

# Preprocessing

```
#Restore workspace - includes all the objects/libraries from the last run. This way we don't have to run
library(session)
restore.session(file='Preprocessing.Rda')

## Loading all data...
## Loading packages...

##
## Welcome to installr version 0.22.0
##
## More information is available on the installr project website:
## https://github.com/talgalili/installr/
##
## Contact: <tal.galili@gmail.com>
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/installr/issues
##
##           To suppress this message use:
##           suppressPackageStartupMessages(library(installr))

## rgdal: version: 1.4-4, (SVN revision 833)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 2.2.3, released 2017/11/20
## Path to GDAL shared files: C:/Users/Priyu/Documents/R/win-library/3.6/rgdal/gdal
## GDAL binary built with GEOS: TRUE
## Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]
## Path to PROJ.4 shared files: C:/Users/Priyu/Documents/R/win-library/3.6/rgdal/proj
## Linking to sp version: 1.3-1

## Checking rgeos availability: TRUE

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess

## randomForest 4.6-14

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

##
## Attaching package: 'ggplot2'

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

##
## Attaching package: 'e1071'

## The following object is masked from 'package:raster':
##
##     interpolate

##
```

```

## Attaching package: 'neuralnet'
## The following object is masked from 'package:ROCR':
##
##      prediction
##
## Attaching package: 'MASS'
## The following objects are masked from 'package:raster':
##
##      area, select
##
## rgeos version: 0.5-1, (SVN revision 614)
## GEOS runtime version: 3.6.1-CAPI-1.10.1
## Linking to sp version: 1.3-1
## Polygon checking: TRUE
##
## Attaching package: 'lubridate'
## The following object is masked from 'package:base':
##
##      date
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:lubridate':
##
##      intersect, setdiff, union
## The following objects are masked from 'package:rgeos':
##
##      intersect, setdiff, union
## The following object is masked from 'package:MASS':
##
##      select
## The following object is masked from 'package:neuralnet':
##
##      compute
## The following object is masked from 'package:randomForest':
##
##      combine
## The following objects are masked from 'package:raster':
##
##      intersect, select, union
## The following objects are masked from 'package:stats':
##
##      filter, lag
## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union
## Registered S3 method overwritten by 'R.oo':
##   method      from

```

Download additional observation data for all species from GBIF (Global Biodiversity Information Facility)  
<https://www.gbif.org>

```

#Loading the data. High reliability = values that are acceptable or confirmed
#If you filter by acceptable, confirmed, and high reliability some species have 0 observation points or
#For the purpose of training, we choose points that are either acceptable/confirmed
data = read.csv("Monash_sample_VBA.csv", stringsAsFactors = FALSE)
data = data[data$RELIABILITY != "Unconfirmed" & data$RELIABILITY != "",]

#Separate each of the species into their own dataframe
Agile = data[data$COMMON_NME == "Agile Antechinus",]
BrownTreeCreeper = data[data$COMMON_NME == "Brown Treecreeper",]
CommonBeardheath = data[data$COMMON_NME == "Common Beard-heath",]
SmallTriggerplant = data[data$COMMON_NME == "Small Triggerplant",]
SouthernBrownTreeFrog = data[data$COMMON_NME == "Southern Brown Tree Frog",]
WhitebrowedTreeCreeper = data[data$COMMON_NME == "White-browed Treecreeper",]

#Get latitude and longitude values; longitude is the first and latitude the second column
Agile.c = Agile[,c(13,12)]
BrownTreeCreeper.c= BrownTreeCreeper[,c(13,12)]
CommonBeardheath.c= CommonBeardheath[,c(13,12)]
SmallTriggerplant.c= SmallTriggerplant[,c(13,12)]
SouthernBrownTreeFrog.c = SouthernBrownTreeFrog[,c(13,12)]
WhitebrowedTreeCreeper.c= WhitebrowedTreeCreeper[,c(13,12)]

----- NO NEED TO RUN THIS AGAIN. RESULT IS ALREADY SAVED -----

# #Additional coordinates downloaded from Global Biodiversity Information Facility
# Agile.gbif = gbif('Antechinus', 'agilis') #https://www.gbif.org/species/2435369
# BrownTreeCreeper.gbif = gbif('Climacteris', 'picumnus') #https://www.gbif.org/species/2494126
# SmallTriggerplant.gbif = gbif("Stylium", "despectum") #https://www.gbif.org/species/5543531
# SouthernBrownTreeFrog.gbif = gbif("Litoria", "ewingii")#https://www.gbif.org/species/2427705
# WhitebrowedTreeCreeper.gbif = gbif("Climacteris", "affinis") #https://www.gbif.org/en/species/2494130
# CommonBeardheath.gbif = CommonBeardheath.gbif[CommonBeardheath.gbif$acceptedScientificName == "Leucophaea_melanosticta"]

#save downloaded gbif data
# write.csv(Agile.gbif, "Agile.csv")
# write.csv(BrownTreeCreeper.gbif, "BrownTreeCreeper.csv")
# write.csv(CommonBeardheath.gbif, "CommonBeardheath.csv")
# write.csv(SmallTriggerplant.gbif, "SmallTriggerplant.csv")
# write.csv(SouthernBrownTreeFrog.gbif, "SouthernBrownTreeFrog.csv")
# write.csv(WhitebrowedTreeCreeper.gbif, "WhitebrowedTreecreeper.csv")



#Read saved species data from gbif
Agile.gbif = read.csv("Agile.csv")
BrownTreeCreeper.gbif = read.csv("BrownTreeCreeper.csv")
CommonBeardheath.gbif = read.csv("CommonBeardheath.csv")
SmallTriggerplant.gbif = read.csv("SmallTriggerplant.csv")
SouthernBrownTreeFrog.gbif = read.csv("SouthernBrownTreeFrog.csv")
WhitebrowedTreeCreeper.gbif = read.csv("WhitebrowedTreecreeper.csv")

#Filter years in the downloaded GBIF data so the earliest year => earliest year of given DELWP dataset

```

```

earliest_year = min(year(dmy(Agile$SURVEY_START_DATE)))
Agile.gbif = Agile.gbif[Agile.gbif$year >= earliest_year,]

earliest_year = min(year(dmy(BrownTreecreeper$SURVEY_START_DATE)))
BrownTreeCreeper.gbif = BrownTreeCreeper.gbif[BrownTreeCreeper.gbif$year >= earliest_year,]

earliest_year = min(year(dmy(CommonBeardheath$SURVEY_START_DATE)))
CommonBeardheath.gbif = CommonBeardheath.gbif[CommonBeardheath.gbif$year >= earliest_year,]

earliest_year = min(year(dmy(SmallTriggerplant$SURVEY_START_DATE)))
SmallTriggerplant.gbif = SmallTriggerplant.gbif[SmallTriggerplant.gbif$year >= earliest_year,]

earliest_year = min(year(dmy(SouthernBrownTreeFrog$SURVEY_START_DATE)))
SouthernBrownTreeFrog.gbif = SouthernBrownTreeFrog.gbif[SouthernBrownTreeFrog.gbif$year >= earliest_year,]

earliest_year = min(year(dmy(WhitebrowedTreecreeper$SURVEY_START_DATE)))
WhitebrowedTreecreeper.gbif = WhitebrowedTreecreeper.gbif[WhitebrowedTreecreeper.gbif$year >= earliest_year,]

```

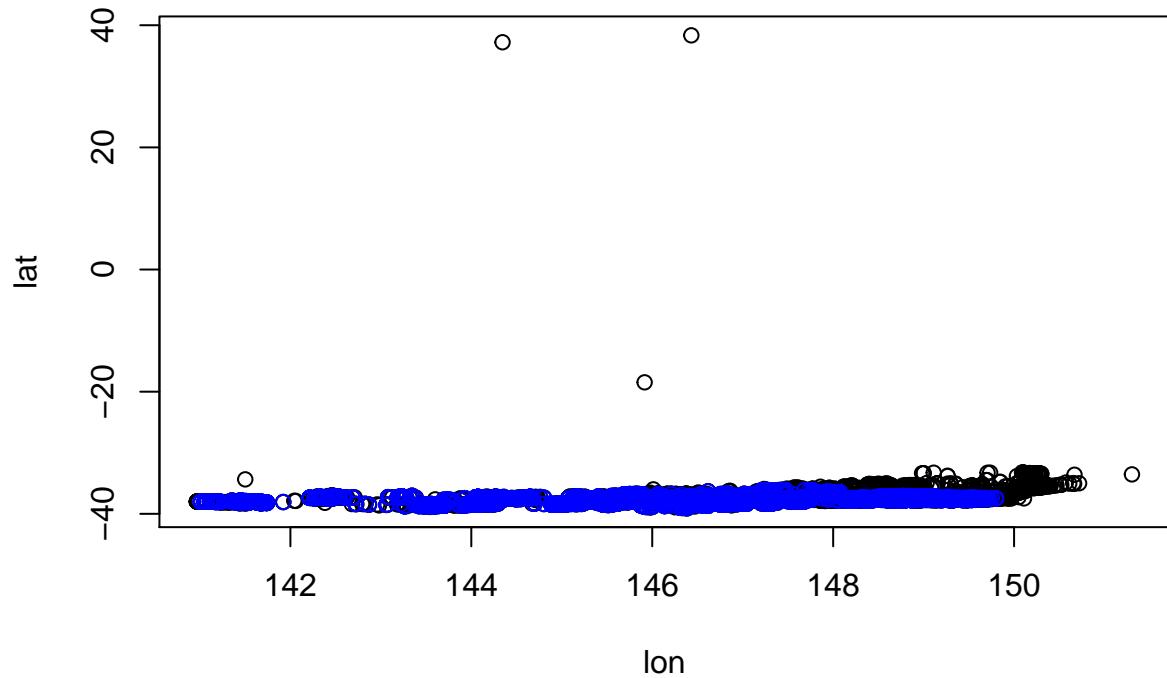
## AGILE

```

#Merge lon/lat values with gbif coords and clean the data.
Agile.cgbif = subset(Agile.gbif, !is.na(lon) & !is.na(lat)) #Select non-empty lat/lon columns
Agile.cgbif = Agile.cgbif[,c("lon","lat")] #Get latitude and longitude values
colnames(Agile.c) = c("lon","lat")
Agile.cmerged = rbind(Agile.c,Agile.cgbif) #Merge GBIF values with data from DEWLP
Agile.cmerged = distinct(Agile.cmerged) #Remove duplicated matching lat/lon values

#We see a few outliers
plot(Agile.cmerged)
points(Agile.c, col = 'blue')

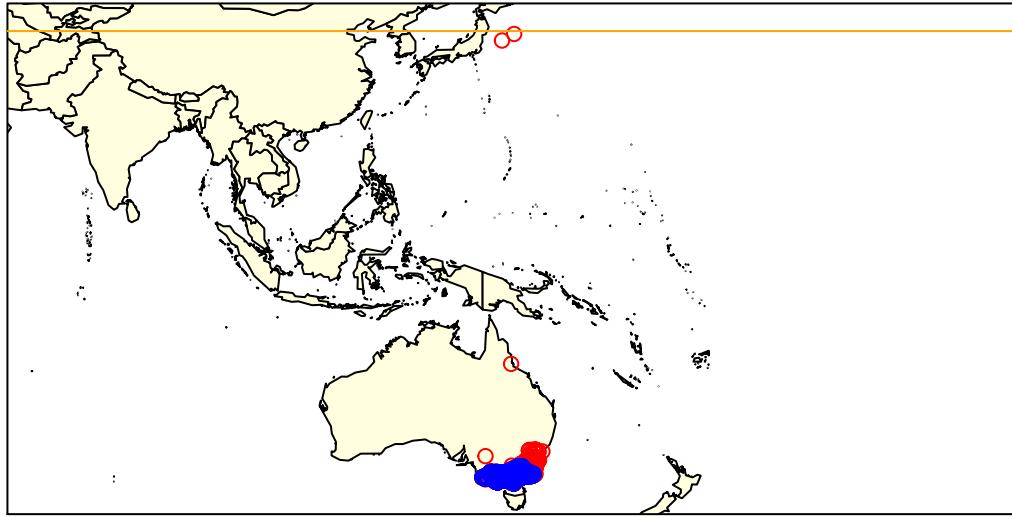
```



```

#See overlap of merged points and points given just by DELWP
#RED POINTS = additional observations from gbif, Blue points = observations in DELWP dataset
data("wrld_simpl")
plot(wrld_simpl, ylim = c(min(Agile.cmerged$lat - 2), max(Agile.cmerged$lat) + 2 ), xlim = c(min(Agile
points(Agile.cmerged, col = "red")
points(Agile.c, col = "blue")
abline(h = (max(Agile.cmerged$lat)) + 0.5, col = 'orange')

