Tutorial: Species Distribution Modelling with GBIF and Worldclim data

Timings

Section times as follows, based on using coarse environmental data (10 arc minutes):

1. Package install (ca. 3 min)
2. Download and prepare environmental data (ca. 2 min)
3. Download and prepare future environmental data (ca. 2 min)
4. Obtain species occurrence data from GBIF and spatially rarefy (ca. 5 min)
5. Build SDMs (ca. 10 min)
6. Project SDMs from current to future climate (ca. 5 min)
7. Evaluate present vs future models

Other info

- If you are on a PC, directory paths need to be separated by back-slashes instead of forward-slashes
- Try not to resize windows/display panels while R is busy calculating/plotting - this tends to cause problems

1. Initial setup and package install

The following block of code will install all required R packages for this tutorial, you can copy and paste this in your R Studio and run it. All of this code works with the latest R version (4.1.0), and possibly older ones (e.g. 3.5.0), but I cannot guarantee it will work, so best to have R v.4.1.0.

```
#install.packages(c('sdm','raster','rgdal','rgbif','readr','dismo','dplyr','spThin'))
```

Now we will create some directories where things will be saved

```
base.dir <- '/Users/cb76kecu/Dropbox (iDiv)/Micro-Macro_course/2021/Tutorial/Part_1_SDMs/'

dir.create(paste0(base.dir))
dir.create(paste0(base.dir,'env_data/'))
```

```
dir.create(paste0(base.dir,'env_data/'))
dir.create(paste0(base.dir,'env_data/present_climate'))
dir.create(paste0(base.dir,'env_data/future_climate_2070_RCP_8.5'))
dir.create(paste0(base.dir,'sdm_R/'))
dir.create(paste0(base.dir,'sdm_R/ensembles'))
dir.create(paste0(base.dir,'sdm_R/GBIF_data'))
dir.create(paste0(base.dir,'sdm_R/GBIF_data/thinned'))
```

2. Download and prepare the environmental data

This will prepare the environmnental data you will use for the next steps. First load some packages we need, and set our working directory

```
library(raster)
library (rgdal)

setwd <- base.dir
```

Then download the data for the 19 bioclim variables at a given arc minute resolution (here 10 arc minute degrees, ~20x20km at equator) globally - bear in mind the lower numbers = higher resolution so will take more time to do analyses. A summary of what the bioclim variables are can be found here: https://www.worldclim.org/data/bioclim.html

```
climate <- raster::getData('worldclim', var='bio', res=10)
```

Now we will clip the data to a specific extent (xmin,xmax,ymin,ymax)

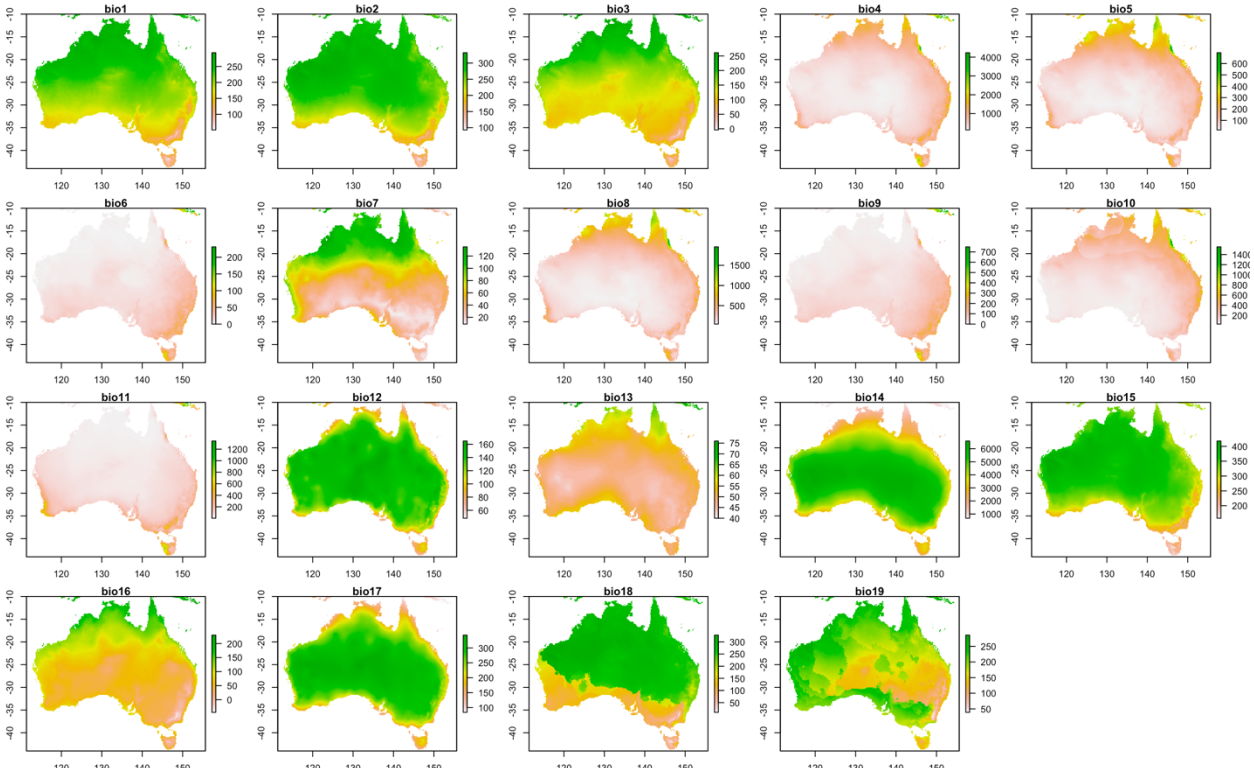
```
template <- extent(112, 155, -44, -10)
climate.crop <- crop(climate, template, snap="out")
```

And save each of the bioclim variables as .asc grid files

```
for(i in 1:nlayers(climate.crop)){
  writeRaster(climate.crop[[i]], paste0(base.dir,
"env_data/present_climate/bioclim_", i), "ascii", overwrite = T)
  cat('Writing bioclim',i,
      '... \n')
}
```

Now load them from file and plot them, just to check they look ok

```
env_data <- paste0(base.dir,'env_data/present_climate')
lst <- list.files(path=env_data,pattern='asc$',full.names = T)
climate <- stack(lst)

par(mar=c(1,1,1,1))
par(mfrow=c(4,5))
for(i in 1:nlayers(climate)){
  plot(climate[[i]], main= paste0("bio",i))
}
```



