Spatial Data Analysis in R

Misc. Concepts 1: ggplot2, NetCDF, Modeling Species Distributions

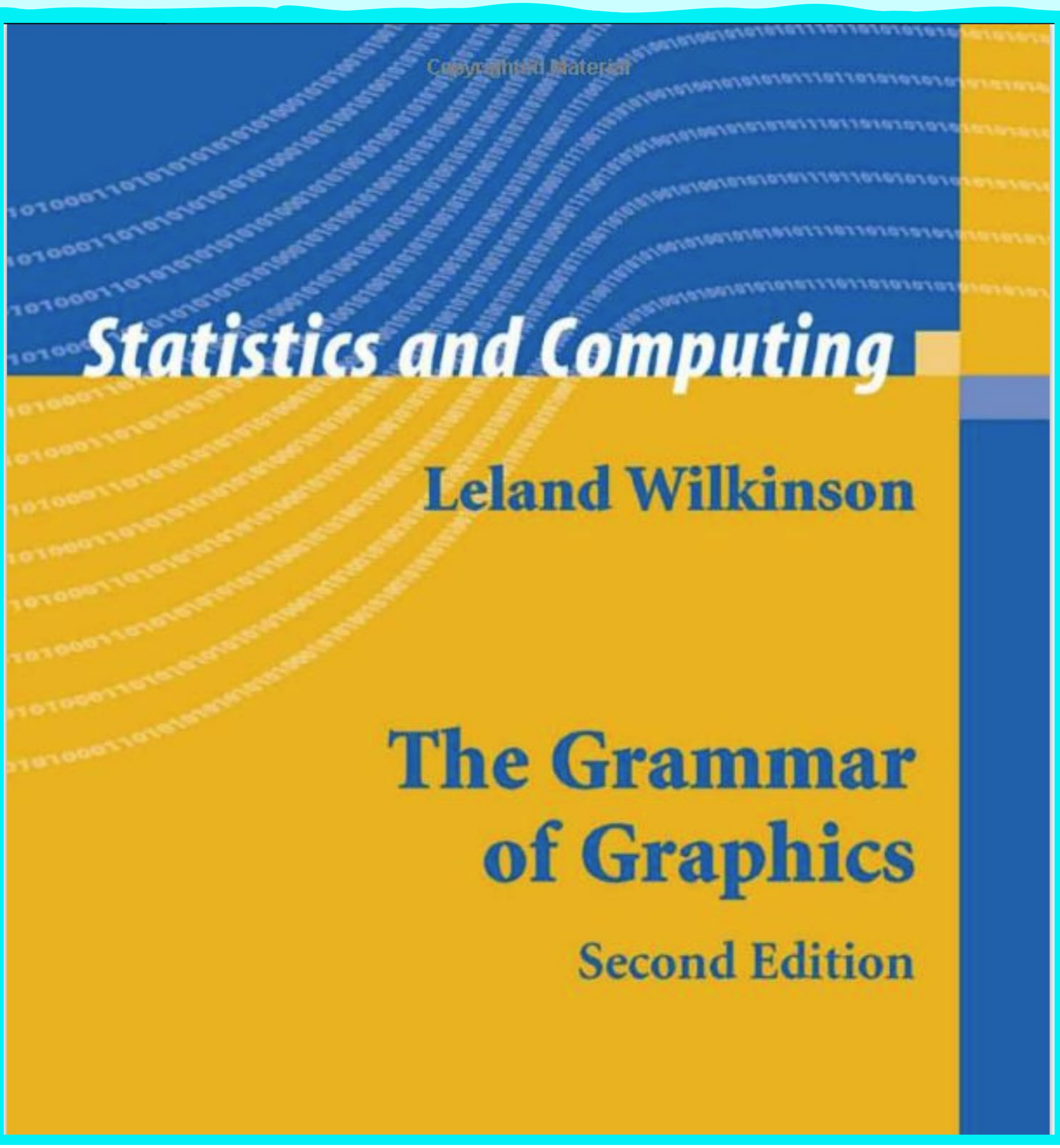
Eco 697DR – University of Massachusetts, Amherst – Spring 2022
Michael France Nelson

