.129249           46.28033
5              20.75752                   10.070734           47.49575
6              21.00405                    9.129249           46.28033
  BIO4_Temperature_Seasonality BIO5_Max_Temp_Warmest_Month
1                     367.6388                    29.73501
2                     379.8710                    30.47122
3                     384.8382                    30.53297
4                     363.1416                    29.72843
5                     378.2968                    30.20078
6                     363.1416                    29.72843
  BIO6_Min_Temp_Coldest_Month BIO7_Temperature_Annual_Range
1                    9.386996                     20.34801
2                    8.940289                     21.53093
3                    8.487451                     22.04552
4                    9.958574                     19.76986
5                    8.959633                     21.24114
6                    9.958574                     19.76986
  BIO8_Mean_Temp_Wettest_Quarter BIO9_Mean_Temp_Driest_Quarter
1                       22.98827                      18.36451
2                       23.55152                      18.09120
3                       23.31077                      17.78633
4                       23.01914                      18.89500
5                       23.32381                      18.13480
6                       23.01914                      18.89500
  BIO10_Mean_Temp_Warmest_Quarter BIO11_Mean_Temp_Coldest_Quarter
1                        24.94183                        15.92675
2                        25.18869                        15.87509
3                        24.99932                        15.57214
4                        25.18043                        16.27002
5                        25.08320                        15.80299
6                        25.18043                        16.27002
  BIO12_Annual_Precipitation BIO13_Precip_Wettest_Month
1                   1238.362                   327.7901
```

```
2                   1165.235                   327.1804
3                   1161.138                   327.4446
4                   1222.618                   311.3793
5                   1109.354                   305.5583
6                   1222.618                   311.3793
  BIO14_Precip_Driest_Month BIO15_Precip_Seasonality
1                 10.800349                  92.89355
2                  8.330686                  98.80420
3                  8.048150                  99.45329
4                 11.913177                  88.93709
5                  8.327950                  97.22968
6                 11.913177                  88.93709
  BIO16_Precip_Wettest_Quarter BIO17_Precip_Driest_Quarter
1                     606.3058                    94.95189
2                     598.7009                    77.30684
3                     598.9535                    75.86132
4                     583.3614                   100.51526
5                     558.9021                    77.90620
6                     583.3614                   100.51526
  BIO18_Precip_Warmest_Quarter BIO19_Precip_Coldest_Quarter       x        y
1                     449.8074                     189.9305 1870064 -3265048
2                     444.0957                     147.6409 1849568 -3234305
3                     445.4260                     145.7036 1844444 -3234305
4                     448.1904                     200.5528 1875188 -3259924
5                     408.9403                     153.3001 1859816 -3259924
6                     448.1904                     200.5528 1875188 -3259924
  Presence
1        1
2        1
3        1
4        1
5        1
6        1
```

Thin the koala presence points (for tutorial only)

We now thin the presences to reduce the number of points to a manageable size for plotting and modelling. This is not a recommended step for real data, but is done here to make the tutorial run faster and to make the plots clearer.

```
train_PB_covs_pres <- train_PB_covs %>% filter(Presence == 1)
train_PB_covs_bg <- train_PB_covs %>% filter(Presence == 0)

# Thin the presences for plotting
```

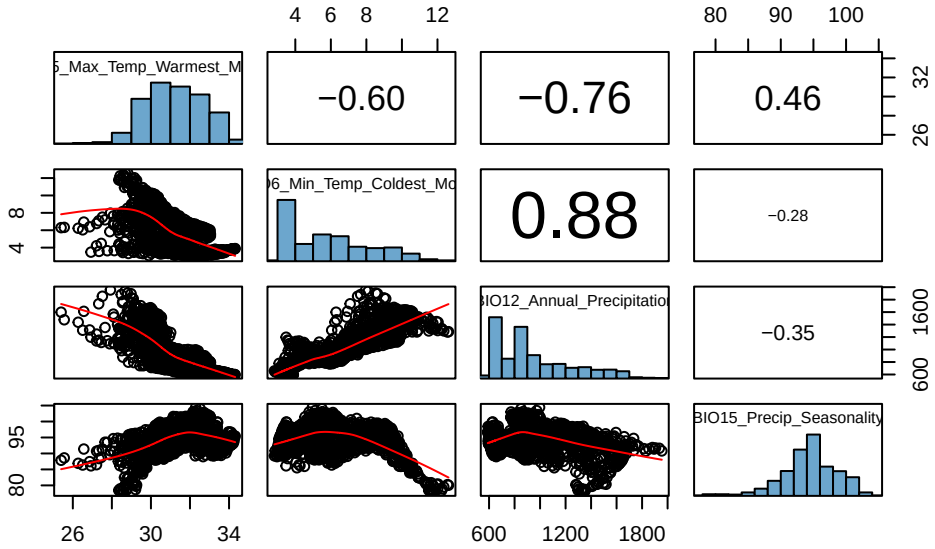
```
train_PB_covs_pres_thin <- train_PB_covs_pres[sample(nrow(train_PB_covs_pres), 10000), ]

# Combine back into both presence and background
train_PB_covs_thinned <- rbind(train_PB_covs_pres_thin, train_PB_covs_bg)
```

Check correlation and multicollinearity of covariates

Correlation plot

There are several different methods for creating correlation plots.

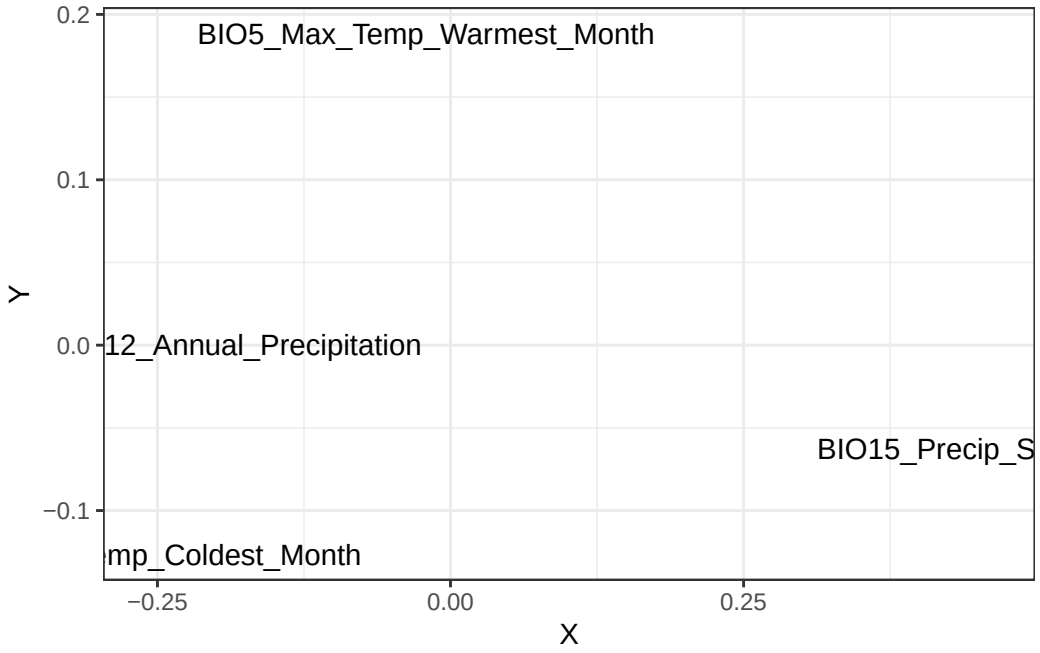
```
ecospat.cor.plot(covs_current_expert)
```



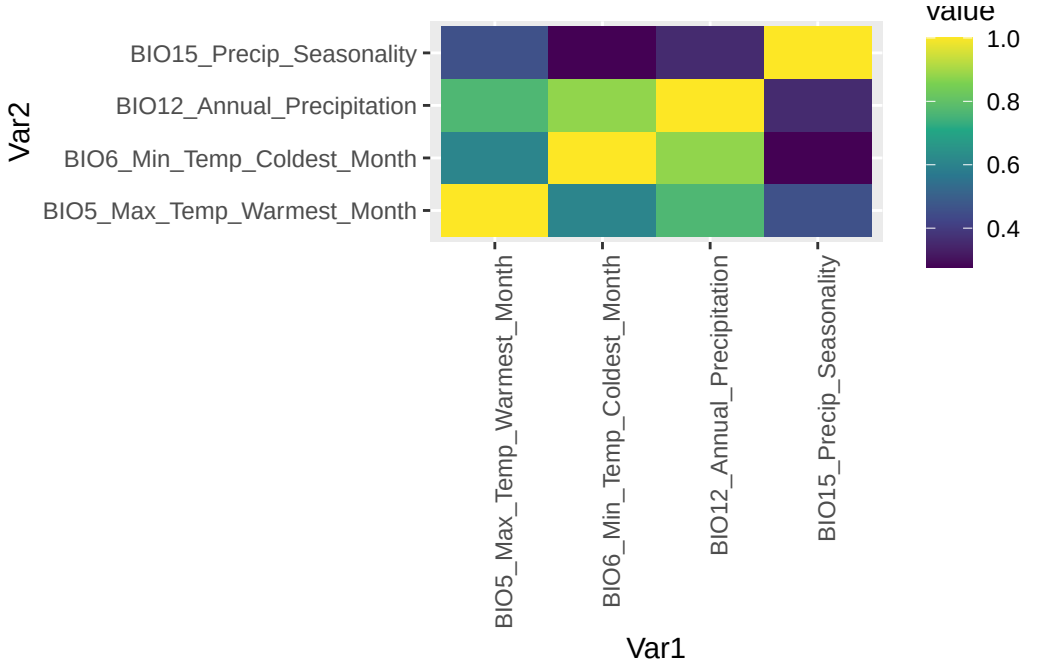
Using the corplots package

For a simple and quick plot.

```
corplots <- ENMTools::raster.cor.plot(covs_current_expert)
corplots$cor.mds.plot
```




```
corplots$cor.heatmap
```

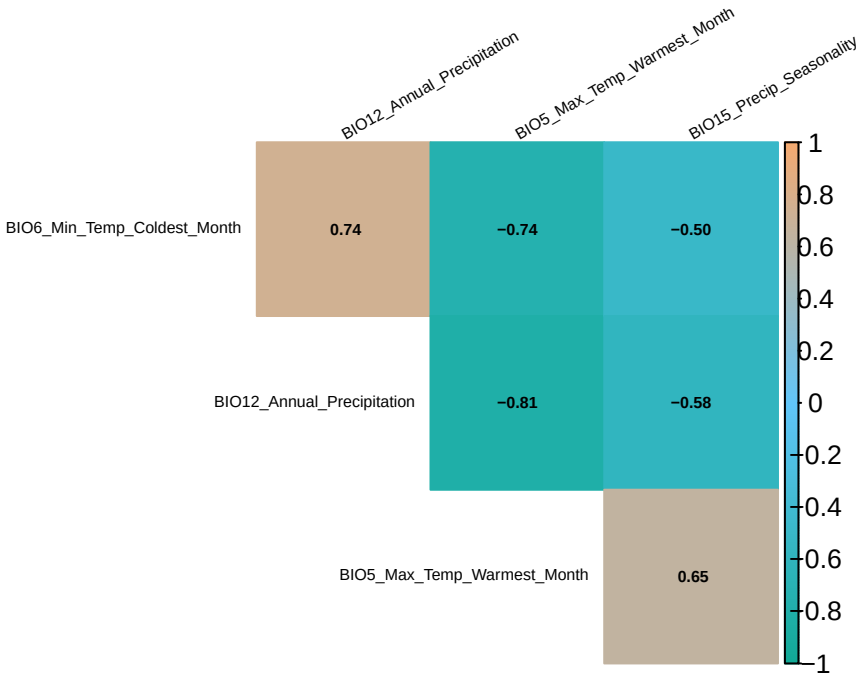


Here, we use the `corrplot` package to create a correlation plot of the selected covariates (this is taken from an EcoCommons Australia notebook).

```r
# Select columns by their names
cor_data <- train_PB_covs[, names(train_PB_covs) %in% c("BIO5_Max_Temp_Warmest_Month",
                                                        "BIO6_Min_Temp_Coldest_Month",
                                                        "BIO12_Annual_Precipitation",
                                                        "BIO15_Precip_Seasonality")]

# Check the structure of the numeric data
str(cor_data)
```

```
'data.frame':   88136 obs. of  4 variables:
 $ BIO5_Max_Temp_Warmest_Month: num  29.7 30.5 30.5 29.7 30.2 ...
 $ BIO6_Min_Temp_Coldest_Month: num  9.39 8.94 8.49 9.96 8.96 ...
 $ BIO12_Annual_Precipitation : num  1238 1165 1161 1223 1109 ...
 $ BIO15_Precip_Seasonality   : num  92.9 98.8 99.5 88.9 97.2 ...
```

```r
# Calculate the correlation matrix for the numeric columns
cor_matrix <- cor(cor_data, use = "complete.obs", method = "pearson")


corrplot(cor_matrix,
         method = "color",           # Use colored squares for correlation
         type = "upper",             # Show upper triangle only
         order = "hclust",           # Reorder variables hierarchically
         addCoef.col = "black",      # Show correlation coefficients in black
         number.cex = 0.5,           # Reduce the size of correlation labels
         tl.col = "black",           # Text label color
         tl.srt = 30,                # Rotate labels slightly for readability
         tl.cex = 0.5,               # Reduce text size of variable labels (set smaller val
         cl.cex = 0.8,               # Reduce text size of color legend
         diag = FALSE,               # Hide diagonal
         col = colorRampPalette(c("#11aa96", "#61c6fa", "#f6aa70"))(200),
         sig.level = 0.01, insig = "blank")
```



Variance Inflation Factor (VIF)

If you find corrplot is hard for you to make decisions, we can use Variance Inflation Factor (VIF). VIF is another statistical measure used to detect multicollinearity in a set of explanatory (independent) variables in a regression model.

Interpretation:

- VIF = 1: No correlation
- VIF > 1 and <= 5: Moderate correlation; may not require corrective action.
- VIF > 5: Indicates high correlation. Multicollinearity may be problematic, and further investigation is recommended.
- VIF > 10: Strong multicollinearity. The variable is highly collinear with others, and steps should be taken to address this.

```
# usdm::vif(covs_current_expert) # just VIF for all covariates
usdm::vifstep(covs_current_expert) # Variance Inflation Factor and test for multicollinearit
```

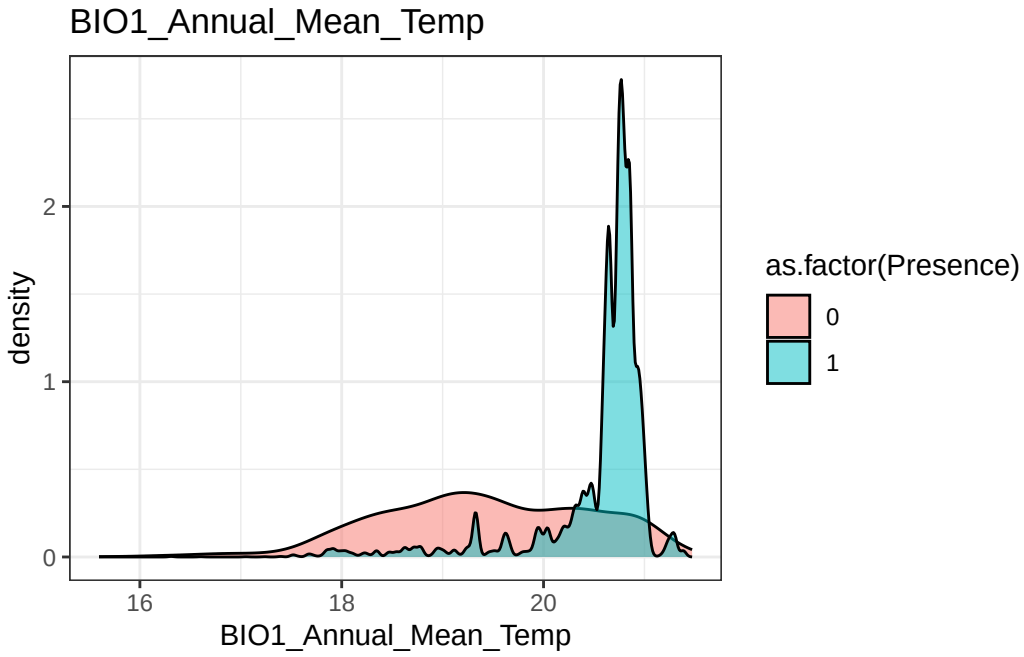
```
No variable from the 4 input variables has collinearity problem.

The linear correlation coefficients ranges between:
min correlation ( BIO15_Precip_Seasonality ~ BIO6_Min_Temp_Coldest_Month ):  -0.2756697
max correlation ( BIO12_Annual_Precipitation ~ BIO6_Min_Temp_Coldest_Month ):  0.8769718

---------- VIFs of the remained variables --------
```

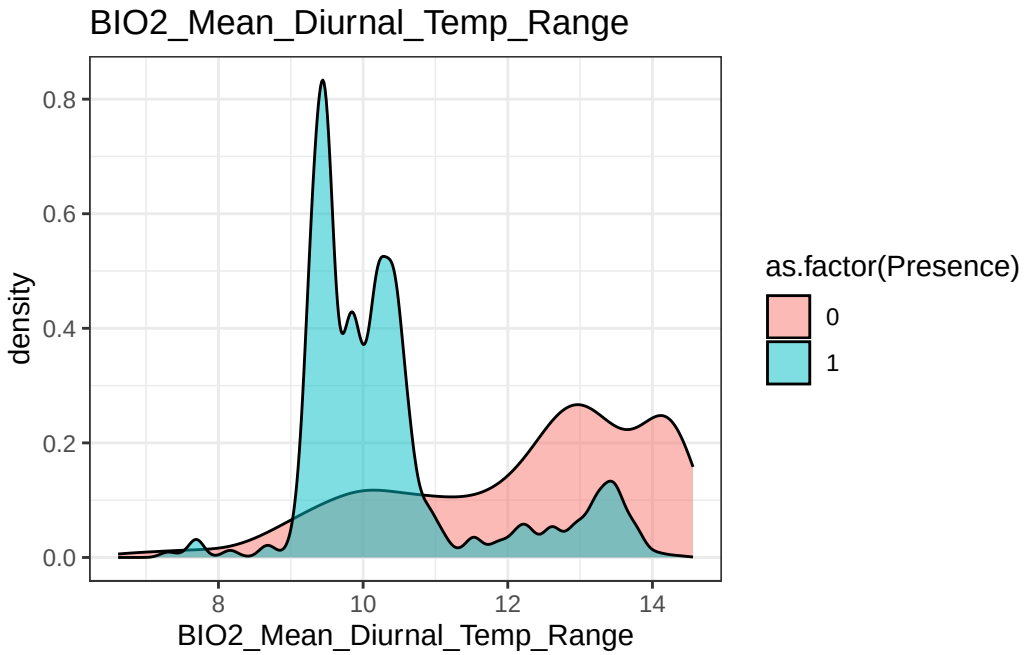
```
                    Variables      VIF
1 BIO5_Max_Temp_Warmest_Month 2.751370
2 BIO6_Min_Temp_Coldest_Month 4.528719
3  BIO12_Annual_Precipitation 6.888077
4    BIO15_Precip_Seasonality 1.263385
```

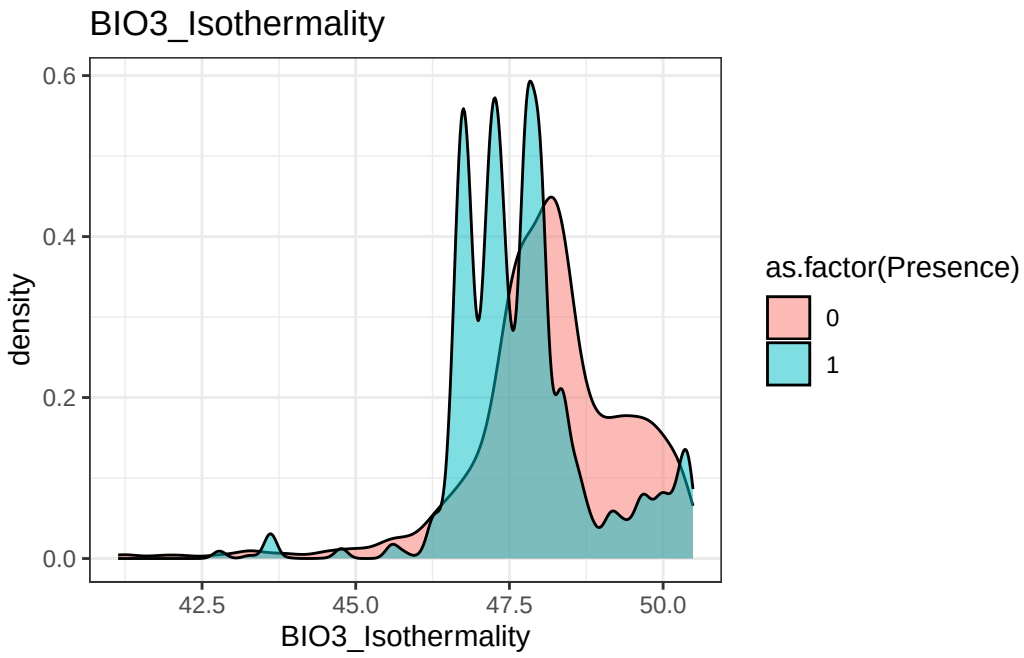
Exploration of the koala presence and background data

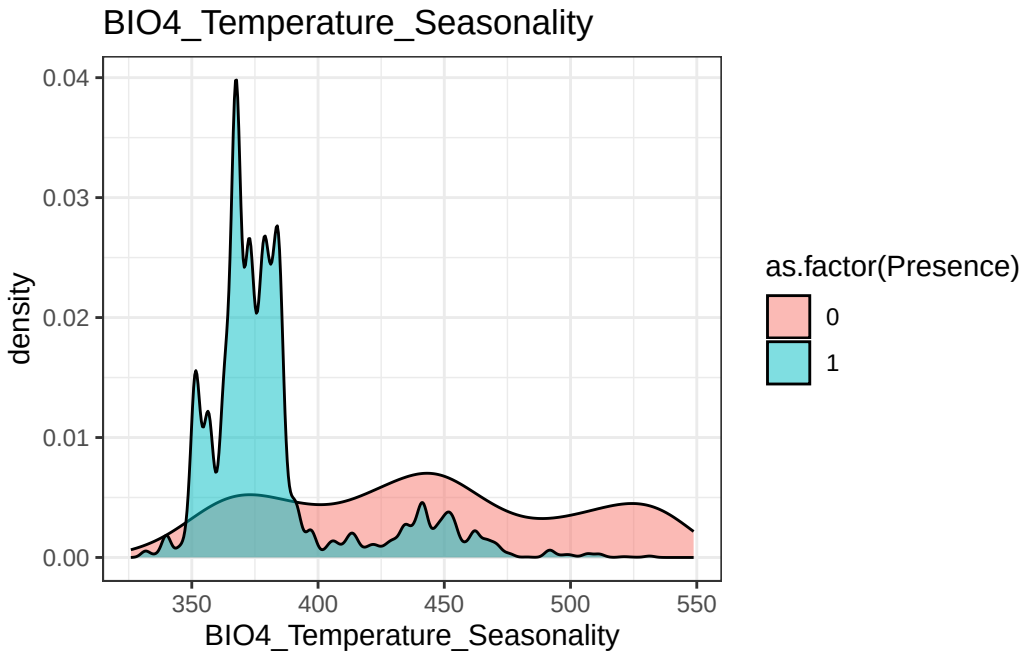
It is good practice to assess where in the environmental space the presence and background points are located. This can help to identify if there are any potential issues with the data, such as a lack of background points in certain areas of environmental space, and should show any patterns in the data that the model should pick up.

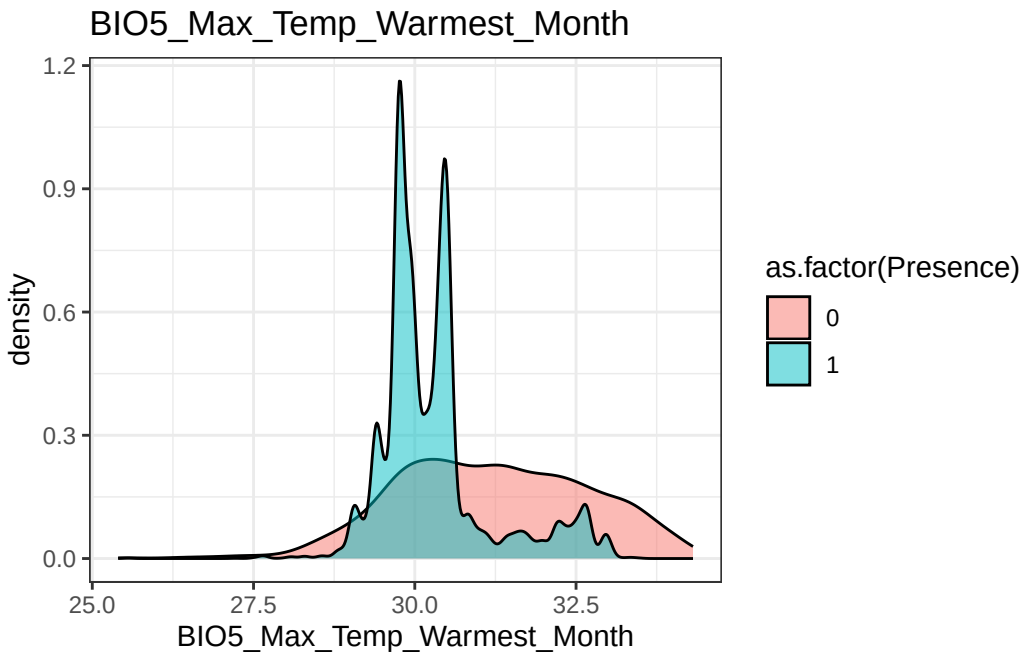
```
# Iterate over all of the variables to create density plots of the background and presence d
for(i in 1:ncol(train_PB_covs_thinned)) {

  print(ggplot() +
          geom_density(data = train_PB_covs_thinned,
                       aes(x = .data[[names(train_PB_covs_thinned)[i]]], fill = as.factor(Pr
                       alpha = 0.5) +
    theme_bw() +
    labs(title = names(train_PB_covs_thinned)[i]))