3 Download and prepare future climate data

Download the FUTURE worldclim data at 10 arc minute resolution globally (matching the present climate) 'model' should be one of "AC", "BC", "CC", "CE", "CN", "GF", "GD", "GS", "HD", "HG", "HE", "IN", "IP", "MI", "MR", "MC", "MP", "MG", or "NO". 'rcp' should be one of 26, 45, 60, or 85. 'year' should be 50 or 70. Not all combinations are available - see www.worldclim.org for details.

```
future_climate <- getData('CMIP5', var='bio', res=10, rcp=85, model='AC',
year=70)
```

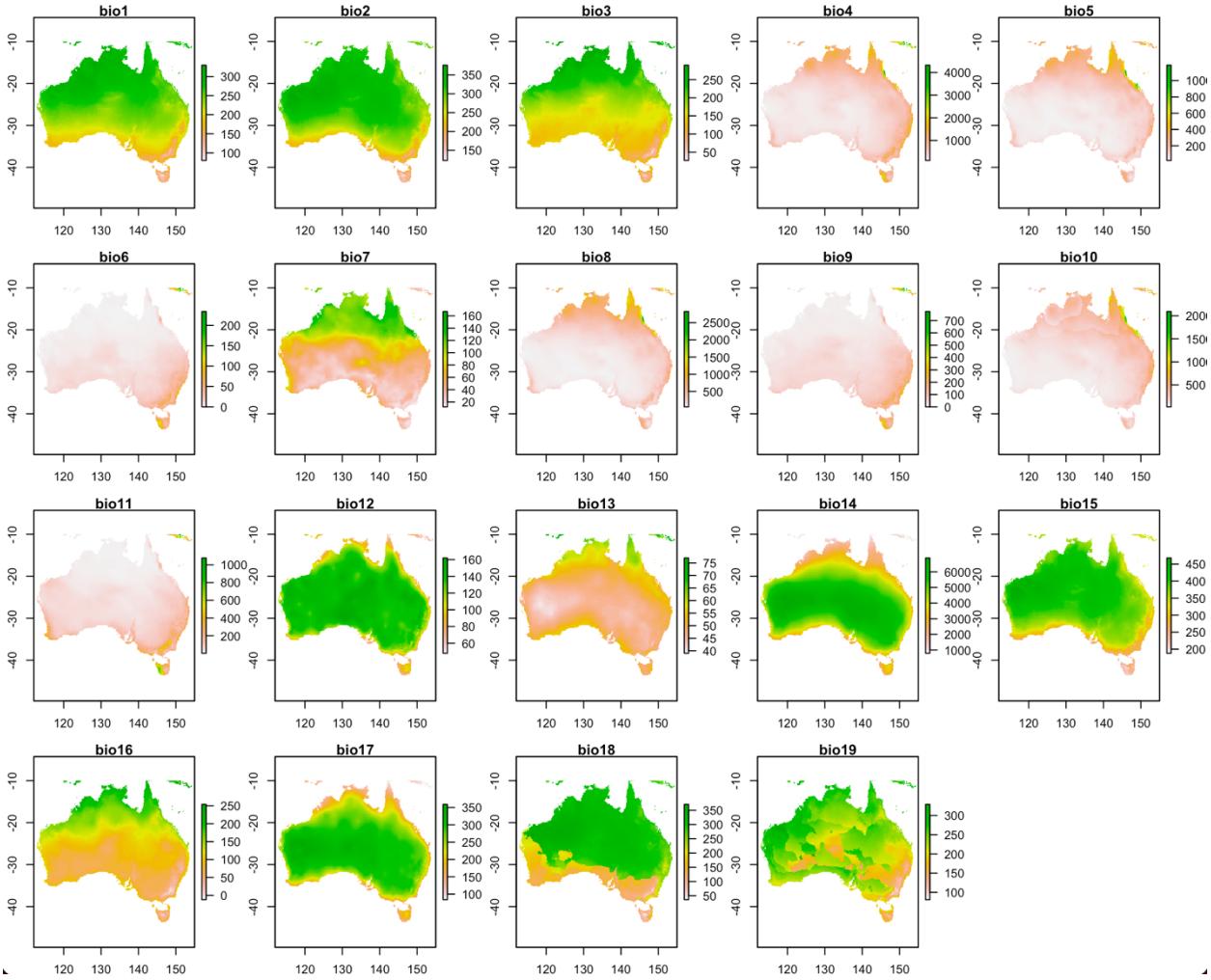
Clip the data to the same extent as defined for the present climate data (xmin,xmax,ymin,ymax)

```
future_climate.crop <- crop(future_climate, template, snap="out")
```

And again save each of the bioclim variables as .asc grid files, reading and plotting them to check they are ok

```
for(i in 1:nlayers(future_climate.crop)){
  writeRaster(future_climate.crop[[i]], paste0(base.dir,
"env_data/future_climate_2070_RCP_8.5/bioclim_", i), "ascii", overwrite = T)
  cat('Writing bioclim',i,
      '... \n')
}
```

```
env_data <- paste0(base.dir,'env_data/future_climate_2070_RCP_8.5/')
# lists of pred variable locations
lst <- list.files(path=env_data,pattern='asc$',full.names = T)
future_climate <- stack(lst)

par(mar=c(1,1,1,1))
par(mfrow=c(4,5))
for(i in 1:nlayers(future_climate)){
  plot(future_climate[[i]], main= paste0("bio",i))
}
```



Just some housekeeping to remove the worldclim climate data that you don't need any more (you already saved it for your geographic area of interest)

```
unlink('./cmip5/', recursive=TRUE)
unlink('./wc10/', recursive=TRUE)
```

4. Obtain species data from GBIF and spatially rarefy the data

Now we will download some spatial occurrence data (presences) from the Global Biodiversity Information Facility (https://www.gbif.org) using the rgbif R package. This can take a long time if you do not set a limit on the number of occurrences to download (set as 1000 here). We use the rgbif package (already loaded), and use the occ_search function, then store it as a dataframe