Crash Course in ggplot

Just the basics

The Grammar of Graphics

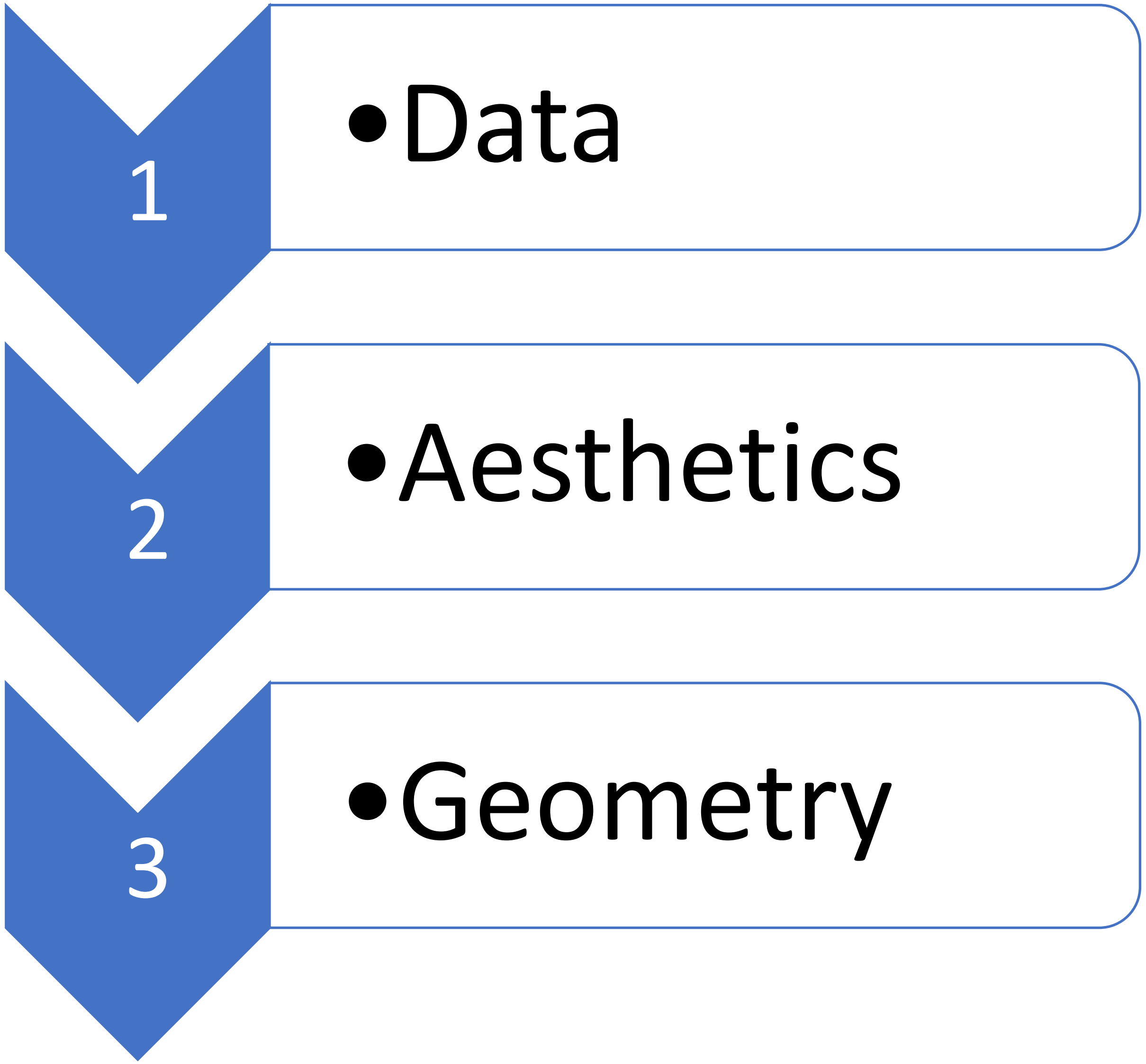
- A logical, layered approach to creating graphics.
- The Grammar of Graphics is a philosophy.
- Being intentional about graphic elements.
- Grammar of graphics is a great fit for spatial data.



Grammar of Graphics in R

- The Grammar of Graphics is a philosophy.
- It is *implemented* in R with the **ggplot2** package.

Grammar of Graphics in R



1
- Data

2
- Aesthetics

3
- Geometry

ggplot: Data (Frames) In Row Format

Data in row-format is key to success with ggplot!

