Exercise 1: Obtaining and Cleaning Biodiversity Data

Adam B. Smith, Danielle Christianson, & Camilo Sanín

July 6, 2018

In this first exercise you will learn:

1. **Where** to obtain biodiversity data;
2. What aspects of data are **relevant to distribution modeling**;
3. How to **download** data;
4. How to **merge** data from different sources;
5. How to **inspect** biodiversity data;
6. How to **clean** or flag biodiversity data for:
   * Nomenclatural error and **taxonomic uncertainty**;
   * **Error in geographic coordinates**;
   * **Errors in dates**;
   * And lots of other things that arise when using data from many different sources collected in a non-standardized manner.

Biodiversity data encompasses anything that represents biological diversity, but here we will specifically explore methods for specimen-based "point" records which represent particular places on a map where an organism was collected or observed. Point records are the fundamental type of data used in distribution and niche modeling.

We will be modeling the Columbian ground squirrel, *Urocitellus columbianus*.



Urocitellus columbianus (Wikimedia)

**Working folder**: There are *many* steps in modeling, so getting the workflow right is essential to creating **reproducible research**. To begin please create a folder somewhere on your computer system. Copy all of the files and folders for the workshop into this folder. Hereafter we'll refer to this as your "working" folder. *All* of the input and output files will be saved in this folder, including files you download.

# Taxonomy and species' names

Many species have multiple names because different taxonomists worked on the species. Usually one of these is the "accepted" name and the others are "synonyms". As a result, species are often represented in databases under the name in the taxonomic system used by the particular collector at the time specimens were collected. Thus, going to a database and searching for just one name may give you a subset of the records that are really available for that species.

First let's see how synonymy might affect your selected species. We'll use [Encyclopedia of Life](http://eol.org/) as our taxonomic authority. Lately there seems to be a battle for authoritativeness, with different taxa (plants, ants, mammals etc.) having one or more "authorities" to which one can go. Regardless of which authority you use, ensure that you search for alternative names for your species.

For taxonomically authoritative sources see <www.earthskysea.org/biodiversity>. There we've provided an incomplete list of biodiversity data sources, including **Taxonomic Name Resolution Services (TNRS)**, which can assist in searching for species' alternative names.

1. Go to the Encyclopedia of Life at <www.eol.org>.
2. Enter "*Urocitellus columbianus*" in the search box.
3. You will see that it directs you to the page for *Spermophilis columbianus*. This is direct evidence of taxonomic confusion!
4. Now click the "Names" tab then the "3 synonyms" tab along the left side. Notice that two out of the three taxonomic name resolution systems prefer *Urocitellus columbianus*. Write down the three names given to this species (just the binomials--not the authors). It will be important to look under each of these names when searching (some databases do this automatically, though).

# Obtaining raw biodiversity data

In your working folder please create a subfolder entitled Species Records. We'll be saving the output from each of four data repositories in this folder.

Unfortunately there is no single data portal for all biodiversity data... in fact, there are *several* attempts to create "one-stop shops" for this kind of data. So where should you go? There are two strategies, both with downsides. You could go to just a single database, preferably the one that best covers the taxon/region of interest. However, many databases are incomplete, so you might end up missing a lot of records by relying on just one. But many databases share data, too, so you risk including duplicate records by using more than one!

There are scores of sites where you can download some data relevant to distribution modeling. You can find an abbreviated, haphazardly amassed list at <www.earthskysea.org/biodiversity>. There we've organized portals by taxonomic identity--note that general portals that offer data for any taxon are listed last and are almost always worth visiting. Also note that many millions of specimens have yet to be digitized or at least submitted to these large data clearinghouses. So if you're interested in a particular species and want to get as comprehensive as coverage as possible, it is likely worth your time to contact state and regional source (herbaria, museums, etc.). Small institutions often have spreadsheets of their holdings with data that they are willing to share if you contact them, but these data are rarely incorporated into larger portals, although there is increasing effort to do so.

We are going to use two databases: **GBIF** and **VertNet**. Normally I'd also check **iDigBio** (a new initiative to digitize hithertofore overlooked biodiversity collections), **Canadensys** (a Canadian portal--because the species occurs partly in Canada), and **BISON** (an American portal) but we'll just use **GBIF** and **VertNet** to illustrate the process of obtaining and cleaning data.

### GBIF (Global Biodiversity Information Facility)

**URL**: <http://www.gbif.org/>

**Description**: The purported "master" database that attempts to aggregate all of the world's biodiversity data.

**Answer to that question bothering you**: If GBIF aggregates all biodiversity data, why not just go to GBIF instead of checking multiple sources? Politely said, much of the data in GBIF has quality problems so a lot of it is less useful in the end. Likewise, many of the databases contributing data on to GBIF (called GBIF "nodes") don't do so in real time, so there may be records in the country-level nodes that aren't in GBIF. Also, some nodes use different fields which are very useful for cleaning this kind of data and yet absent from GBIF.