```



```

#Remove 'outliers'
Agile.cmerged = Agile.cmerged[Agile.cmerged$lat < -30,]
# Agile.cmerged = Agile.cmerged[Agile.cmerged$lon >= min(Agile.c$lon) & Agile.cmerged$lon <= max(Agile.c$lon),]
# Agile.cmerged = Agile.cmerged[Agile.cmerged$lat >= min(Agile.c$lat) & Agile.cmerged$lat <= max(Agile.c$lat),]

#Cross checking - this ensures coordinates are valid and within Australia
data("wrld_simpl")
Agile.cmerged2 = Agile.cmerged
coordinates(Agile.cmerged2) = ~lon+lat
crs(Agile.cmerged2) <- crs(wrld_simpl)
ovr <- over(Agile.cmerged2, wrld_simpl)
unique(ovr$NAME)

## [1] <NA>      Australia
## 246 Levels: Aaland Islands Afghanistan Albania Algeria ... Zimbabwe
#There are NA entries, i.e. not from any country which should be removed
i <- which(ovr$NAME == "Australia")

#Easier is to only select points that are from australia since this species is only in Australia
Agile.cmerged = Agile.cmerged[c(i),]

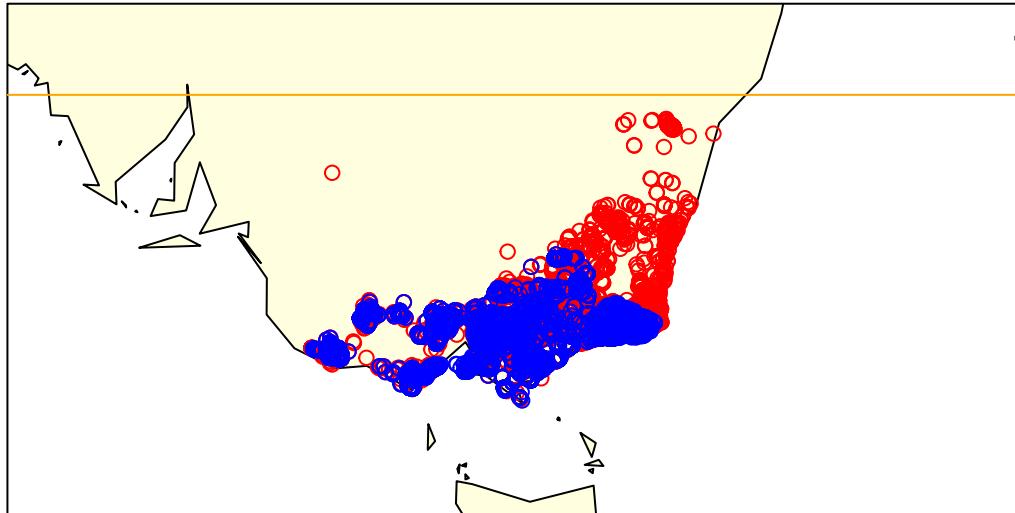
#Final inspection of all observation points
#RED POINTS = additional observations from gbif, Blue points = observations in DELWP dataset
data("wrld_simpl")
plot(wrld_simpl, ylim = c(min(Agile.cmerged$lat - 2), max(Agile.cmerged$lat) + 2 ), xlim = c(min(Agile.cmerged$lon - 2), max(Agile.cmerged$lon) + 2 ))

```

```

points(Agile.cmerged, col = "red")
points(Agile.c, col = "blue")
abline(h = (max(Agile.cmerged$lat)) + 0.5, col = 'orange')

```



```

#Save the merged, preprocess coordinates as a CSV file, only needs to be run once
# write.csv(Agile.cmerged, "AgileC.csv")

```

Procedure is similar for the others

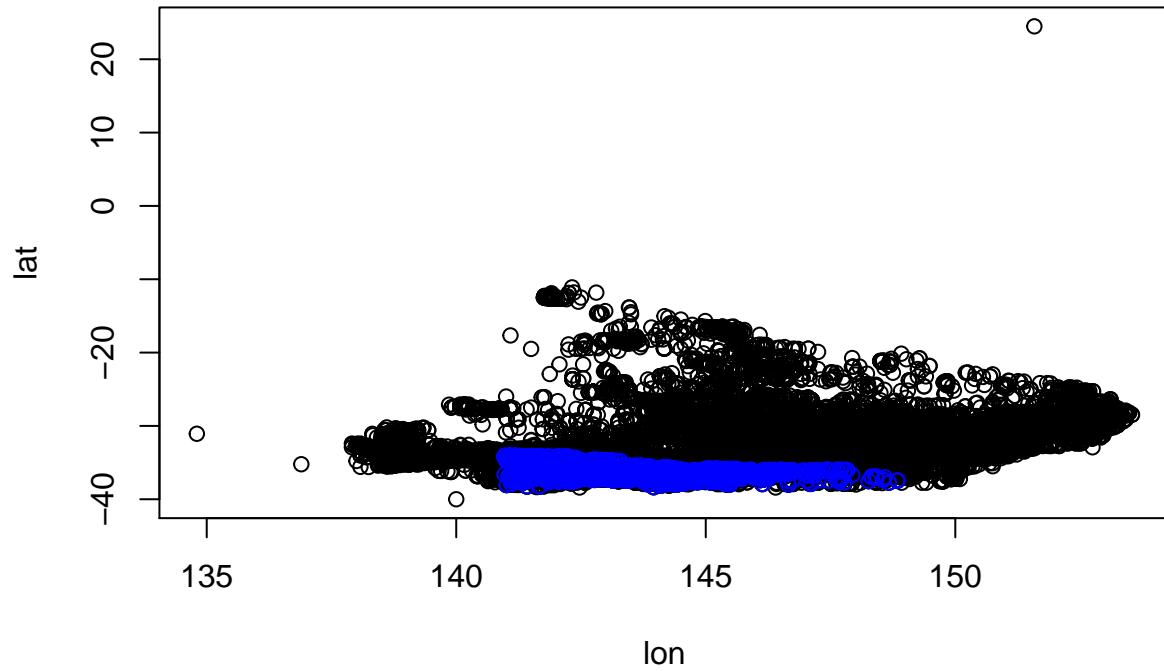
BrownTreeCreeper

```

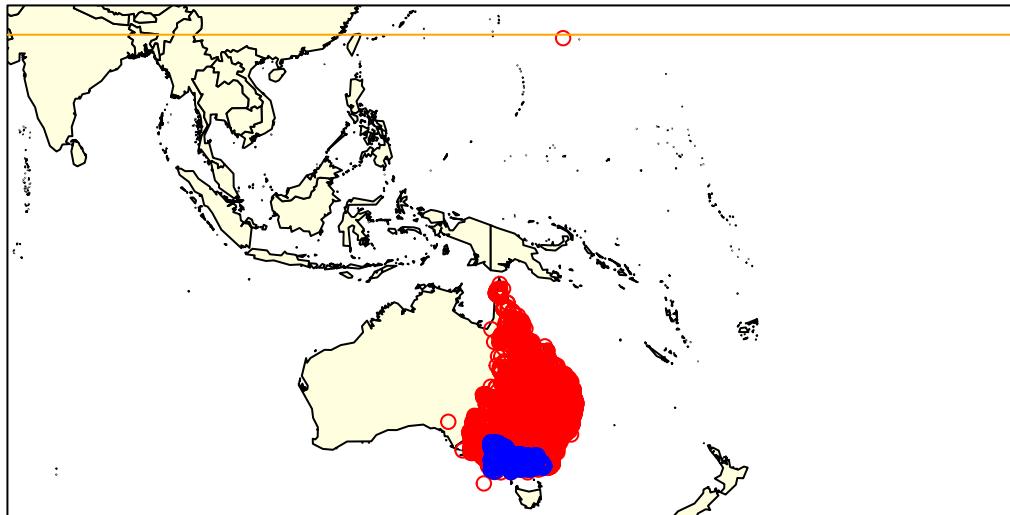
BrownTreeCreeper.cgbif = subset(BrownTreeCreeper.gbif, !is.na(lon) & !is.na(lat)) #Select non-empty lat/long
BrownTreeCreeper.cgbif = BrownTreeCreeper.cgbif[,c("lon","lat")] #Get latitude and longitude values
colnames(BrownTreeCreeper.c) = c("lon","lat")
BrownTreeCreeper.cmerged = rbind(BrownTreeCreeper.c,BrownTreeCreeper.cgbif) #Merge GBIF values with data
BrownTreeCreeper.cmerged = distinct(BrownTreeCreeper.cmerged) #Remove duplicated matching lat/lon values

#A few outliers
plot(BrownTreeCreeper.cmerged)
points(BrownTreeCreeper.c, col = 'blue')