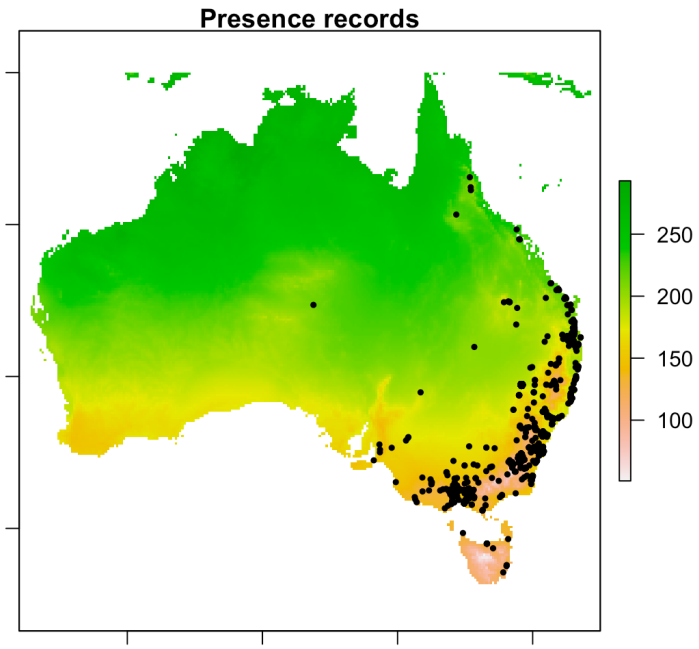
```
library(rgbif)
library(dplyr)
species_name <- "Macropus_giganteus"
species_name_for_GBIF <- gsub("_", " ", species_name)
dat <- occ_search(scientificName = species_name_for_GBIF, hasCoordinate = T,
limit = 3000)
dat_ne_lapply <- lapply(dat, "as.data.frame")
dat_ne <- bind_rows(dat_ne_lapply)
```

We will reduce it to three columns (latitude, longitude and species presence (=1)). We remove any record without coordinates, and write it out as a csv file

```
sp <- dat_ne[,c('decimalLongitude','decimalLatitude')]
sp$species <- 1
sp <- na.omit(sp)
write.csv(sp,paste0(base.dir,'sdm_R/GBIF_data/',species_name,'.csv'))
```

Now we tell R which fields are representing geo-coordinates, and plot the locations on a map (over one of the environmental variables)

```
coordinates(sp) <- ~ decimalLongitude + decimalLatitude
par(mar=c(1,1,1,1))
par(mfrow=c(1,1))
plot(climate[[1]], main = paste("Presence records"))
plot(sp, pch=21, cex=0.5, add=T)
```



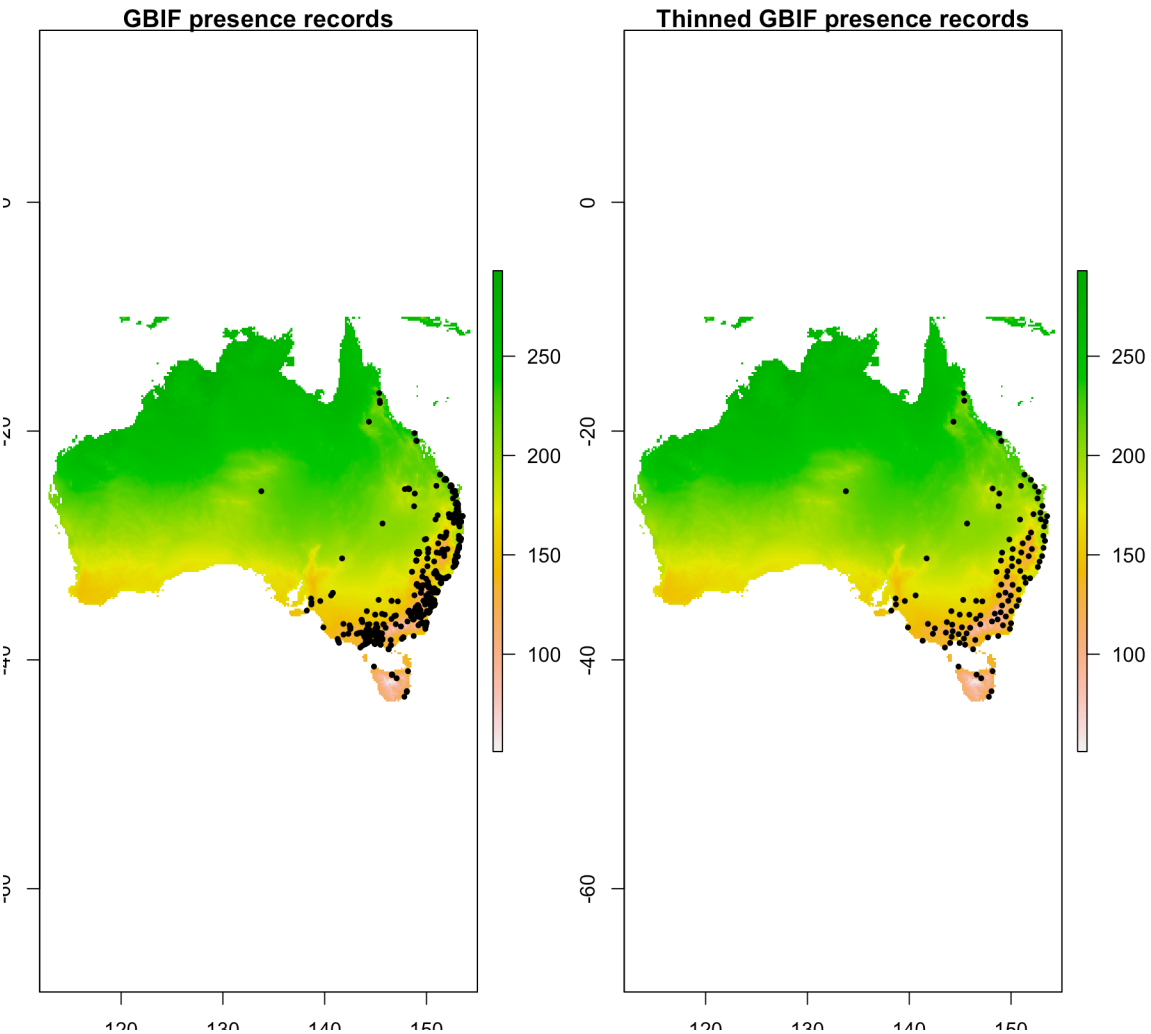
Much of your occurrence data downloaded will have a lot of spatial aurocorrelation, leading to sampling bias (e.g. points very close together), which will introduce bias in the SDMs you build. We will deal with this by spatially rarefying the occurrence data using the spThin R package. The thin.par argument defines the min number of km apart samples should be to be kept in the dataset (in this case let's choose 50km apart)

```
library(spThin)
sp_to_thin <-
read.csv(paste0(base.dir,'sdm_R/GBIF_data/',species_name,'.csv'))
out.dir <- paste0(base.dir,'sdm_R/GBIF_data/thinned/')

thinned_dataset <- thin( loc.data = sp_to_thin,
                         lat.col = "decimalLatitude", long.col =
"decimalLongitude",
                         spec.col = "species",
                         thin.par = 50, reps = 1,
                         locs.thinned.list.return = TRUE,
                         write.files = TRUE,
                         max.files = 5,
                         out.dir = out.dir,
                         out.base = paste0(species_name),
                         write.log.file = TRUE,
                         log.file =
paste0(base.dir,'sdm_R/GBIF_data/thinned/',species_name,'_log_file.txt'))

file.rename(paste0(base.dir,'sdm_R/GBIF_data/thinned/',species_name,'_thin1.c
sv'), paste0(base.dir,'sdm_R/GBIF_data/thinned/',species_name,'_thinned.csv')
)

sp_thinned <-
```

```
read.csv(paste0(base.dir,'sdm_R/GBIF_data/thinned/',species_name,'_thinned.csv'))
coordinates(sp_thinned) <- ~ decimalLongitude + decimalLatitude
```