**Useful fact**: You can use the package [rgbif](https://cran.r-project.org/web/packages/rgbif) to download data directly into R. We'll do it by hand, though, because it illustrates several important issues.

**Instructions**:

1. In your Species Records folder create a subfolder named GBIF. We'll be creating a different directory for each data source. This is good practice because data sets are often downloaded as multiple files and it's better to keep them separate from one another. It also enables you to easily see from where you've gotten data.
2. Go to [www.gbif.org/](http://www.gbif.org/). Save a link to this website in the GBIF folder. Again, this is good practice. **Take home**: Ensure you can relocate the source of each data set you use in case you need to get an updated version or cite the source. For databases that get updated constantly (e.g., biodiversity databases), I often record the date I downloaded the data so I know what "version" I got.
3. At this point it will help to create an account (link on the upper right). You will not receive spam from them.
4. Click "Data" (top right) --> "Explore species". Search for "*Urocitellus columbianus*". Click the link of the same name.
5. On this page you'll see a list of synonyms that GBIF also checked for your species. Note that "*Spermophilus columbiansus*" was automatically checked and included in your search. However, also note that "*Citellus columbianus*" was not checked. Normally you would do two searches, one for each name, but we'll just tell you now that searching for "*Citellus columbianus*" returns no results, so we'll skip it. Note that GBIF checks for synonyms using the "GBIF Backbone Taxonomy," which may or may not include all of the taxonomic synonyms for your species.

**Take-home**: Check with taxonomic authorities like specialists in the taxon you're interested in or online at places like [EOL](www.eol.org) to ensure you're using all of the potential names for species (see www.earthskysea.org/biodiversity for some sources). Ensure the data provider either includes synonymy in the searches or do separate searches for each synonym.

1. Click "View occurrences" (top right) then "Download" (top right). Select "Darwin Core Archive".

**Take-home**: [Darwin Core](http://rs.tdwg.org/dwc/) (also see [Wieczorek et al. 2012](http://dx.doi.org/10.1371/journal.pone.0029715) is a set of biodiversity data standards that are used to enable sharing of biodiversity databases. It provides a set list of agreed-upon fields each with acceptable data formats. Note all Darwin Core database will have all of these fields, and they might also include non-Core fields.

1. When you get the email notice, save your results to the GBIF folder inside the Species Records folder.

**Take-home**: Notice the download page on GBIF has a "Cite As" link. This enables you or someone else to download the exact same data again. This is important for reproducibility because these databases are constantly being updated.

Cheating: it's OK in this case. If the download is taking too long to get, you can just copy the file from the .../Backup/Species Records/GBIF directory into your GBIF directory. We've included files like this in the Backup directory in case there's a problem with Internet connectivity.

1. Unzip the files if necessary. The one named occurrences has the data we're seeking. The others contain metadata and information on usage rights.
2. Open the file in Excel.

Please note: GBIF data files can be really finicky… If you have problems in subsequent steps you may have to do some manual processing to get R to load it properly. If this is the case, try the steps at the GBIF FAQ on [opening GBIF CSV files](https://www.gbif.org/faq?question=opening-gbif-csv-in-excel). When you are done with these steps, save it as a CSV file (ironic, I know).

### VertNet

**URL**: <http://vertnet.org/>

**Description**: VertNet aggregates data on vertebrates from multiple institutions. For these kinds of databases, we've found it often has the highest-quality data.

**Fun fact**: VertNet subsumed four other vertebrate-focused portals: HerpNet (reptiles and amphibians), ORNIS (birds), FishNet (fishes), and MANIS (mammals).

**Useful fact**: You can also use the package [rvertnet](https://cran.r-project.org/web/packages/rvertnet) to download data directly into R. Again, we'll do it by hand, though.

**Instructions**:

1. Go to <http://vertnet.org/> then search for "\*Urocitellus columbianus" (lower left part of page).
2. Look over the results then click "Download" (right side of page). Fill in the form, and download!
3. In the Species Records folder create a subfolder named VertNet and save the data there. Unzip the file if necessary. This is either a text ('txt) file or a tab-separated value (.tsv) file. You can open either in Excel.

## Reflection

1. What issues did you find when you searched each portal for data?
2. How did synonymy affect your searches?
3. What irregularities did you find in the data you downloaded? Did you correct them, and how?
4. What other data sources could you have consulted?
5. Species' names in databases (esp. GBIF) are often misspelled (e.g., "*Urocitellus columbiansu*"). How could you search to capture these misspelled records?

# Merging the data from different sources

There are a just a few relevant fields necessary for distribution modeling. However, many of the other fields contain information relevant to the quality of the record. When possible, we try to include as much of the "extraneous" data as I can in the combined data set. This really helps later when we spot an erroneous record and need to check its validity--we'll be doing this! It's much easier to open the combined database in Excel and look at the fields then try to locate the offending record in the original source file, especially if you used many different sources.

This said, for the purposes of this tutorial we'll just use a few relevant fields.