}
```

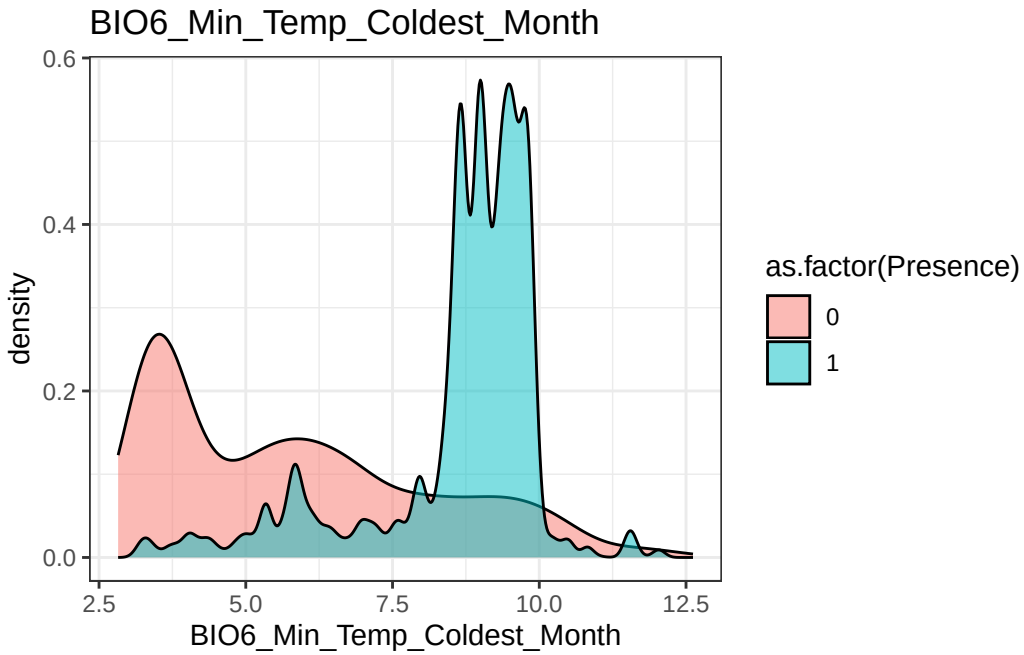


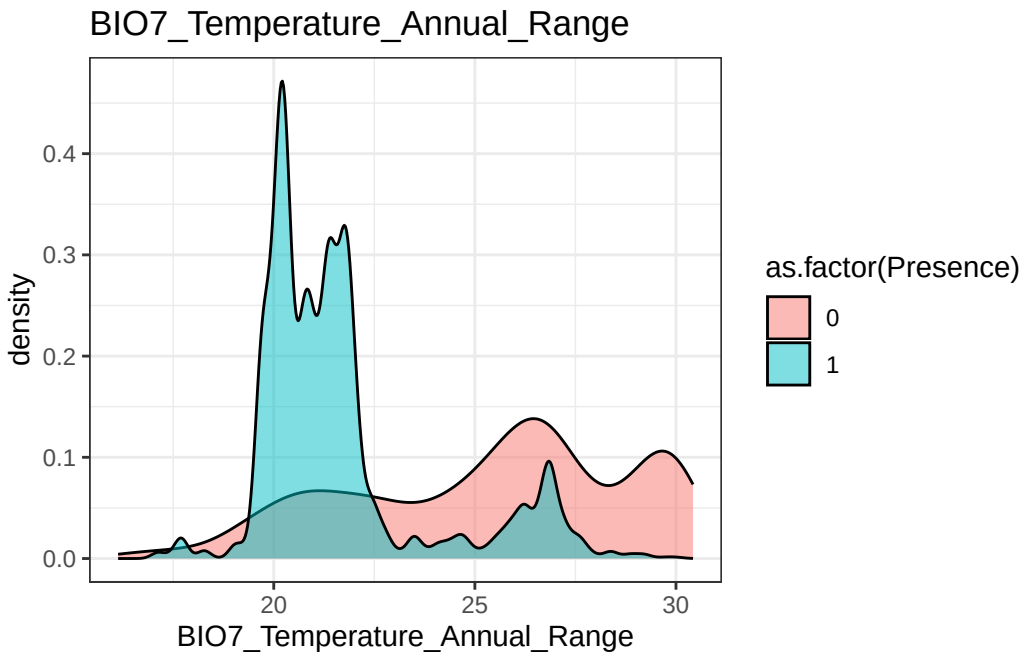

BIO2_Mean_Diurnal_Temp_Range
density
0.0
0.2
0.4
0.6
0.8
8
10
12
14
BIO2_Mean_Diurnal_Temp_Range
as.factor(Presence)
0
1

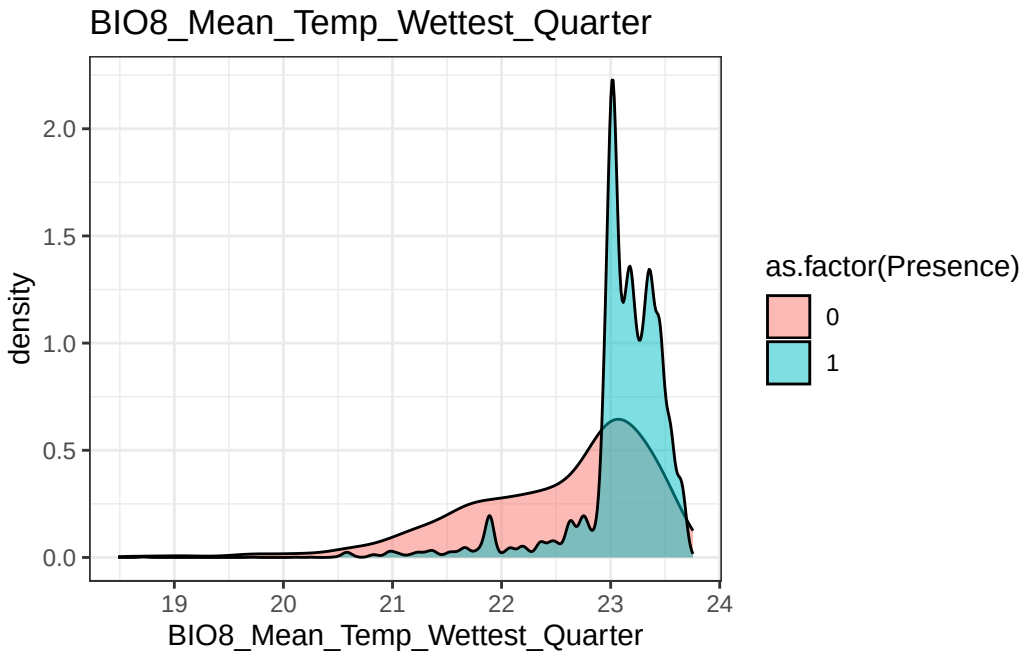

BIO3_Isothermality
density
0.0
0.2
0.4
0.6
42.5
45.0
47.5
50.0
BIO3_Isothermality
as.factor(Presence)
0
1

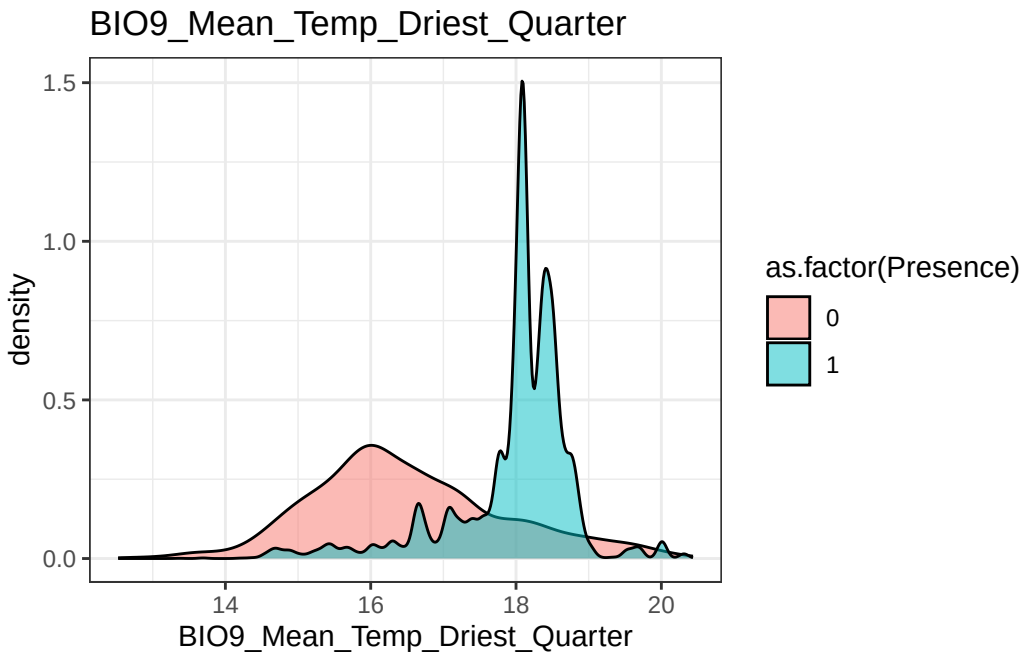

BIO4_Temperature_Seasonality
density
0.04
0.03
0.02
0.01
0.00
350
400
450
500
550
BIO4_Temperature_Seasonality
as.factor(Presence)
0
1

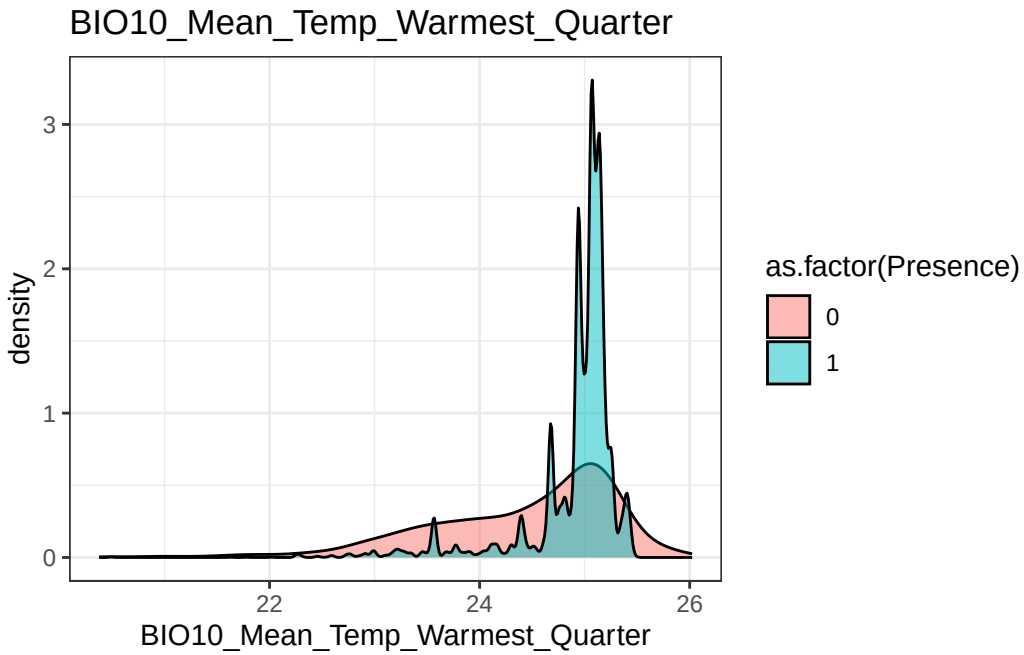

BIO5_Max_Temp_Warmest_Month
density
1.2
0.9
0.6
0.3
0.0
25.0
27.5
30.0
32.5
BIO5_Max_Temp_Warmest_Month
as.factor(Presence)
0
1

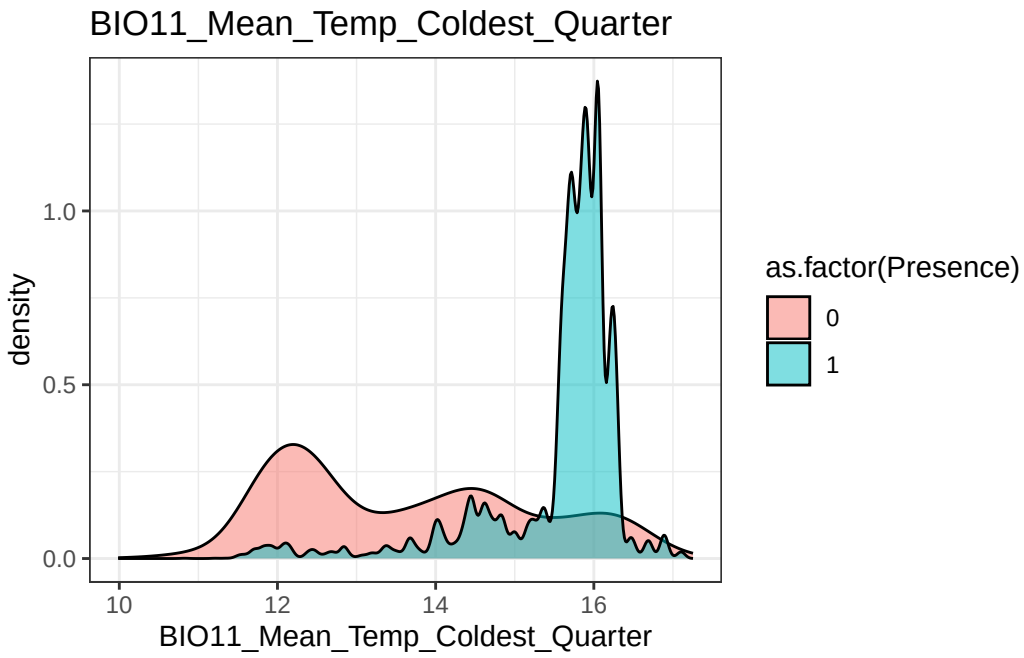

BIO6_Min_Temp_Coldest_Month
density
0.6
0.4
0.2
0.0
2.5
5.0
7.5
10.0
12.5
BIO6_Min_Temp_Coldest_Month
as.factor(Presence)
0
1

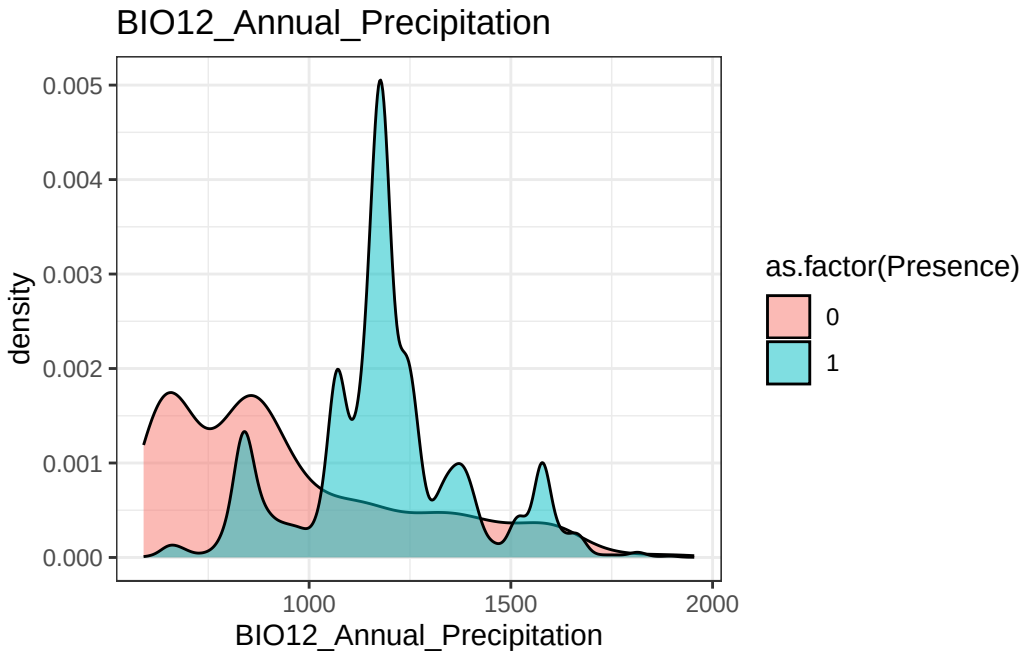

BIO7_Temperature_Annual_Range
density
0.4
0.3
0.2
0.1
0.0
20
25
30
BIO7_Temperature_Annual_Range
as.factor(Presence)
0
1

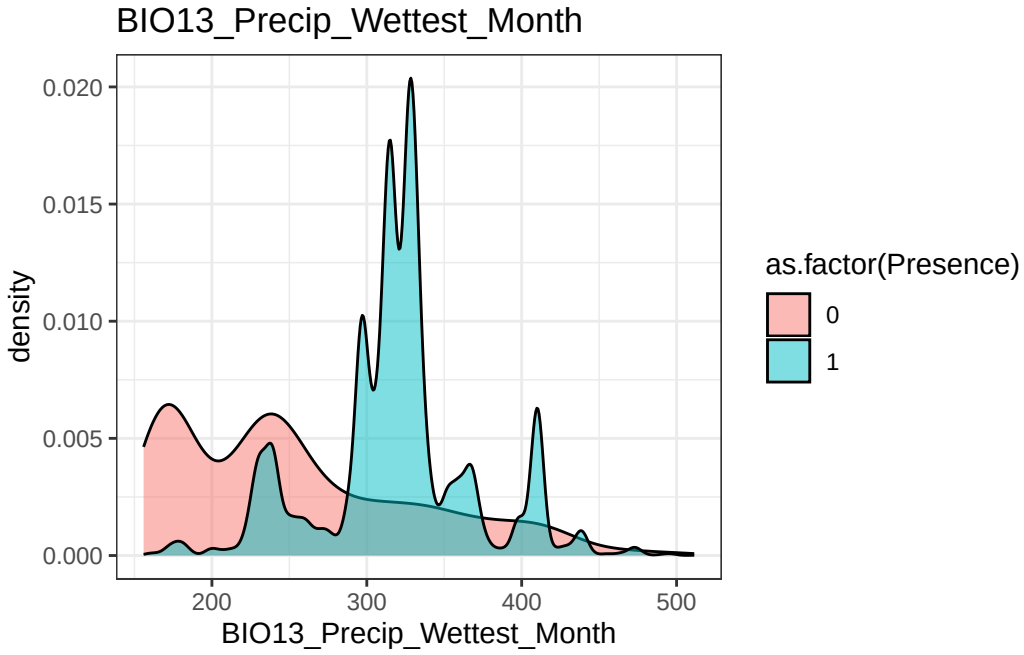

BIO8_Mean_Temp_Wettest_Quarter
density
2.0
1.5
1.0
0.5
0.0
19
20
21
22
23
24
BIO8_Mean_Temp_Wettest_Quarter
as.factor(Presence)
0
1

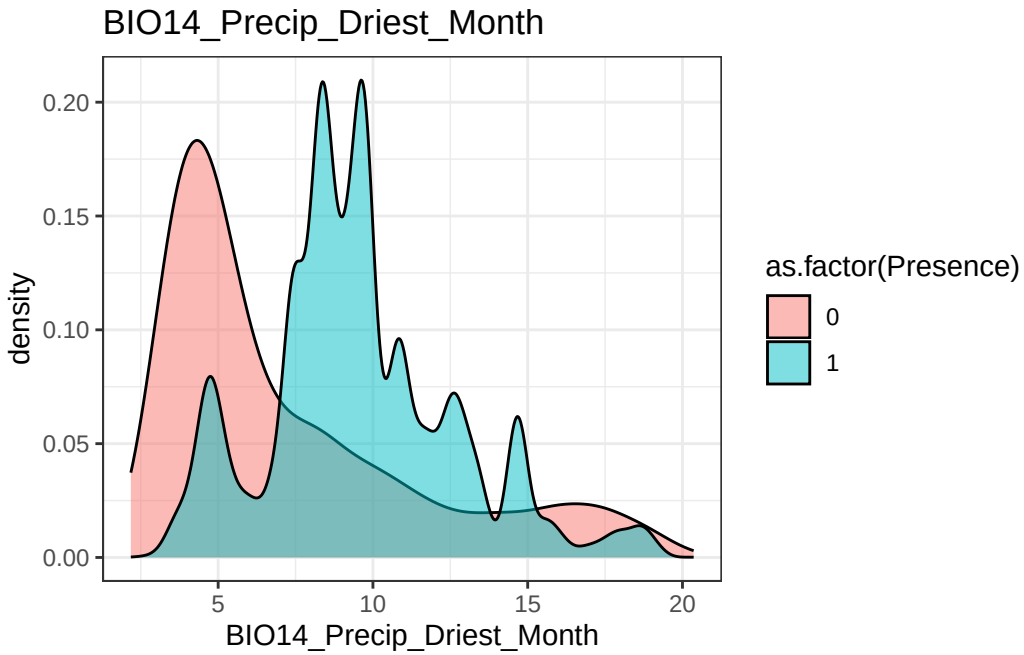

BIO9_Mean_Temp_Driest_Quarter
density
1.5
1.0
0.5
0.0
14
16
18
20
BIO9_Mean_Temp_Driest_Quarter
as.factor(Presence)
0
1

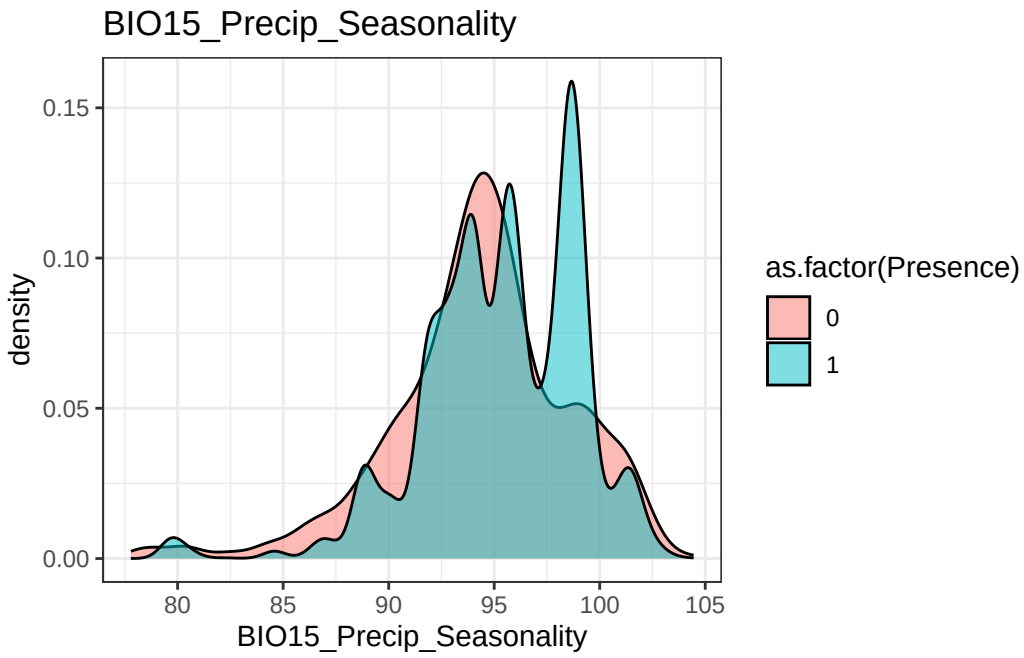

BIO10_Mean_Temp_Warmest_Quarter
density
3
2
1
0
22
24
26
BIO10_Mean_Temp_Warmest_Quarter
as.factor(Presence)
0
1

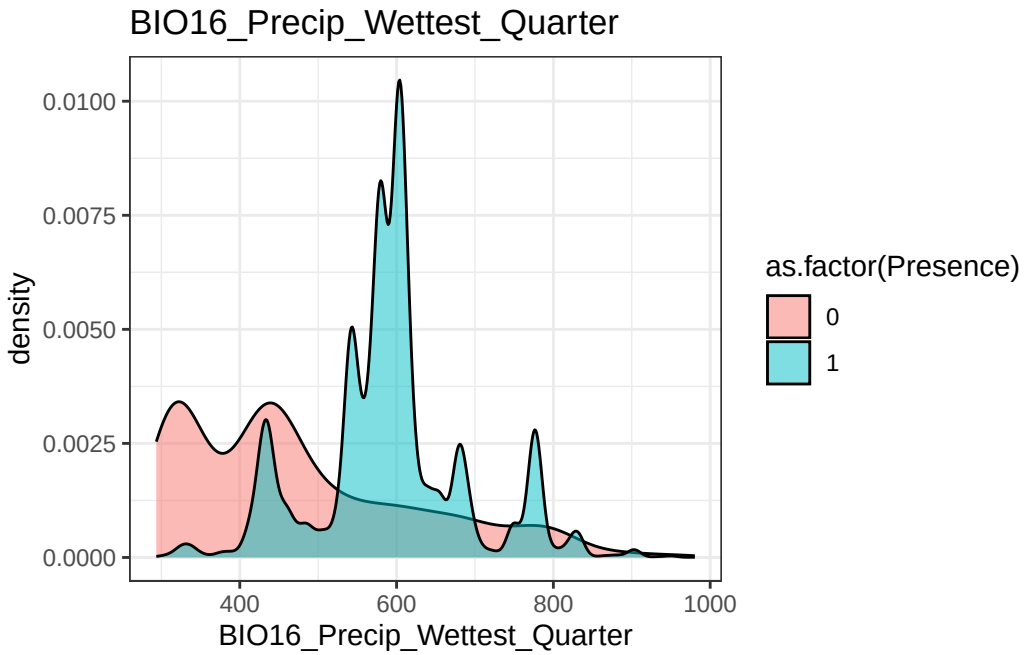

BIO11_Mean_Temp_Coldest_Quarter
density
1.0
0.5
0.0
10
12
14
16
BIO11_Mean_Temp_Coldest_Quarter
as.factor(Presence)
0
1

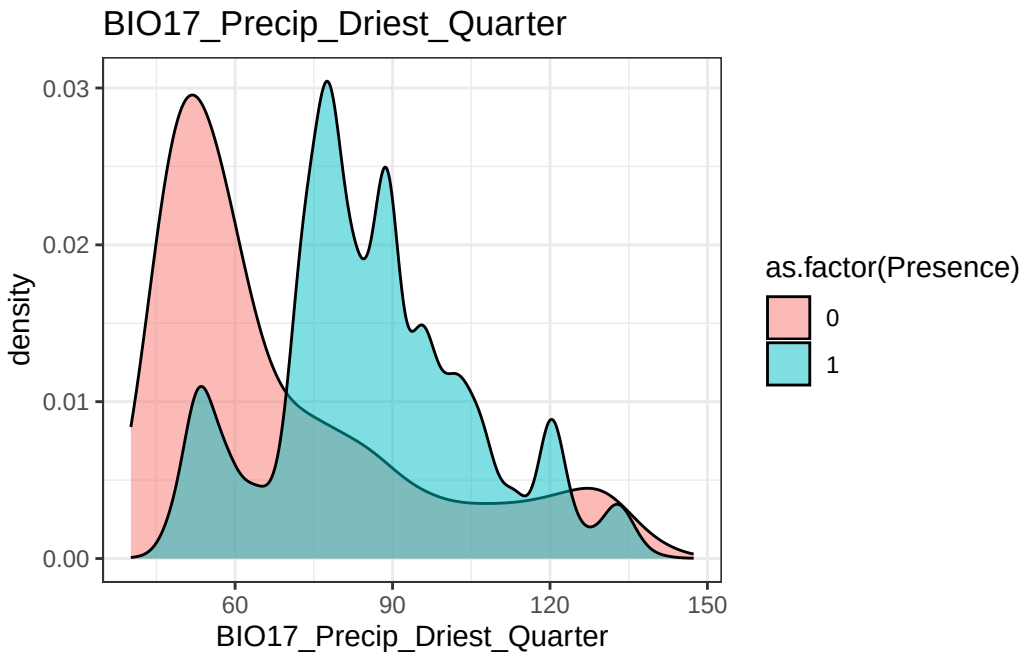

BIO12_Annual_Precipitation
density
0.005
0.004
0.003
0.002
0.001
0.000
1000
1500
2000
BIO12_Annual_Precipitation
as.factor(Presence)
0
1

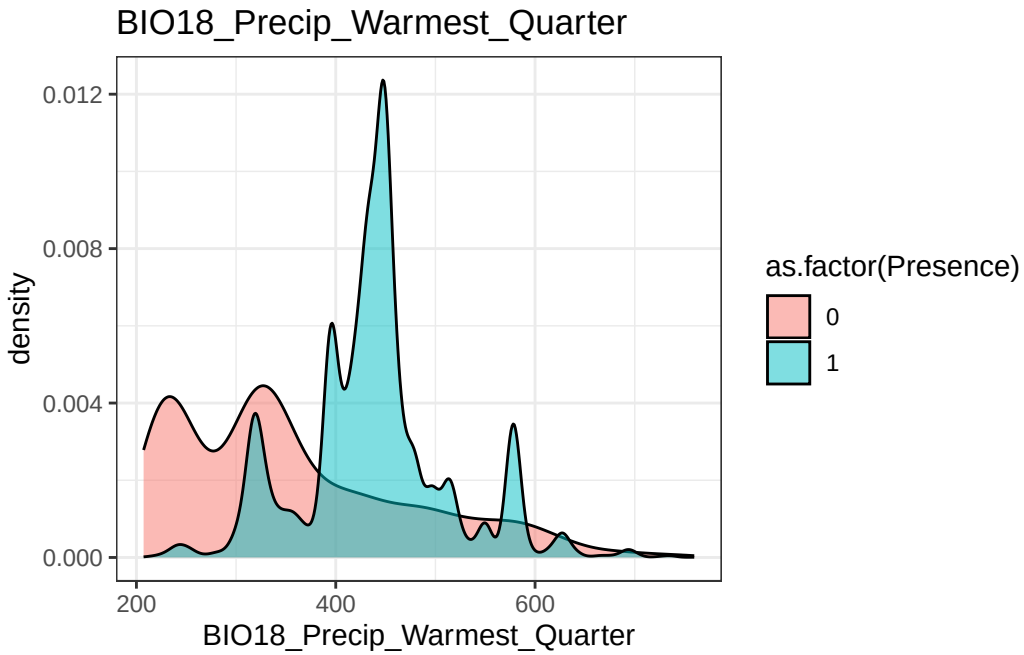

BIO13_Precip_Wettest_Month
density
0.020
0.015
0.010
0.005
0.000
200
300
400
500
BIO13_Precip_Wettest_Month
as.factor(Presence)
0
1

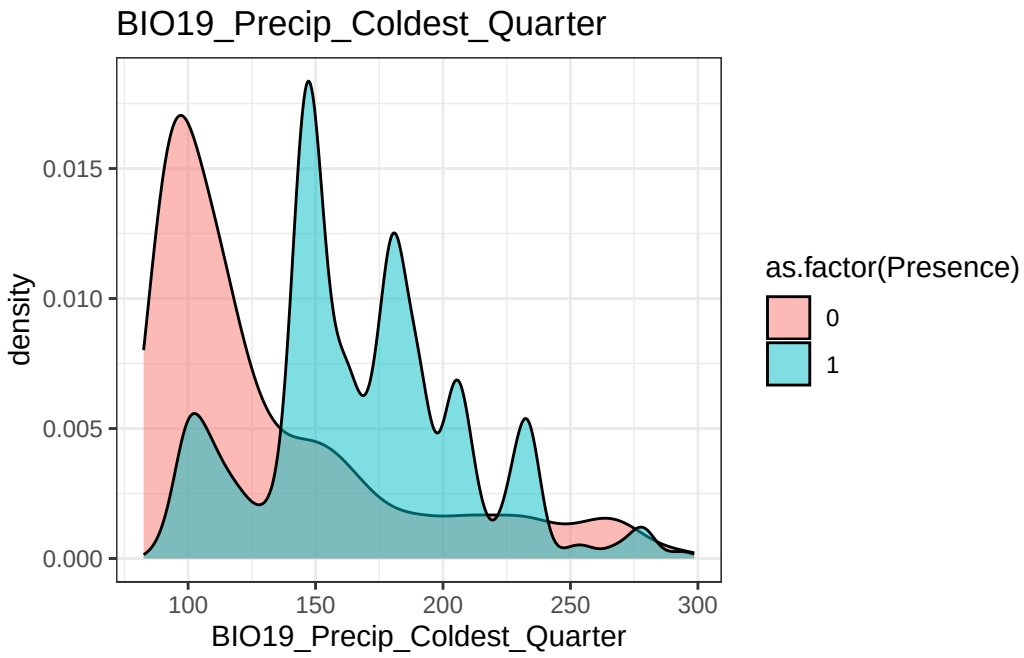

BIO14_Precip_Driest_Month
density
0.20
0.15
0.10
0.05
0.00
5
10
15
20
BIO14_Precip_Driest_Month
as.factor(Presence)
0
1

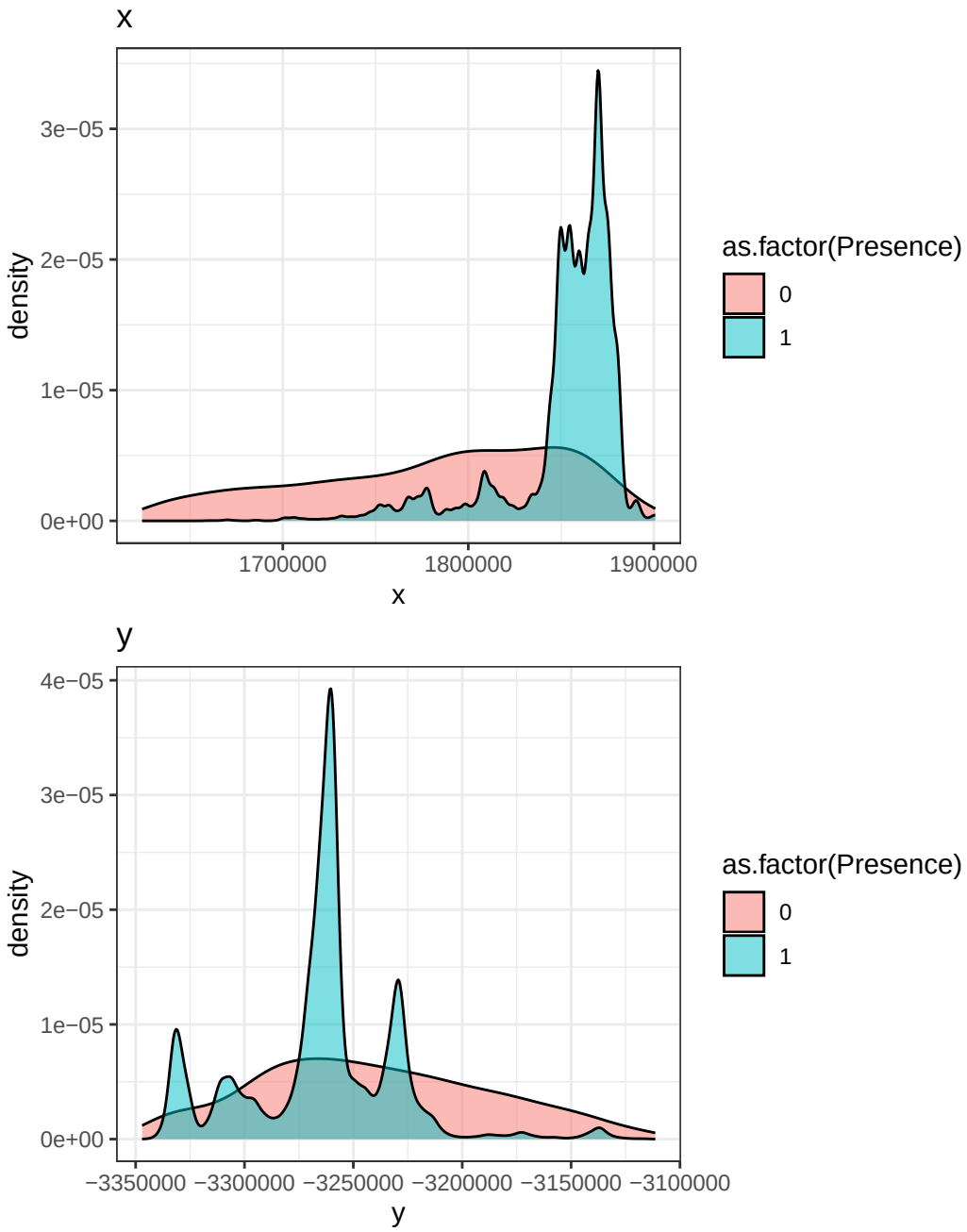

BIO15_Precip_Seasonality
density
0.15
0.10
0.05
0.00
80
85
90
95
100
105
BIO15_Precip_Seasonality
as.factor(Presence)
0
1

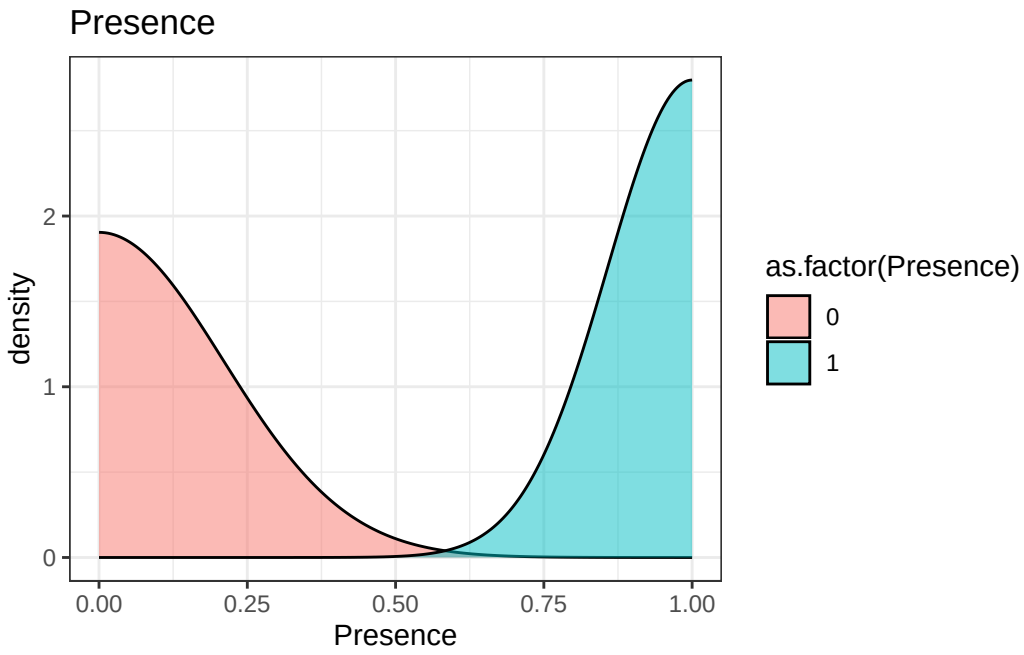

BIO16_Precip_Wettest_Quarter
density
0.0100
0.0075
0.0050
0.0025
0.0000
400
600
800
1000
BIO16_Precip_Wettest_Quarter
as.factor(Presence)
0
1


BIO17_Precip_Driest_Quarter
density
0.03
0.02
0.01
0.00
60
90
120
150
BIO17_Precip_Driest_Quarter
as.factor(Presence)
0
1


BIO18_Precip_Warmest_Quarter
density
0.012
0.008
0.004
0.000
200
400
600
BIO18_Precip_Warmest_Quarter
as.factor(Presence)
0
1


BIO19_Precip_Coldest_Quarter
density
0.015
0.010
0.005
0.000
100
150
200
250
300
BIO19_Precip_Coldest_Quarter
as.factor(Presence)
0
1


x
density
3e−05
2e−05
1e−05
0e+00
1700000
1800000
1900000
x
as.factor(Presence)
0
1
y
density
4e−05
3e−05
2e−05
1e−05
0e+00
−3350000
−3300000
−3250000
−3200000
−3150000
−3100000
y
as.factor(Presence)
0
1



First Model: GLM model fitting

