BioClim Data

Alton Barbehenn

December 3, 2015

Getting BioClim Data

The easiest way to get the actual BioClim data is to look it up online at http://www.worldclim.org. This website will give you the necessary definitions of each BioClim and each map at three different levels of precision. Once you have downloaded the .bil and corresponding .hdr file for your desired BioClim zone, you are all set to extract the BioClim data. Using the BioClim data will be easier if the .bil and .hdr files are in your working directory (I will assume this in the examples).

Using the `raster` package we can access the BioClim data. To start we need to get the data from the .bil file. For the sake of an example I will be using BioClim9 (Mean Temperature of Driest Quarter).

```
require(raster)
```

```
## Loading required package: raster
## Loading required package: sp
```

```
bio9 = raster("bio_9.bil")
fromDisk(bio9)
```

```
## [1] TRUE
```

The first line takes the raster data from the .bil file and puts it in the usable variable `bio9`. The second line checks to see if `bio9` was created successfully. This step is not necessary, but is a good precaution to take when dealing with this large and important variable.

Plotting the BioClim Data

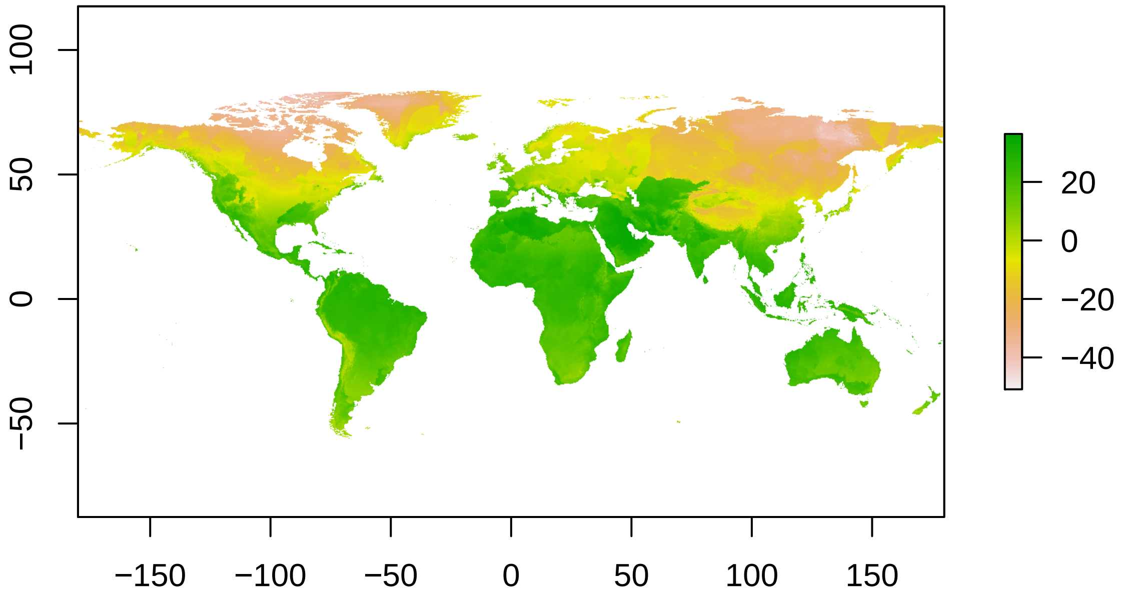
We need to do one more important thing before we can use this data. For the sake of saving memory while maintaining precision, each point of the raster data is stored as a integer value ten times larger than actual. So we need to make the correction:

```
bio9 <- bio9/10
```

This step will take a long time because you are making this calculation hundreds of millions of times (almost a trillion times at the highest resolution).

Now that we have the BioClim data in a usable form, we can plot it.

```
plot(bio9)
```



Analyzing BioClim Data

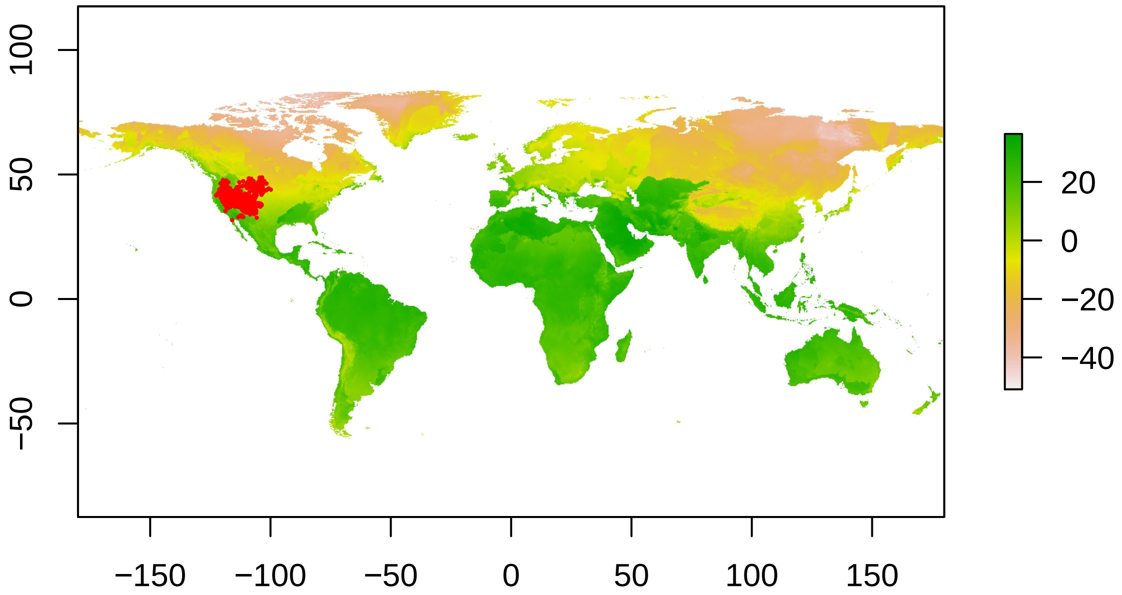
Alternately, if we want BioClim values at specific coordinates we can do that too. Assume `gis_data` is a data frame containing latitude and longitude coordinates as well as color. We must first make the location coordinates into a spatial data frame object, through the following process:

```
require(sp)
coordinates(gis_data) <- ~longitude+latitude
proj4string(gis_data) <- CRS("+proj=longlat +datum=WGS84 +no_defs
                             +ellps=WGS84 +towgs84=0,0,0")
```