In R set your working directory. Note that we're showing you *our* working folder's directory, but you will have to change this to the working directory you created.

# for Windows  
setwd('C:/SDM Workshop')  
  
# for a Mac  
setwd('/Users/userName/SDM Workshop')

Load each database:

# You may have to change some of the file names to match yours!  
gbif <- read.csv('./Species Records/GBIF/occurrence.txt',  
 as.is=TRUE, sep='\t')  
vertnet <- read.csv('./Species Records/VertNet/MyResults.txt',  
 as.is=TRUE, sep='\t')

*Special note*: Sometimes the file from VertNet doesn't open as it should in R. Here's a trick: open it in Excel (start Exel first then load the document--don't just click the document to open it). Then save the document with the exact same name. R should behave better now. You should find ~1000 records in the vertnet data frame (use nrow(vertnet)).

Across the data sets, combine each set of columns needed for modeling. These columns pertain to:

* Data portal
* A unique identifier for the specimen for that database--helps find it in the original if needed
* Species name as given in database
* Longitude
* Latitude
* Coordinate uncertainty (we'll ignore coordinate precision, though it could also be used)
* Record type (specimen, observation, etc.)
* Country of collection
* State/province of collection
* County/parish/district of collection
* Locality -- useful in case you need to double-check the coordinates
* Date of collection (often called "eventDate" in data bases)
* Year of collection
* Institution housing the specimen
* Person who identified the specimen

We'll use most of these to either model the species and/or check coordinates for gross errors. The code below "stacks" each column from each database, GBIF first followed by VertNet. In some cases some data sets don't include a particular field, so we'll use NA's.

records <- data.frame(  
 dataSet=c(rep('GBIF', nrow(gbif)), rep('VertNet', nrow(vertnet))),  
 idNum=c(gbif$gbifID, vertnet$recordnumber),  
 rawSpecies=c(gbif$scientificName, vertnet$scientificname),  
 longitude=c(gbif$decimalLongitude, vertnet$decimallongitude),  
 latitude=c(gbif$decimalLatitude, vertnet$decimallatitude),  
 coordUncer=c(gbif$coordinateUncertaintyInMeters, vertnet$coordinateuncertaintyinmeters),  
 recordType=c(gbif$type, vertnet$basisofrecord),  
 country=c(gbif$countryCode, vertnet$country),  
 state=c(gbif$stateProvince, vertnet$stateprovince),  
 county=c(gbif$county, vertnet$county),  
 locality=c(gbif$locality, vertnet$locality),  
 date=c(gbif$eventDate, vertnet$eventdate),  
 year=c(gbif$year, vertnet$year),  
 institution=c(gbif$institutionCode, vertnet$institutioncode),  
 identifiedBy=c(gbif$identifiedBy, vertnet$identifiedby)  
)  
  
head(records) # look at first 6 lines

## dataSet idNum rawSpecies longitude  
## 1 GBIF 1571177692 Urocitellus columbianus (Ord, 1815) -113.8693  
## 2 GBIF 1571173508 Urocitellus columbianus (Ord, 1815) -114.1324  
## 3 GBIF 1571153940 Urocitellus columbianus (Ord, 1815) -119.2962  
## 4 GBIF 1571149711 Urocitellus columbianus (Ord, 1815) -113.6685  
## 5 GBIF 1571149565 Urocitellus columbianus (Ord, 1815) -113.6405  
## 6 GBIF 1571130535 Urocitellus columbianus (Ord, 1815) -117.1286  
## latitude coordUncer recordType country state county locality  
## 1 49.13227 31 CA   
## 2 49.34467 522 CA   
## 3 49.50982 7 CA   
## 4 48.79202 974 US   
## 5 47.43769 1678 US   
## 6 45.28111 NA US   
## date year institution identifiedBy  
## 1 2017-06-03T10:30Z 2017 iNaturalist   
## 2 2017-06-22T19:58Z 2017 iNaturalist   
## 3 2017-06-18T15:00Z 2017 iNaturalist   
## 4 2016-07-23T14:45Z 2016 iNaturalist   
## 5 2016-04-10T15:51Z 2016 iNaturalist   
## 6 2017-06-12T18:20Z 2017 iNaturalist

tail(records) # look at last 6 lines

## dataSet idNum rawSpecies longitude latitude  
## 3995 VertNet Urocitellus columbianus columbianus NA NA  
## 3996 VertNet Urocitellus columbianus columbianus NA NA  
## 3997 VertNet Citellus columbianus NA NA  
## 3998 VertNet Citellus columbianus NA NA  
## 3999 VertNet Citellus columbianus NA NA  
## 4000 VertNet Citellus columbianus NA NA  
## coordUncer recordType country state county  
## 3995 NA PreservedSpecimen Canada Alberta   
## 3996 NA PreservedSpecimen USA Washington Pend Oreille County  
## 3997 NA Occurrence   
## 3998 NA Occurrence   
## 3999 NA Occurrence   
## 4000 NA Occurrence   
## locality date year institution identifiedBy  
## 3995 Skoki Lodge NA YPM   
## 3996 Monumental Mountain NA YPM   
## 3997 NA ANSP   
## 3998 NA ANSP   
## 3999 NA ANSP   
## 4000 NA ANSP

nrow(records) # how many records?