```
# Make a folder to save outputs
dir.create("outputs/GLM_outputs", showWarnings = F)
```

Null model

Null model: no explanatory variables or predictors are included.

It is always helpful to create a null model as a benchmark to assess how the inclusion of explanatory variables improves the model.

```
# Fit a null model with only the intercept
null_model <- glm(Presence ~ 1,
                  data = train_PB_covs,
                  family = binomial(link = "logit"))

# Check the model results
summary(null_model)
```

```
Call:
glm(formula = Presence ~ 1, family = binomial(link = "logit"),
```

```
    data = train_PB_covs)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.08096    0.02636   154.8   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 14902  on 88135  degrees of freedom
Residual deviance: 14902  on 88135  degrees of freedom
AIC: 14904

Number of Fisher Scoring iterations: 7
```

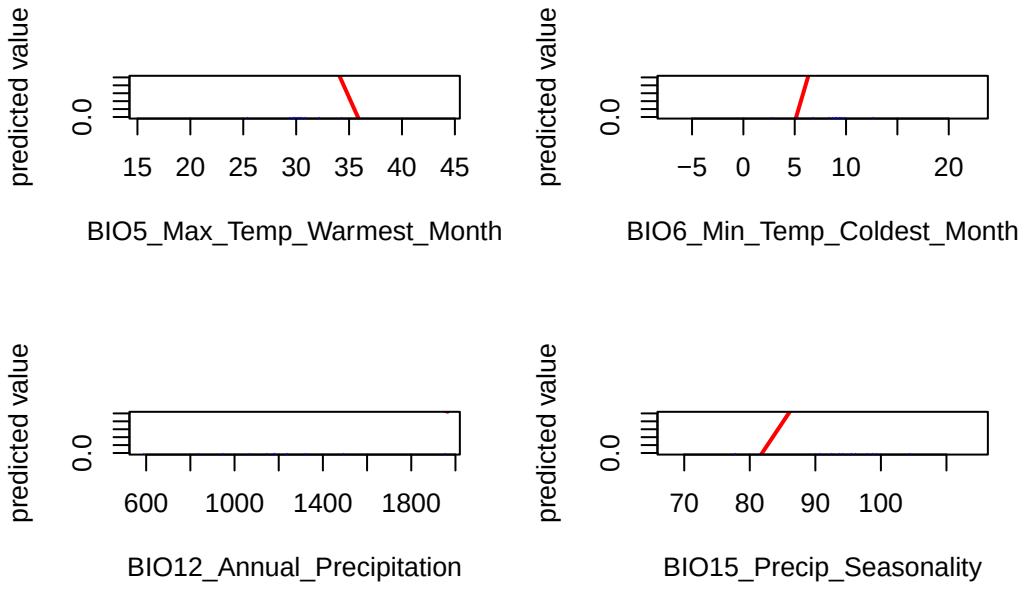
GLM Model 1 - expert variables

```
glm_model_1 <- glm(Presence ~
                     BIO5_Max_Temp_Warmest_Month +
                     BIO6_Min_Temp_Coldest_Month +
                     BIO12_Annual_Precipitation +
                     BIO15_Precip_Seasonality,
                   data=train_PB_covs_thinned,
                   family = binomial(link = "logit"))

# Check the model results
summary(glm_model_1)
```

```
Call:
glm(formula = Presence ~ BIO5_Max_Temp_Warmest_Month + BIO6_Min_Temp_Coldest_Month +
    BIO12_Annual_Precipitation + BIO15_Precip_Seasonality, family = binomial(link = "logit"),
    data = train_PB_covs_thinned)

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                -6.651059   1.562034  -4.258 2.06e-05 ***
BIO5_Max_Temp_Warmest_Month -0.596618   0.051512 -11.582  < 2e-16 ***
BIO6_Min_Temp_Coldest_Month  0.880896   0.028613  30.786  < 2e-16 ***
BIO12_Annual_Precipitation  -0.002949   0.000275 -10.724  < 2e-16 ***
BIO15_Precip_Seasonality     0.247330   0.010880  22.733  < 2e-16 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8758.5  on 11463  degrees of freedom
Residual deviance: 5710.0  on 11459  degrees of freedom
AIC: 5720

Number of Fisher Scoring iterations: 6
```

```r
# These response curves don't look very helpful
dismo::response(glm_model_1)
```



GLM Model 2 - expert variables with quadratics

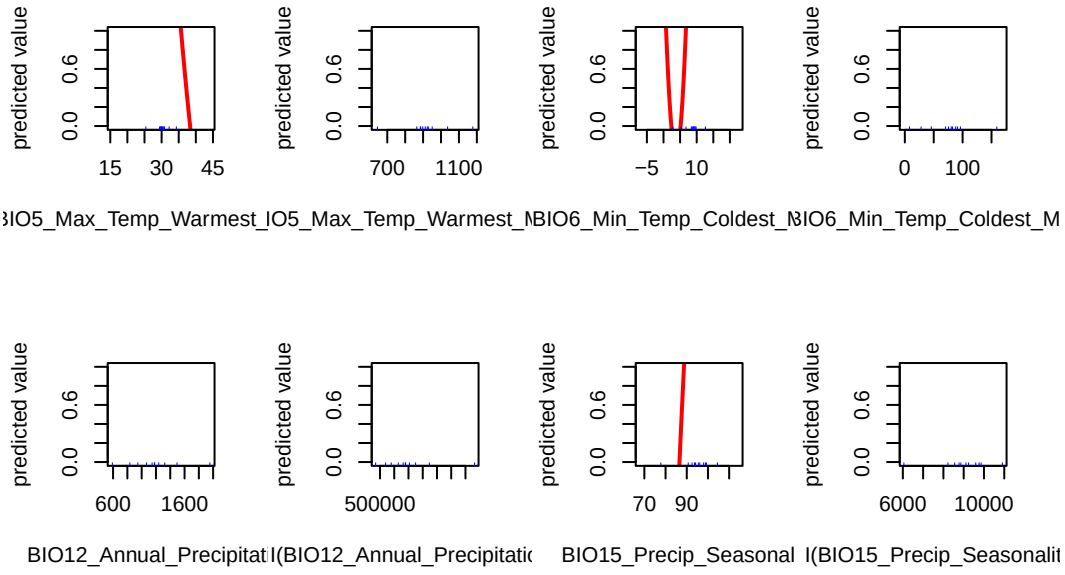
In this model, we include quadratic terms for the covariates. This is a common approach in species distribution modelling to account for non-linear relationships between the predictors and the response variable. This increases the complexity of the model and allows for more flexibility in fitting the data.

```r
glm_model_2 <- glm(Presence ~
                     BIO5_Max_Temp_Warmest_Month + I(BIO5_Max_Temp_Warmest_Month^2) +
                     BIO6_Min_Temp_Coldest_Month + I(BIO6_Min_Temp_Coldest_Month^2) +
                     BIO12_Annual_Precipitation + I(BIO12_Annual_Precipitation^2) +
                     BIO15_Precip_Seasonality + I(BIO15_Precip_Seasonality^2),
                   data=train_PB_covs_thinned,
                   family = binomial(link = "logit"))
```

```r
# Check the model results
summary(glm_model_2)
```

```
Call:
glm(formula = Presence ~ BIO5_Max_Temp_Warmest_Month + I(BIO5_Max_Temp_Warmest_Month^2) +
    BIO6_Min_Temp_Coldest_Month + I(BIO6_Min_Temp_Coldest_Month^2) +
    BIO12_Annual_Precipitation + I(BIO12_Annual_Precipitation^2) +
    BIO15_Precip_Seasonality + I(BIO15_Precip_Seasonality^2),
    family = binomial(link = "logit"), data = train_PB_covs_thinned)

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -8.396e+01  1.828e+01  -4.593 4.37e-06 ***
BIO5_Max_Temp_Warmest_Month    -1.334e+00  1.180e+00  -1.130   0.2585
I(BIO5_Max_Temp_Warmest_Month^2)  1.292e-02  1.910e-02   0.676   0.4988
BIO6_Min_Temp_Coldest_Month    -1.096e+00  1.914e-01  -5.728 1.02e-08 ***
I(BIO6_Min_Temp_Coldest_Month^2)  1.456e-01  1.293e-02  11.260  < 2e-16 ***
BIO12_Annual_Precipitation      7.378e-04  1.941e-03   0.380   0.7039
I(BIO12_Annual_Precipitation^2) -1.266e-06  6.923e-07  -1.828   0.0675 .
BIO15_Precip_Seasonality        2.084e+00  2.380e-01   8.758  < 2e-16 ***
I(BIO15_Precip_Seasonality^2)  -9.202e-03  1.269e-03  -7.253 4.09e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8758.5  on 11463  degrees of freedom
Residual deviance: 5546.0  on 11455  degrees of freedom
AIC: 5564

Number of Fisher Scoring iterations: 6
```

```r
# These response curves don't look very helpful
dismo::response(glm_model_2)
```



Model effect evaluation

Here we use a function presented in an EcoCommons Australia notebook to evaluate the model performance. The notebook can be found on their GitHub: https://github.com/EcoCommonsAustralia/n

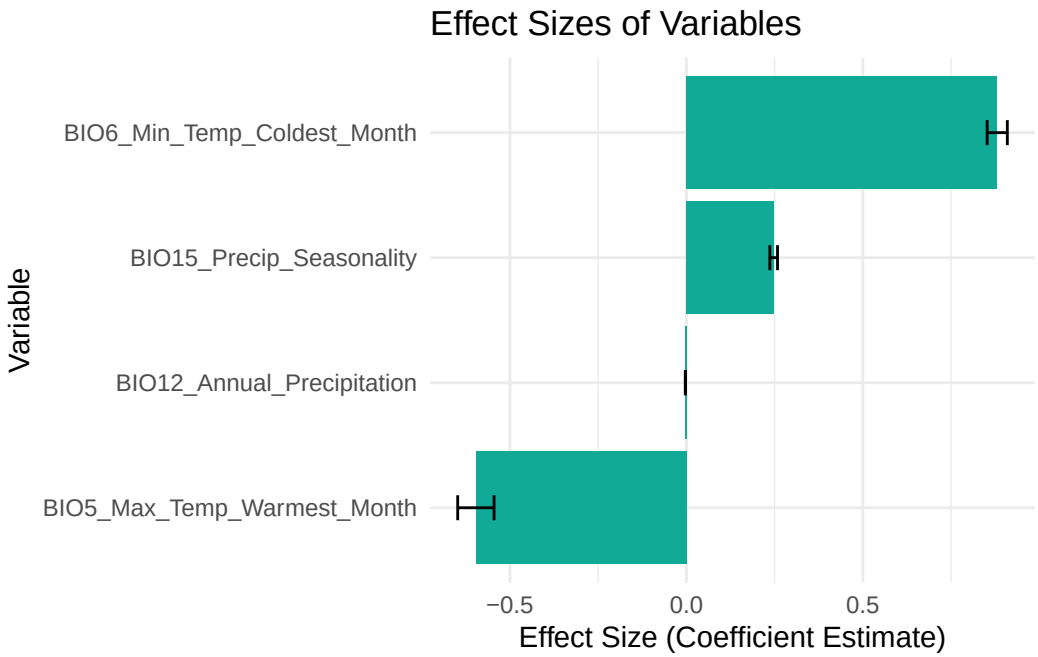
```r
# Function to plot effect size graph
plot_effect_size <- function(glm_model) {
  # Check if required libraries are installed
  if (!requireNamespace("ggplot2", quietly = TRUE)) {
    stop("Please install the 'ggplot2' package to use this function.")
  }
  library(ggplot2)

  # Extract effect sizes (coefficients) from the model
  coefs <- summary(glm_model)$coefficients
  effect_sizes <- data.frame(
    Variable = rownames(coefs)[-1],  # Exclude the intercept
    Effect_Size = coefs[-1, "Estimate"],
    Std_Error = coefs[-1, "Std. Error"]
  )

  # Sort by effect size
  effect_sizes <- effect_sizes[order(-abs(effect_sizes$Effect_Size)), ]

  # Plot the effect sizes with error bars
  ggplot(effect_sizes, aes(x = reorder(Variable, Effect_Size), y = Effect_Size)) +
    geom_bar(stat = "identity", fill = "#11aa96") +
```

```
    geom_errorbar(aes(ymin = Effect_Size - Std_Error, ymax = Effect_Size + Std_Error), width
    coord_flip() +
    labs(
      title = "Effect Sizes of Variables",
      x = "Variable",
      y = "Effect Size (Coefficient Estimate)"
    ) +
    theme_minimal()
}
```

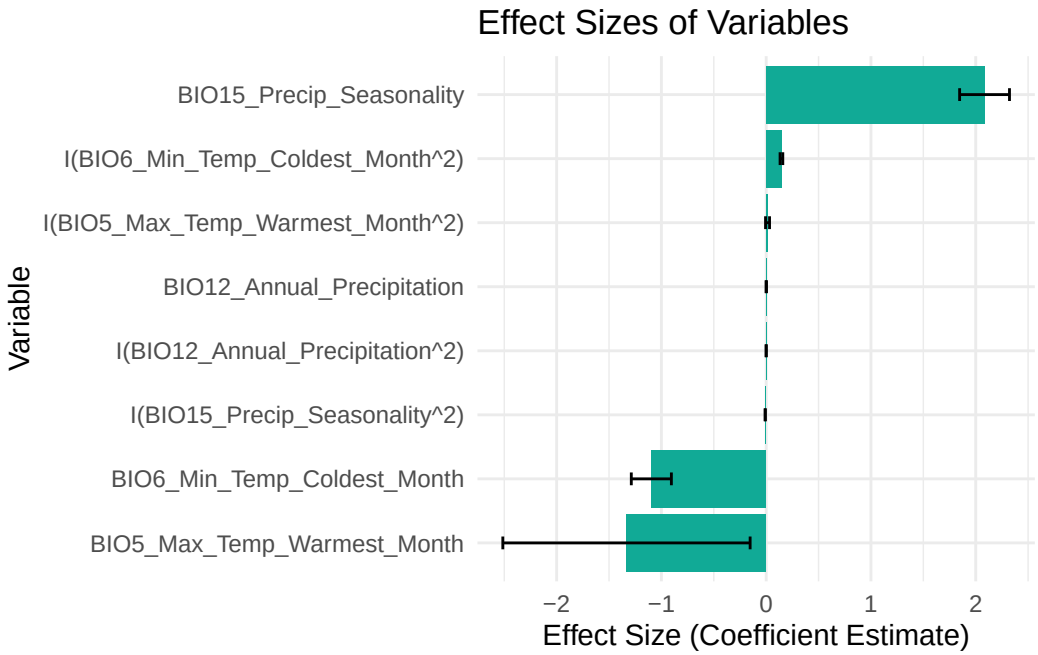
Use the function to check the effect sizes

We need to be careful when interpreting the effect sizes of models with quadratic terms however, as the response curve depends on the linear and the quadratic term.