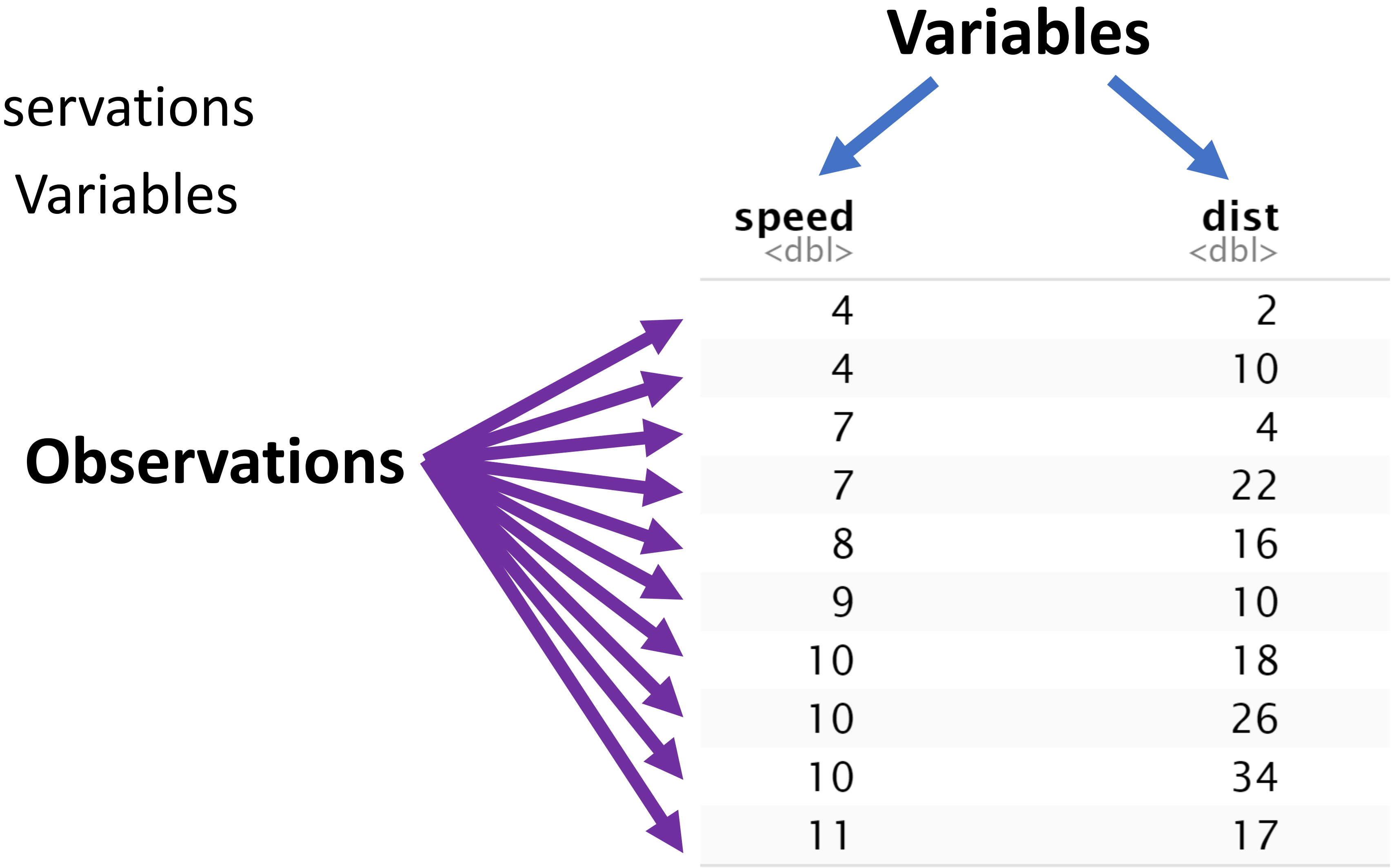
But… what is the row data paradigm?

The row data paradigm:

- Rows are observations, or features.
- Columns are attributes, i.e. variables.
- This sounds a lot like an attribute table in Arc GIS…