## [1] 4000

We have thousands of records! But not all of these are usable owing to various issues. The next section shows how to address these issues to produce a clean version of the data set.

Before we continue, let's save our records.

saveRDS(records, './Species Records/00 Species Records - Merged All Raw Data Sets.rds')

Notice three things:

* We prepended a version number on the file name. Why prefix the file with "00"? It can literally take months to create a reliable, cleaned data set. The process is iterative because you'll find errors that you didn't observe at first but which must be addressed before the step you're currently taking. The entire process involves many steps and issues always arise that you didn't anticipate. For example, I often find that the remarks field indicates some specimens were held in captivity, which makes their use in distribution/niche modeling dubious. How do you search for these kinds of records? Do a keyword search for "captiv", "grow", "zoo", "garden", "experiment", etc. Later, you may find another record that was also for a captive organism but wasn't caught by this keyword search, so to keep things clean you have to go back to the step where you removed captive specimens. Saving versions enables you to go back without having to lose all of your work. Later we'll make versions "01", "02", etc.
* We included a description of the cleaning stage in the file name. You can't back up effectively if you don't know what step to go back to!
* We put the version number on the front of the file name. This makes it sort in sequential order when viewed on your computer. I also used two digits--since sometimes operating systems sort "1" then "10", "11", "12", etc. then "2", "20", etc.

**Take home**: Cleaning species' data is a very iterative process. Labeling species' record files with version numbers and a description of the cleaning procedure taken in that step enables you to backtrack to a given point then start again without having to restart from scratch.

# Cleaning biodiversity data

As you can already see, biodiversity data can be messy. In this section we'll pare down the database to something that is intended to be more reliable. Specifically, we're looking to:

* Avoid false identifications
* Use only specimens collected in a period relevant to the climate data we'll be utilizing
* Have reliable locational certainty

## Removing unreliable specimens

Misidentifications are a special concern because a specimen that was identified as the species of interest, but wasn't really that species, may falsely indicate that the species of interest prefers the habitat represented by the record. Short of having a taxonomic authority actually inspect each specimen, it's usually not possible to know with certainty that identifications are accurate (and even experts can be unsure). Nonetheless, let's remove any records that may be especially subject to misidentifications.

In general there are three kinds of biodiversity records:

1. Specimens collected and identified by a professional and deposited in a museum or herbarium. These are the most reliable because they offer potential for checking the original identification.
2. Observational records by experts in the taxon. These are usually also good, but because no one else can verify an observation (unless it was recorded), they're of lesser quality. In general though, I would use them if you can verify the observer was a trained specialist and the species is easy to identify by sight.
3. Citizen science data, often collected by interested people of various skill. These can be reliable for some easily-identified species, but unreliable for rare species few people have experience seeing ([Lozier et al. 2009](http://dx.doi.org/10.1111/j.1365-2699.2009.02152.x); [Miller et al. 2013](http://dx.doi.org/10.1371/journal.pone.0065808); [Lin et al. 2015](http://dx.doi.org/10.1016/j.biocon.2014.11.012)). Again, since they are observations they are inherently irreproducible. However, citizen science databases usually include photos or recordings contributors took of the species. The link to this media is often included in data downloads, enabling you to visually inspect what the contributor thought they saw.

**Take home**: Not all records are reliably identified. Take steps to minimize this problem, especially for taxa that would be hard to identify. If possible, show an expert on the species a map of the records you obtained. They can often identify odd records that were probably misidentified or mis-geolocated.

In our case we're going to take a conservative approach and exclude observational data (including video) from the records.

Note the column named recordType. Look at the values this field takes:

unique(records$recordType)

## [1] PhysicalObject Physical Object specimen   
## [5] Sound MovingImage PreservedSpecimen Occurrence   
## 8 Levels: MovingImage Occurrence Physical Object ... specimen

Let's assume that the types "PhysicalObject", "specimen", "PRESERVED\_SPECIMEN", "PreservedSpecimen", and "Occurrence" refer to actual specimens deposited in a natural history museum. Let's keep these. But let's remove observational records, videos, and sound recordings:

# Using the "if" functions because each species won't necessarily have this type  
records <- subset(records,  
 recordType=='PhysicalObject' |  
 recordType=='specimen' |  
 recordType=='PRESERVED\_SPECIMEN' |  
 recordType=='PreservedSpecimen' |  
 recordType=='Occurrence'  
)  
nrow(records)

## [1] 3692

We've lost several hundred records by excluding observations.