```



```
#See overlap of merged points and points given just by DELWP
#RED POINTS = additional observations from gbif, Blue points = observations in DELWP dataset
plot(wrld_simpl, ylim = c(min(BrownTreeCreeper.cmerged$lat - 2), max(BrownTreeCreeper.cmerged$lat) + 2)
points(BrownTreeCreeper.cmerged, col = "red")
points(BrownTreeCreeper.c, col = "blue")
abline(h = (max(BrownTreeCreeper.cmerged$lat)) + 0.5, col = 'orange')
```



```
#Huge variance between given DELWP data and gbif. Also, too many points, will take long to train models
BrownTreeCreeper.cmerged = BrownTreeCreeper.cmerged[BrownTreeCreeper.cmerged$lat < 0 & BrownTreeCreeper.cmerged$lon >= min(BrownTreeCreeper.cmerged$lon) & BrownTreeCreeper.cmerged$lon <= max(BrownTreeCreeper.cmerged$lon), drop = TRUE]

#Cross checking
data("wrld_simpl")
BrownTreeCreeper.cmerged2 = BrownTreeCreeper.cmerged
coordinates(BrownTreeCreeper.cmerged2) = ~lon+lat
crs(BrownTreeCreeper.cmerged2) <- crs(wrld_simpl)
ovr <- over(BrownTreeCreeper.cmerged2, wrld_simpl)
unique(ovr$NAME)

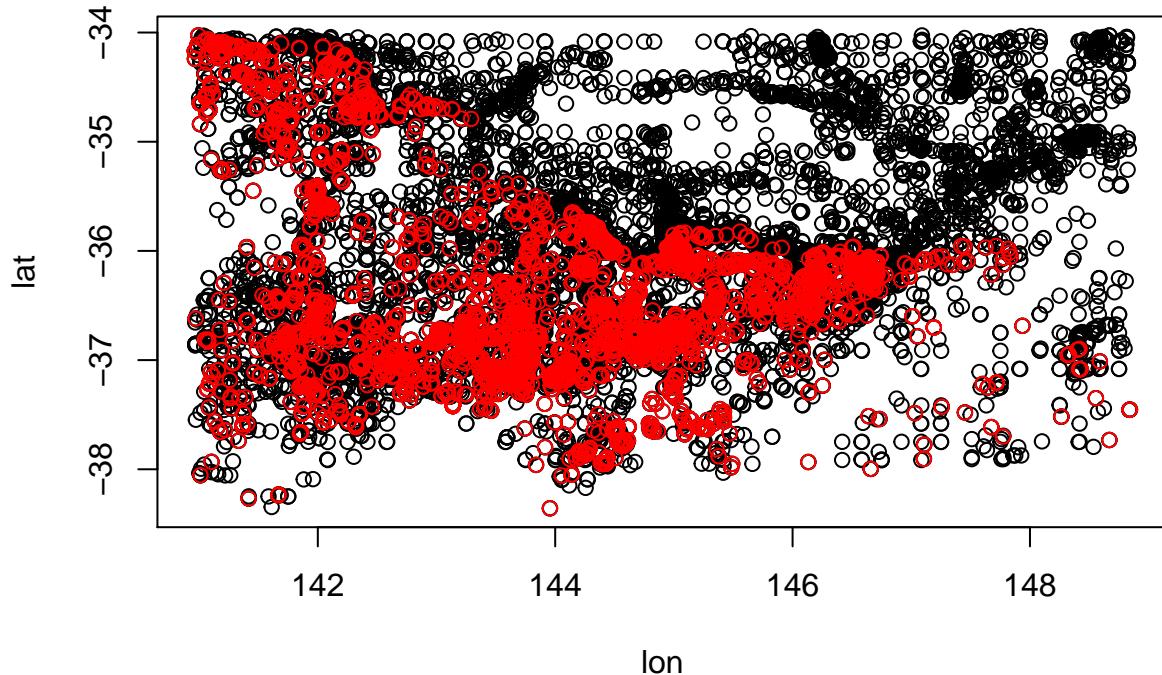
## [1] Australia <NA>
## 246 Levels: Aaland Islands Afghanistan Albania Algeria ... Zimbabwe
#There are NA entries, i.e. not from any country which should be removed
i <- which(ovr$NAME == "Australia")

#Easier is to only select points that are from australia since this species is only in Australia
```

```
BrownTreeCreepers.cmerged = BrownTreeCreepers.cmerged[c(i),]
```

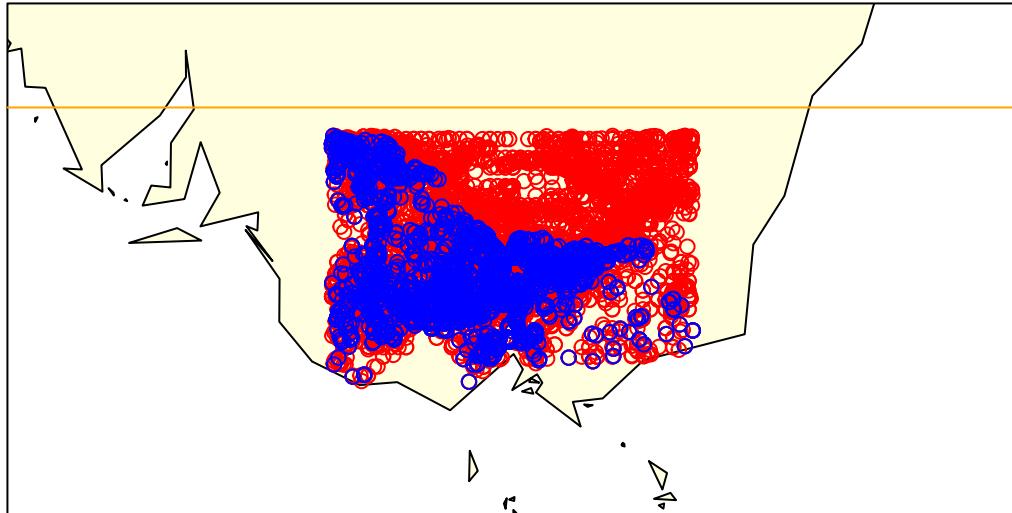
```
#Let's see how the coordinates look now.. red = only points from DELWP. black = points from both DELWP
```

```
plot(BrownTreeCreepers.cmerged)
points(BrownTreeCreepers.c, col = "red")
```



```
#Final inspection of all observation points
```

```
#RED POINTS = additional observations from gbif, Blue points = observations in DELWP dataset
plot(wrld_simpl, ylim = c(min(BrownTreeCreepers.cmerged$lat - 2), max(BrownTreeCreepers.cmerged$lat) + 2)
points(BrownTreeCreepers.cmerged, col = "red")
points(BrownTreeCreepers.c, col = "blue")
abline(h = (max(BrownTreeCreepers.cmerged$lat)) + 0.5, col = 'orange')
```