Let's see how that data looks now compared to the original over-clustered data

```
par(mar=c(1,1,1,1))
par(mfrow=c(1,2))
plot(climate[[1]], main = paste("GBIF presence records"))
plot(sp, pch=21, cex=0.5, add=T)
plot(climate[[1]], main = paste("Thinned GBIF presence records"))
plot(sp_thinned, pch=21, cex=0.5, add=T)
```



5. Build SDMs

We're almost ready to start building some SDMs. First load the sdm package, then use the installAll() function to setup the environment and install dependencies. This will take a

while (ca. 5 mins) and needs you to answer 'yes' when the questions come up about installing from sources. We only need to do installAll() once, no need to do it again.

```
library(sdm)
installAll()
```

Before we build models, we will first remove highly correlated predictor variables from our bioclim data. To do this we use Variable Inflation Factors in the usdm package, which needs loading first, followed by extracting all the values of each climate layer for every occurrence point for our species

```
library(usdm)
uncorrelated_vars <- extract(climate, sp)
uncorrelated_vars <- data.frame(uncorrelated_vars)
```

Then we measure the variable inflation factor for this dataframe, and then exclude all highly correlated variables from the climate layers we will use

```
vif <- vifstep(uncorrelated_vars)
vif
climate <- exclude(climate, vif)
climate
```

```
12 variables from the 19 input variables have collinearity problem:

bioclim_5 bioclim_10 bioclim_1 bioclim_7 bioclim_16 bioclim_2 bioclim_12 bioclim_11 bioclim_13 bioclim_17 bioclim_8 b
ioclim_19

After excluding the collinear variables, the linear correlation coefficients ranges between:
min correlation ( bioclim_9 ~ bioclim_3 ):  0.02117659
max correlation ( bioclim_9 ~ bioclim_4 ):  -0.6148112

---------- VIFs of the remained variables --------
   Variables      VIF
1 bioclim_14 3.939049
2 bioclim_15 3.091031
3 bioclim_18 4.494917
4  bioclim_3 1.917855
5  bioclim_4 4.188830
6  bioclim_6 3.338816
7  bioclim_9 3.051843
```

```
class      : RasterStack
dimensions : 204, 258, 52632, 7  (nrow, ncol, ncell, nlayers)
resolution : 0.1666667, 0.1666667  (x, y)
extent     : 112, 155, -44, -10  (xmin, xmax, ymin, ymax)
crs        : NA
names      : bioclim_14, bioclim_15, bioclim_18, bioclim_3, bioclim_4, bioclim_6, bioclim_9
```

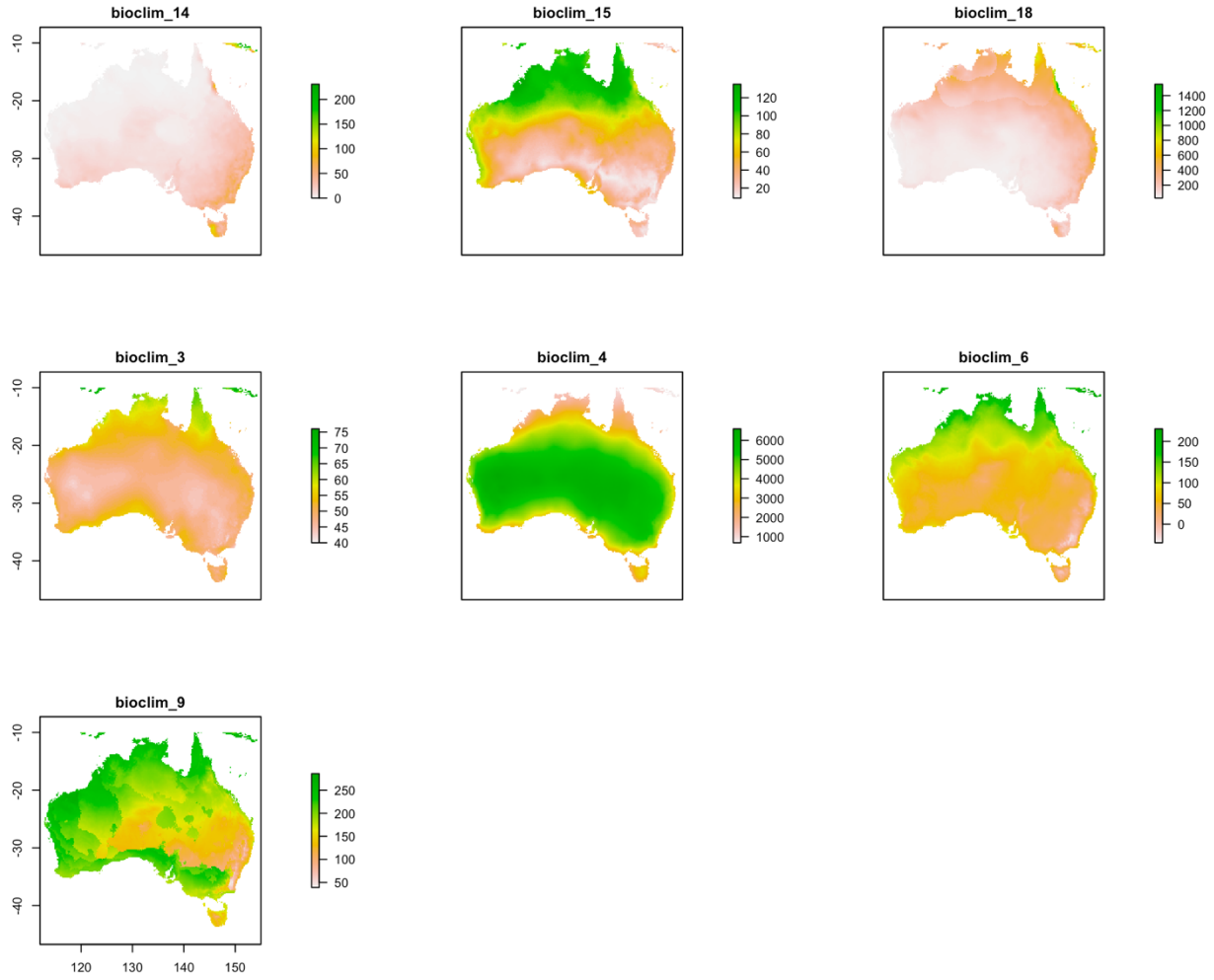
We'll do the same for the future climate dataset you have downloaded, so that we now have a set of uncorrelated variables both for the present time period, and for a future time period