Simple Data Example: Cars

- Rows = Observations
- Columns = Variables



Variables

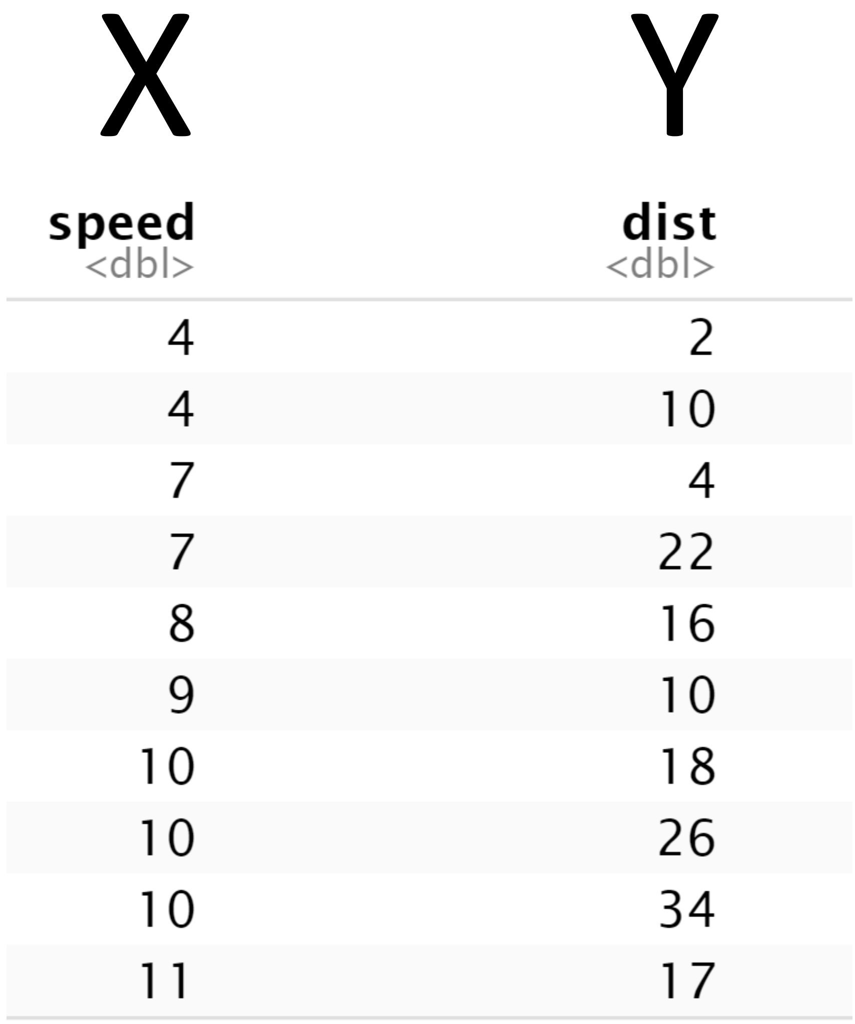
Observations

speed <dbl>	**dist** <dbl>
4	2
4	10
7	4
7	22
8	16
9	10
10	18
10	26
10	34
11	17

ggplot: Aesthetics

- In ggplot, aesthetics specify how variables are mapped to components of a plot. For example:
  - X- and Y- values
  - Groups
  - Color and fill
- Aesthetics are specified with the aes() function.