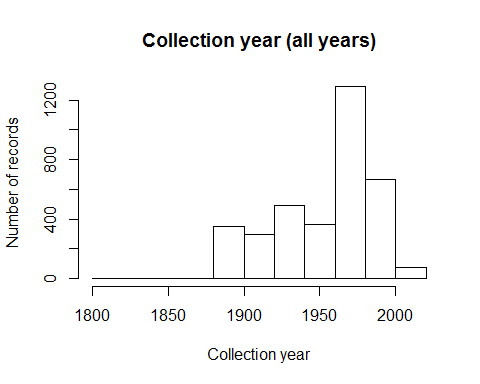
Save your results!

saveRDS(records, './Species Records/01 Species Records - Removed Observations.rds')

## Dates

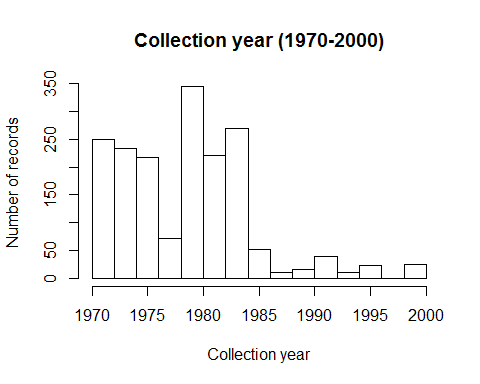
Later we will be matching these records to climate data that was calculated for the period 1970-2000. Since climate is constantly changing, we will generally want to keep records that have only been collected across this time period. First, though let's look at the distribution of years in which our specimens were collected.

hist(records$year,  
 xlab='Collection year',  
 ylab='Number of records',  
 main='Collection year (all years)'  
)



Sometimes peaks and troughs in collection activity correspond to activity and inactivity of a single avid collector. Let's retain just records collected from 1970 to 2000.

records <- records[records$year >=1970 & records$year <=2000, ]  
hist(records$year,  
 xlab='Collection year',  
 ylab='Number of records',  
 main='Collection year (1970-2000)'  
)



nrow(records)

## [1] 1942

How many records have we lost?

saveRDS(records, './Species Records/02 Species Records - Retained Records between 1970-2000.rds')

## Ensuring locational reliability

Ensuring locational reliability is the most troublesome step, but very important because wrongly-placed records can mislead a model. Again, short of having been at the collection site when the specimen was collected, you can't verify location. Even the specimen tag may be wrong--we've seen tags with place names plus verbal descriptions of directions to get to those places, and the two were many miles apart from one another. So the most we can do is reduce the chance that records are incorrectly geolocated.

First, let's remove records with no longitude or latitude:

records <- records[-which(is.na(records$longitude) | is.na(records$latitude)), ]  
nrow(records)

## [1] 1012

How many records had no associated longitude or latitude? Note, though that many of these same records do have locality descriptions. If you're needing more records, I would start by georeferencing these records using standard methods if you can.

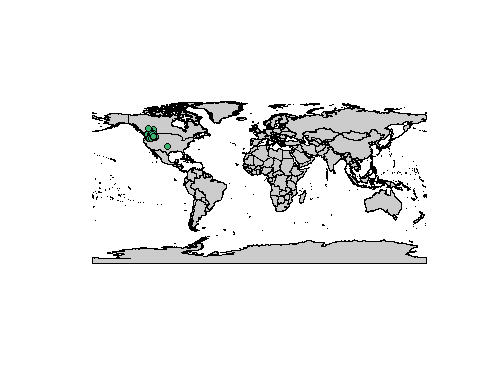
Let's plot these records on a map. By themselves the locations of points will be a little hard to interpret, so we'll use a shapefile of countries. Normally I suggest using the [Global Administrative Areas Database (GADM)](http://www.gadm.org/) map because it's constantly updated and contains detailed information on counties/parishes/districts, etc. However, it's very large so we'll use another for this exercise that we've already downloaded.

The [World Borders Dataset](http://thematicmapping.org/downloads/world_borders.php)(<http://thematicmapping.org/downloads/world_borders.php>) can be found in the folder of the same name. Load this into R.

countries <- rgdal::readOGR(  
 './World Borders Dataset',  
 'TM\_WORLD\_BORDERS-0.3'  
)

## OGR data source with driver: ESRI Shapefile   
## Source: "./World Borders Dataset", layer: "TM\_WORLD\_BORDERS-0.3"  
## with 246 features  
## It has 11 fields  
## Integer64 fields read as strings: POP2005

sp::plot(countries, col='gray80')  
points(records$longitude, records$latitude, bg='mediumseagreen', pch=21)



What do you see? What stands out? Does everything make sense?