```
# Example usage of effect size plot
plot_effect_size(glm_model_1)
```




```
plot_effect_size(glm_model_2)
```



Response curves

Again, we can use a function from the EcoCommons notebook to plot the response curves from the model, although for quadratics we need to adjust the function or use something else.

```
plot_species_response <- function(glm_model, predictors, data) {
  # Check if required libraries are installed
  if (!requireNamespace("ggplot2", quietly = TRUE) || !requireNamespace("gridExtra", quietly
    stop("Please install the 'ggplot2' and 'gridExtra' packages to use this function.")
  }
  library(ggplot2)
  library(gridExtra)

  # Create empty list to store response plots
  response_plots <- list()

  # Loop through each predictor variable
  for (predictor in predictors) {
    # Create new data frame to vary predictor while keeping others constant
    pred_range <- seq(
      min(data[[predictor]], na.rm = TRUE),
      max(data[[predictor]], na.rm = TRUE),
      length.out = 100
    )
```

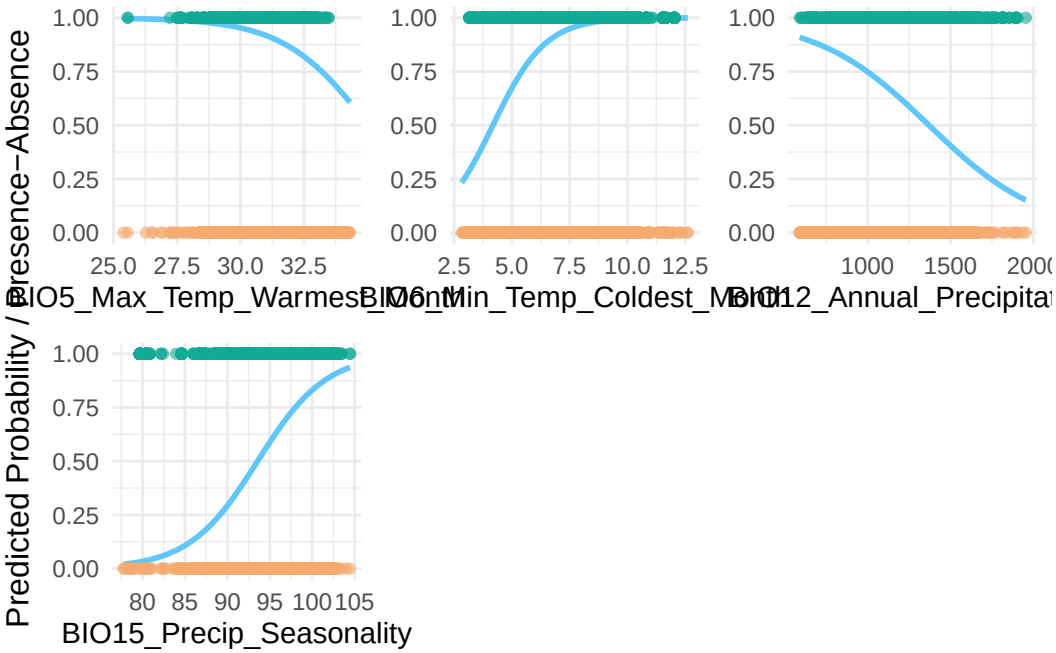
```r
  const_data <- data[1, , drop = FALSE]  # Use first row to keep other predictors constant
  response_data <- const_data[rep(1, 100), ]  # Duplicate the row
  response_data[[predictor]] <- pred_range

  # Predict probabilities
  predicted_response <- predict(glm_model, newdata = response_data, type = "response")

  # Create data frame for plotting
  plot_data <- data.frame(
    Predictor_Value = pred_range,
    Predicted_Probability = predicted_response
  )

  # Add presence and absence data
  presence_absence_data <- data.frame(
    Predictor_Value = data[[predictor]],
    Presence_Absence = data$Presence
  )

  # Generate the response plot
  p <- ggplot() +

    geom_line(data = plot_data,
              aes(x = Predictor_Value, y = Predicted_Probability),
              color = "#61c6fa", linewidth = 1) +

    geom_point(data = presence_absence_data[presence_absence_data$Presence_Absence == 1, ],
               aes(x = Predictor_Value, y = Presence_Absence),
               color = "#11aa96", alpha = 0.6) +

    geom_point(data = presence_absence_data[presence_absence_data$Presence_Absence == 0, ],
               aes(x = Predictor_Value, y = Presence_Absence),
               color = "#f6aa70", alpha = 0.6) +

    labs(x = predictor, y = NULL) +
    theme_minimal() +
    theme(axis.title.y = element_blank())

  # Store the plot in the list
  response_plots[[predictor]] <- p
}

# Arrange all plots in one combined plot with a single shared y-axis label
grid.arrange(
```

```
    grobs = response_plots,
    ncol = 3,
    left = "Predicted Probability / Presence-Absence"
  )
}
```

Use the response curve function

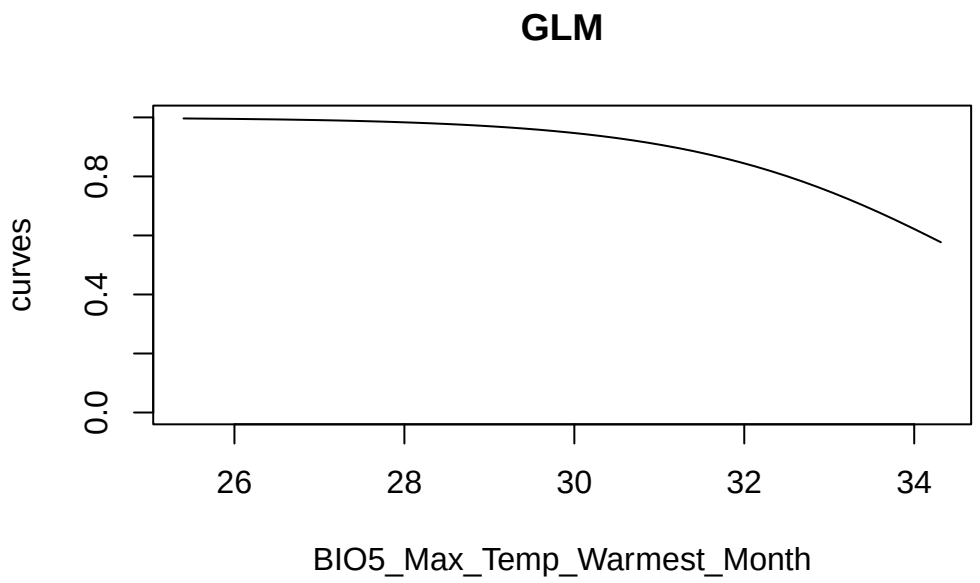
```
predictors <- c("BIO5_Max_Temp_Warmest_Month",
                "BIO6_Min_Temp_Coldest_Month",
                "BIO12_Annual_Precipitation",
                "BIO15_Precip_Seasonality")

plot_species_response(glm_model_1, predictors, train_PB_covs_thinned)
```

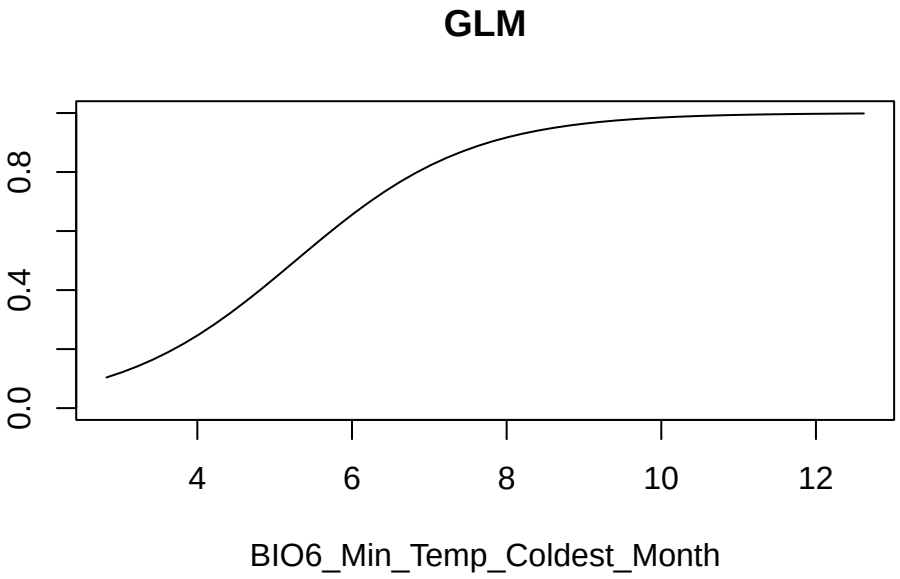
```
Attaching package: 'gridExtra'

The following object is masked from 'package:randomForest':

    combine

The following object is masked from 'package:dplyr':

    combine
```

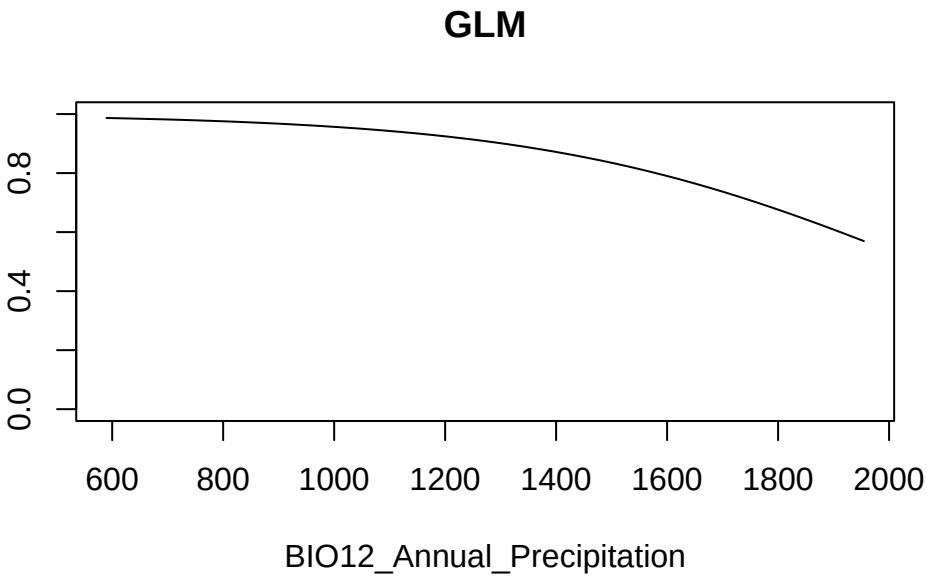


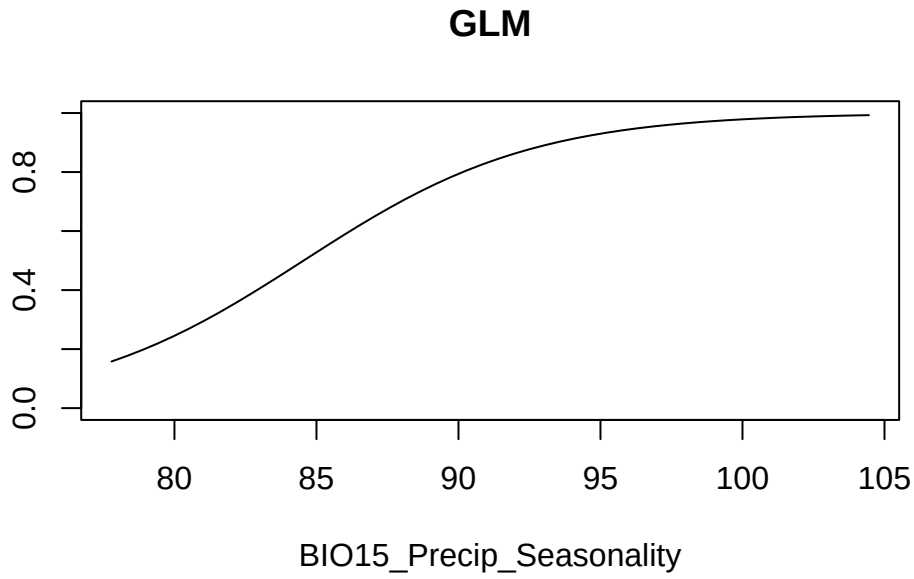
Model 1 partial responses

```
# Plot the partial responses
partial_response(glm_model_1, predictors = train_PB_covs_thinned[,predictors], main='GLM')
```

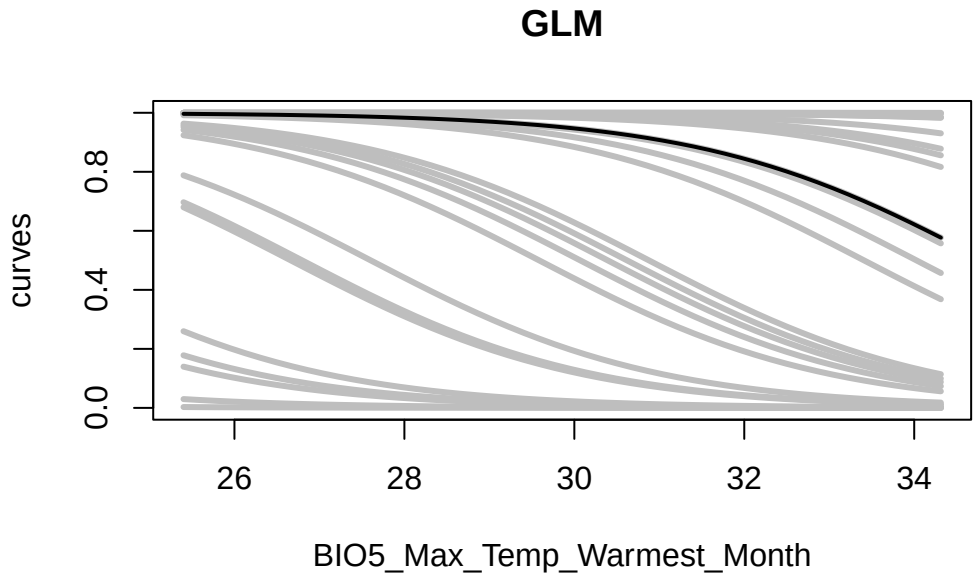


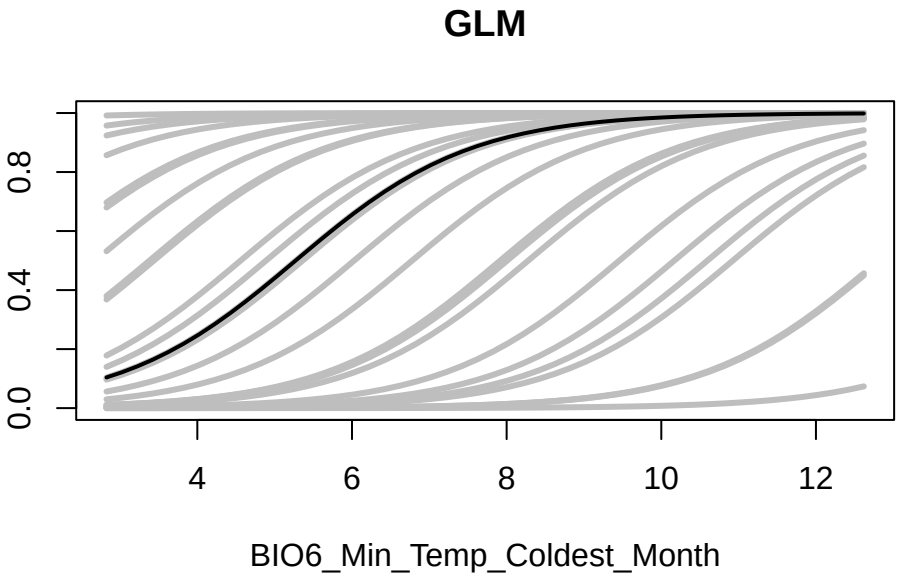

GLM
0.0
0.4
0.8
4
6
8
10
12
BIO6_Min_Temp_Coldest_Month

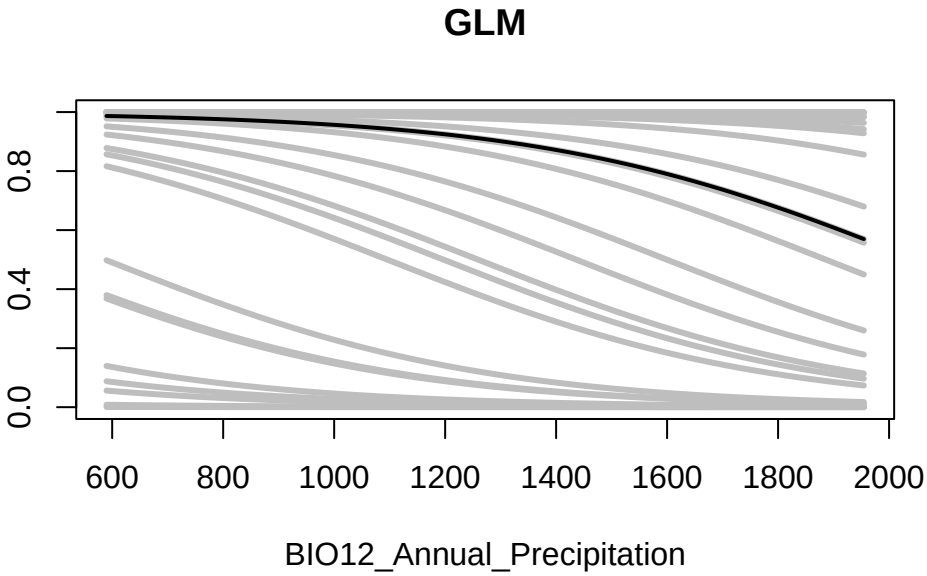

GLM
0.0
0.4
0.8
600
800
1000
1200
1400
1600
1800
2000
BIO12_Annual_Precipitation

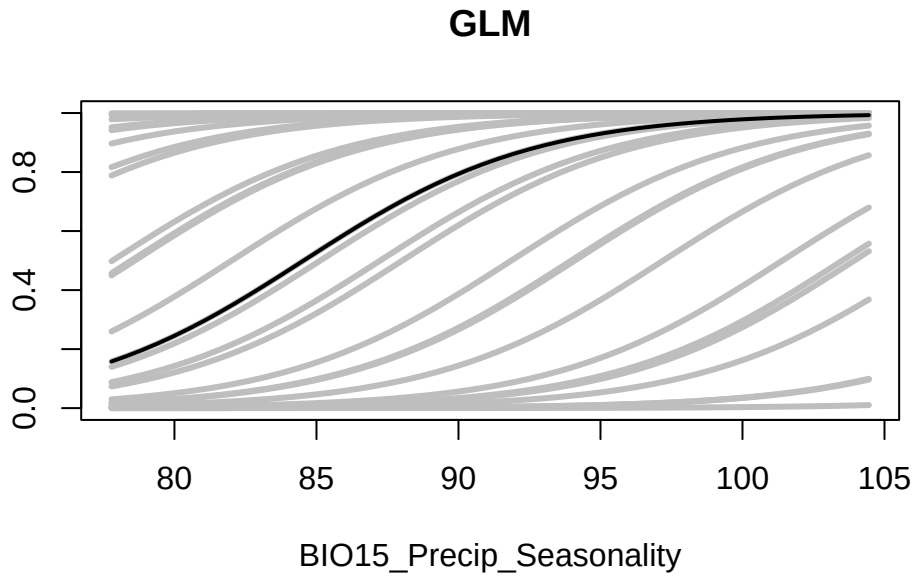



```
# Plot inflated response curves:
inflated_response(glm_model_1, predictors = train_PB_covs_thinned[,predictors], method = "st
```



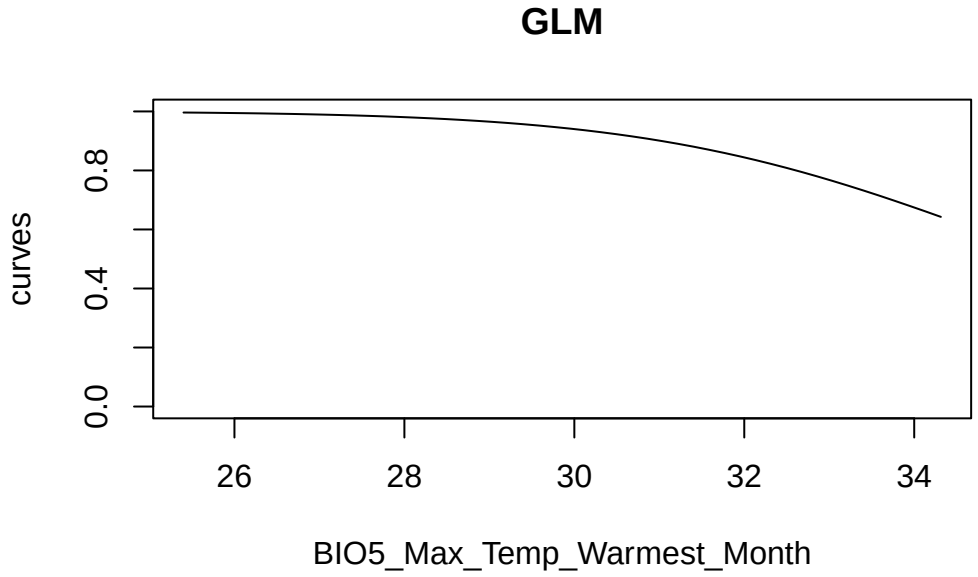

GLM
0.0
0.4
0.8
4
6
8
10
12
BIO6_Min_Temp_Coldest_Month

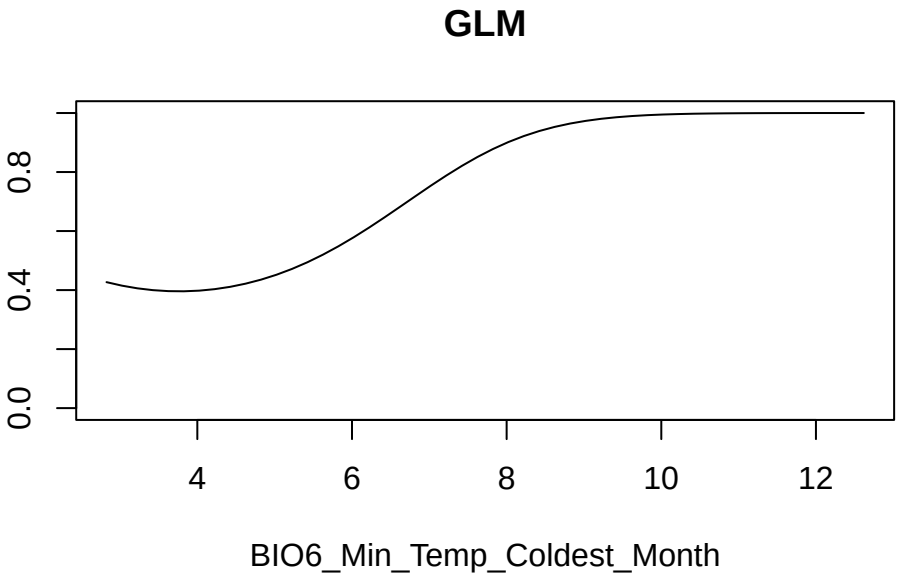

GLM
0.0
0.4
0.8
600
800
1000
1200
1400
1600
1800
2000
BIO12_Annual_Precipitation

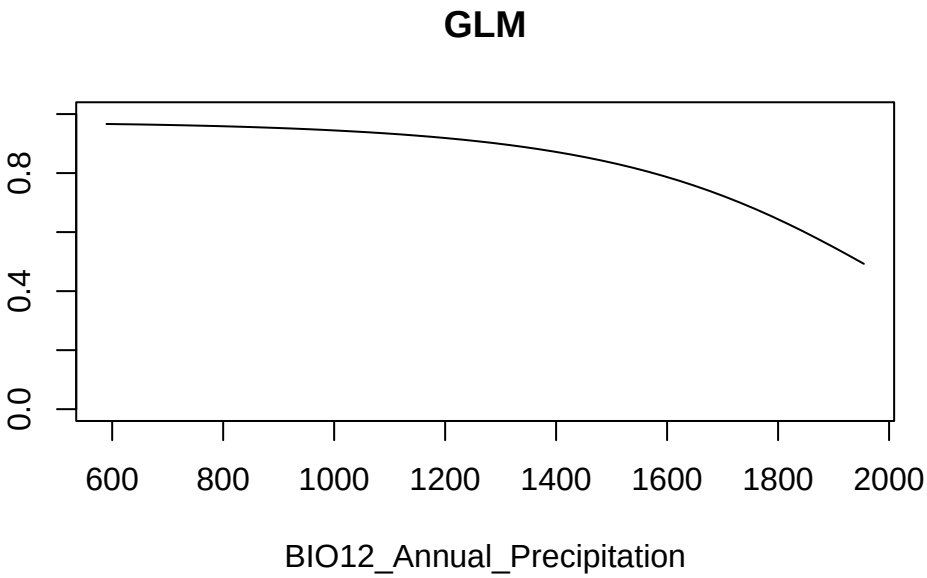


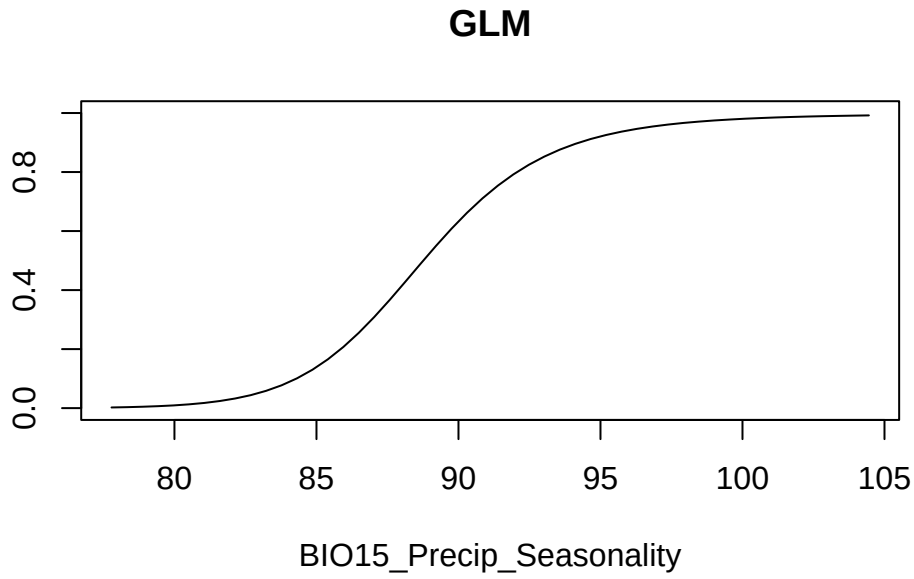
Model 2 partial responses