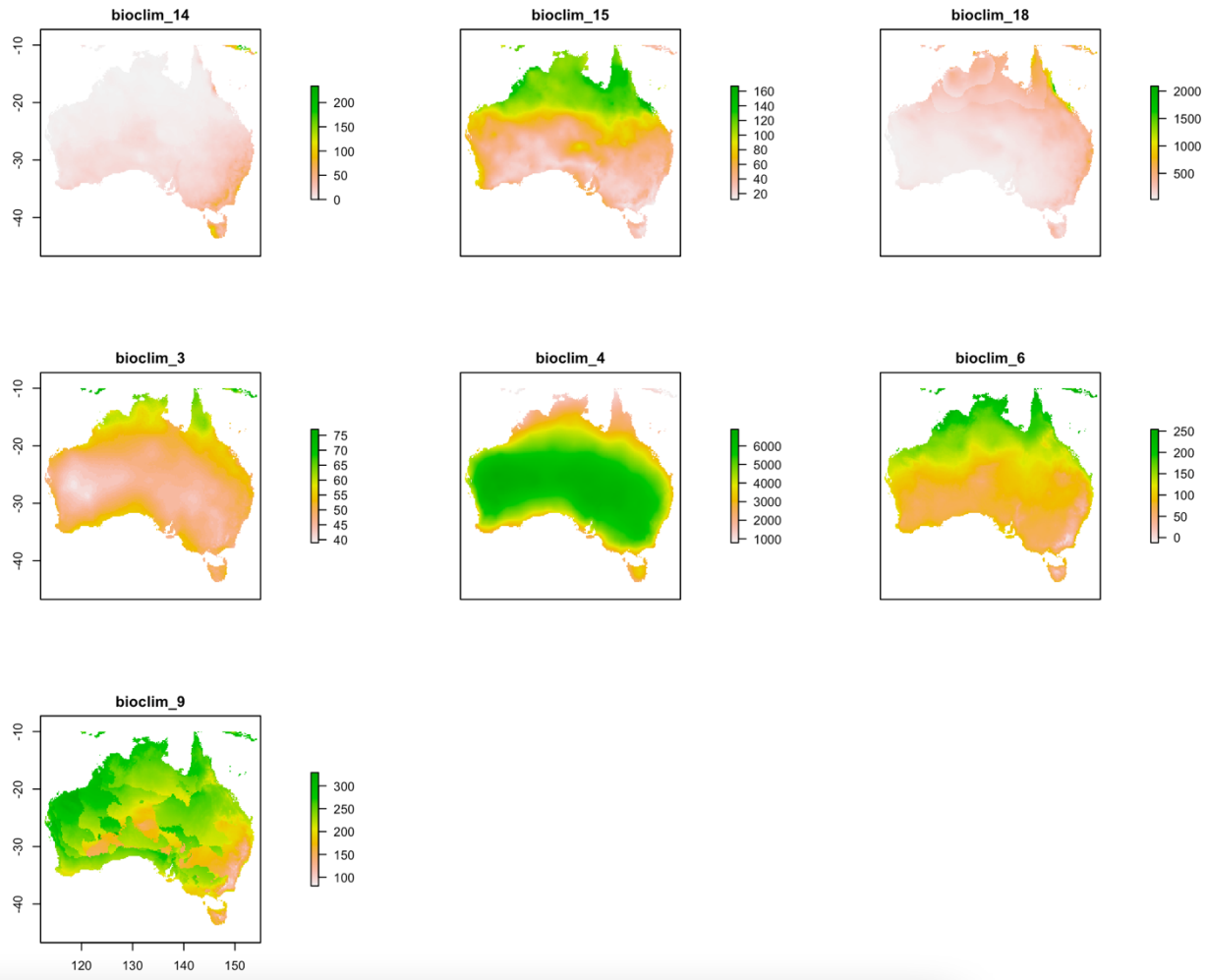
```
future_climate <- exclude(future_climate, vif)
future_climate
```

```
class       : RasterStack
dimensions  : 204, 258, 52632, 7  (nrow, ncol, ncell, nlayers)
resolution  : 0.1666667, 0.1666667  (x, y)
extent      : 112, 155, -44, -10  (xmin, xmax, ymin, ymax)
crs         : NA
names       : bioclim_14, bioclim_15, bioclim_18, bioclim_3, bioclim_4, bioclim_6, bioclim_9
```

Let's have a quick look at the remaining variables

```
par(mar=c(1,1,1,1))
par(mfrow=c(3,3))
plot(climate)
plot(future_climate)
```