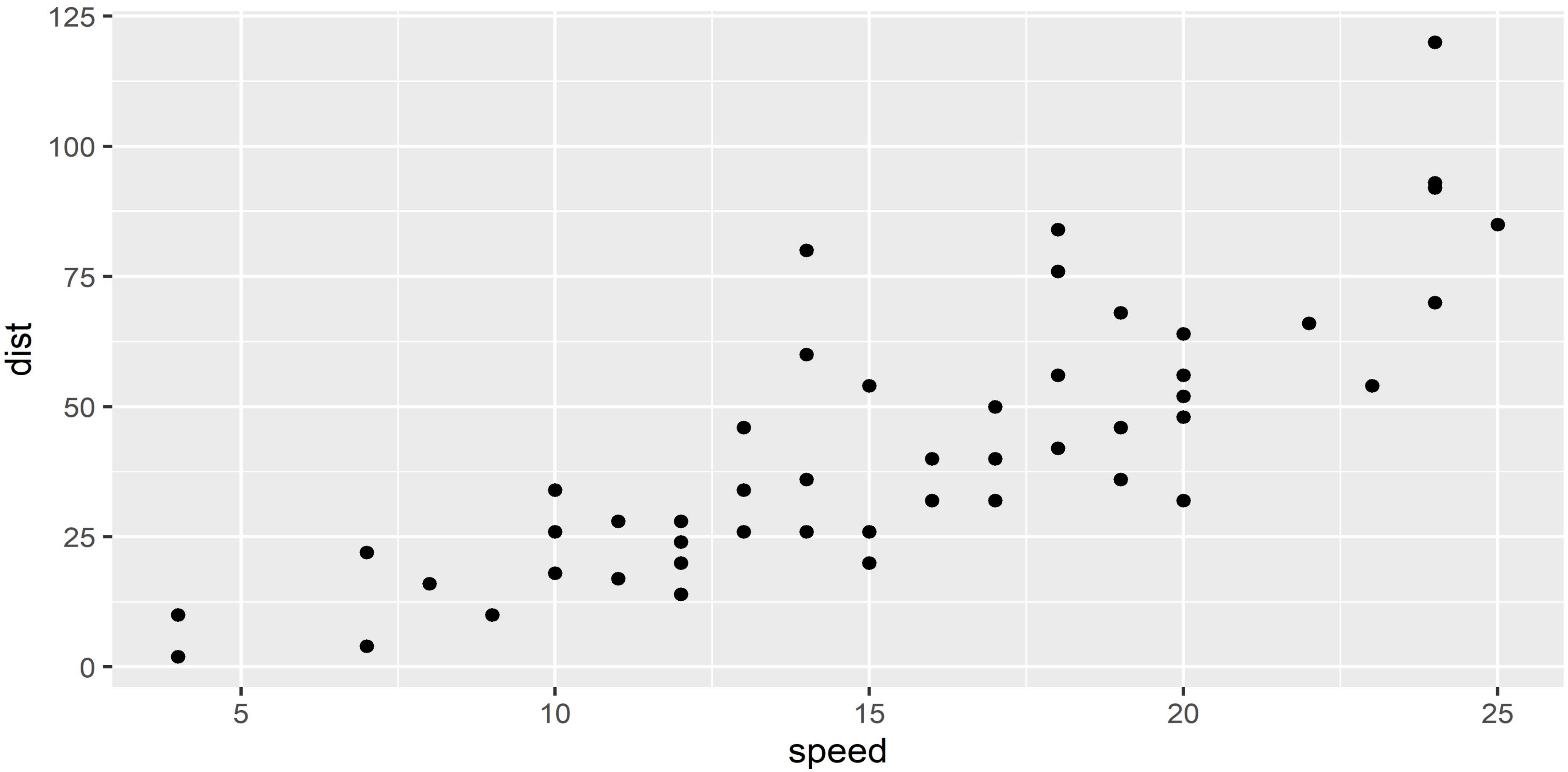
```
ggplot(cars, aes(x = speed, y = dist))
```



X	Y
speed	dist
<dbl>	<dbl>
4	2
4	10
7	4
7	22
8	16
9	10
10	18
10	26
10	34
11	17

Cars Scatterplot: X- and Y- Aesthetics

Y

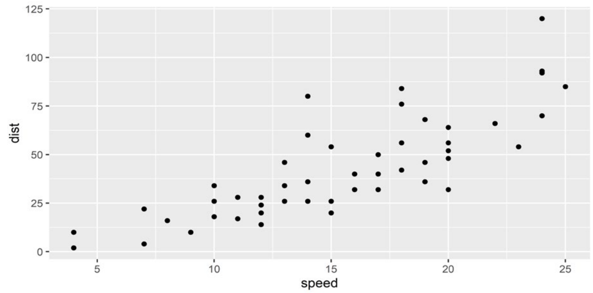


X

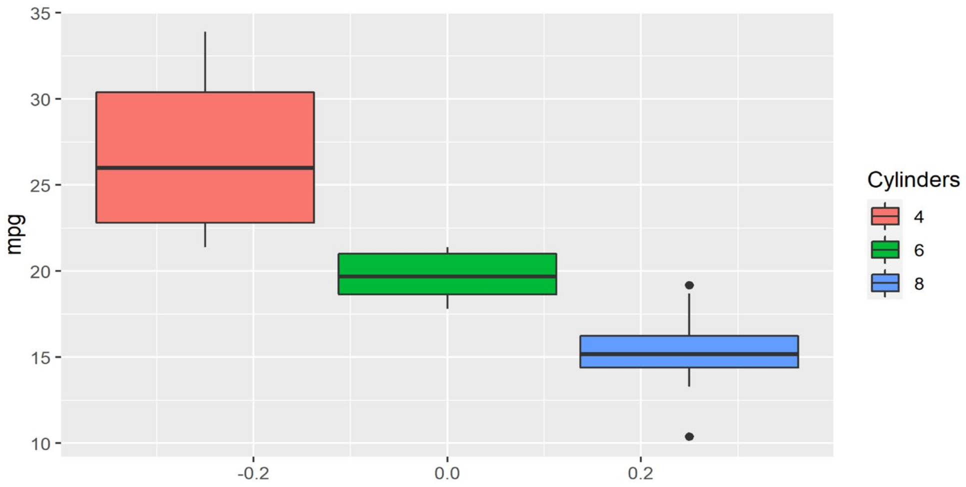
ggplot: Geometries

Geometries specify the type of plot. For example:

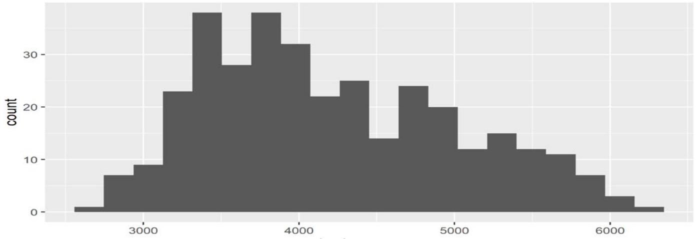
- Scatterplots



- Boxplots



- Histograms

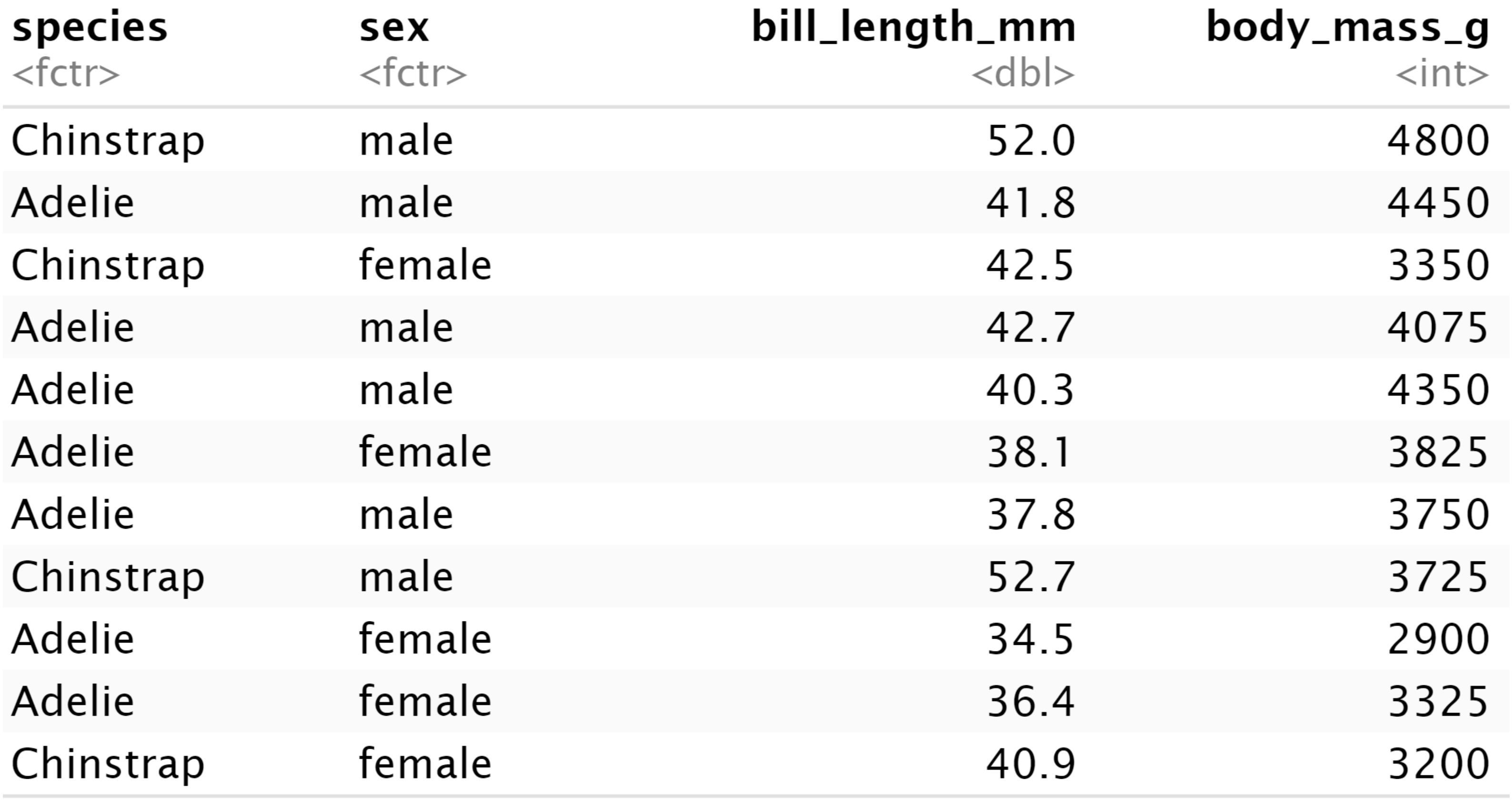


There are tons of different geometries, each one recognizes a different set of one or more aesthetics.