```
# Plot the partial responses
partial_response(glm_model_2, predictors = train_PB_covs_thinned[,predictors], main='GLM')
```

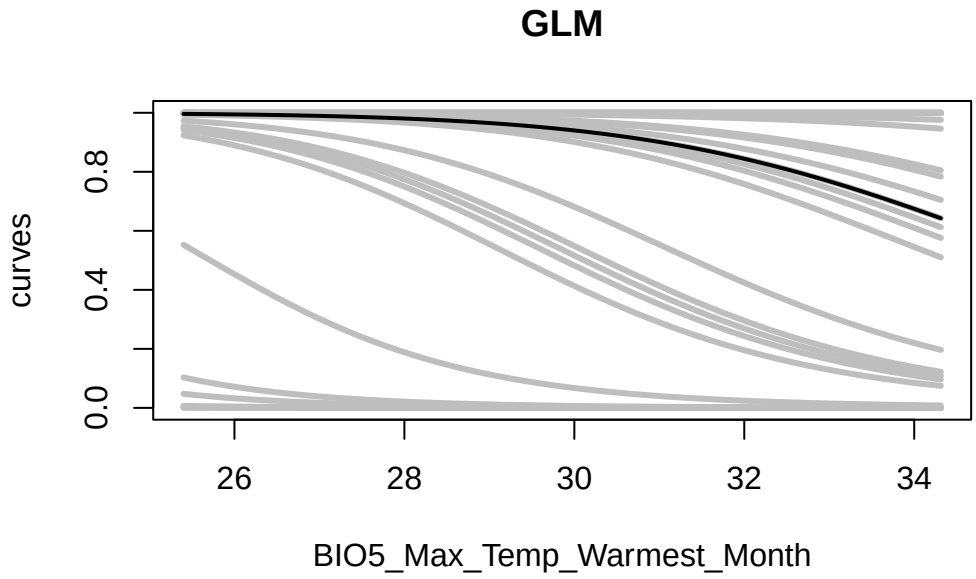


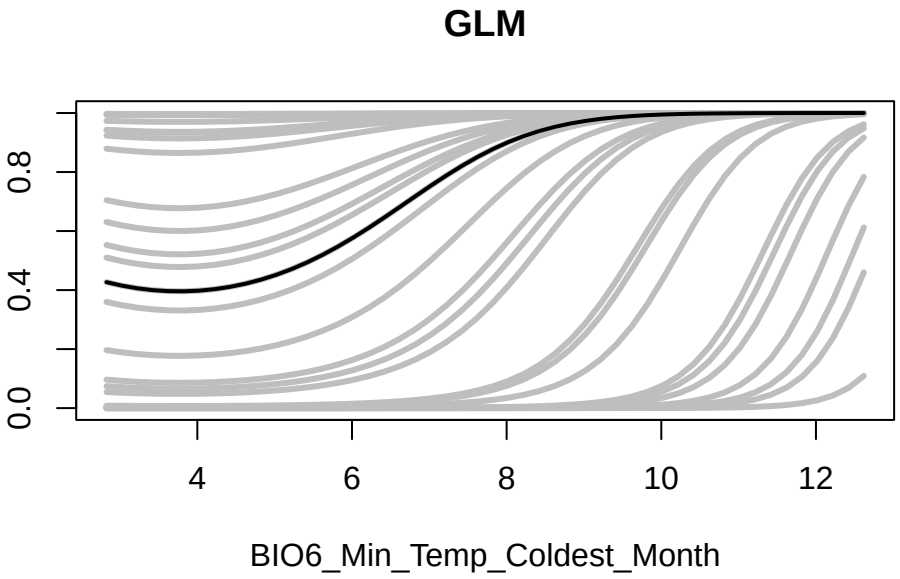

GLM
0.0
0.4
0.8
4
6
8
10
12
BIO6_Min_Temp_Coldest_Month

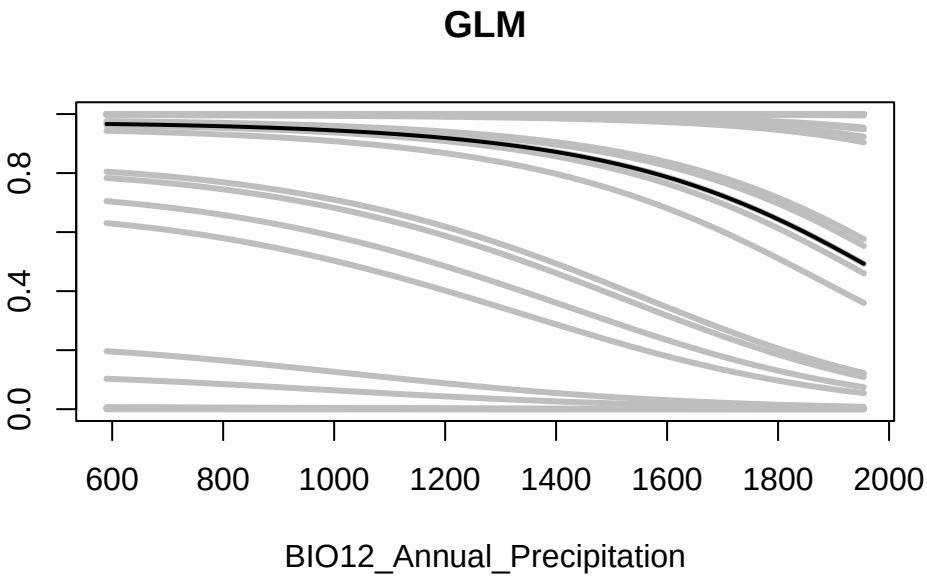

GLM
0.0
0.4
0.8
600
800
1000
1200
1400
1600
1800
2000
BIO12_Annual_Precipitation

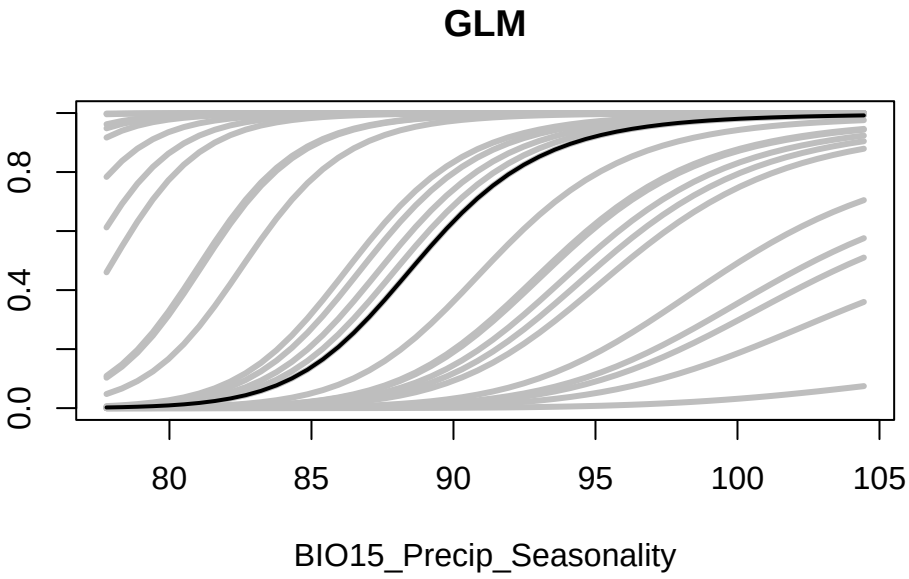



```
# Plot inflated response curves:
inflated_response(glm_model_2, predictors = train_PB_covs_thinned[,predictors], method = "st
```




GLM
0.0
0.4
0.8
4
6
8
10
12
BIO6_Min_Temp_Coldest_Month

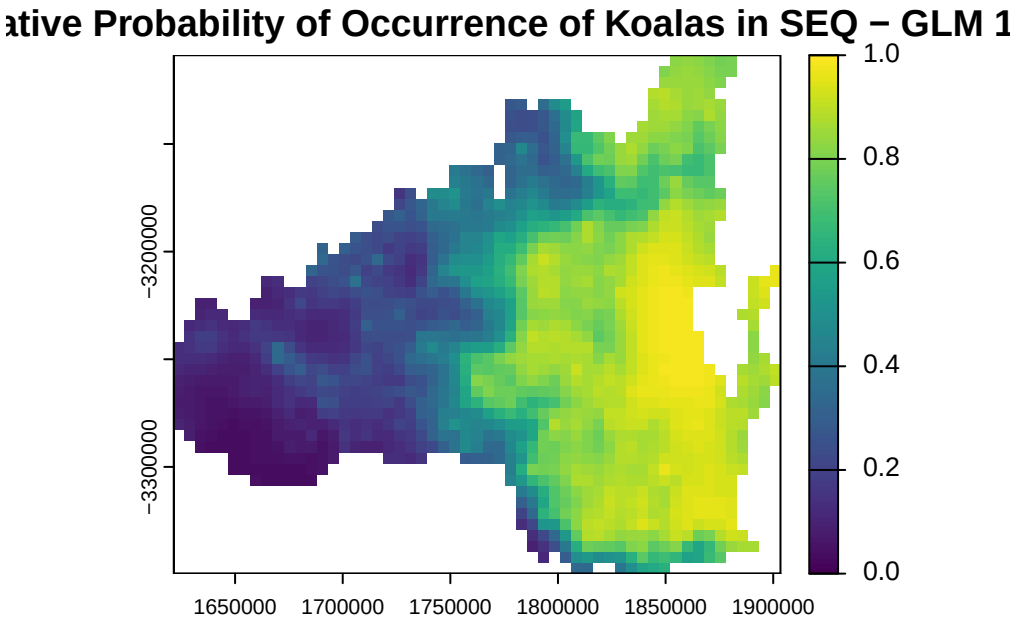

GLM
0.0
0.4
0.8
600
800
1000
1200
1400
1600
1800
2000
BIO12_Annual_Precipitation



GLM predictions to current environment

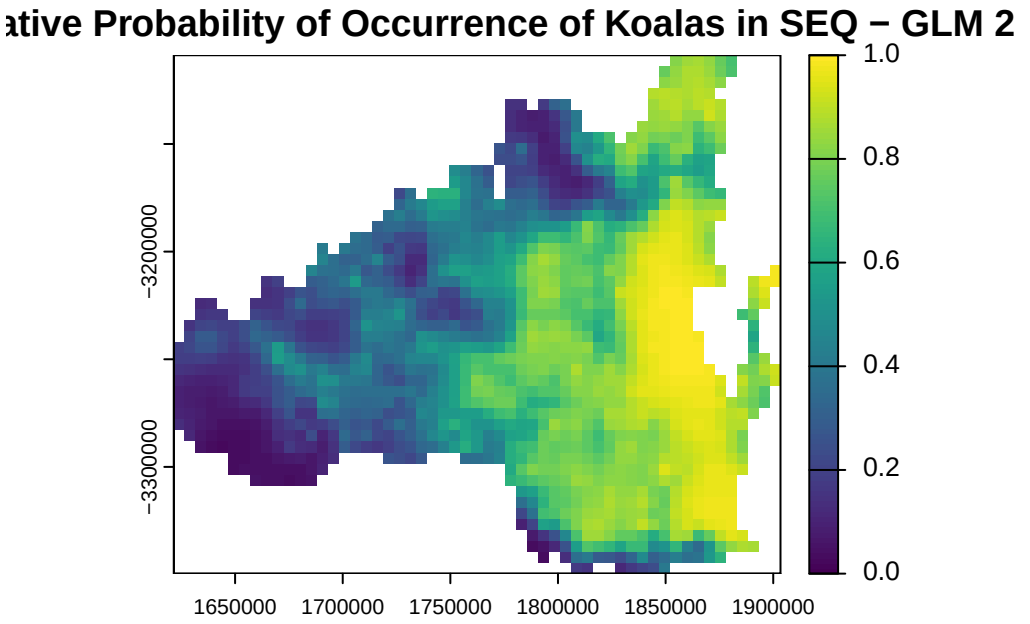
Model 1

```
# Predict the presence probability across the entire raster extent
predicted_raster_model_1 <- predicts::predict(covs_current_expert, glm_model_1, type = "resp

# Plot the species distribution raster
plot(
  predicted_raster_model_1,
  range = c(0, 1),  # Set min and max values for the color scale
  main = "Relative Probability of Occurrence of Koalas in SEQ - GLM 1"
)
```



Model 2

```
# Predict the presence probability across the entire raster extent
predicted_raster_model_2 <- predicts::predict(covs_current_expert, glm_model_2, type = "resp

# Plot the species distribution raster
plot(
  predicted_raster_model_2,
  range = c(0, 1),  # Set min and max values for the color scale
  main = "Relative Probability of Occurrence of Koalas in SEQ - GLM 2"
)
```



Random forest

```
# Make a folder to save outputs

dir.create("outputs/RF_outputs", showWarnings = F)
```

Data preparation

Calculate the case weights (down/up-weighting)

Because we have a 'class imbalance' (uneven number of presences and background points), we are making their 'weight' in the model equal.

```
presNum <- as.numeric(table(train_PB_covs_thinned$Presence)["1"]) # number of presences
bgNum <- as.numeric(table(train_PB_covs_thinned$Presence)["0"]) # number of backgrounds
weight <- ifelse(train_PB_covs_thinned$Presence == 1, 1, presNum / bgNum) # down-weighting
```

Prepare for fitting the RF model(s)

```r
# If wanting to fit a classification model
# Convert the response to factor for producing class relative likelihood
train_PB_covs_thinned$Presence_Factor <- as.factor(train_PB_covs_thinned$Presence)

# For down-sampling, the number of background (0s) in each bootstrap sample should the same
# (1s). For this, we use sampsize argument to do this.
# We need to choose the SMALLER number out of the two classes
sample_size <- c("0" = bgNum, "1" = bgNum)
sample_size <- c(bgNum)
```

Random Forest Model - expert covariates

Classification or regression model?

```r
# # Specify the model formula - classification
# model_formula <- as.formula(Presence_Factor ~
#                               BIO5_Max_Temp_Warmest_Month +
#                               BIO6_Min_Temp_Coldest_Month +
#                               BIO12_Annual_Precipitation +
#                               BIO15_Precip_Seasonality)

# Specify the model formula - regression
model_formula <- as.formula(Presence ~
                              BIO5_Max_Temp_Warmest_Month +
                              BIO6_Min_Temp_Coldest_Month +
                              BIO12_Annual_Precipitation +
                              BIO15_Precip_Seasonality)
```

Fit the random forest model

This will throw a warning as we only have two unique values (0s and 1s), but in our case that is fine, and you can ignore the warning.

```r
rf_1 <- randomForest::randomForest(formula = model_formula,
                                   data = train_PB_covs_thinned,
                                   weights = weight,
                                   ntree = 1000,
                                   sampsize = sample_size,
                                   replace = T,
                                   importance=TRUE)
```

```
Warning in randomForest.default(m, y, ...): The response has five or fewer
unique values.  Are you sure you want to do regression?
```

Check the model results

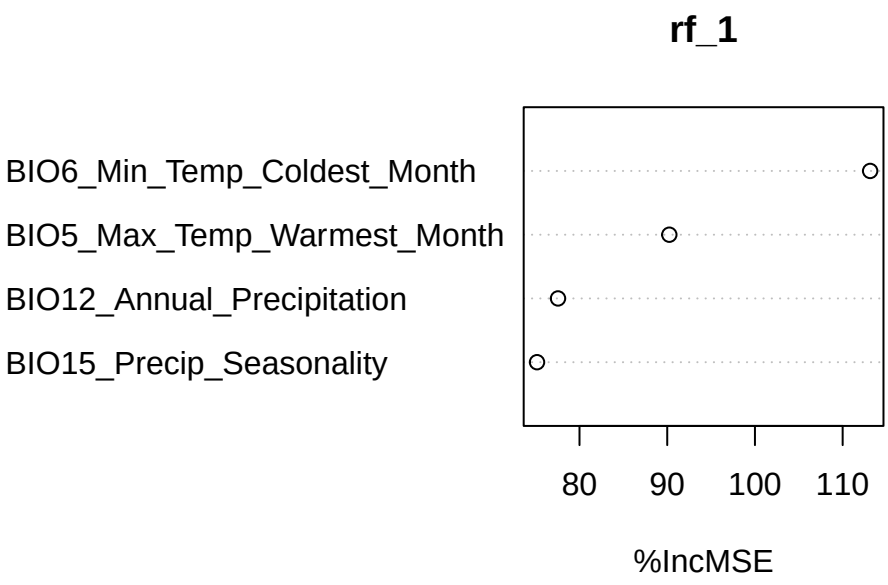
```
# Model summary
rf_1
```

```
Call:
 randomForest(formula = model_formula, data = train_PB_covs_thinned,      weights = weight,
               Type of random forest: regression
                     Number of trees: 1000
No. of variables tried at each split: 1

          Mean of squared residuals: 0.08568366
                    % Var explained: 23.08
```

```
# Variable importance:
importance(rf_1)
```

```
                               %IncMSE IncNodePurity
BIO5_Max_Temp_Warmest_Month  90.23977      69.47576
BIO6_Min_Temp_Coldest_Month 113.13609     102.85306
BIO12_Annual_Precipitation   77.54665      82.91868
BIO15_Precip_Seasonality     75.15556      47.28076
```

```
# Plot variable importance
varImpPlot(rf_1, type = 1)
```

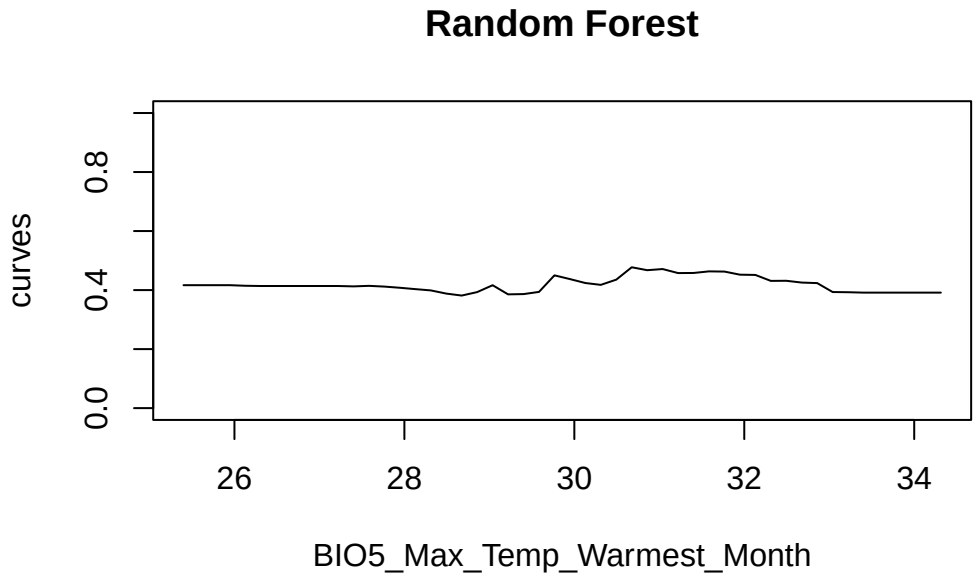


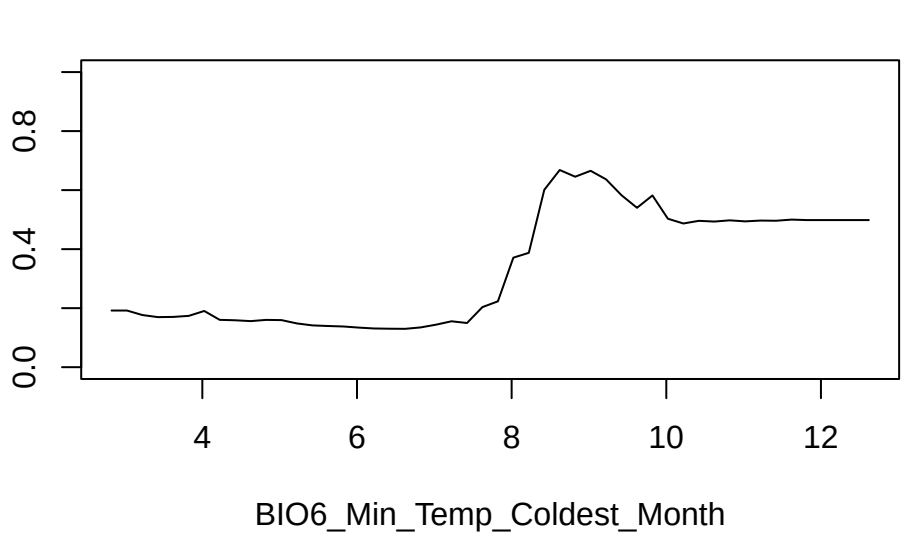

```
# Look at single trees:
head(getTree(rf_1,1,T))
```

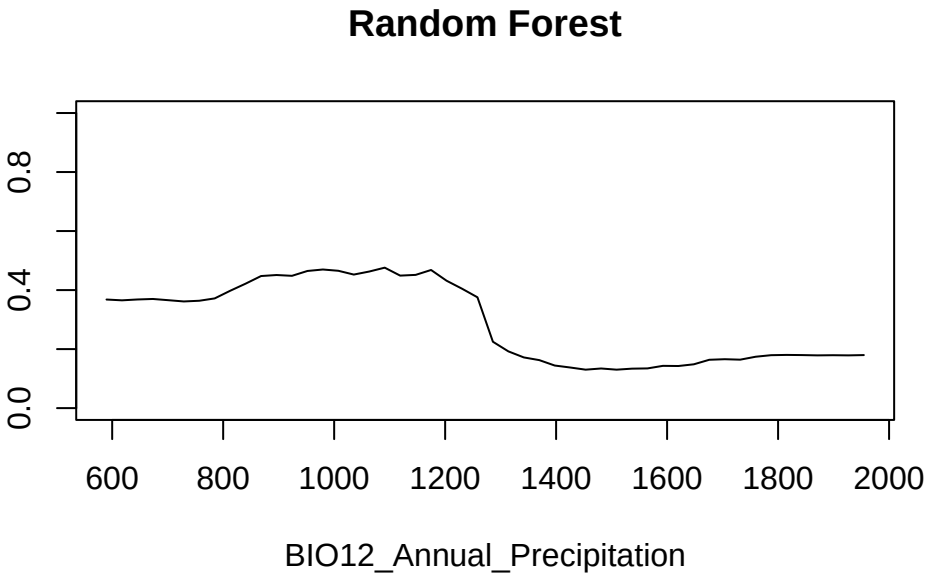
```
  left daughter right daughter                  split var split point status
1             2              3 BIO6_Min_Temp_Coldest_Month    8.319836     -3
2             4              5    BIO15_Precip_Seasonality   96.030197     -3
3             6              7 BIO6_Min_Temp_Coldest_Month    9.960938     -3
4             8              9 BIO6_Min_Temp_Coldest_Month    6.763289     -3
5            10             11  BIO12_Annual_Precipitation 1342.408508     -3
6            12             13 BIO6_Min_Temp_Coldest_Month    8.922107     -3
  prediction
1  0.4965847
2  0.1905405
3  0.8093923
4  0.1025057
5  0.3189369
6  0.8502269
```

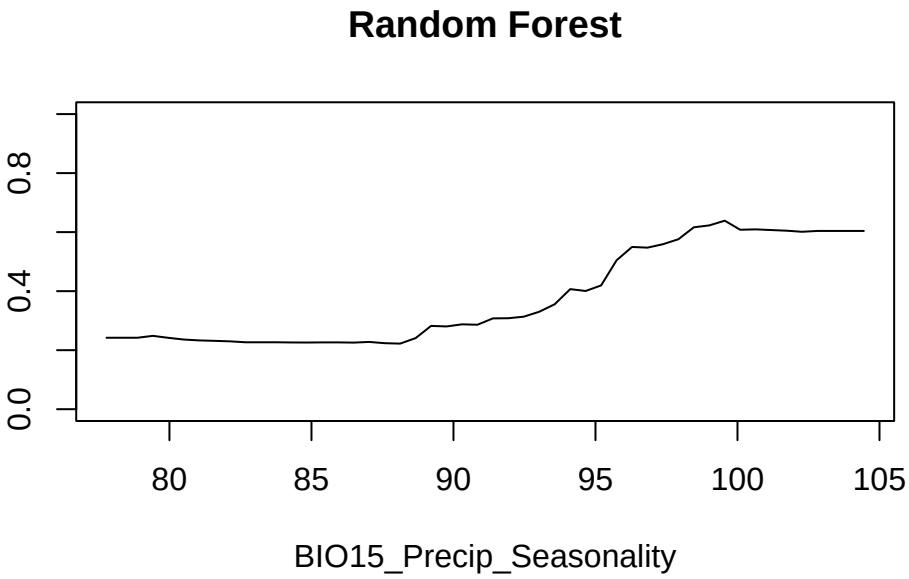
Partial dependence plots