Now you are ready to start building SDMs! Let's begin by making sure our models are stored in the ensembles folder. We'll read in the species .csv file with presence data and double check that coordinates are assigned

```
setwd(paste0(base.dir,'sdm_R/ensembles/'))

sp <-
read.csv(paste0(base.dir,'sdm_R/GBIF_data/thinned/',species_name,'_thinned.cs
v'))
sp <- sp[,c('decimalLongitude','decimalLatitude')]
sp <- na.omit(sp)
sp$presence <- 1
coordinates(sp) <- ~ decimalLongitude + decimalLatitude
```

We will tell the sdm R package which field lists the data (occurence points) to be used, along with training data to calibrate the model. In this case we will use the data itself, but we will partition it into training and evaluation data later on. We also define the environmental variables (predictors) that the model will use and the number of background points (i.e. places where the species is 'absent').

```
d <- sdmData(presence~., train=sp, predictors=climate, bg=list(n=250))
```

If you examine this object you'll see some info on the model object that will be run

```
d
```

```
class                                 : sdmdata
============================================================
number of species                     :  1
species names                         :  presence
number of features                    :  7
feature names                         :  bioclim_14, bioclim_15, bioclim_18, ...
type                                  :  Presence-Background
has independet test data?             :  FALSE
number of records                     :  364
has Coordinates?                      :  TRUE
```

Then we will list which SDM algorithms are available to use in the package

```
getmethodNames()
```

Let's try quickly running four algorithms - Bioclim, Boosted Regression Trees, Generalised Additive Models and Random Forest, each for 3 runs, using the subsampling method. Here we'll also define the test percentage of the data (70% of the data for training the model, 30% for testing)

```
methods <- c('bioclim','brt','gam','rf')
m <- sdm(presence ~., d, methods=methods, replication='sub', test.p=30, n=3)
```

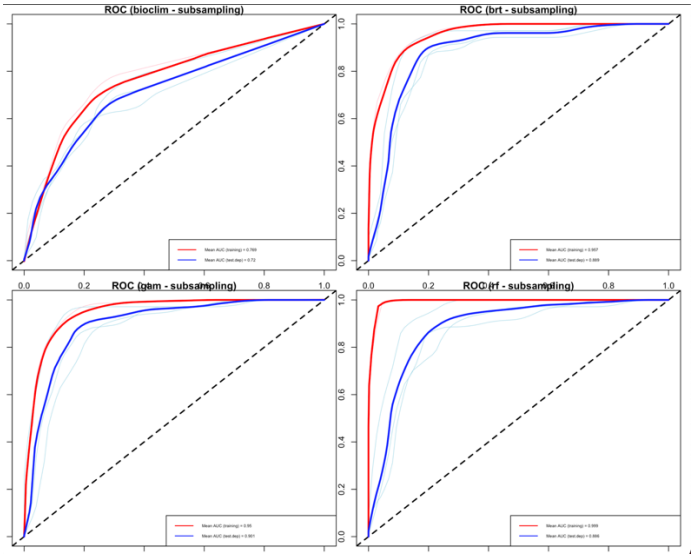
When the models are complete you can visualise a table with their model evaluation scores - AUC, TSS and COR.

```
m
```

```
class                                 : sdmModels
=======================================================
number of species                     :  1
number of modelling methods           :  4
names of modelling methods            :  bioclim, brt, gam, rf
replicate.methods (data partitioning) :  subsampling
number of replicates (each method)    :  3
toral number of replicates per model  :  3 (per species)
test percentage (in subsampling)      :  30
----------------------------------------
model run success percentage (per species)  :
----------------------------------------
method          presence
----------------------
bioclim    :        100   %
brt        :        100   %
gam        :        100   %
rf         :        100   %

#####################################################################
model Mean performance (per species), using test dataset (generated using partitioning):
-------------------------------------------------------------------------------

 ## species   :  presence
=========================

methods    :     AUC     |     COR     |     TSS     |     Deviance
-------------------------------------------------------------------------
bioclim    :     0.72    |     0.36    |     0.42    |     1.83
brt        :     0.89    |     0.67    |     0.74    |     0.84
gam        :     0.9     |     0.69    |     0.74    |     0.88
rf         :     0.89    |     0.66    |     0.71    |     0.84
```

You can see that some algorithms perform better than others (depending on input data), also evident by inspecting the AUC curves. If you want to you can explore the results interactively using the gui (commented out here, just remove the # if you want to look at it)

```
roc(m, smooth=T)
#gui(m)
```

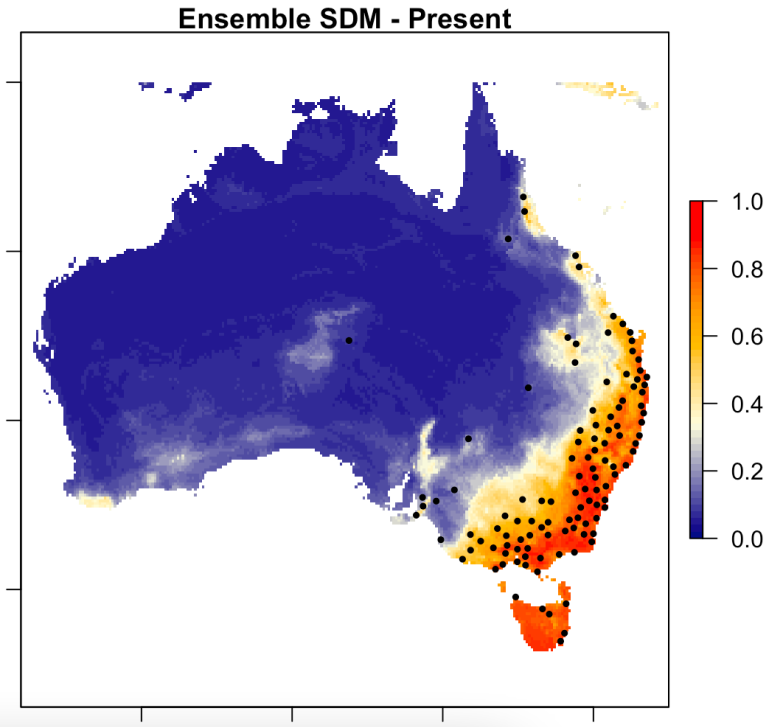


We can see here that most algorithms worked pretty well, with AUC scores fairly ok, with the exception of the bioclim algorithm. Let's take all of these models into our final ensemble, except bioclim - this is easy because the first 3 models represent bioclim, and the following 9 represent brt, gam and rf (all models have 3 replicates each)

```
ensemble_model  <- ensemble(m,climate, setting=list(id=c(4:12),
method='weighted', stat='AUC', opt=2))
```