Example Scatterplot: Penguins Data

The Palmer penguins dataset contains lots of variables, we'll concentrate on:

- Species
- Sex
- Bill Length
- Body Mass



species <fctr>	sex <fctr>	bill_length_mm <dbl>	body_mass_g <int>
Chinstrap	male	52.0	4800
Adelie	male	41.8	4450
Chinstrap	female	42.5	3350
Adelie	male	42.7	4075
Adelie	male	40.3	4350
Adelie	female	38.1	3825
Adelie	male	37.8	3750
Chinstrap	male	52.7	3725
Adelie	female	34.5	2900
Adelie	female	36.4	3325
Chinstrap	female	40.9	3200

Scatterplot 1: Body Mass and Bill Length

1: Data

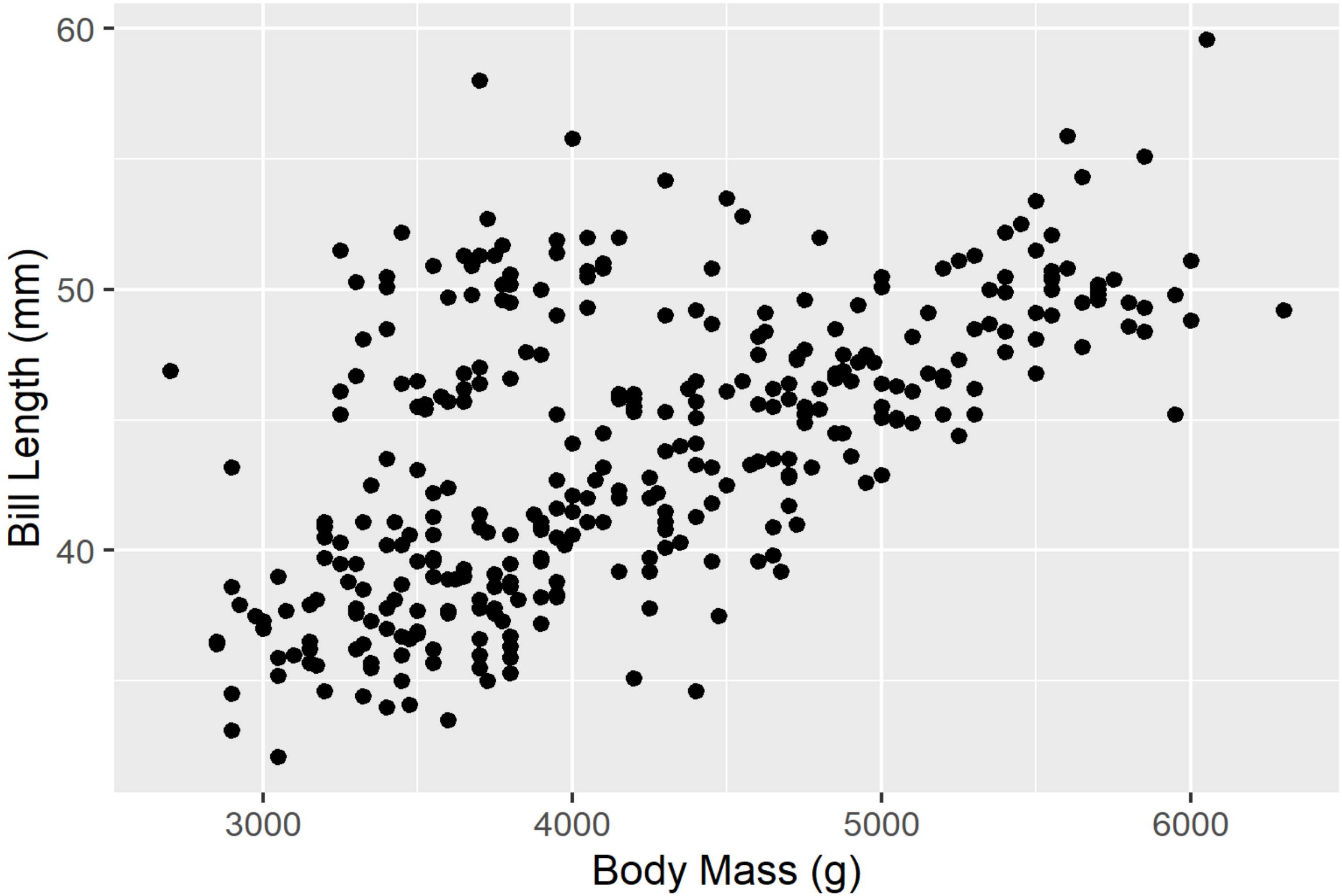
- Penguins

2: Aesthetics

- X: Body Mass
- Y: Bill Length

3: Geometry

- Points (scatterplot)