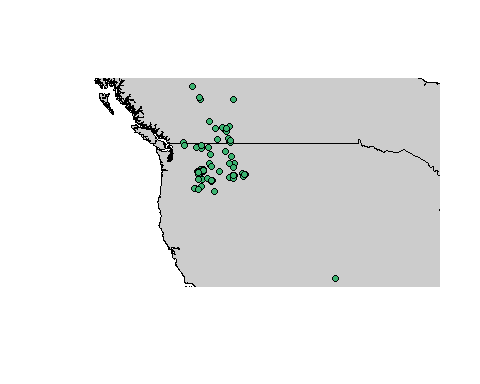
This probably won't occur in this example, but quite often there's at least one erroneous record in the Gulf of Guinea at coordinate (0, 0). This occurs When someone entering the coordinate data found no coordinates for this record, and so used to indicate "no data". But this is obviously an error--zero does not mean no data! If you were to map all known biodiversity records you would probably find (0, 0) is the most biodiverse place on Earth.

What else do you see that may be concerning? How about that record in the Great Plains (southeast of the main body of presences)? It's possible that the species was there, but it seems highly improbable. Let's take a closer look by zooming in on the records that remain. This will let us highlight that record. **Take home:** Closely examine maps of specimen records to ensure a terrestrial species' records aren't in the water, for example.

We'll manually create a bounding box around the points then use that box to zoom in.

1. First, we'll coerce the points into a spatial object (i.e., like a points shapefile):

# The "proj4string"" argument is a code for the coordinate system of unprojected  
# (WGS84) coordinates.  
recordsSpatial <- SpatialPointsDataFrame(  
 coords=cbind(records$longitude, records$latitude),   
 data=records,  
 proj4string=CRS('+proj=longlat +datum=WGS84 +no\_defs +ellps=WGS84 +towgs84=0,0,0')  
)  
plot(recordsSpatial, col='white')  
sp::plot(countries, add=TRUE, col='gray80')  
points(recordsSpatial, pch=21, bg='mediumseagreen')



Here we've taken good advantage of the system for plotting spatial objects. We plot the spatial object first--R automatically adjusts the width and height of the plot to maintain the proper aspect ratio so longitude isn't "squeezed" in relation to latitude, for example. Then we plot the countries over these points, then plot the points again.

1. Now let's define a band of longitude dividing the outlying point from the others. We'll do this by using the mouse to click in the space between the outlying point and the rest of the records.

# Click somewhere along a line of latitude that would separate the erroneous  
# record from the rest of the records.  
lineInSand <- click(n=1)  
lineInSand

Note that lineInSand is a 2-column matrix. In the code below we'll assume you want to separate records by a particular line of longitude. If you want to separate records by latitude, change lineInSand[1] to lineInSand[2]. Depending on where you clicked, **you may also need to change > to <**.

records[which(records$longitude > lineInSand[1]), ]

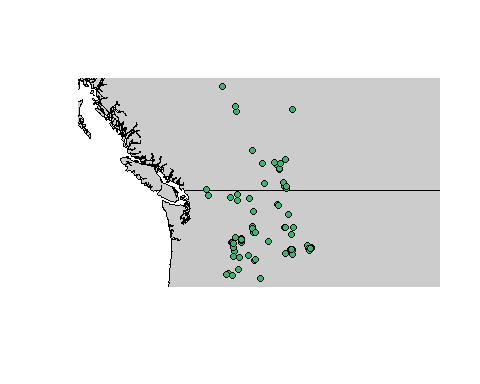
What do you see? First, notice that the outlying record has a place description. A quick search on Wikipedia reveals that Elgin, Oklahoma is indeed a place, so it's probably not a result of a mistaken place name. However, there is an Elgin in Oregon... "Oregon", "Oklahoma", they're the same, right? Should we keep this record? It's fairly clear we should not, but often problem records are not so easily identified. You can check the collector (i.e., are they a respected taxonomist?) and locality description (earlier versions of this exercise used a species georeferenced to Karlsruhe, Germany when in fact it was collected in Karlsruhe, North Dakota).

Remove these records:

records <- records[-which(records$longitude > lineInSand[1]), ]

Let's replot the data.

recordsSpatial <- SpatialPointsDataFrame(  
 coords=cbind(records$longitude, records$latitude),   
 data=records,  
 proj4string=CRS('+proj=longlat +datum=WGS84 +no\_defs +ellps=WGS84 +towgs84=0,0,0')  
)  
plot(recordsSpatial, col='white')  
sp::plot(countries, add=TRUE, col='gray80')  
points(recordsSpatial, pch=21, bg='mediumseagreen')

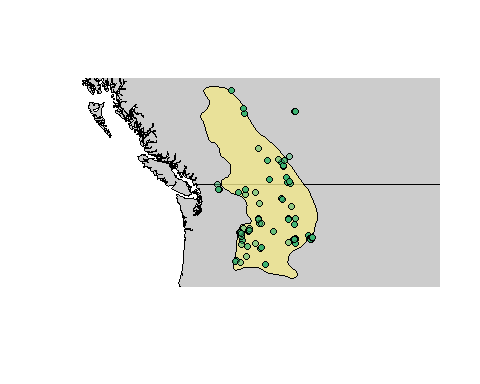


How well do these records represent the species' actual range? It's tempting to see if a range map exists and compare it to the points. In general this is a good idea, but realize that the range map was originally drawn from biodiversity collections so isn't an independent test of points' validity. For this particular species there is indeed a range map available from the [IUCN]('http://www.iucnredlist.org/technical-documents/spatial-data'). The actual file is large because it contains all the world's mammals, so we've included the range map just for this species in the folder Urocitellus columbianus.