```
# Now, we plot response curves in the same way as we did for GLMs above:
partial_response(rf_1, predictors = train_PB_covs_thinned[,predictors], main='Random Forest'
```

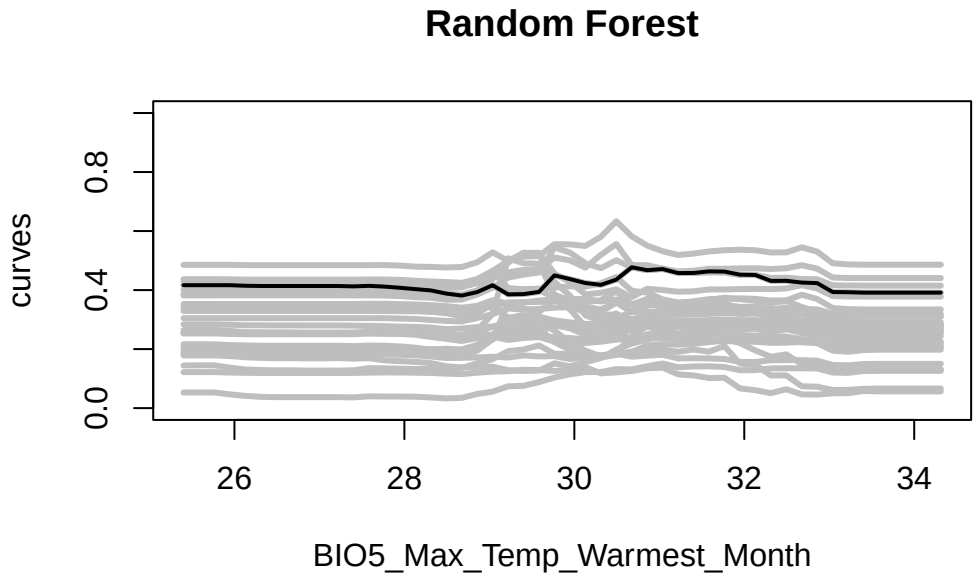

Random Forest
curves
0.0
0.4
0.8
26
28
30
32
34
BIO5_Max_Temp_Warmest_Month

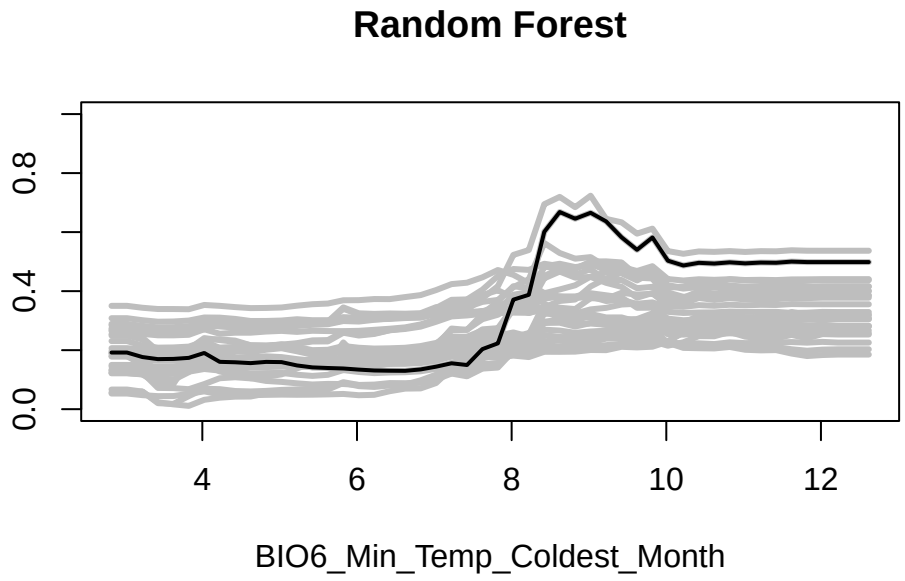

Random Forest
0.0
0.4
0.8
4
6
8
10
12
BIO6_Min_Temp_Coldest_Month

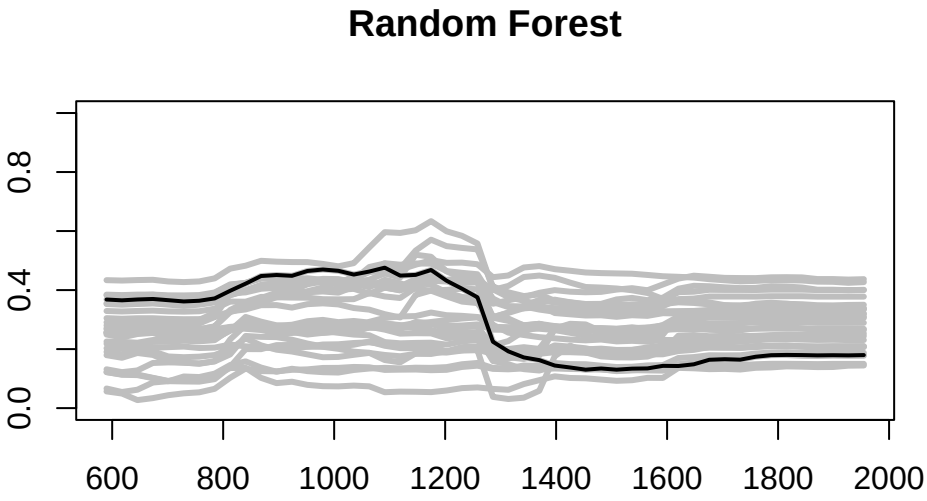


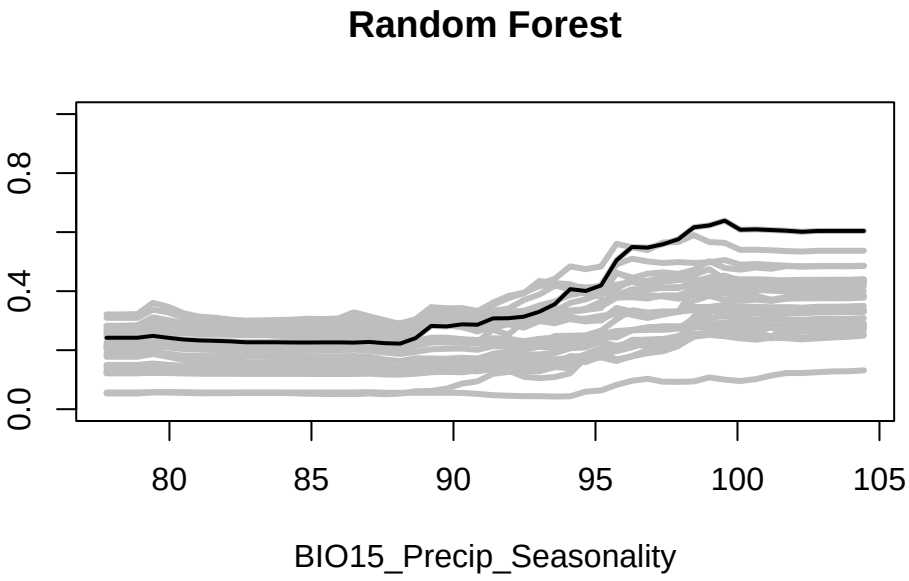



```
# Plot inflated response curves:
inflated_response(rf_1, predictors = train_PB_covs_thinned[,predictors], method = "stat3", l
```


Random Forest
curves
0.0
0.4
0.8
26
28
30
32
34
BIO5_Max_Temp_Warmest_Month


Random Forest
0.0
0.4
0.8
4
6
8
10
12
BIO6_Min_Temp_Coldest_Month

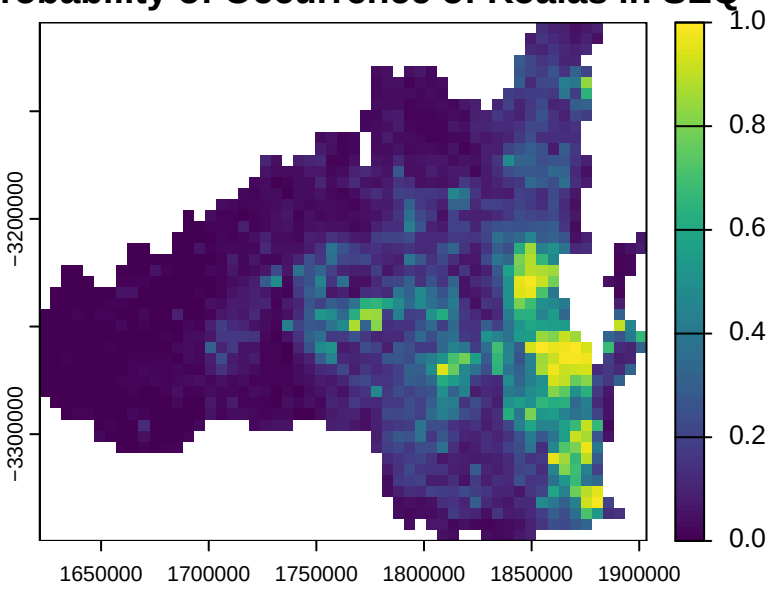




Random Forest predictions

```
# Predict the presence probability across the entire raster extent
predicted_raster_RF_1 <- predicts::predict(covs_current_expert, rf_1, type = "response")

# Plot the species distribution raster
plot(
  predicted_raster_RF_1,
  range = c(0, 1),  # Set min and max values for the color scale
```

```
  main = "Relative Probability of Occurrence of Koalas in SEQ - RF 1"
)
```




```
writeRaster(predicted_raster_RF_1,
            filename = "outputs/current_distribution_RF_1.tif",
            overwrite = TRUE)
```

Model evaluation with spatial block cross-validation

```
# Convert training data to sf
train_PB_covs_thinned_sf <- st_as_sf(train_PB_covs_thinned[, c("x", "y", "Presence")], coord

# Generate spatial blocks
spblock <- cv_spatial(x = train_PB_covs_thinned_sf,
                      column = "Presence",
                      r = NULL,
                      size = 50000, # Size of the blocks in metres
                      k = 5,
                      hexagon = TRUE,
                      selection = "random",
                      iteration = 100, # to find evenly-dispersed folds
                      biomod2 = FALSE)
```

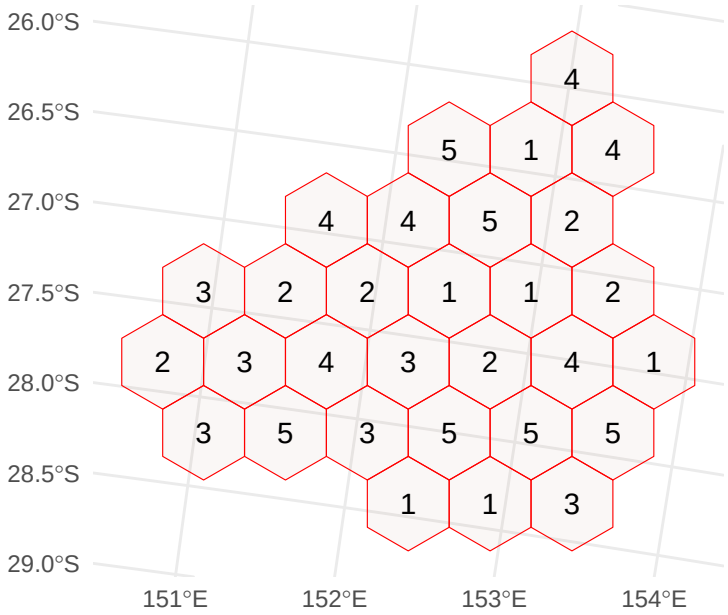
```
|
|                                                                      |   0%
|
|=                                                                     |   1%
|
|=                                                                     |   2%
|
|==                                                                    |   3%
|
|===                                                                   |   4%
|
|====                                                                  |   5%
|
|====                                                                  |   6%
|
|=====                                                                 |   7%
|
|======                                                                |   8%
|
|======                                                                |   9%
|
|=======                                                               |  10%
|
|========                                                              |  11%
|
|========                                                              |  12%
|
|=========                                                             |  13%
|
|==========                                                            |  14%
|
|==========                                                            |  15%
|
|===========                                                           |  16%
|
|============                                                          |  17%
|
|=============                                                         |  18%
|
|=============                                                         |  19%
|
|==============                                                        |  20%
|
|===============                                                       |  21%
|
```

```
|===============                                                       |  22%
|
|================                                                      |  23%
|
|=================                                                     |  24%
|
|==================                                                    |  25%
|
|==================                                                    |  26%
|
|===================                                                   |  27%
|
|====================                                                  |  28%
|
|====================                                                  |  29%
|
|=====================                                                 |  30%
|
|======================                                                |  31%
|
|======================                                                |  32%
|
|=======================                                               |  33%
|
|========================                                              |  34%
|
|========================                                              |  35%
|
|=========================                                             |  36%
|
|=========================                                             |  37%
|
|==========================                                            |  38%
|
|==========================                                            |  39%
|
|===========================                                           |  40%
|
|===========================                                           |  41%
|
|============================                                          |  42%
|
|============================                                          |  43%
|
|=============================                                         |  44%
```

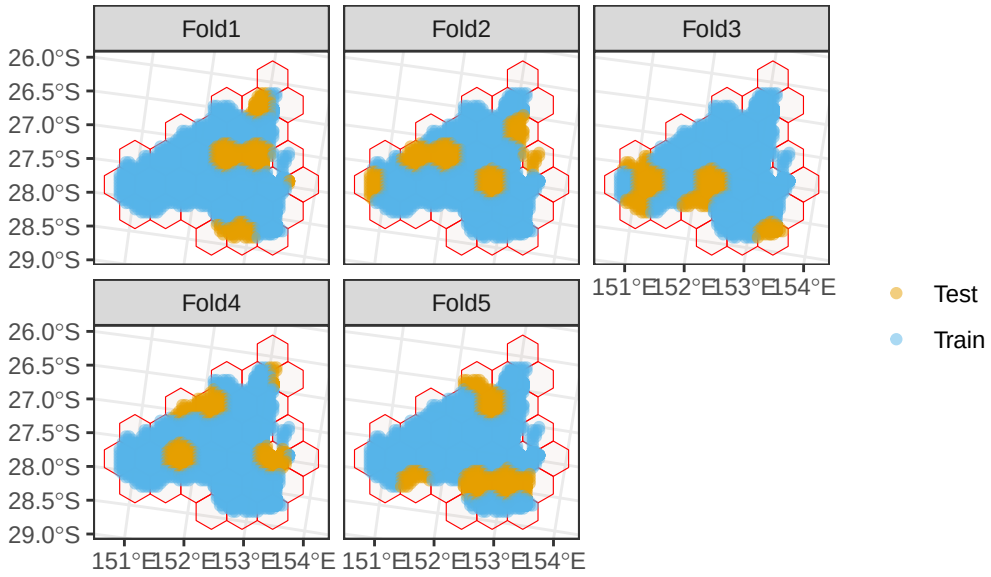
```
|
|================================                                 |  45%
|
|================================                                 |  46%
|
|=================================                                |  47%
|
|=================================                                |  48%
|
|=================================                                |  49%
|
|==================================                               |  50%
|
|==================================                               |  51%
|
|==================================                               |  52%
|
|===================================                              |  53%
|
|===================================                              |  54%
|
|===================================                              |  55%
|
|====================================                             |  56%
|
|====================================                             |  57%
|
|====================================                             |  58%
|
|=====================================                            |  59%
|
|=====================================                            |  60%
|
|=====================================                            |  61%
|
|======================================                           |  62%
|
|======================================                           |  63%
|
|======================================                           |  64%
|
|=======================================                          |  65%
|
|=======================================                          |  66%
|
```

```
|=============================================                         |  67%
|
|==============================================                        |  68%
|
|===============================================                       |  69%
|
|================================================                      |  70%
|
|=================================================                     |  71%
|
|==================================================                    |  72%
|
|===================================================                   |  73%
|
|====================================================                  |  74%
|
|=====================================================                 |  75%
|
|======================================================                |  76%
|
|=======================================================               |  77%
|
|========================================================              |  78%
|
|=========================================================             |  79%
|
|==========================================================            |  80%
|
|===========================================================           |  81%
|
|============================================================          |  82%
|
|=============================================================         |  83%
|
|==============================================================        |  84%
|
|===============================================================       |  85%
|
|================================================================      |  86%
|
|=================================================================     |  87%
|
|==================================================================    |  88%
|
|===================================================================   |  89%
```

```
  |
  |============================================================       |  90%
  |
  |=============================================================      |  91%
  |
  |==============================================================     |  92%
  |
  |===============================================================    |  93%
  |
  |================================================================   |  94%
  |
  |=================================================================  |  95%
  |
  |================================================================== |  96%
  |
  |=================================================================== |  97%
  |
  |==================================================================== |  98%
  |
  |===================================================================== |  99%
  |
  |======================================================================| 100%
  train_0 train_1 test_0 test_1
1    1206    8207    258   1793
2    1129    9274    335    726
3    1176    8653    288   1347
4    1202    5192    262   4808
5    1143    8674    321   1326
```




```
cv_plot(cv = spblock,
        x = train_PB_covs_thinned_sf,
        points_alpha = 0.5,
        nrow = 2)
```




```
# Extract the folds to save
spfolds <- spblock$folds_list

# We now have a list of 5 folds, where the first object is the training data, and the second
str(spfolds)
```

```
List of 5
 $ :List of 2
  ..$ : int [1:9413] 10859 10860 10965 11019 11020 10706 10656 10707 10657 11021 ...
  ..$ : int [1:2051] 11421 11423 11382 6511 11442 11422 11402 11381 11401 10738 ...
 $ :List of 2
  ..$ : int [1:10403] 11022 11264 11223 11265 11263 11074 4920 11075 11180 11224 ...
  ..$ : int [1:1061] 10859 10860 10965 11019 11020 10706 10656 10707 10657 11021 ...
 $ :List of 2
  ..$ : int [1:9829] 10859 10860 10965 11019 11020 10706 10656 10707 10657 11021 ...
  ..$ : int [1:1635] 11022 11264 11223 11265 11263 11074 4920 11075 11180 11224 ...
 $ :List of 2
  ..$ : int [1:6394] 10859 10860 10965 11019 11020 10706 10656 10707 10657 11021 ...
  ..$ : int [1:5070] 10982 11140 3731 9771 3978 5636 294 10774 10981 3251 ...
 $ :List of 2
  ..$ : int [1:9817] 10859 10860 10965 11019 11020 10706 10656 10707 10657 11021 ...
  ..$ : int [1:1647] 11230 11271 11229 11231 11270 11137 11269 11233 11136 11228 ...
```

Run the model for every fold and evaluate

Model evaluation - metrics

Typically, it helps to evaluate your model with several metrics that describe different features of model performance and prediction. Here, we define a function to feed in a model prediction and calculate several evaluation metrics.

The metrics are:

-Area under the receiver operating characteristic curve (AUC ROC)

Higher values of this (closer to 1) suggest a model is good at distinguishing presence points from the background.

-Continuous boyce index

Higher values of this (closer to 1) suggest a model is good at predicting higher suitability at spots where there were presences.

```
# Start a dataframe to save results
eval_df <- data.frame(fold = numeric(),
                      ROC = numeric(),
                      boyce = numeric())

for(f in seq_along(spfolds)) {

  # Subset the training and testing data (spatial cross validation) (for the fth fold)

  train_PB_covs_scv <- train_PB_covs_thinned[spfolds[[f]][[1]], ]
```

```
  test_PB_covs_scv <- train_PB_covs_thinned[spfolds[[f]][[2]], ]

  glm_model_1 <- glm(Presence ~
                       BIO5_Max_Temp_Warmest_Month +
                       BIO6_Min_Temp_Coldest_Month +
                       BIO12_Annual_Precipitation +
                       BIO15_Precip_Seasonality,
                     data=train_PB_covs_scv,
                     family = binomial(link = "logit"))

    # Predict to the testing data of fold f
  test_PB_covs_scv$pred <- predict(glm_model_1, newdata = test_PB_covs_scv, type = "response

  # Evaluate prediction on test set
  ROC = precrec::auc(precrec::evalmod(scores = test_PB_covs_scv$pred, labels = test_PB_covs_

  boyce = ecospat::ecospat.boyce(fit = test_PB_covs_scv$pred,
                                 obs = test_PB_covs_scv$pred[which(test_PB_covs_scv$Presence
                                 nclass = 0, # Calculate continuous index
                                 method = "pearson",
                                 PEplot = F)[["cor"]]

  # Add results to dataframe
  eval_df <- eval_df %>% add_row(fold = f, ROC = ROC, boyce = boyce)


}
```

Summarise the evaluation metrics

```
# Mean AUC & boyce
eval_df %>%
  summarise(mean_AUC = mean(ROC),
            mean_boyce = mean(boyce),
            sd_AUC = sd(ROC),
            sd_boyce = sd(boyce))
```

```
   mean_AUC mean_boyce     sd_AUC  sd_boyce
1 0.8470138     0.7584 0.08467198 0.1327057
```

Model evaluation for Random Forest with spatial block cross-validation

We're going to use the same spatial folds as before.

Run the RF model for every fold and evaluate

```
# Start a dataframe to save results
eval_df.RF <- data.frame(fold = numeric(),
                         ROC = numeric(),
                         boyce = numeric())

for(f in seq_along(spfolds)) {

  # Subset the training and testing data (spatial cross validation) (for the fth fold)

  train_PB_covs_scv <- train_PB_covs_thinned[spfolds[[f]][[1]], ]
  test_PB_covs_scv <- train_PB_covs_thinned[spfolds[[f]][[2]], ]

  presNum <- as.numeric(table(train_PB_covs_scv$Presence)["1"]) # number of presences
  bgNum <- as.numeric(table(train_PB_covs_scv$Presence)["0"]) # number of backgrounds
  weight <- ifelse(train_PB_covs_scv$Presence == 1, 1, presNum / bgNum) # down-weighting

  sample_size <- c("0" = bgNum, "1" = bgNum)
  sample_size <- c(bgNum)


  rf_1 <- randomForest::randomForest(formula = model_formula,
                                     data = train_PB_covs_scv,
                                     weights = weight,
                                     ntree = 1000,
                                     sampsize = sample_size,
                                     replace = T,
                                     importance=TRUE)


    # Predict to the testing data of fold f
  test_PB_covs_scv$pred <- predict(rf_1, newdata = test_PB_covs_scv, type = "response")

  # Evaluate prediction on test set
  ROC = precrec::auc(precrec::evalmod(scores = test_PB_covs_scv$pred, labels = test_PB_covs_

  boyce = ecospat::ecospat.boyce(fit = test_PB_covs_scv$pred,
```

```
                                  obs = test_PB_covs_scv$pred[which(test_PB_covs_scv$Presence
                                  nclass = 0, # Calculate continuous index
                                  method = "pearson",
                                  PEplot = F)[["cor"]]

  # Add results to dataframe
  eval_df.RF <- eval_df.RF %>% add_row(fold = f, ROC = ROC, boyce = boyce)


}
```

```
Warning in randomForest.default(m, y, ...): The response has five or fewer
unique values.  Are you sure you want to do regression?
Warning in randomForest.default(m, y, ...): The response has five or fewer
unique values.  Are you sure you want to do regression?
Warning in randomForest.default(m, y, ...): The response has five or fewer
unique values.  Are you sure you want to do regression?
Warning in randomForest.default(m, y, ...): The response has five or fewer
unique values.  Are you sure you want to do regression?
Warning in randomForest.default(m, y, ...): The response has five or fewer
unique values.  Are you sure you want to do regression?
```

Summarise the evaluation metrics

```
# Mean AUC & boyce
eval_df.RF %>%
  summarise(mean_AUC = mean(ROC),
            mean_boyce = mean(boyce),
            sd_AUC = sd(ROC),
            sd_boyce = sd(boyce))
```

```
   mean_AUC mean_boyce     sd_AUC  sd_boyce
1 0.8121285       0.64 0.08989678 0.2091853
```

Make predictions to future climates

Load future environmental data

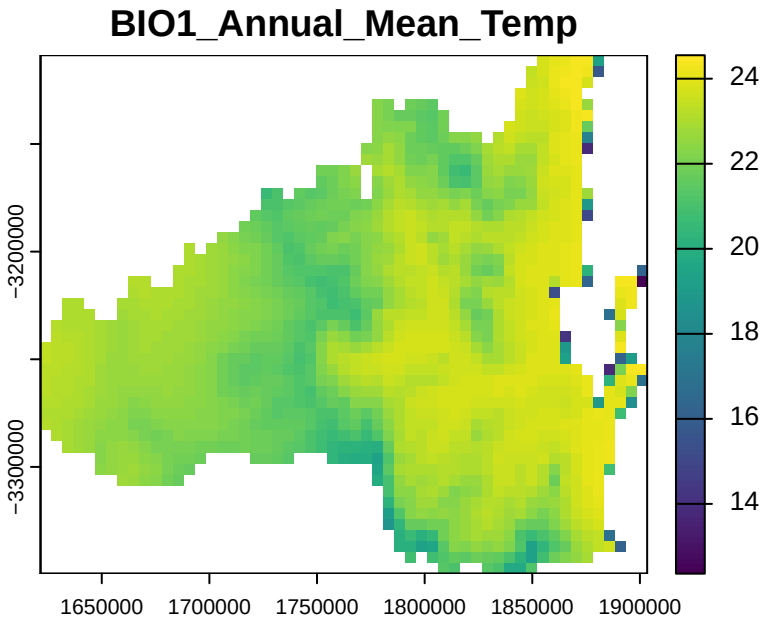
```
covs_future <- rast("Data/Environmental_variables/future_bioclim.2090.SSP370.tif")
names(covs_future) <- layer_names
covs_future
```

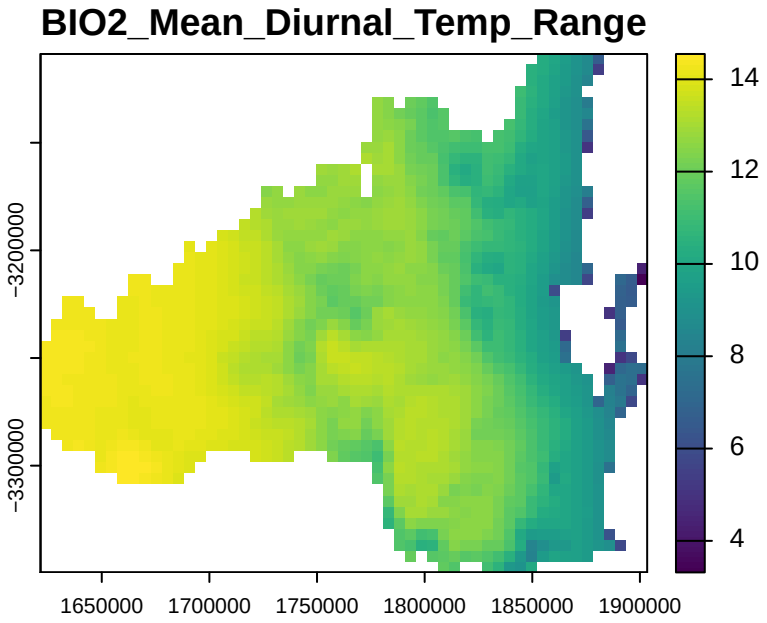
```
class       : SpatRaster
dimensions  : 47, 55, 19  (nrow, ncol, nlyr)
resolution  : 5123.954, 5123.954  (x, y)
extent      : 1621552, 1903370, -3349594, -3108768  (xmin, xmax, ymin, ymax)
coord. ref. : GDA94 / Geoscience Australia Lambert (EPSG:3112)
source      : future_bioclim.2090.SSP370.tif
names       : BIO1_~_Temp, BIO2_~Range, BIO3_~ality, BIO4_~ality, BIO5_~Month, BIO6_~Month,
min values  :   0.8177757,   0.3121004,    1.564732,    12.43535,    1.130271,   0.4521889,
max values  :  24.5521488,  14.5486765,   50.989292,   528.63293,   37.389927,  15.7586107,
```

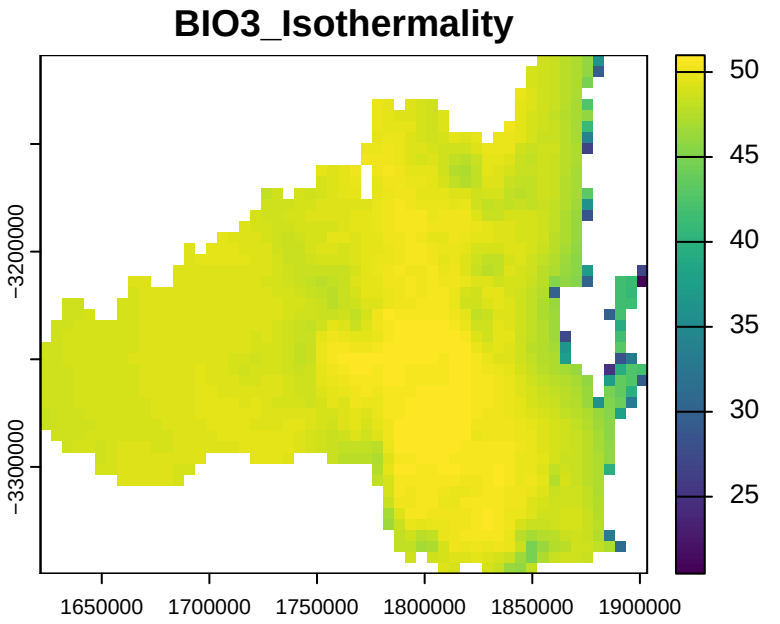
```
covs_future <- terra::mask(covs_future, covs_current) # Crop to SEQ extent