This process changes `gis_data` from a standard data frame to a spatial data frame with `coordinates()`, then it produces coordinates out of the two columns of `gis_data`: `longitude` and `latitude`. These coordinate pairs are then transformed according to the Mercator Projection in the next line by the `proj4string()` and `CRS()` functions.

Now we can map onto the raster plot of the BioClim data in order to visualize the climates in which our `gis_data` is being found. Also note that you can also plot shapes onto this plot using so long as the shape is the correct projection; this can be done by the same process as we used with the points.

```
plot(bio9)
points(gis_data, col = "red", pch = 20, cex = 0.25)
```

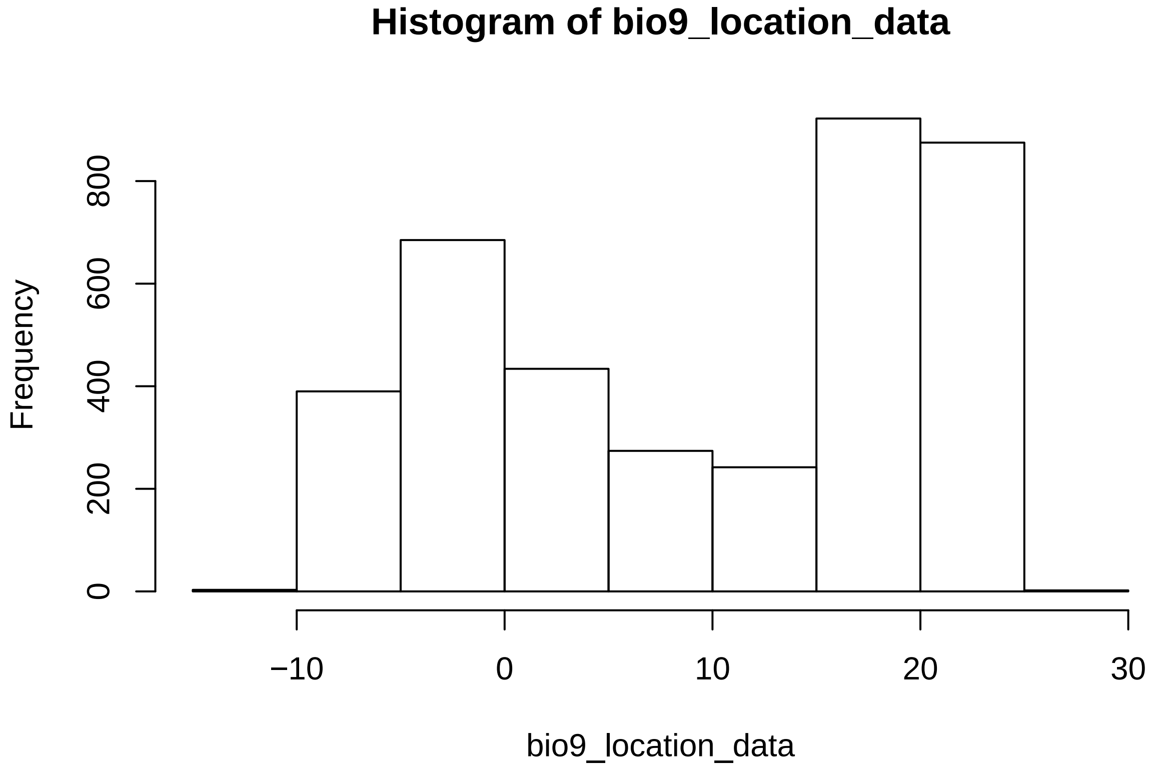


Now that our location data is properly formatted, we can extract values at each location on the BioClim map.

```
bio9_location_data = extract(bio9, gis_data)
```

This data can be used in the normal way. For example, we can produce a histogram of the BioClim value at each of our locations:

```
hist(bio9_location_data)
```


Histogram of bio9_location_data
Frequency
bio9_location_data