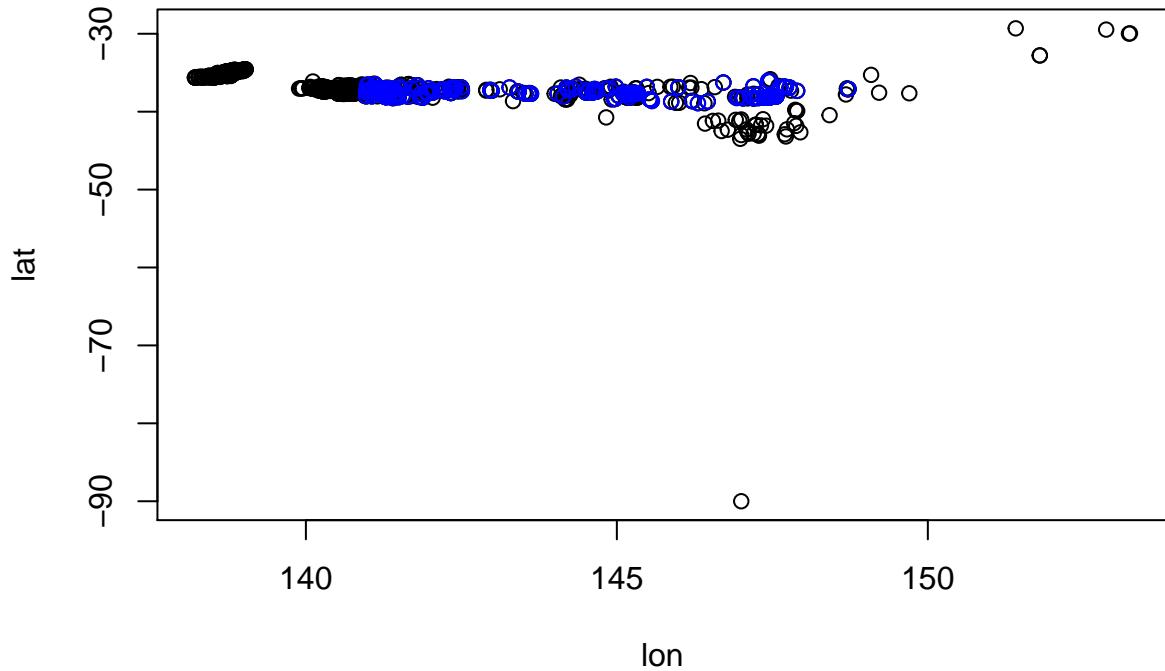


```
#Save the merged, preprocess coordinates as a CSV file, only needs to be run once
# write.csv(BrownTreeCreeper.cmerged, "BrownTreeCreeperC.csv")

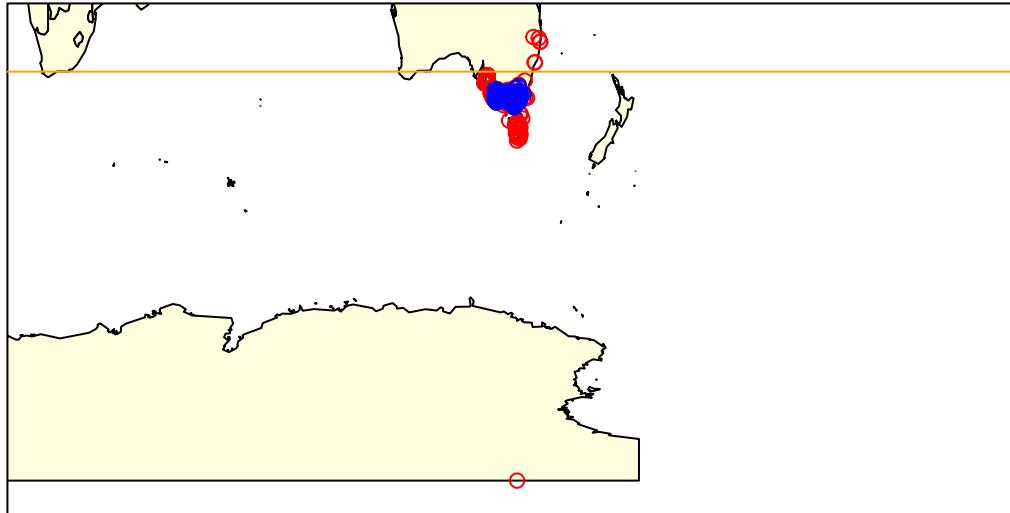
CommonBeardheath

CommonBeardheath.cgbif = subset(CommonBeardheath.gbif, !is.na(lon) & !is.na(lat)) #Select non-empty lat/lon
CommonBeardheath.cgbif = CommonBeardheath.cgbif[,c("lon","lat")] #Get latitude and longitude values
colnames(CommonBeardheath.c) = c("lon","lat")
CommonBeardheath.cmerged = rbind(CommonBeardheath.c,CommonBeardheath.cgbif) #Merge GBIF values with data
CommonBeardheath.cmerged = distinct(CommonBeardheath.cmerged) #Remove duplicated matching lat/lon values

plot(CommonBeardheath.cmerged)
points(CommonBeardheath.c, col = 'blue')
```



```
#See overlap of merged points and points given just by DELWP
#RED POINTS = additional observations from gbif, Blue points = observations in DELWP dataset
plot(wrld_simpl, ylim = c(min(CommonBeardheath.cmerged$lat - 2), max(CommonBeardheath.cmerged$lat) + 2)
points(CommonBeardheath.cmerged, col = "red")
points(CommonBeardheath.c, col = "blue")
abline(h = (sort(CommonBeardheath.cmerged$lat, TRUE)[7]) + 0.5, col = 'orange')
```



```

# CommonBeardheath.cmerged = CommonBeardheath.cmerged[CommonBeardheath.cmerged$lat >-45 & CommonBeardheath.cmerged$lon <= min(CommonBeardheath.cmerged$lon),]
# CommonBeardheath.cmerged = CommonBeardheath.cmerged[CommonBeardheath.cmerged$lat >-39.5 & CommonBeardheath.cmerged$lon <= min(CommonBeardheath.cmerged$lon),]
# CommonBeardheath.cmerged = CommonBeardheath.cmerged[CommonBeardheath.cmerged$lon >= min(CommonBeardheath.cmerged$lon) & CommonBeardheath.cmerged$lat <= max(CommonBeardheath.cmerged$lat),]
# CommonBeardheath.cmerged = CommonBeardheath.cmerged[CommonBeardheath.cmerged$lat >= min(CommonBeardheath.cmerged$lat) & CommonBeardheath.cmerged$lon <= max(CommonBeardheath.cmerged$lon),]

#Cross checking

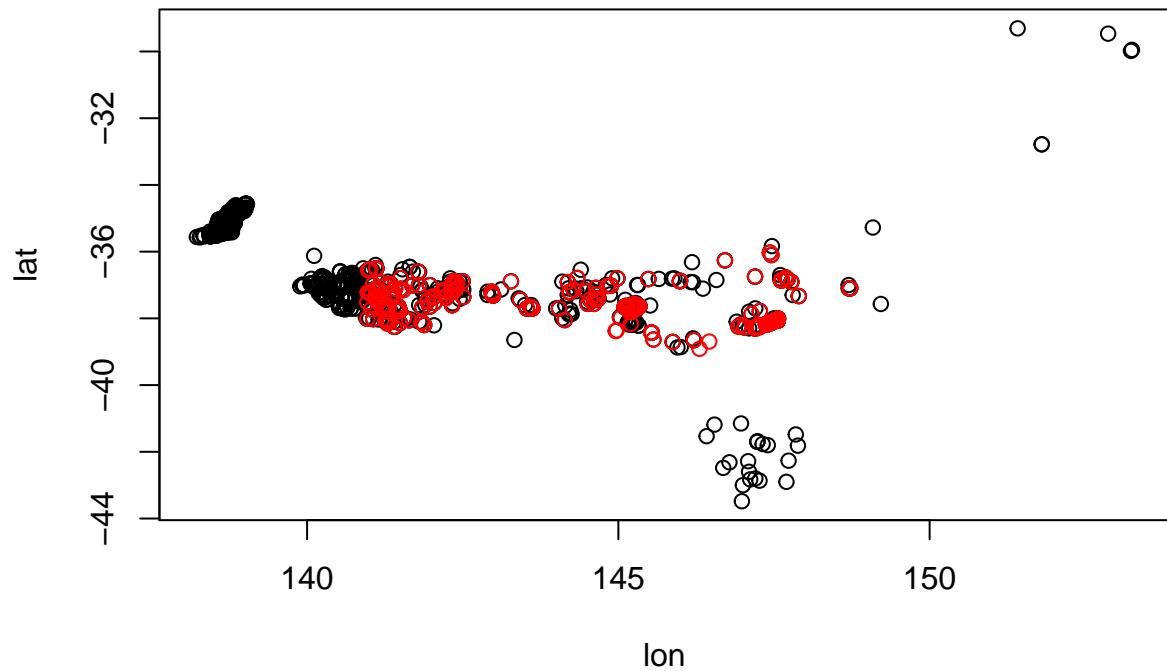
data("wrld_simpl")
CommonBeardheath.cmerged2 = CommonBeardheath.cmerged
coordinates(CommonBeardheath.cmerged2) = ~lon+lat
crs(CommonBeardheath.cmerged2) <- crs(wrld_simpl)
ovr <- over(CommonBeardheath.cmerged2, wrld_simpl)
unique(ovr$NAME)

## [1] Australia      Antarctica
## 246 Levels: Aaland Islands Afghanistan Albania Algeria ... Zimbabwe
#There are NA entries, i.e. not from any country which should be removed
i <- which(ovr$NAME == "Australia")

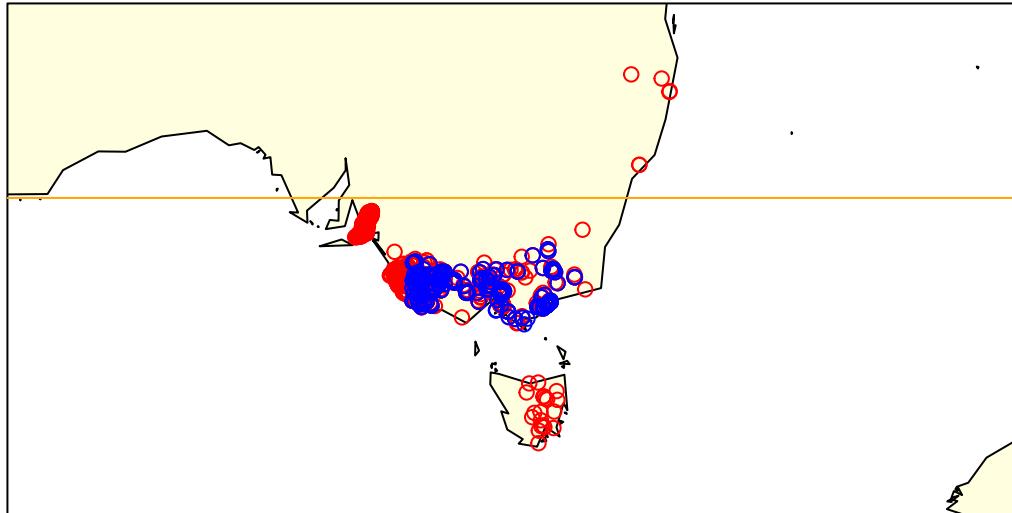
#Easier is to only select points that are from australia since this species is only in Australia
CommonBeardheath.cmerged = CommonBeardheath.cmerged[c(i),]

#Let's see how the coordinates look now.. red = only points from DELWP. black = points from both DELWP and
plot(CommonBeardheath.cmerged)
points(CommonBeardheath.cmerged, col = "red")

```



```
#Final inspection of all observation points
#RED POINTS = additional observations from gbif, Blue points = observations in DELWP dataset
plot(wrld_simpl, ylim = c(min(CommonBeardheath.cmerged$lat - 2), max(CommonBeardheath.cmerged$lat) + 2)
points(CommonBeardheath.cmerged, col = "red")
points(CommonBeardheath.c, col = "blue")
abline(h = (sort(CommonBeardheath.cmerged$lat, TRUE)[7]) + 0.5, col = 'orange')
```