covs_future_expert <- subset(covs_future, names(covs_future) %in% c("BIO5_Max_Temp_Warmest_M
                                                                    "BIO6_Min_Temp_Coldest_M
                                                                    "BIO12_Annual_Precipitat
                                                                    "BIO15_Precip_Seasonalit
```

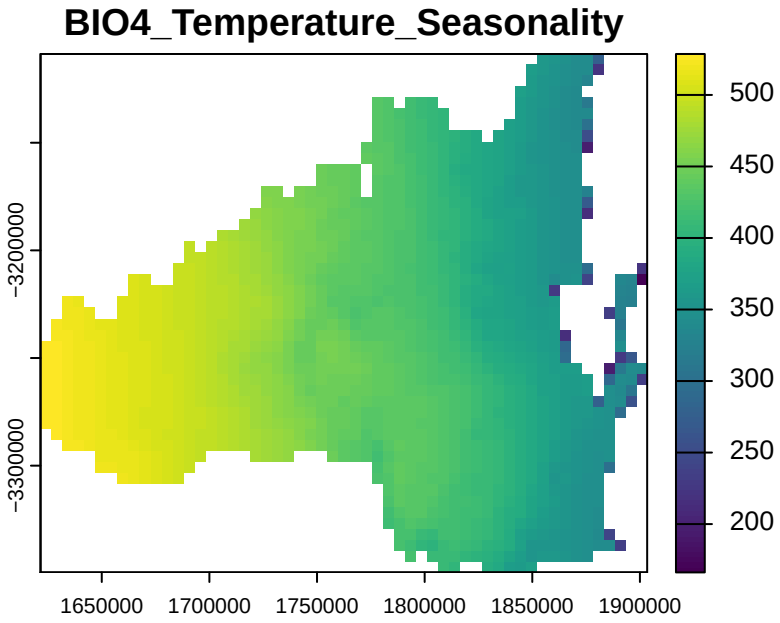
Plot the future rasters

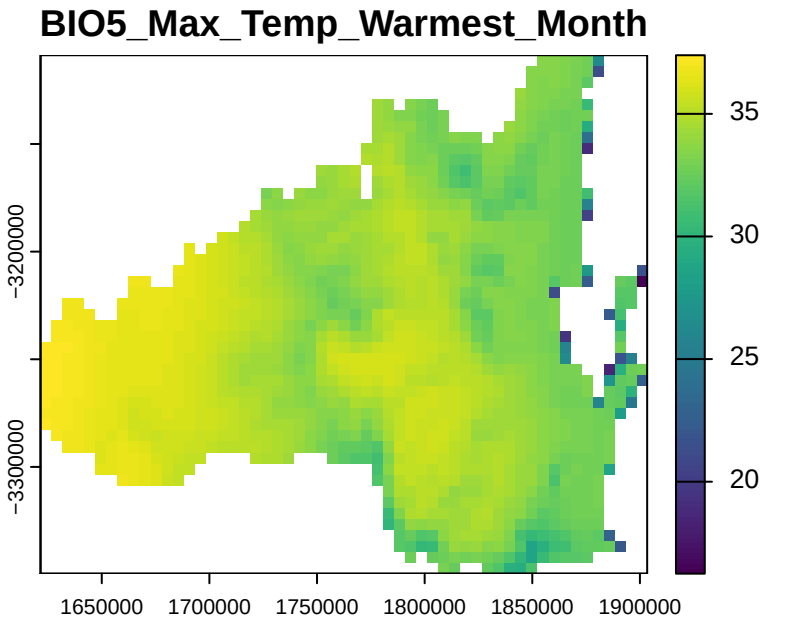
```
for(i in 1:nlyr(covs_future)) {
  terra::plot(covs_future[[i]], main = names(covs_future)[[i]])
}
```

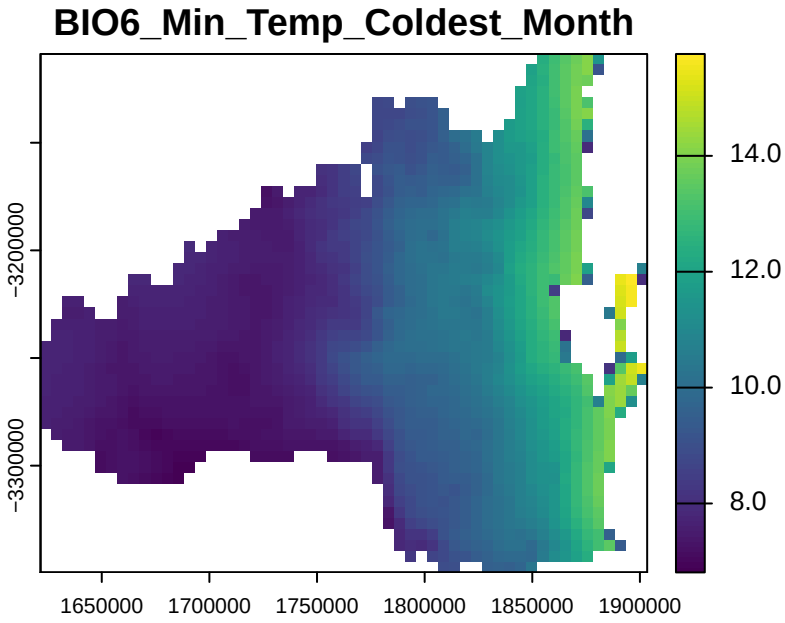

BIO1_Annual_Mean_Temp
24
22
20
18
16
14
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000

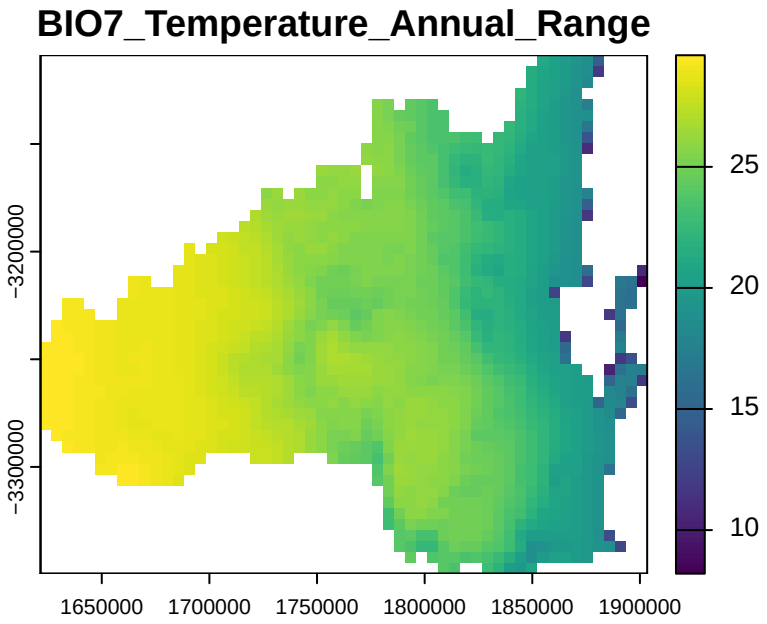

BIO2_Mean_Diurnal_Temp_Range
14
12
10
8
6
4
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000

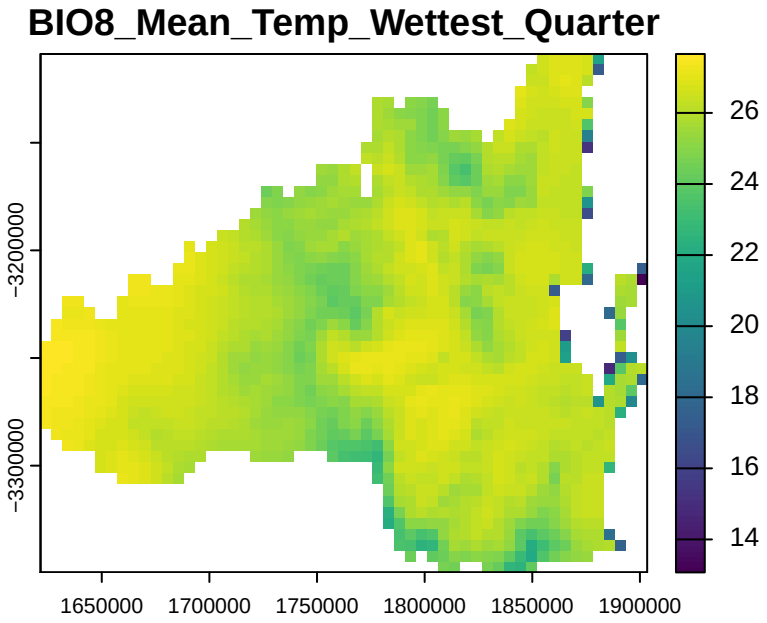

BIO3_Isothermality
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
50
45
40
35
30
25

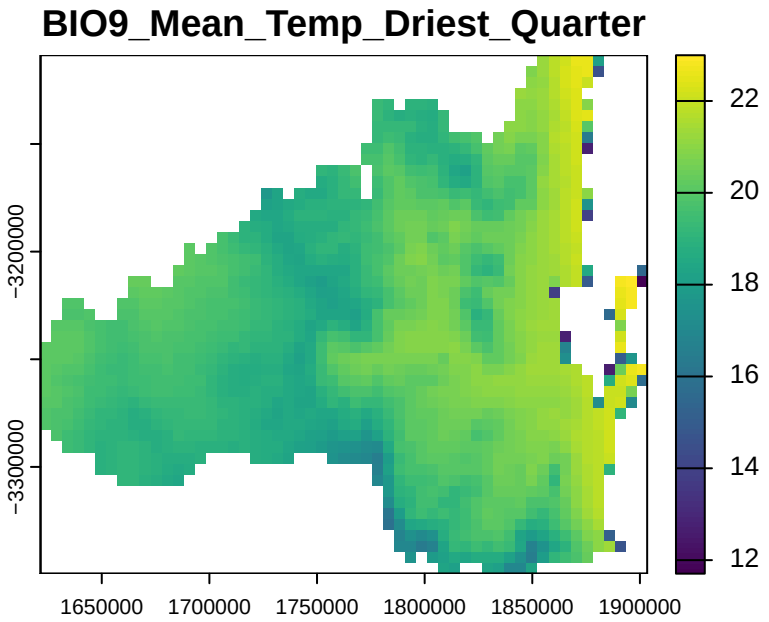

BIO4_Temperature_Seasonality
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
500
450
400
350
300
250
200

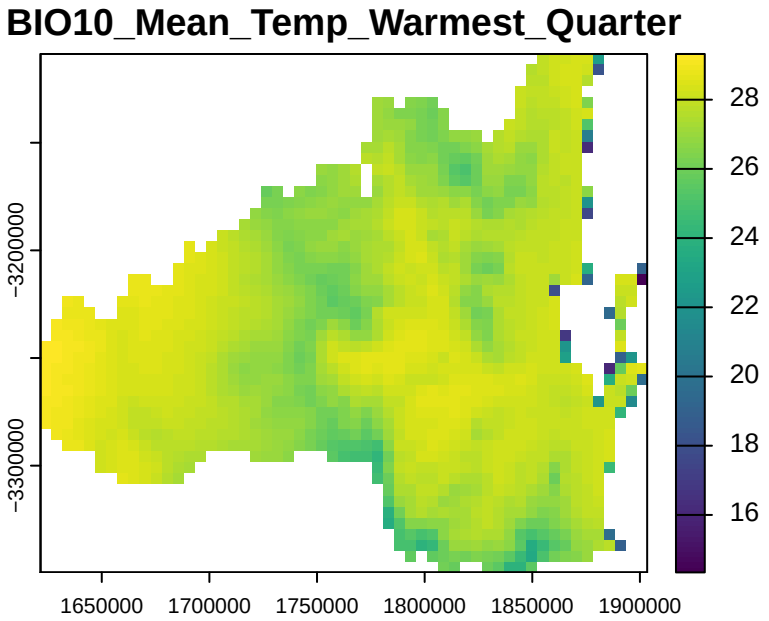

BIO5_Max_Temp_Warmest_Month
35
30
25
20
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000

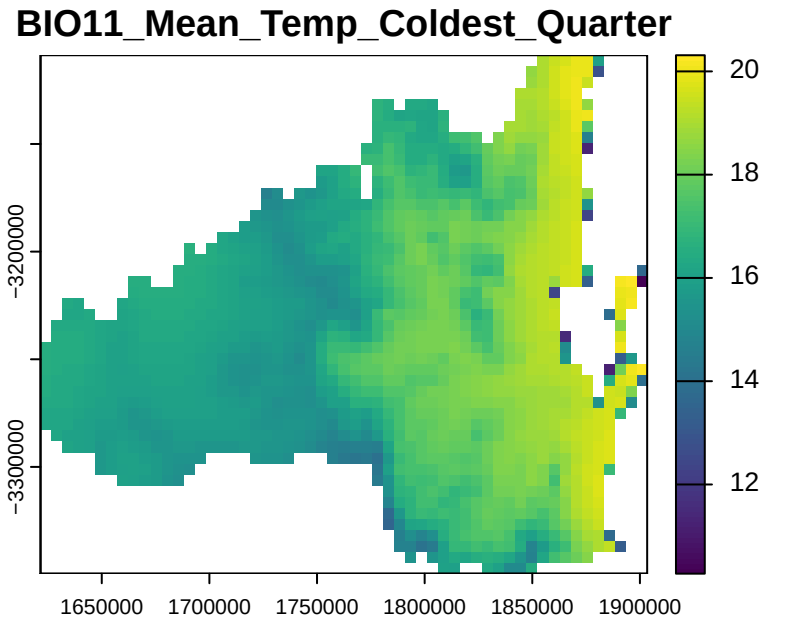

BIO6_Min_Temp_Coldest_Month
14.0
12.0
10.0
8.0
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000

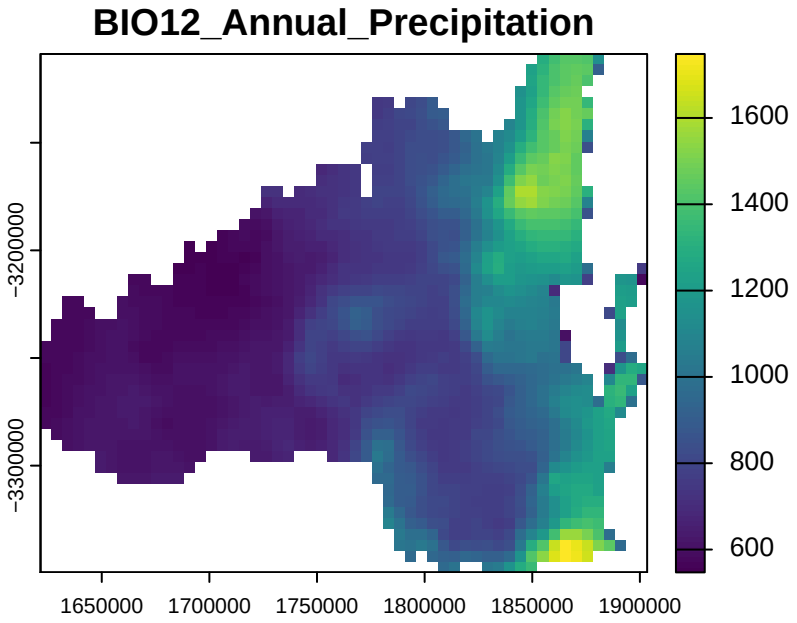

BIO7_Temperature_Annual_Range
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
25
20
15
10

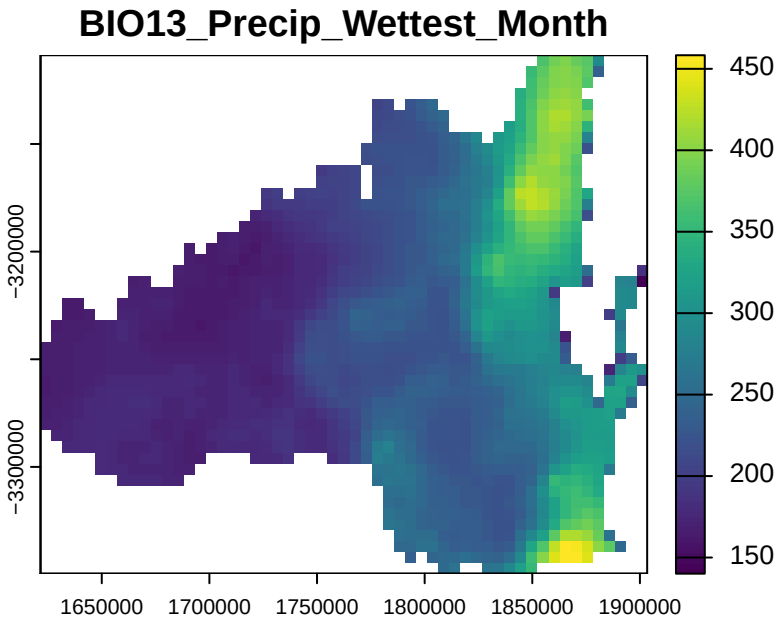

BIO8_Mean_Temp_Wettest_Quarter
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
26
24
22
20
18
16
14

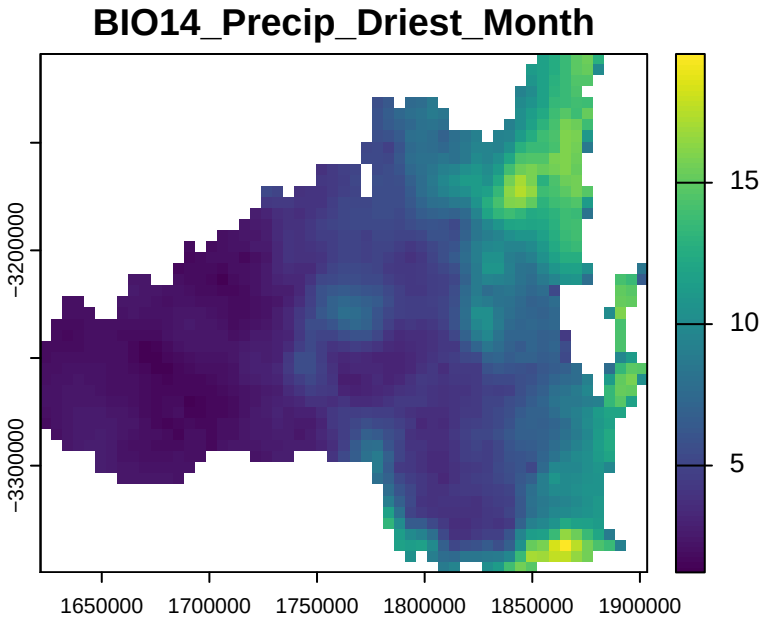

BIO9_Mean_Temp_Driest_Quarter
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
22
20
18
16
14
12

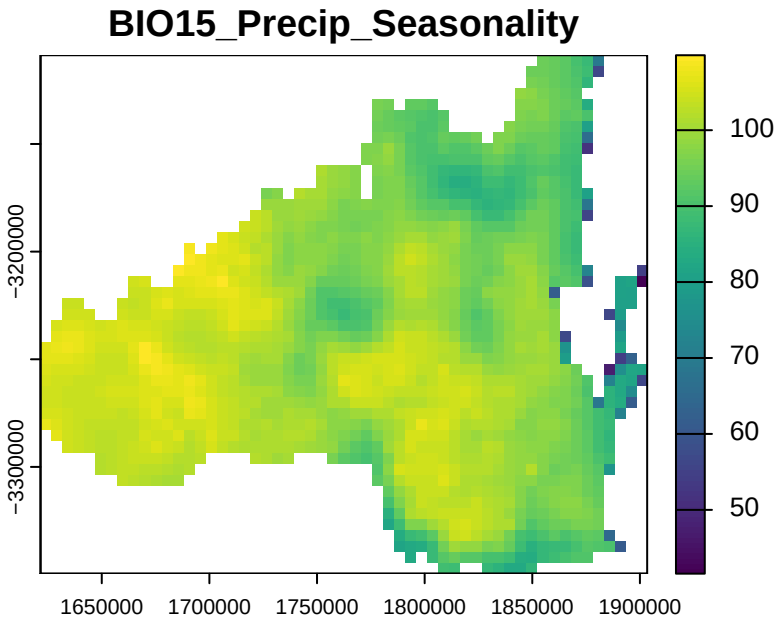

BIO10_Mean_Temp_Warmest_Quarter
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
28
26
24
22
20
18
16

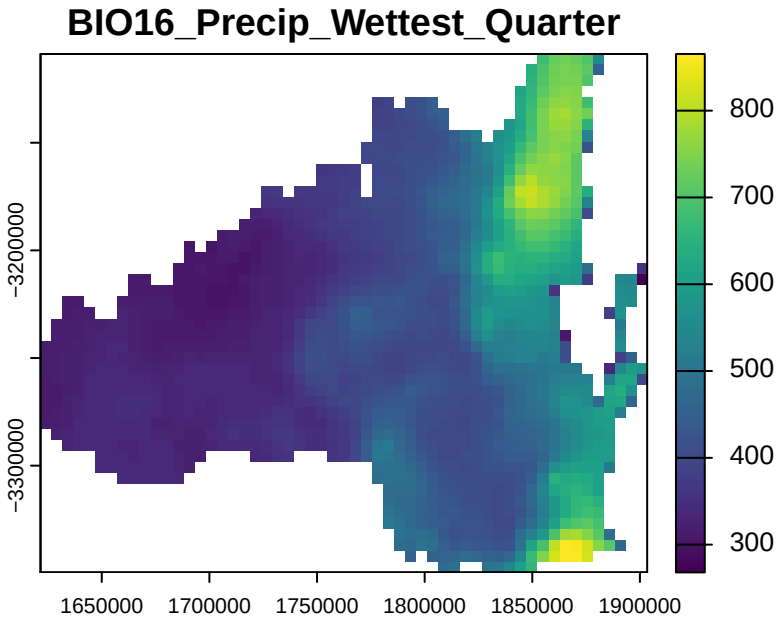

BIO11_Mean_Temp_Coldest_Quarter
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
20
18
16
14
12

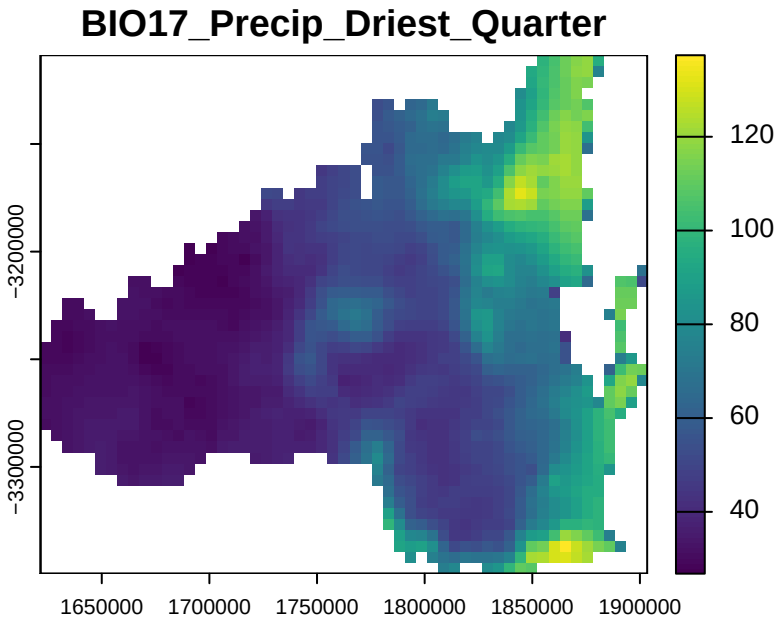

BIO12_Annual_Precipitation
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
1600
1400
1200
1000
800
600

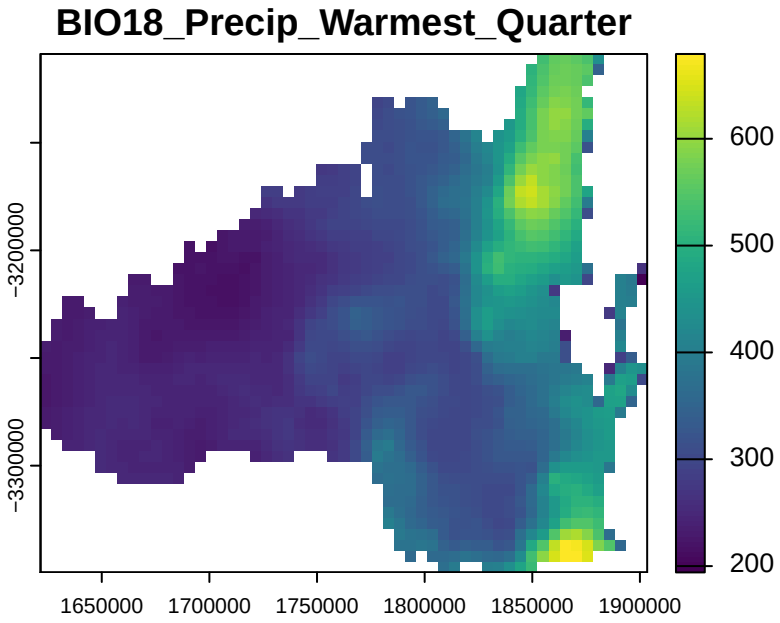

BIO13_Precip_Wettest_Month
450
400
350
300
250
200
150
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000

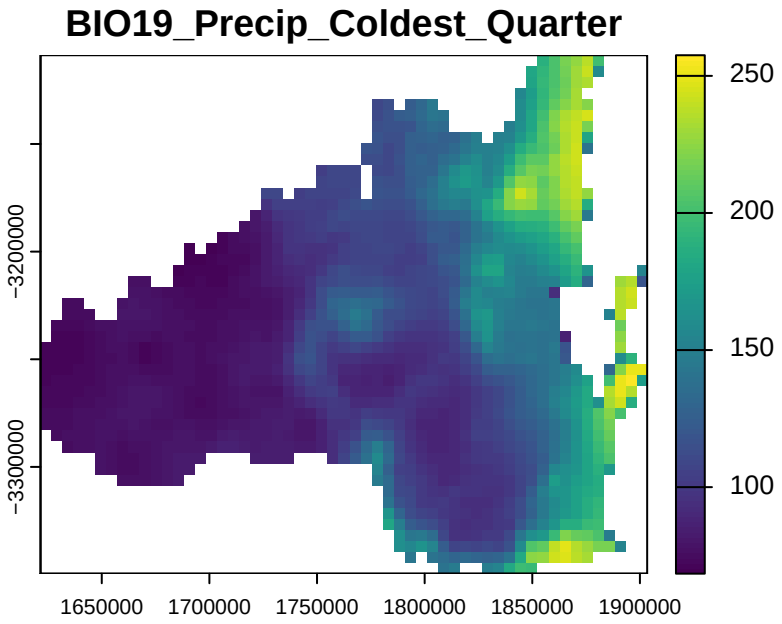

BIO14_Precip_Driest_Month
15
10
5
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000


BIO15_Precip_Seasonality
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
100
90
80
70
60
50

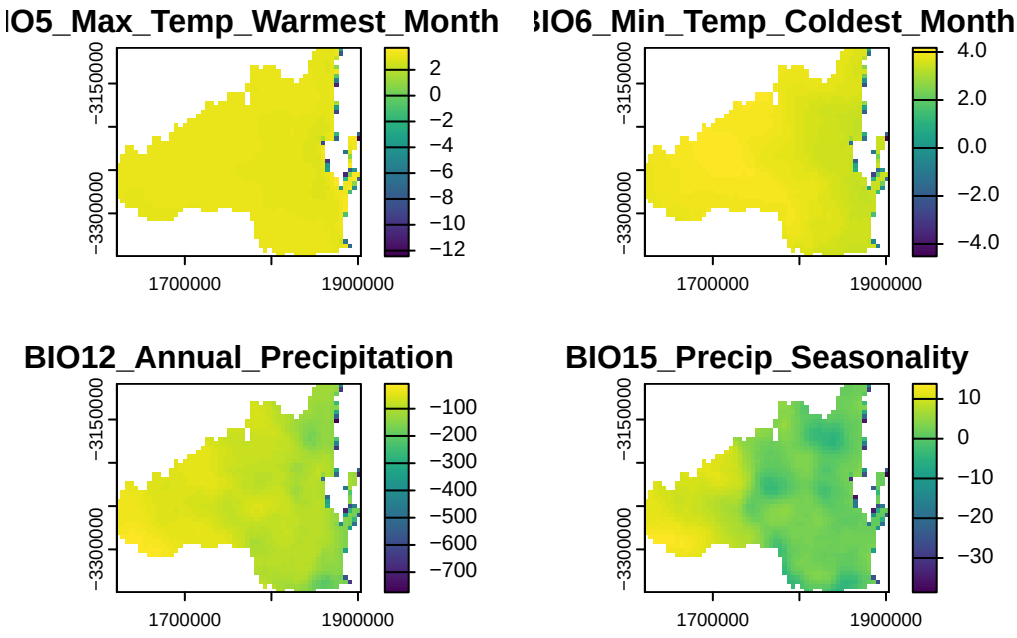

BIO16_Precip_Wettest_Quarter
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
800
700
600
500
400
300


BIO17_Precip_Driest_Quarter
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
120
100
80
60
40


BIO18_Precip_Warmest_Quarter
−3200000
−3300000
1650000
1700000
1750000
1800000
1850000
1900000
600
500
400
300
200

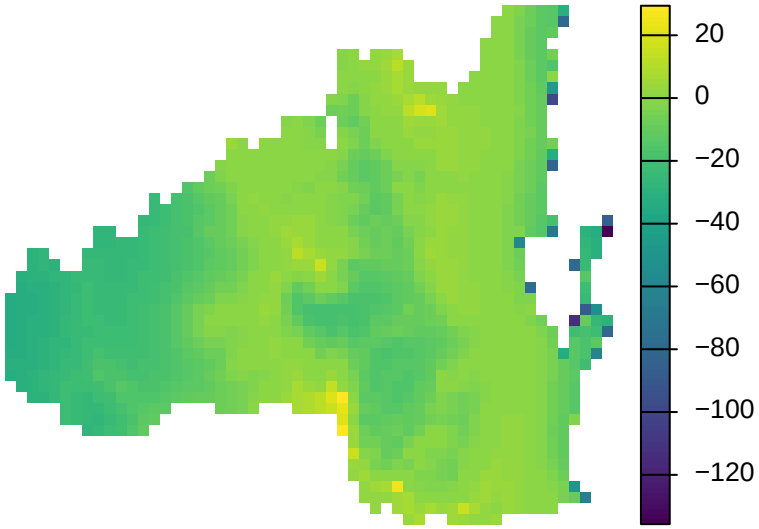