We can plot the final ensemble model with our own colour scheme and write it as an asc file

```
setwd(paste0(base.dir))

colours <- colorRampPalette(c('navy','lightyellow','orange','red'))(50)
par(mar=c(1,1,1,1))
par(mfrow=c(1,1))
plot(ensemble_model, main = "Ensemble SDM - Present", col=colours,
zlim=c(0,1))
points(sp_thinned, pch=19, cex=0.5, add=TRUE)
writeRaster(ensemble_model, './sdm_R/ensembles/present.asc', format="ascii",
overwrite=TRUE)
file.rename('./sdm_R/ensembles/present.asc',
paste0('./sdm_R/ensembles/',species_name,'_ensemble_present.asc'))
```



The sdm R package always writes temporary files as .grd and .gri - we will clean these up by deleting them as we go along (we already wrote the files we need as .asc)

```
lst1 <- list.files(path=getwd(),pattern='grd$',full.names = T)
lst2 <- list.files(path=getwd(),pattern='gri$',full.names = T)
file.remove(lst1)
file.remove(lst2)
```

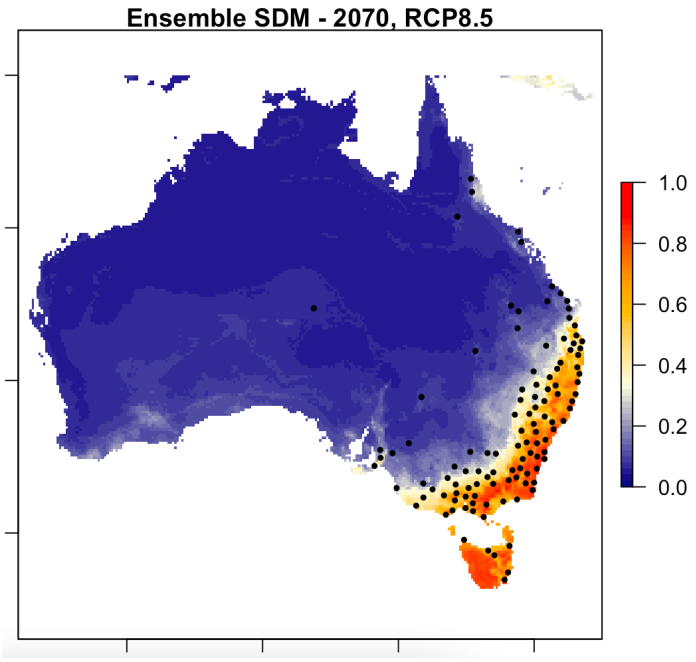
6. Project SDMs from present to future climate

Now that we have our ensemble SDM for our species, we can use the future climate data we already prepared to project how our species distribution will be in the future (the year 2070, under the RCP 8.5 scenario)

```
future_ensemble_model  <- ensemble(m,newdata=future_climate,
setting=list(id=4:12, method='weighted', stat='AUC', opt=2))
```

Just like before we can plot the final ensemble model with our own colour scheme and write it as an asc file

```
colours <- colorRampPalette(c('navy','lightyellow','orange','red'))(50)
par(mar=c(1,1,1,1))
par(mfrow=c(1,1))
plot(future_ensemble_model, main = "Ensemble SDM - 2070, RCP8.5",
col=colours, zlim=c(0,1))
points(sp_thinned, pch=19, cex=0.5, add=TRUE)
writeRaster(future_ensemble_model,
'./sdm_R/ensembles/future_2070_RCP_8.5.asc', format="ascii", overwrite=TRUE)
file.rename('./sdm_R/ensembles/future_2070_RCP_8.5.asc',paste0('./sdm_R/ensem
bles/',species_name,'_ensemble_future_2070_RCP_8.5.asc'))
```



And again clear up temporary files when exiting

```
lst1 <- list.files(path=getwd(),pattern='grd$',full.names = T)
lst2 <- list.files(path=getwd(),pattern='gri$',full.names = T)
file.remove(lst1)
file.remove(lst2)
```

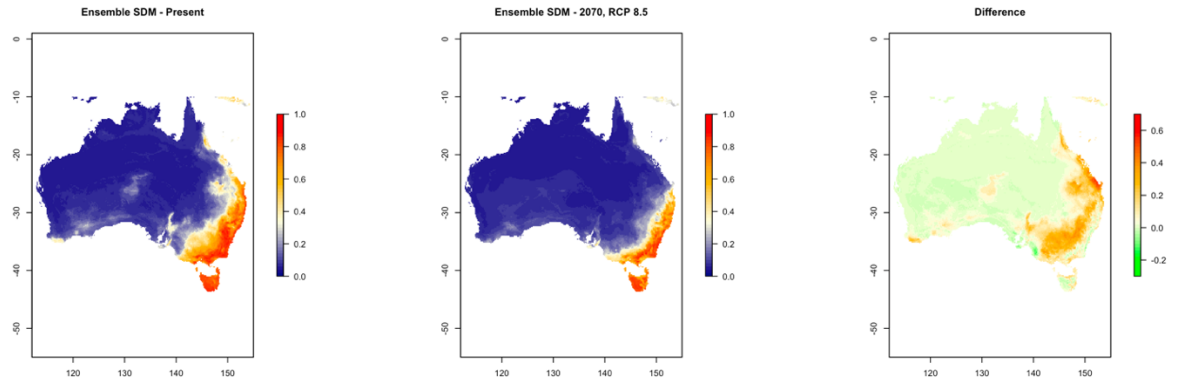
7. Now compare present vs future models and evaluate them

Now we can take a detailed look at the present vs the future models. We will read in the files from the rasters that we wrote, and calculate the difference in change between then (per pixel)

```
ensemble_model <-
raster(paste0('./sdm_R/ensembles/',species_name,'_ensemble_present.asc'))
future_ensemble_model <-
raster(paste0('./sdm_R/ensembles/',species_name,'_ensemble_future_2070_RCP_8.
5.asc'))
difference = ensemble_model-future_ensemble_model
```