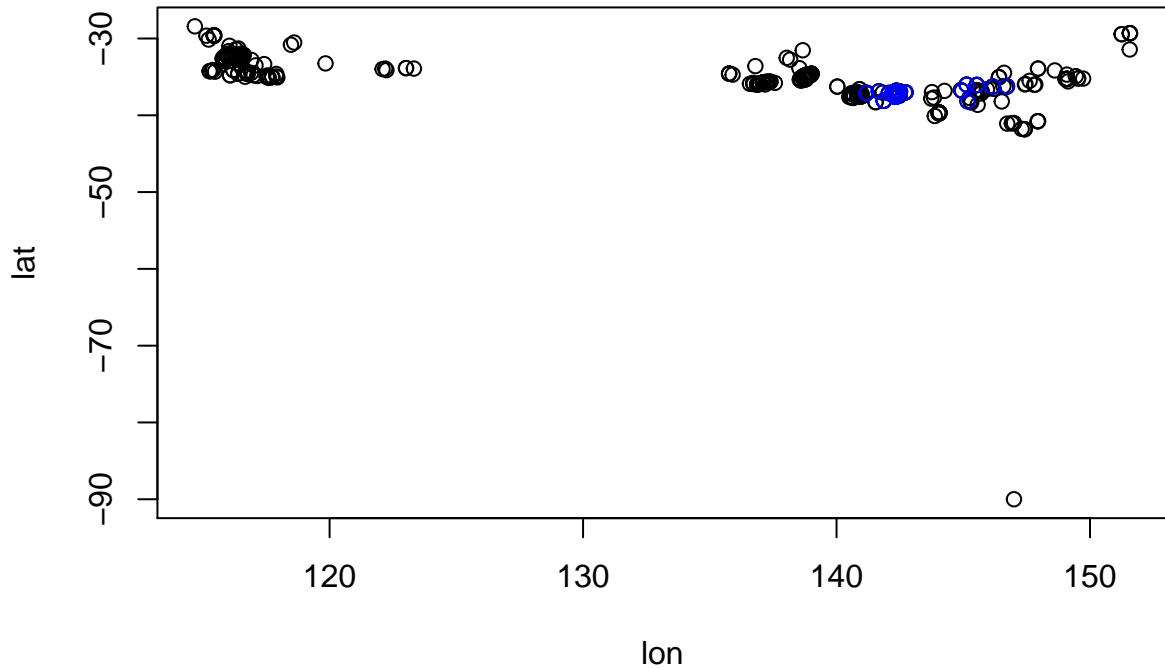
```

#Save the merged, preprocessed coordinates as a CSV file, only needs to be run once
# write.csv(CommonBeardheath.cmerged, "CommonBeardheathC.csv")

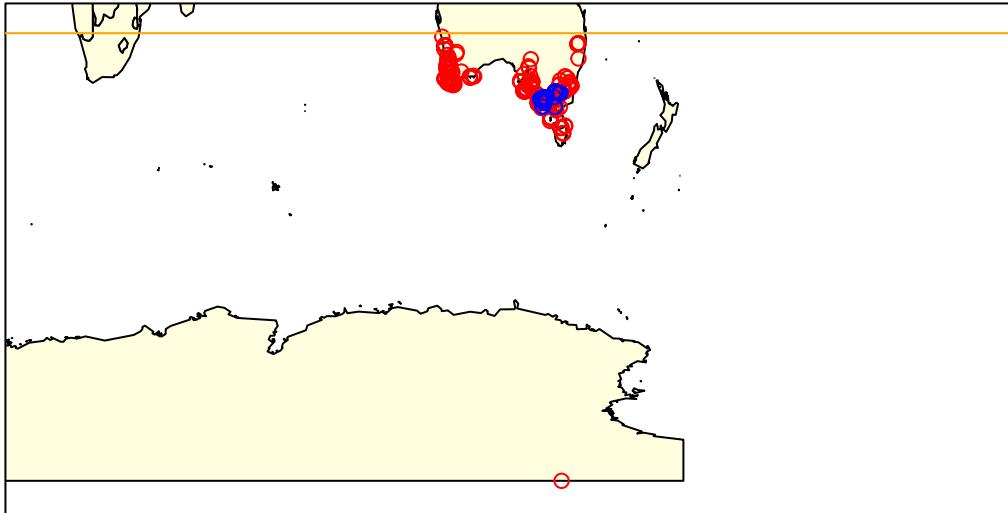
SmallTriggerplant.cgbif = subset(SmallTriggerplant.gbif, !is.na(lon) & !is.na(lat)) #Select non-empty locations
SmallTriggerplant.cgbif = SmallTriggerplant.cgbif[,c("lon","lat")] #Get latitude and longitude values
colnames(SmallTriggerplant.c) = c("lon","lat")
SmallTriggerplant.cmerged = rbind(SmallTriggerplant.c,SmallTriggerplant.cgbif) #Merge GBIF values with our own
SmallTriggerplant.cmerged = distinct(SmallTriggerplant.cmerged) #Remove duplicated matching lat/lon values

plot(SmallTriggerplant.cmerged)
points(SmallTriggerplant.c, col = 'blue')

```



```
#See overlap of merged points and points given just by DELWP
#RED POINTS = additional observations from gbif, Blue points = observations in DELWP dataset
plot(wrld_simpl, ylim = c(min(SmallTriggerplant.cmerged$lat) - 2), max(SmallTriggerplant.cmerged$lat) +
points(SmallTriggerplant.cmerged, col = "red")
points(SmallTriggerplant.c, col = "blue")
abline(h = (sort(SmallTriggerplant.cmerged$lat, TRUE)[1]) + 0.5, col = 'orange')
```



```
#Keep all GBIF ones otherwise we lack data, <100 observations

# SmallTriggerplant.cmerged = SmallTriggerplant.cmerged[SmallTriggerplant.cmerged$lat >-45 & SmallTriggerplant.cmerged$lon >= min(SmallTriggerplant.cmerged$lon) & SmallTriggerplant.cmerged$lat <= max(SmallTriggerplant.cmerged$lat)]
# SmallTriggerplant.cmerged = SmallTriggerplant.cmerged[SmallTriggerplant.cmerged$lat >-40 & SmallTriggerplant.cmerged$lon >= min(SmallTriggerplant.cmerged$lon) & SmallTriggerplant.cmerged$lat <= max(SmallTriggerplant.cmerged$lat)]
# SmallTriggerplant.cmerged = SmallTriggerplant.cmerged[SmallTriggerplant.cmerged$lat <-36 & SmallTriggerplant.cmerged$lon >= min(SmallTriggerplant.cmerged$lon) & SmallTriggerplant.cmerged$lat <= max(SmallTriggerplant.cmerged$lat)]

# SmallTriggerplant.cmerged = SmallTriggerplant.cmerged[SmallTriggerplant.cmerged$lon >= min(SmallTriggerplant.cmerged$lon) & SmallTriggerplant.cmerged$lat >= min(SmallTriggerplant.cmerged$lat) & SmallTriggerplant.cmerged$lat <= max(SmallTriggerplant.cmerged$lat) & SmallTriggerplant.cmerged$lon <= max(SmallTriggerplant.cmerged$lon)]
# SmallTriggerplant.cmerged = SmallTriggerplant.cmerged[SmallTriggerplant.cmerged$lat >= min(SmallTriggerplant.cmerged$lat) & SmallTriggerplant.cmerged$lon <= max(SmallTriggerplant.cmerged$lon) & SmallTriggerplant.cmerged$lat <= max(SmallTriggerplant.cmerged$lat) & SmallTriggerplant.cmerged$lon >= min(SmallTriggerplant.cmerged$lon)]


data("wrld_simpl")
SmallTriggerplant.cmerged2 = SmallTriggerplant.cmerged
coordinates(SmallTriggerplant.cmerged2) = ~lon+lat
crs(SmallTriggerplant.cmerged2) <- crs(wrld_simpl)
ovr <- over(SmallTriggerplant.cmerged2, wrld_simpl)
unique(ovr$NAME)

## [1] Australia      Antarctica
## 246 Levels: Aaland Islands Afghanistan Albania Algeria ... Zimbabwe
#There are NA entries, i.e. not from any country which should be removed
ovr[is.na(ovr$NAME),]

##      FIPS ISO2 ISO3 UN NAME AREA POP2005 REGION SUBREGION LON LAT
## 51    <NA> <NA> <NA> NA <NA>   NA     NA     NA   NA   NA
## 63    <NA> <NA> <NA> NA <NA>   NA     NA     NA   NA   NA
## 64    <NA> <NA> <NA> NA <NA>   NA     NA     NA   NA   NA
## 65    <NA> <NA> <NA> NA <NA>   NA     NA     NA   NA   NA
```