# This line reads the shapefile. The first argument is the directory, the second the file name.  
# Note that if you include a trailing "/" in the first argument you will get an error. Took  
# me an hour to figure this out!  
rangeMap <- rgdal::readOGR(  
 './Urocitellus columbianus',  
 'urocitellusColumbianus\_iucnRangeMap'  
)

## OGR data source with driver: ESRI Shapefile   
## Source: "./Urocitellus columbianus", layer: "urocitellusColumbianus\_iucnRangeMap"  
## with 1 features  
## It has 19 fields

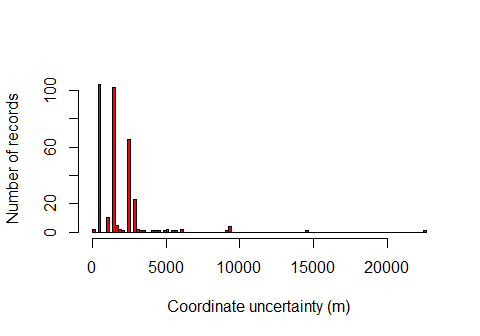
sp::plot(rangeMap)  
sp::plot(countries, add=TRUE, col='gray80')  
sp::plot(rangeMap, add=TRUE, col=alpha('khaki', 0.8))  
points(recordsSpatial, pch=21, bg=alpha('mediumseagreen', 0.5))



There are several points that fall outside the range map, but most of them are probably OK given the coarse nature of the range map. It would be especially important to investigate that site lying to the northeast of the range polygon, but for now we'll let it be. Remember--the range map isn't the "truth"--it's ultimately derived from specimen data!

Many records were obtained before the advent of GPS. As a result, they have been georeferenced post-de facto by someone who was not the collector. Depending on the institution doing the georeferencing, this may have been done with or without accounting for uncertainty in interpretation of the locality description where the specimen was collected. There is a standard methodology for assessing error around points, and if this was practiced, then the value can be found in the field coordUncer in the records data frame. These values represent the best guess of the radius of a circle around the coordinate in which the specimen was actually likely to have been collected. Let's look at the distribution of errors around coordinates.

hist(records$coordUncer,  
 breaks=100,  
 xlab='Coordinate uncertainty (m)',  
 ylab='Number of records',  
 main='',  
 col='red'  
)

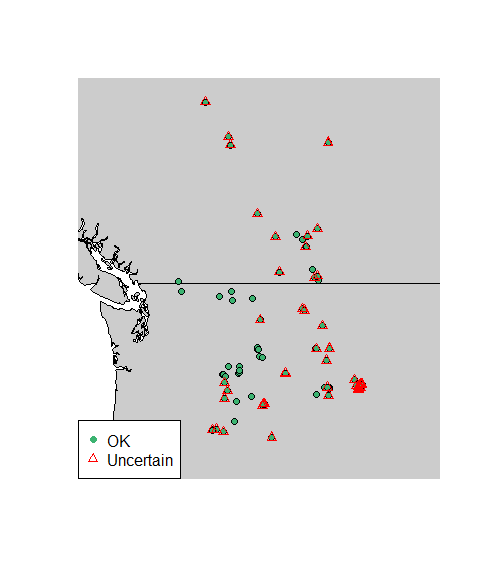


Most points have errors <5 km, but some have much larger errors. Large errors in coordinates arise when a collector simply named a general place (e.g., "eastern Oregon") which the georeferencer then had to interpret.

The most conservative way to ensure locational accuracy of records is to use only those records with a coordinate uncertainty less than some predetermined value. A fairly comprehensive paper by [Graham et al. (2008)](http://dx.doi.org/10.1111/j.1365-2664.2007.01408.x) found that across a wide range of species and modeling algorithms errors in coordinates up to ~10 km had no appreciable effect on model performance. Since then, others have noted that the effect of positional error depends on spatial autocorrelation of the environment around each point--we'll explore this later.

First, let's see where records occur with errors >10000 m or have no stated uncertainty.

recordsUncertain <- records[records$coordUncer > 10000 | is.na(records$coordUncer), ]  
plot(rangeMap)  
sp::plot(countries, add=TRUE, col='gray80')  
points(recordsSpatial, pch=21, bg='mediumseagreen')  
points(recordsUncertain$longitude, recordsUncertain$latitude, pch=2, col='red')  
legend('bottomleft', legend=c('OK', 'Uncertain'), pch=c(16, 2),  
 col=c('mediumseagreen', 'red'), bg='white')



nrow(records)

## [1] 1011

nrow(recordsUncertain)

## [1] 678

There seem to be quite a few records with no associated uncertainty or too much uncertainty!