```
# Plot to compare the difference between the current and future rasters

plot(covs_future_expert - covs_current_expert)
```



Test the environmental distance between current data and future conditions

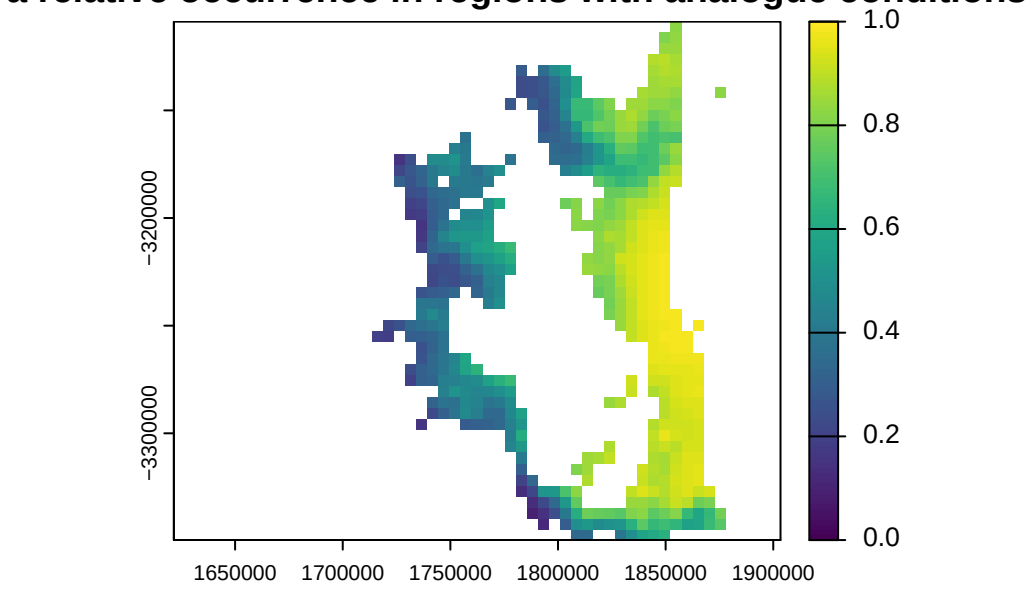
```
mess <- predicts::mess(covs_future_expert,
                       train_PB_covs_thinned[, c("BIO5_Max_Temp_Warmest_Month",
                                                 "BIO6_Min_Temp_Coldest_Month",
                                                 "BIO12_Annual_Precipitation",
                                                 "BIO15_Precip_Seasonality")])

plot(mess, axes = F)
```



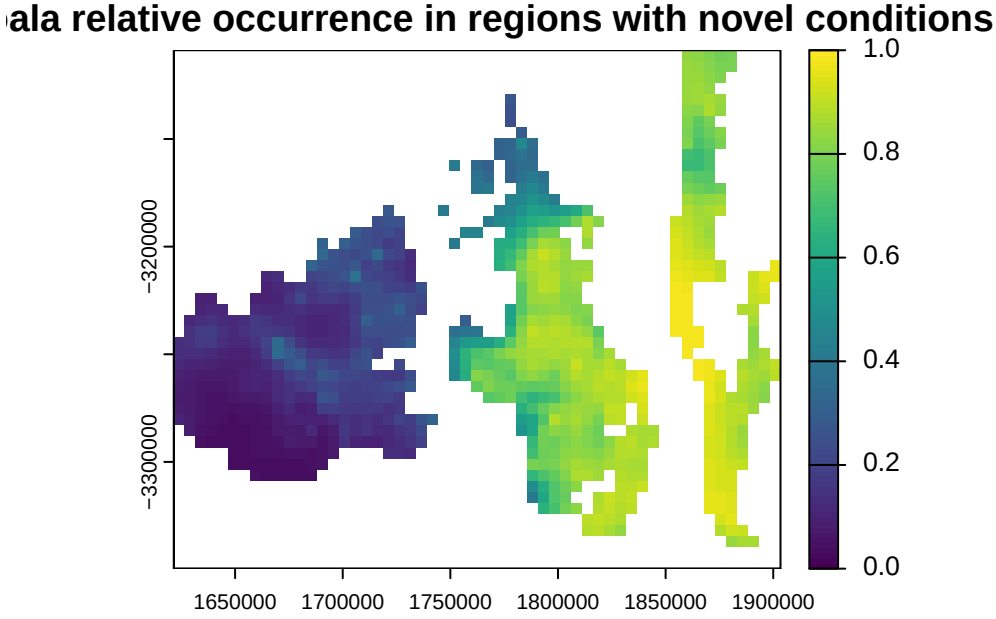

```
r_mess_mask <- mess < 0
plot(r_mess_mask, axes=F)
```

Test which areas you might mask out because they are 'novel' in environmental space and therefore require model extrapolation.

```r
analog_fut <- predicted_raster_model_1

values(analog_fut)[values(mess)<0] <- NA

plot(analog_fut,
     range = c(0, 1),  # Set min and max values for the color scale
     main = "Koala relative occurrence in regions with analogue conditions")
```



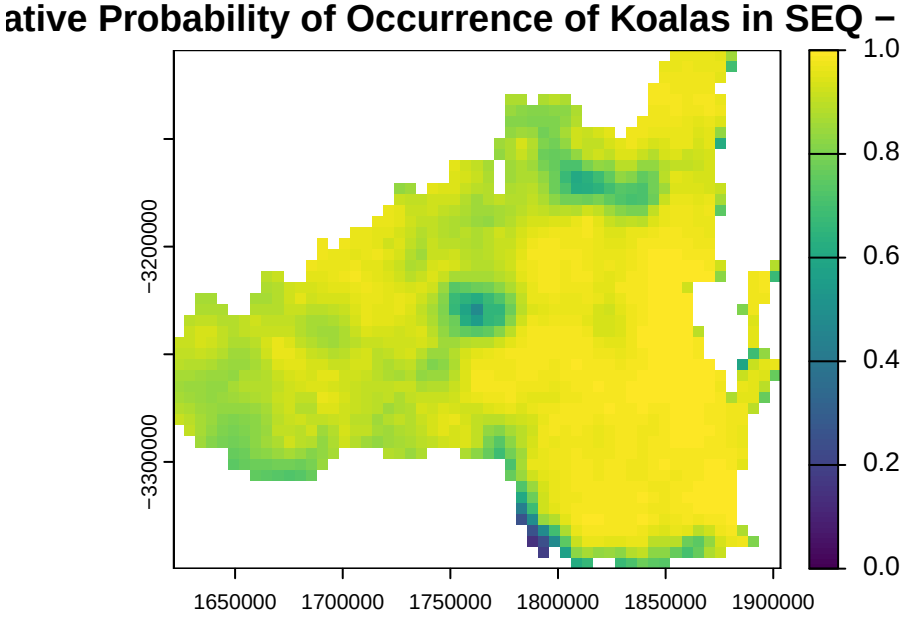

```r
novel_fut <- predicted_raster_model_1

values(novel_fut)[values(mess)>0] <- NA

plot(novel_fut,
     range = c(0, 1),  # Set min and max values for the color scale
     main = "Koala relative occurrence in regions with novel conditions")
```



GLM future predictions

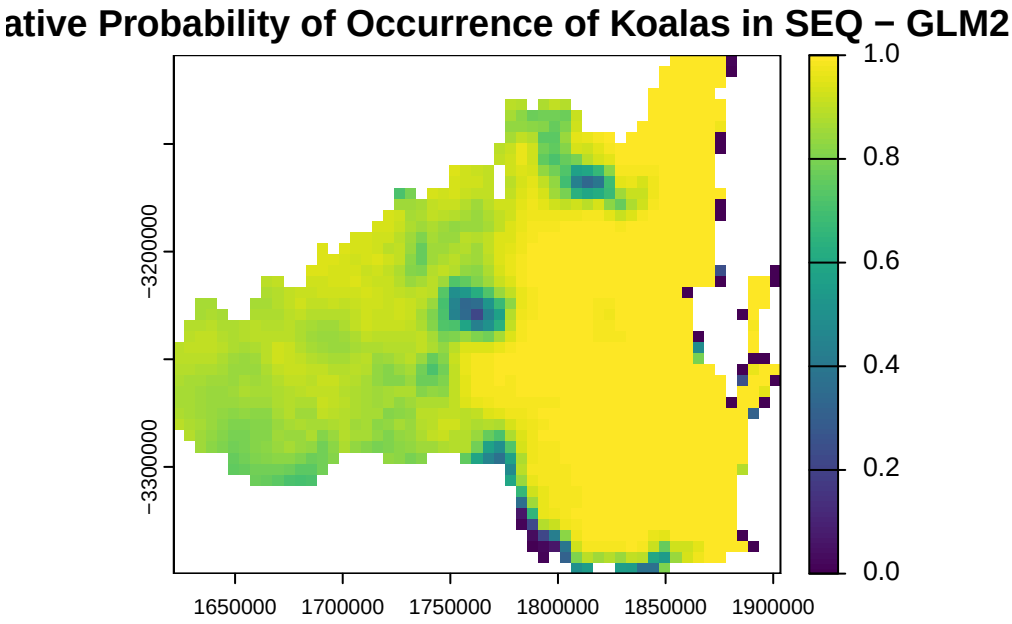
Model 1

```
# Predict the presence probability across the entire raster extent
predicted_raster_model_1 <- predict(covs_future_expert, glm_model_1, type = "response")

# Plot the species distribution raster
plot(
  predicted_raster_model_1,
  range = c(0, 1),  # Set min and max values for the color scale
  main = "Relative Probability of Occurrence of Koalas in SEQ - GLM1"
)
```



Model 2

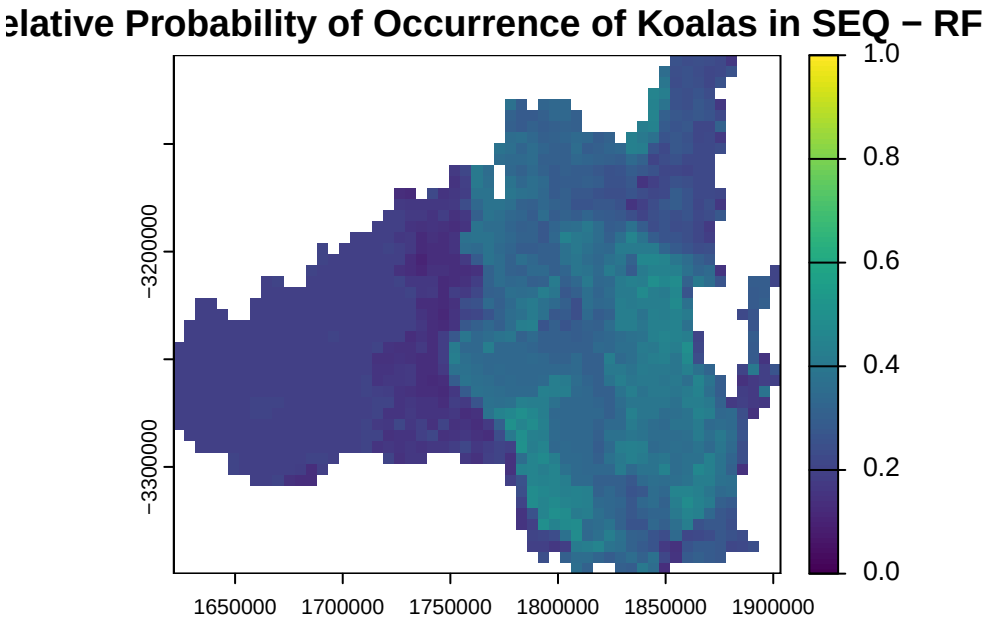
```
# Predict the presence probability across the entire raster extent
predicted_raster_model_2 <- predict(covs_future_expert, glm_model_2, type = "response")

# Plot the species distribution raster
plot(
  predicted_raster_model_2,
  range = c(0, 1),  # Set min and max values for the color scale
  main = "Relative Probability of Occurrence of Koalas in SEQ - GLM2"
)
```



Random Forest future predictions

Model 1

```
# Predict the presence probability across the entire raster extent
predicted_raster_RF_1 <- predicts::predict(covs_future_expert, rf_1, type = "response", na.r

# Plot the species distribution raster
plot(
  predicted_raster_RF_1,
  range = c(0, 1),  # Set min and max values for the color scale
  main = "Relative Probability of Occurrence of Koalas in SEQ - RF"
)
```



Presenting predictions with uncertainty

There are many sources of model uncertainty that should be explored and ideally, presented alongside model predictions.

One that we'll focus on here is climate scenario uncertainty. We do so by fitting a second model to future climate data from a lower emission shared socioeconomic path scenario (SSP 1.26).

Load future environmental data (SSP 1.26)

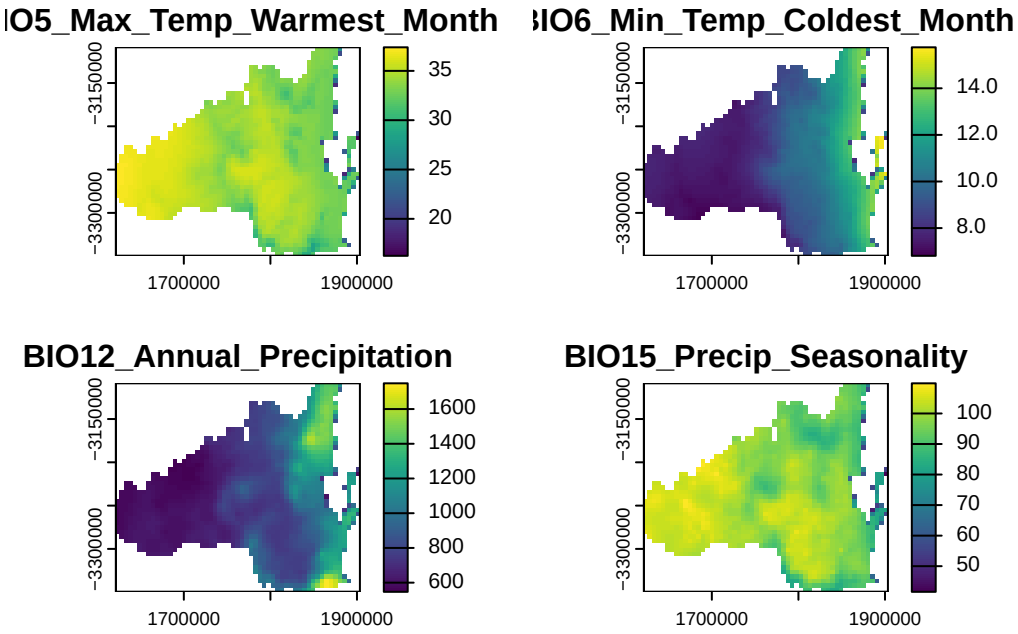
```
covs_future_SSP126 <- rast("Data/Environmental_variables/future_bioclim.2090.SSP126.tif")
names(covs_future_SSP126) <- layer_names
covs_future_SSP126
```

```
class       : SpatRaster
dimensions  : 47, 55, 19  (nrow, ncol, nlyr)
resolution  : 5123.954, 5123.954  (x, y)
extent      : 1621552, 1903370, -3349594, -3108768  (xmin, xmax, ymin, ymax)
coord. ref. : GDA94 / Geoscience Australia Lambert (EPSG:3112)
source      : future_bioclim.2090.SSP126.tif
names       : BIO1_~_Temp, BIO2_~Range, BIO3_~ality, BIO4_~ality, BIO5_~Month, BIO6_~Month,
min values  :   0.7471204,   0.3128485,    1.585051,    12.19436,    1.046093,   0.3754203,
max values  :  22.4384842,  14.8543510,   51.254021,   544.83429,   35.546104,  13.5217409,
```

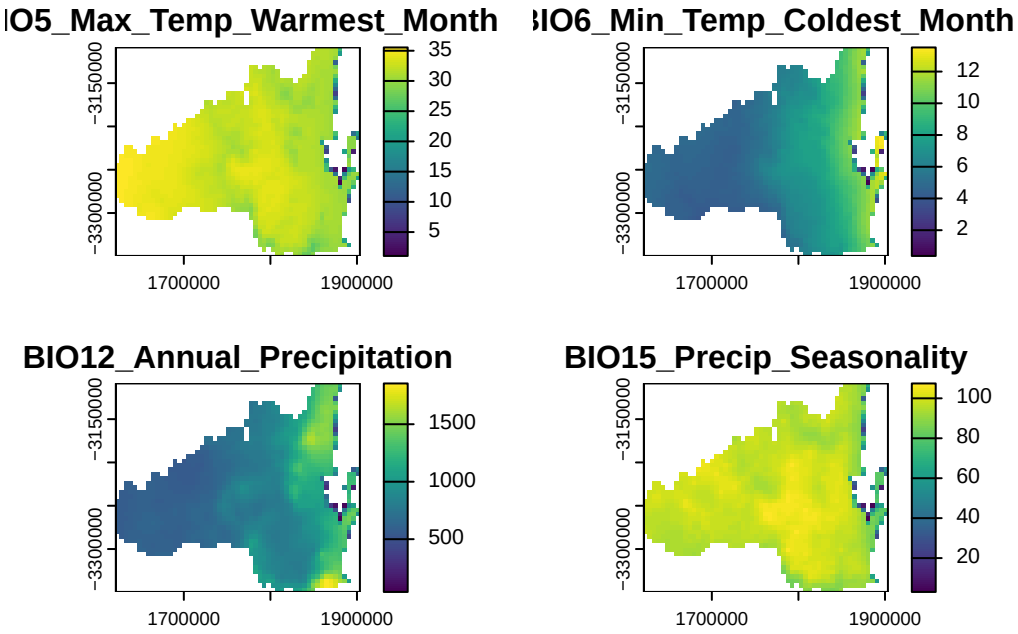
```
covs_future_SSP126_expert <- subset(covs_future_SSP126, names(covs_future_SSP126) %in% c("BI
                                                                                          "B
                                                                                          "B
                                                                                          "B
```

Plot to compare the variables across the two scenarios

```
plot(covs_future_expert)
```



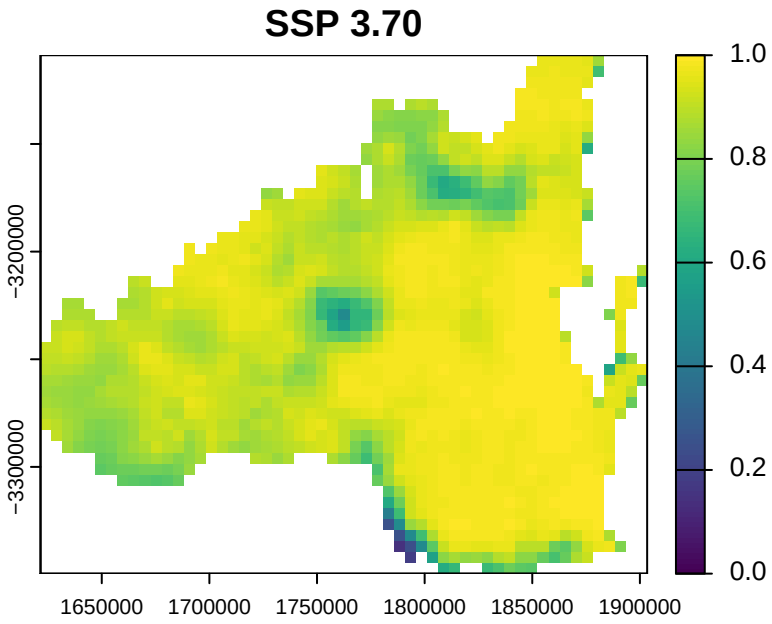

```
plot(covs_future_SSP126_expert)
```



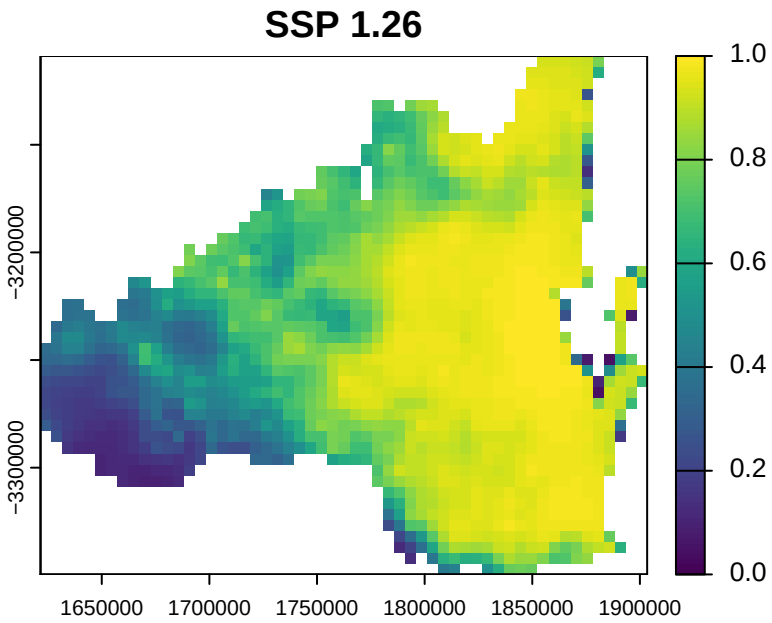
GLM future predictions (SSP 1.26)

Model 1

```
# Predict the presence probability across the entire raster extent
predicted_raster_model_1_SSP126 <- predict(covs_future_SSP126_expert, glm_model_1, type = "r

# Plot the species distribution raster
plot(
  predicted_raster_model_1,
  range = c(0,1),
  main = "SSP 3.70"
)
```




```
plot(
  predicted_raster_model_1_SSP126,
  range = c(0,1),
  main = "SSP 1.26"
)
```



Model uncertainty

Another element of uncertainty that can be represented is model uncertainty, or the standard error around the coefficient estimates.

```
# Extract standard errors of coefficients

coef_se <- summary(glm_model_1)$coefficients[, "Std. Error"]

print(coef_se)
```

```
               (Intercept) BIO5_Max_Temp_Warmest_Month
              1.7073577601                0.0565765829
BIO6_Min_Temp_Coldest_Month  BIO12_Annual_Precipitation
              0.0320840066                0.0003021583
   BIO15_Precip_Seasonality
              0.0118936628
```

```
covs_df <- as.data.frame(covs_future_expert, na.rm = FALSE)

pred_link <- predict(glm_model_1, newdata = covs_df, type = "link", se.fit = TRUE)

# Linear predictor (eta)
eta <- pred_link$fit
se_eta <- pred_link$se.fit

# Confidence intervals (95%)
z <- 1.96
eta_lower <- eta - z * se_eta
eta_upper <- eta + z * se_eta

# Transform back to response scale
linkinv <- glm_model_1$family$linkinv
predicted <- linkinv(eta)
lower_ci <- linkinv(eta_lower)
upper_ci <- linkinv(eta_upper)


# Add to covs_df
covs_df$predicted <- predicted
covs_df$lower_ci <- lower_ci
covs_df$upper_ci <- upper_ci
```

```r
predicted_r <- setValues(rast(covs_future_expert, nlyr = 1), predicted)
lower_ci_r <- setValues(rast(covs_future_expert, nlyr = 1), lower_ci)
upper_ci_r <- setValues(rast(covs_future_expert, nlyr = 1), upper_ci)

# Step 2: Name the layers
names(predicted_r) <- "predicted"
names(lower_ci_r) <- "lower_CI"
names(upper_ci_r) <- "upper_CI"

prediction_w_uncertainty <- c(predicted_r, lower_ci_r, upper_ci_r)

plot(prediction_w_uncertainty, range = c(0, 1))
```