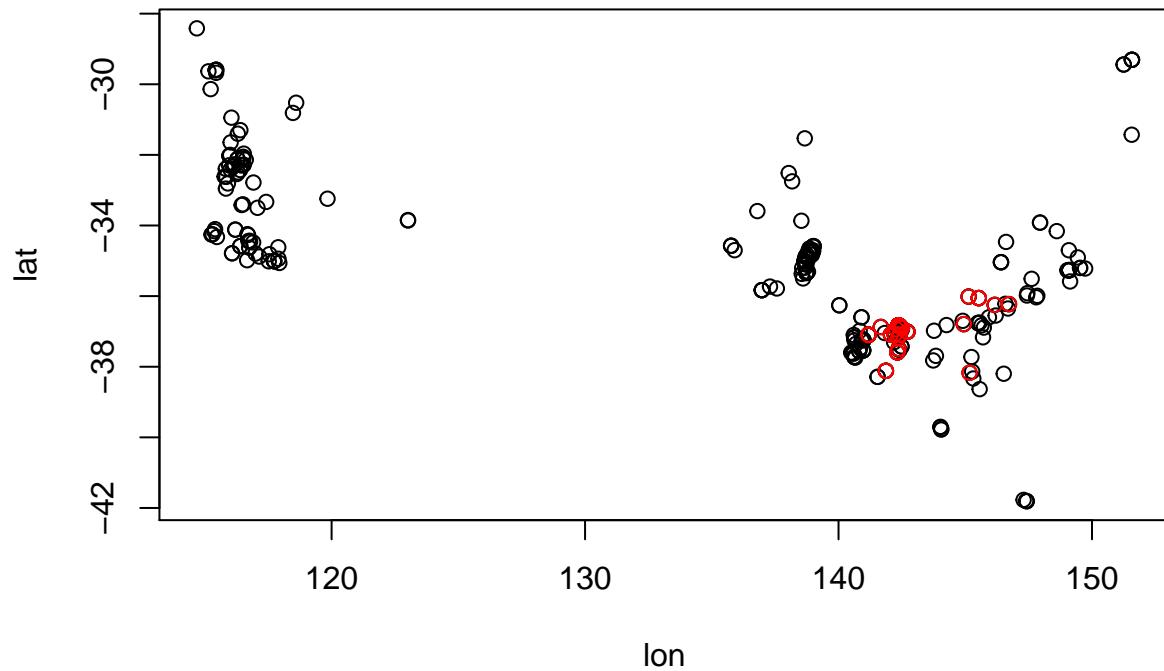
```

## 66 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 82 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 84 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 85 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 89 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 105 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 107 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 110 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 111 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 112 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 121 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 135 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 152 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 156 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 196 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 197 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 200 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 250 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 257 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 268 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 269 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 270 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 283 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 304 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 309 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 310 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 311 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 315 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 319 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 320 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 321 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 322 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 323 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 326 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 339 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 340 <NA> <NA> <NA> NA <NA> NA NA NA NA NA
## 343 <NA> <NA> <NA> NA <NA> NA NA NA NA NA

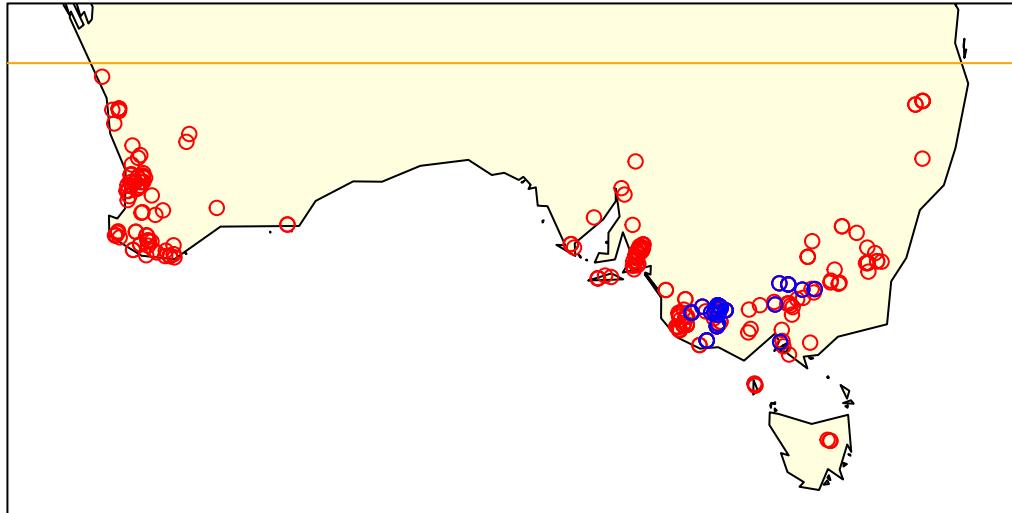
#Easier is to only select points that are from australia since this species is only in Australia
i <- which(ovr$NAME == "Australia")
SmallTriggerplant.cmerged = SmallTriggerplant.cmerged[c(i),]

#Let's see how the coordinates look now.. red = only points from DELWP. black = points from both DELWP
plot(SmallTriggerplant.cmerged)
points(SmallTriggerplant.c, col = "red")

```



```
#Final inspection of all observation points
#RED POINTS = additional observations from gbif, Blue points = observations in DELWP dataset
plot(wrld_simpl, ylim = c(min(SmallTriggerplant.cmerged$lat - 2), max(SmallTriggerplant.cmerged$lat) + 2))
points(SmallTriggerplant.cmerged, col = "red")
points(SmallTriggerplant.c, col = "blue")
abline(h = (sort(SmallTriggerplant.cmerged$lat, TRUE)[1]) + 0.5, col = 'orange')
```

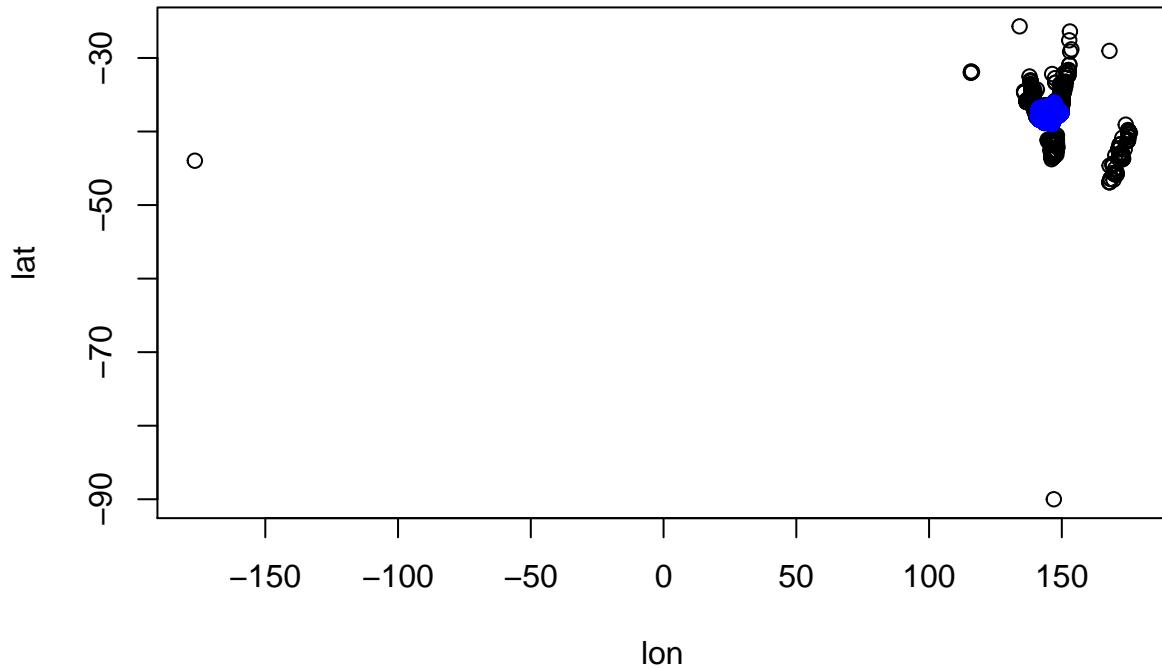


```
#Save the merged, preprocessed coordinates as a CSV file - only needs to be run once
# write.csv(SmallTriggerplant.cmerged, "SmallTriggerplantC.csv")

SouthernBrownTreeFrog

SouthernBrownTreeFrog.cgbif = subset(SouthernBrownTreeFrog.gbif, !is.na(lon) & !is.na(lat)) #Select non
SouthernBrownTreeFrog.cgbif = SouthernBrownTreeFrog.cgbif[,c("lon","lat")] #Get latitude and longitude
colnames(SouthernBrownTreeFrog.c) = c("lon","lat")
SouthernBrownTreeFrog.cmerged = rbind(SouthernBrownTreeFrog.c,SouthernBrownTreeFrog.cgbif) #Merge GBIF +
SouthernBrownTreeFrog.cmerged = distinct(SouthernBrownTreeFrog.cmerged) #Remove duplicated matching lat

plot(SouthernBrownTreeFrog.cmerged)
points(SouthernBrownTreeFrog.c, col = 'blue')
```

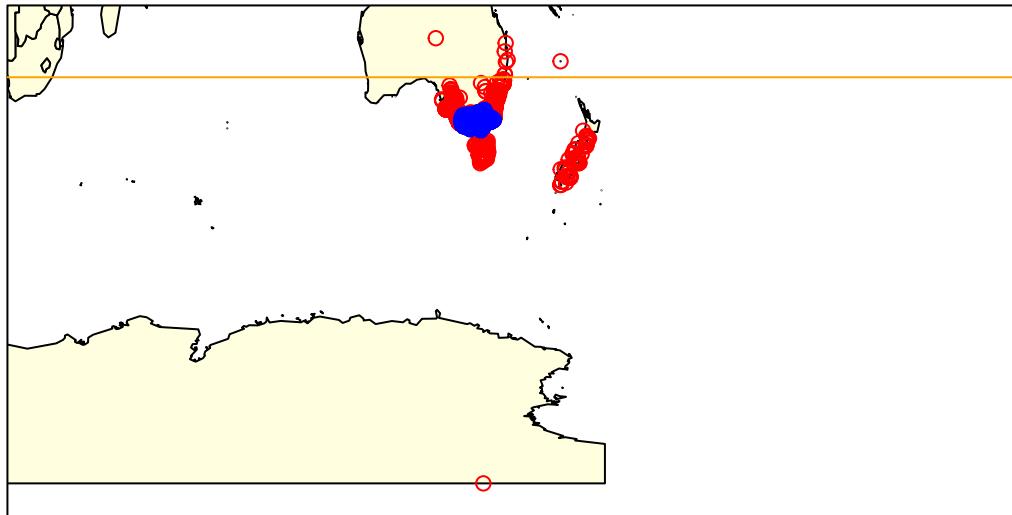


```

#Remove two clear outliers
SouthernBrownTreeFrog.cmerged = SouthernBrownTreeFrog.cmerged[SouthernBrownTreeFrog.cmerged$lon > 130,]

#See overlap of merged points and points given just by DELWP
#RED POINTS = additional observations from gbif, Blue points = observations in DELWP dataset
plot(wrld_simpl, ylim = c(min(SouthernBrownTreeFrog.cmerged$lat - 2), max(SouthernBrownTreeFrog.cmerged$lat + 2)),
points(SouthernBrownTreeFrog.cmerged, col = "red")
points(SouthernBrownTreeFrog.c, col = "blue")
abline(h = (sort(SouthernBrownTreeFrog.cmerged$lat, TRUE)[15]) + 0.5, col = 'orange')