We will still have quite a few records if we remove these uncertain points, so let's be cautious and do that. This is especially warranted for species that live in the mountains like the Columbian Ground Squirrel. With just a few hundred meters' change in elevation climate can change dramatically in montane systems. Thus, it's wise to be extra cautious when modeling montane species for this reason since in mountains even a small horizontal error can move a point very far up or down in elevation. That said, here we'll use 10 km as a cutoff because our climate data has a resolution of ~18 km. Normally you'd probably be using much finer-resolution climate data, in which case you'd use a correspondingly smaller cutoff.

nrow(records)

## [1] 1011

records <- records[which(records$coordUncer <= 10000), ]  
nrow(records)

## [1] 333

How many records did we lose?

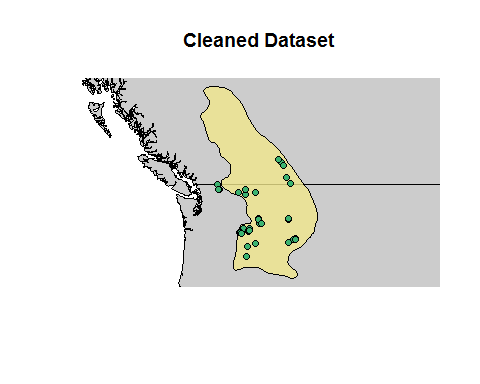
Note that the opposite problem can also occur--some records may be listed with an unbelievably small coordinate error (i.e., a few centimeters or less). This doesn't seem to be the problem in our case, though:

head(sort(unique(records$coordUncer)))

## [1] 50.00 579.36 587.00 1139.00 1142.63 1151.00

Let's map the records we'll be using.

# plot  
recordsSpatial <- SpatialPointsDataFrame(  
 coords=cbind(records$longitude, records$latitude),   
 data=records,  
 proj4string=CRS('+proj=longlat +datum=WGS84 +no\_defs +ellps=WGS84 +towgs84=0,0,0')  
)  
plot(rangeMap, main='Cleaned Dataset')  
sp::plot(countries, add=TRUE, col='gray80')  
plot(rangeMap, col=alpha('khaki', 0.8), add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')



Notice we lost a lot of the records that represent the Canadian portion of the species' range. In this case we might try searching other data portals (especially Canadensys--the Canadian node for GBIF) or contacting local museums. If possible, we could also inspect the specimens that we threw out due to uncertainty to see if we can better locate them and make them usable. This might involve talking to people at museums or even going there in person!

**Take home**: If the lack of reliable of records is so severe in major portions of the species range you should reconsider how you could obtain more records or if it is worth continuing at all.

Save, save, save!

saveRDS(records, './Species Records/03 Species Records - Removed Observations  
 with Bad Coordinates.rds')

Now is also a good time to save your work space!

save.image('./00 Modeling Workspace - Initial Biodiversity Data Collection and Cleaning.Rdata')

### Other methods for checking coordinate uncertainty

There are many other methods for checking coordinate uncertainty:

* Matching the stated tertiary administrative unit (county/parish/district) to the one in which the coordinate actually lies. Matching administrative units can be very helpful but it's arduous because of the many ways in which place names can be written (e.g., "Sedgwick County", "Sedgwick Cty", "Sedg."). At best it takes a combination of machine-assisted manual checking and at worst a lot of manual checking. So for today we'll skip this and rely on other methods. If you do check records by comparing the stated administrative unit to the administrative unit in which the coordinate lies, be aware:
* Many databases automatically populate fields pertaining to country, state/province, etc. based on the coordinates. So checking using these fields will only confirm what was already demonstrated.
* It might have been hard for collectors to correctly identify the county/parish/district they were in if the collection occurred near the border.
* Counties, parishes, and districts have changed borders over the years.
* Diacritics (e.g., a place name written with "?" versus "a") can make machine-matching difficult. We've written a [script](http://www.earthskysea.org/r-code/) that attempts to replace diacritics with their uninfected versions.
* Methods exist for using "atlas" (county-level) data (e.g., [Bombi & D'Amen 2011]((http://dx.doi.org/10.1111/j.1365-2699.2011.02627.x);%20%5BKeil%20et%20al.%202014%5D(http://dx.doi.org/10.1111/ddi.12199))).
* Matching the stated elevation of collection with the elevation of the record's coordinate. Presumably they'd be close to one another. Note, though that most records don't include this information.
* Looking for outliers in environmental space. We'll do this in a later exercise.

## Did you save your work so far?

save.image('SDM Workshop', compress=TRUE)

# Reflection

1. How many records did you start with and how many did you end up with?
2. What kind of unreliable records did your species have?
3. How did you handle records that had unreliable geolocations?
4. What other aspects of the data might be an issue? How would you find "pernicious" errors (records that at first glance appear legitimate but are not actually valid)?