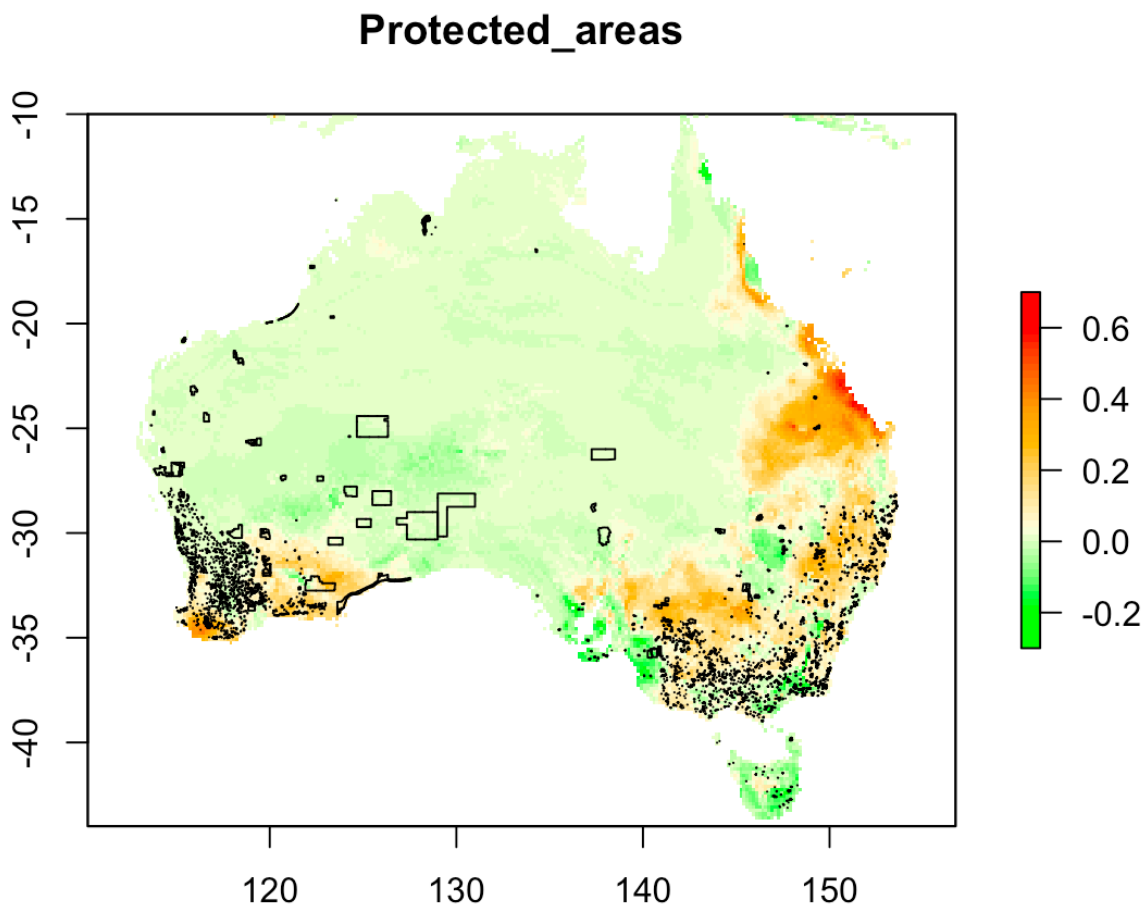
We will then set a different colour legend for the 'difference' raster layer we just made, and plot the three images side by side - Present distribution, Future (2070) distribution and the Difference between the two. We'll save this as a tiff image

```
difference_colours <-
colorRampPalette(c('green','lightyellow','orange','red'))(50)
par(mar=c(5.1, 4.1, 4.1, 2.1))
par(mfrow=c(1,3))
plot(ensemble_model, main = "Ensemble SDM - Present", col=colours,
zlim=c(0,1))
plot(future_ensemble_model, main = "Ensemble SDM - 2070, RCP 8.5",
col=colours, zlim=c(0,1))
plot(difference, main="Difference", col=difference_colours, zlim=c(-0.3,0.7))
dev.copy(tiff, './comparison.tiff', width=2900,height=1600, res=300)
dev.off()
file.rename('./comparison.tiff',
paste0('./',species_name,'_Present_vs_2070_RCP8.5_comparison.tiff'))
```



For applied conservation management you could look at how the projected changes align with the current protected area network for example

```
Australia_protected_areas <-
readOGR('./Protected_areas/Australia_protected_areas.shp')
par(mar=c(5.1, 4.1, 4.1, 2.1))
par(mfrow=c(1,1))
plot(difference, main="Difference", col=difference_colours, zlim=c(-0.3,0.7))
lines(Australia_protected_areas)
dev.copy(tiff, './Protected_Areas_comparison_PAs.tiff',
width=2900,height=1600, res=300)
dev.off()
file.rename('./Protected_Areas_comparison_PAs.tiff',
paste0('./',species_name,'_Protected_Areas_Present_vs_2070_RCP8.5_comparison.
tiff'))
dev.off()
```



Tutorial assignment

Now it's your turn - Use the code above as a basis to do the following:

Select a species that you like - preferably one that has enough records to be able to build a reliable model (i.e. at least over 100 presence records after spatially rarefying the occurrence data to an appropriate resolution, with a good proportion of its distribution range sampled). Build an ensemble SDM and project this into a future climate scenario (i.e. one of the RCP scenarios, 2.6-8.5 in 2050 or 2070). You will need to decide on a suitable template for the extent of your modelling area (i.e. the species range), and a suitable spatial resolution of your predictor variables (e.g. 10 arc degrees, or 5 or 2.5). You may also want to spatially rarefy the species occurrence data in case it is spatially clustered.

Report your findings at the end of the tutorial (max 5 mins, e.g. 5 slides). Here you could include information on: - Your species - is it widespread or does it have a small geographic

range? - Which modelling algorithms did you use to build your ensemble, and how did they perform in terms of AUC and TSS? - How many occurence points did you download? Did you spatially rarefy them? To what distance? And how many are left after spatial rarefaction? - How many background points did you use? - How do you interpret the predicted changes between present and future - How will the species range be affected? (no change/contraction/expansion?) - Do you think your model is accurate? If not, why not? - How could you improve the quality of your modelling approach to make more accurate predictions? (Think back to some of the points we discussed in the lecture regarding this)