```



```

#
# SouthernBrownTreeFrog.cmerged = SouthernBrownTreeFrog.cmerged[SouthernBrownTreeFrog.cmerged$lat >-60 &
# SouthernBrownTreeFrog.cmerged = SouthernBrownTreeFrog.cmerged[SouthernBrownTreeFrog.cmerged$lat >-60 &
#
# SouthernBrownTreeFrog.cmerged = SouthernBrownTreeFrog.cmerged[SouthernBrownTreeFrog.cmerged$lon >= min(SouthernBrownTreeFrog.cmerged$lon) &
# SouthernBrownTreeFrog.cmerged = SouthernBrownTreeFrog.cmerged[SouthernBrownTreeFrog.cmerged$lat >= min(SouthernBrownTreeFrog.cmerged$lat) &
# SouthernBrownTreeFrog.cmerged = SouthernBrownTreeFrog.cmerged[SouthernBrownTreeFrog.cmerged$lon <= max(SouthernBrownTreeFrog.cmerged$lon) &
# SouthernBrownTreeFrog.cmerged = SouthernBrownTreeFrog.cmerged[SouthernBrownTreeFrog.cmerged$lat <= max(SouthernBrownTreeFrog.cmerged$lat) &
# SouthernBrownTreeFrog.cmerged = SouthernBrownTreeFrog.cmerged[SouthernBrownTreeFrog.cmerged$lon <= 180 &
# SouthernBrownTreeFrog.cmerged = SouthernBrownTreeFrog.cmerged[SouthernBrownTreeFrog.cmerged$lat <= 90]

data("wrld_simpl")
SouthernBrownTreeFrog.cmerged2 = SouthernBrownTreeFrog.cmerged
coordinates(SouthernBrownTreeFrog.cmerged2) = ~lon+lat
crs(SouthernBrownTreeFrog.cmerged2) <- crs(wrld_simpl)
ovr <- over(SouthernBrownTreeFrog.cmerged2, wrld_simpl)
unique(ovr$NAME)

## [1] Australia      <NA>          New Zealand    Antarctica
## [5] Norfolk Island
## 246 Levels: Aaland Islands Afghanistan Albania Algeria ... Zimbabwe
#There are NA entries, i.e. not from any country which should be removed
ovr[is.na(ovr$NAME),]

##      FIPS ISO2 ISO3 UN NAME AREA POP2005 REGION SUBREGION LON LAT
## 2    <NA> <NA> <NA> NA <NA>   NA     NA     NA     NA   NA
## 3    <NA> <NA> <NA> NA <NA>   NA     NA     NA     NA   NA

```

















```

## 6999 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7003 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7024 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7135 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7136 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7154 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7174 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7266 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7267 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7275 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7276 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7280 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7282 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7286 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7296 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7297 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7301 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7302 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7328 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7340 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7341 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7344 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7346 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7347 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7351 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7352 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA
## 7359 <NA> <NA> <NA> NA <NA> NA NA NA NA NA NA

#Easier is to only select points that are from australia since this species is only in Australia
NROW(which(ovr$NAME == "Australia"))

## [1] 6838

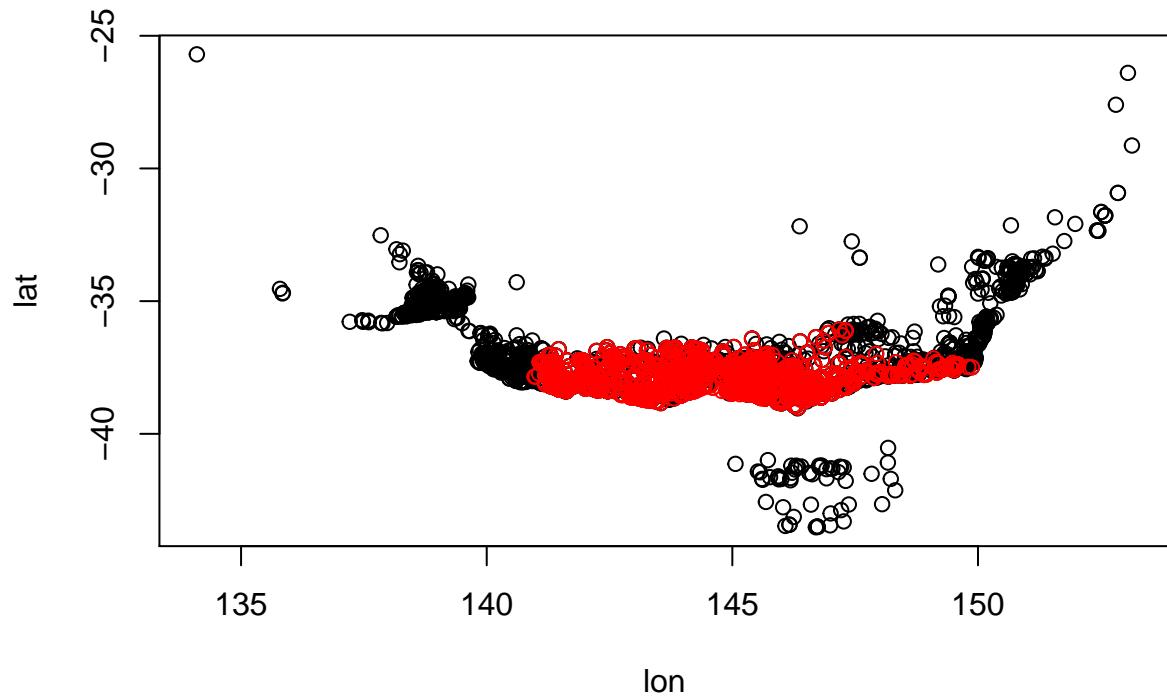
NROW(which(ovr$NAME == "New Zealand"))#Recently introduced to NZ. 58 rows vs 6k. Original DEWLPL database

## [1] 58

i <- which(ovr$NAME == "Australia")
SouthernBrownTreeFrog.cmerged = SouthernBrownTreeFrog.cmerged[c(i),]

#Let's see how the coordinates look now.. red = only points from DELWP. black = points from both DELWP
plot(SouthernBrownTreeFrog.cmerged)
points(SouthernBrownTreeFrog.c, col = "red")

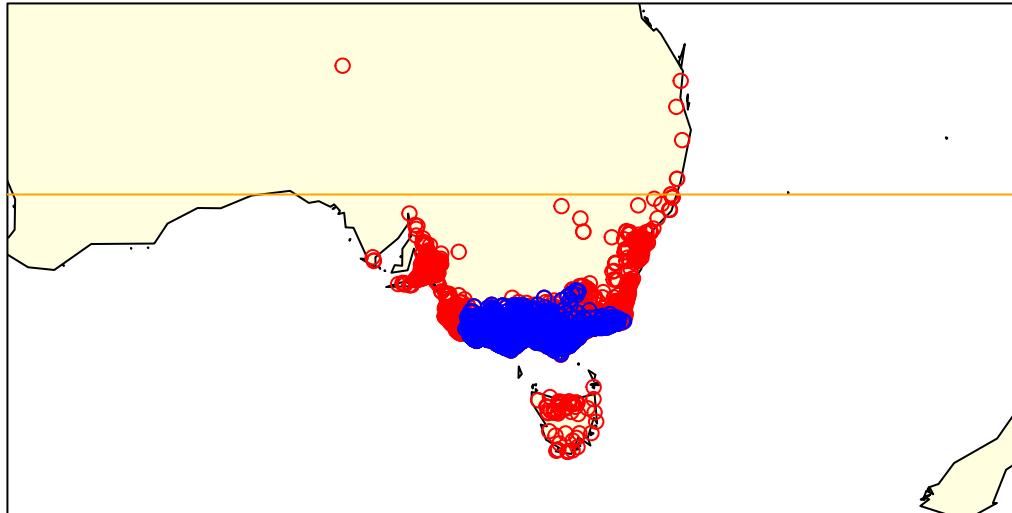
```



```

#Final inspection of all observation points
#RED POINTS = additional observations from gbif, Blue points = observations in DELWP dataset
plot(wrld_simpl, ylim = c(min(SouthernBrownTreeFrog.cmerged$lat - 2), max(SouthernBrownTreeFrog.cmerged$lat + 2)),
points(SouthernBrownTreeFrog.cmerged, col = "red")
points(SouthernBrownTreeFrog.c, col = "blue")
abline(h = (sort(SouthernBrownTreeFrog.cmerged$lat, TRUE)[15]) + 0.5, col = 'orange')

```

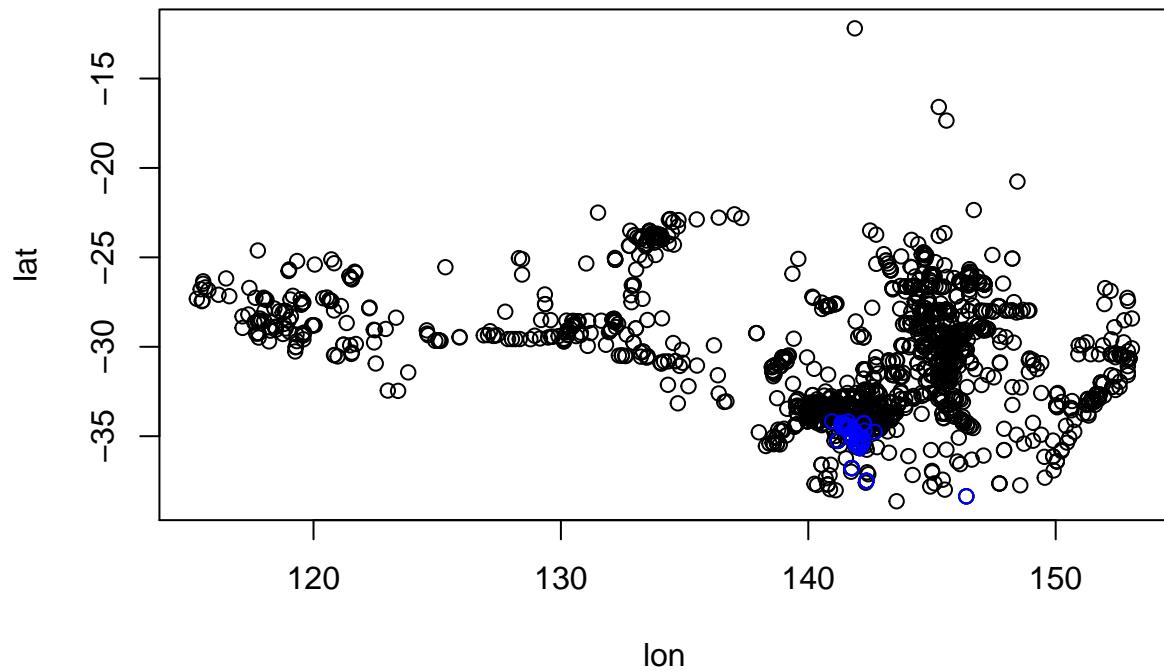


```
#Save the merged, preprocessed coordinates as a CSV file, only needs to be run once
# write.csv(SouthernBrownTreeFrog.cmerged, "SouthernBrownTreeFrogC.csv")
```

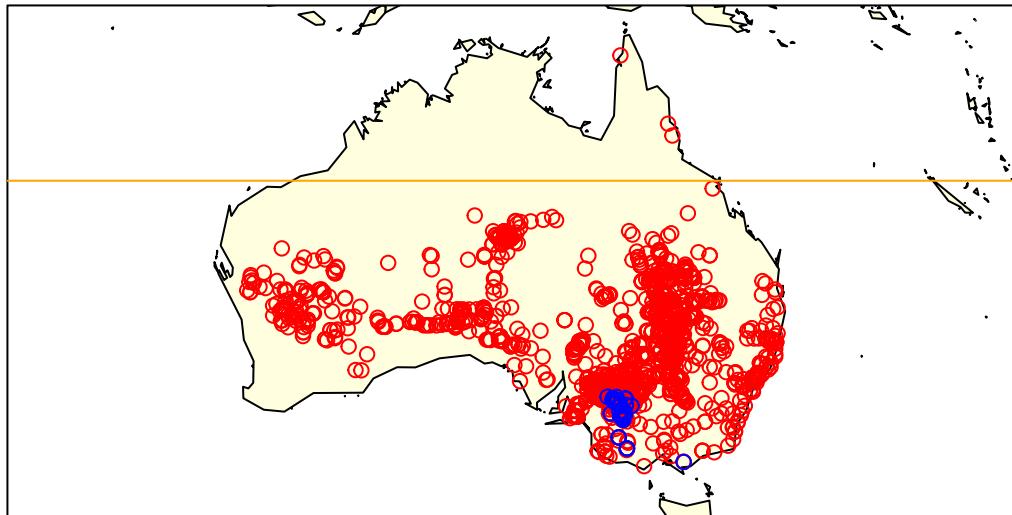
WhitebrowedTreecreeper

```
WhitebrowedTreecreeper.cgbif = subset(WhitebrowedTreecreeper.gbif, !is.na(lon) & !is.na(lat)) #Select non NA rows
WhitebrowedTreecreeper.cgbif = WhitebrowedTreecreeper.cgbif[,c("lon","lat")] #Get latitude and longitude
colnames(WhitebrowedTreecreeper.c) = c("lon","lat")
WhitebrowedTreecreeper.cmerged = rbind(WhitebrowedTreecreeper.c,WhitebrowedTreecreeper.cgbif) #Merge GBIF and Gbif
WhitebrowedTreecreeper.cmerged = distinct(WhitebrowedTreecreeper.cmerged) #Remove duplicated matching locations

plot(WhitebrowedTreecreeper.cmerged)
points(WhitebrowedTreecreeper.c, col = 'blue')
```



```
#See overlap of merged points and points given just by DELWP
#RED POINTS = additional observations from gbif, Blue points = observations in DELWP dataset
plot(wrld_simpl, ylim = c(min(WhitebrowedTreecreeper.cmerged$lat - 2), max(WhitebrowedTreecreeper.cmerged$lat + 2)), col = "red")
points(WhitebrowedTreecreeper.c, col = "blue")
abline(h = (sort(WhitebrowedTreecreeper.cmerged$lat, TRUE)[4]) + 0.5, col = 'orange')
```



```

# WhitebrowedTreecreeper.cmerged = WhitebrowedTreecreeper.cmerged[WhitebrowedTreecreeper.cmerged$lat < -35]
#
# WhitebrowedTreecreeper.cmerged = WhitebrowedTreecreeper.cmerged[WhitebrowedTreecreeper.cmerged$lon >= 115]
# WhitebrowedTreecreeper.cmerged = WhitebrowedTreecreeper.cmerged[WhitebrowedTreecreeper.cmerged$lat >= -25]

data("wrld_simpl")
WhitebrowedTreecreeper.cmerged2 = WhitebrowedTreecreeper.cmerged
coordinates(WhitebrowedTreecreeper.cmerged2) = ~lon+lat
crs(WhitebrowedTreecreeper.cmerged2) <- crs(wrld_simpl)
ovr <- over(WhitebrowedTreecreeper.cmerged2, wrld_simpl)
unique(ovr$NAME)

## [1] Australia <NA>
## 246 Levels: Aaland Islands Afghanistan Albania Algeria ... Zimbabwe
#There are NA entries, i.e. not from any country which should be removed
ovr[is.na(ovr$NAME),]

##          FIPS ISO2 ISO3 UN NAME AREA POP2005 REGION SUBREGION LON LAT
## 213 <NA> <NA> <NA> NA <NA>   NA     NA     NA      NA  NA  NA
## 1913 <NA> <NA> <NA> NA <NA>   NA     NA     NA      NA  NA  NA
## 2062 <NA> <NA> <NA> NA <NA>   NA     NA     NA      NA  NA  NA
## 2076 <NA> <NA> <NA> NA <NA>   NA     NA     NA      NA  NA  NA
## 2239 <NA> <NA> <NA> NA <NA>   NA     NA     NA      NA  NA  NA
## 2351 <NA> <NA> <NA> NA <NA>   NA     NA     NA      NA  NA  NA

#Easier is to only select points that are from australia since this species is only in Australia
i <- which(ovr$NAME == "Australia")

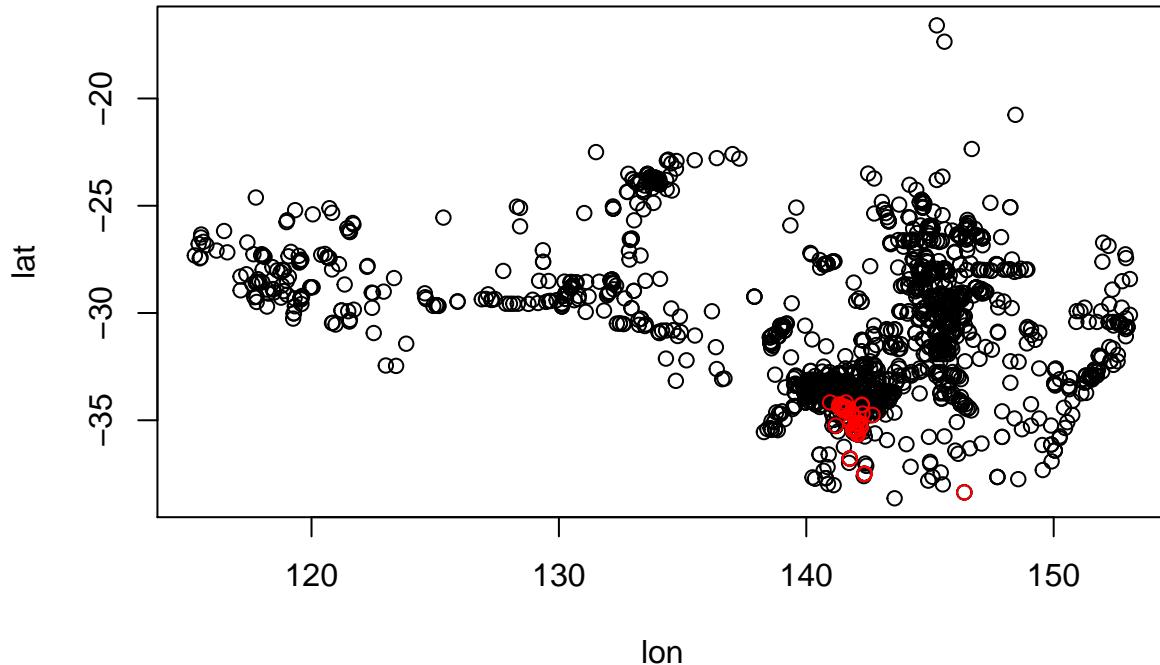
```

```

WhitebrowedTreecreeper.cmerged = WhitebrowedTreecreeper.cmerged[c(i),]

#Let's see how the coordinates look now.. red = only points from DELWP. black = points from both DELWP ...
plot(WhitebrowedTreecreeper.cmerged)
points(WhitebrowedTreecreeper.c, col = "red")

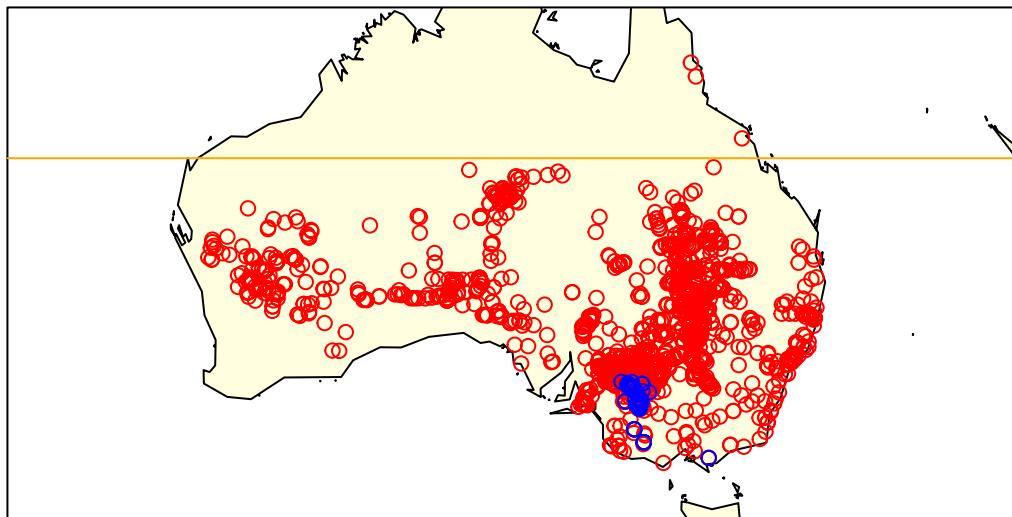
```



```

#Final inspection of all observation points
#RED POINTS = additional observations from gbif, Blue points = observations in DELWP dataset
plot(wrld_simpl, ylim = c(min(WhitebrowedTreecreeper.cmerged$lat - 2), max(WhitebrowedTreecreeper.cmerged$lat + 2)))
points(WhitebrowedTreecreeper.cmerged, col = "red")
points(WhitebrowedTreecreeper.c, col = "blue")
abline(h = (sort(WhitebrowedTreecreeper.cmerged$lat, TRUE)[4]) + 0.5, col = 'orange')

```



```
#Save the merged, preprocessed coordinates as a CSV file. Only needs to be run once  
# write.csv(WhitebrowedTreecreeper.cmerged, "WhitebrowedTreecreeperC.csv")  
  
#Save workspace. So code won't need to be run again in the future.  
# save.session(file='Preprocessing.Rda')
```