Scatterplot 1: Body Mass and Bill Length

1: Data

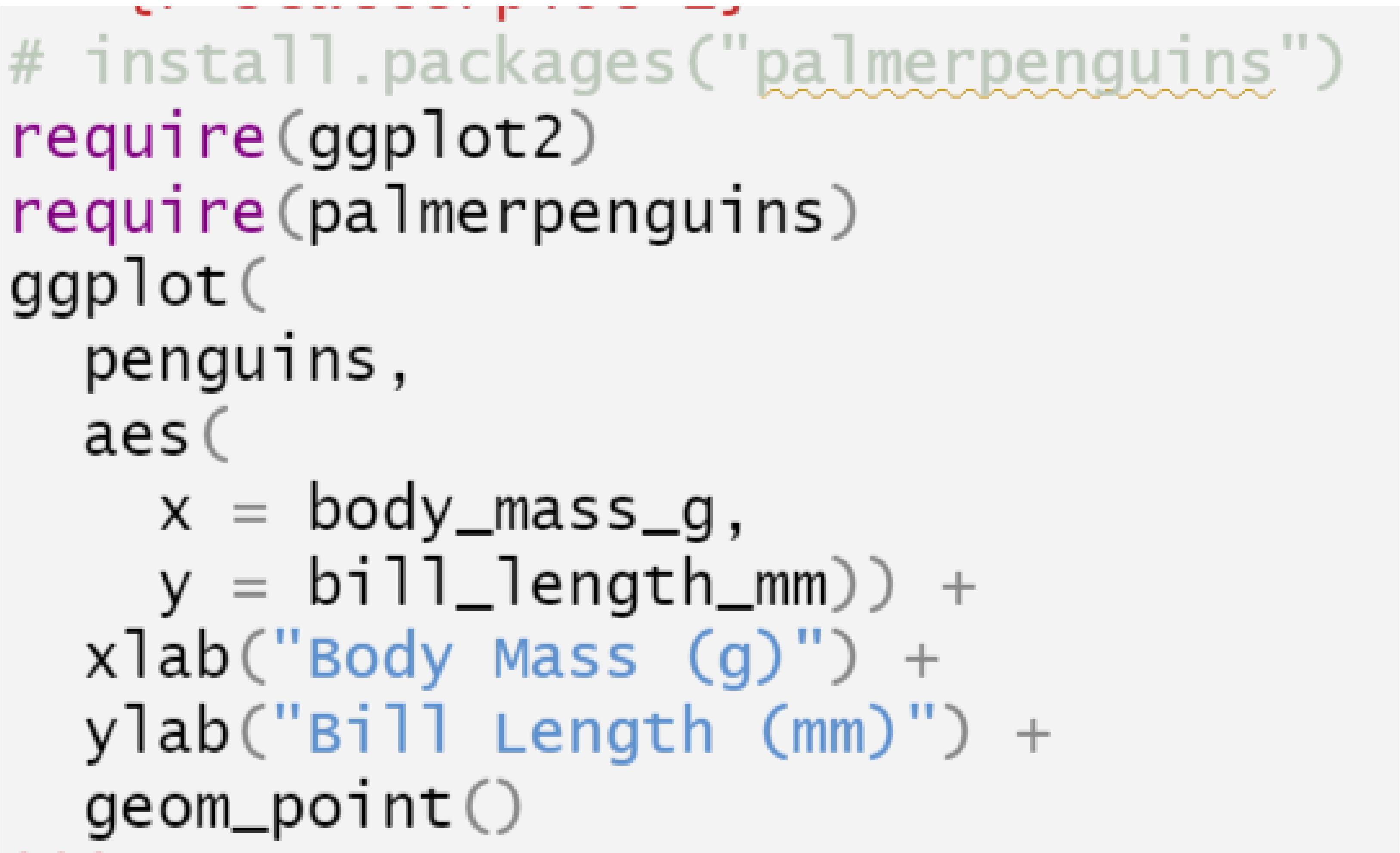
- Penguins

2: Aesthetics

- X: Body Mass
- Y: Bill Length

3: Geometry

- Points (scatterplot)

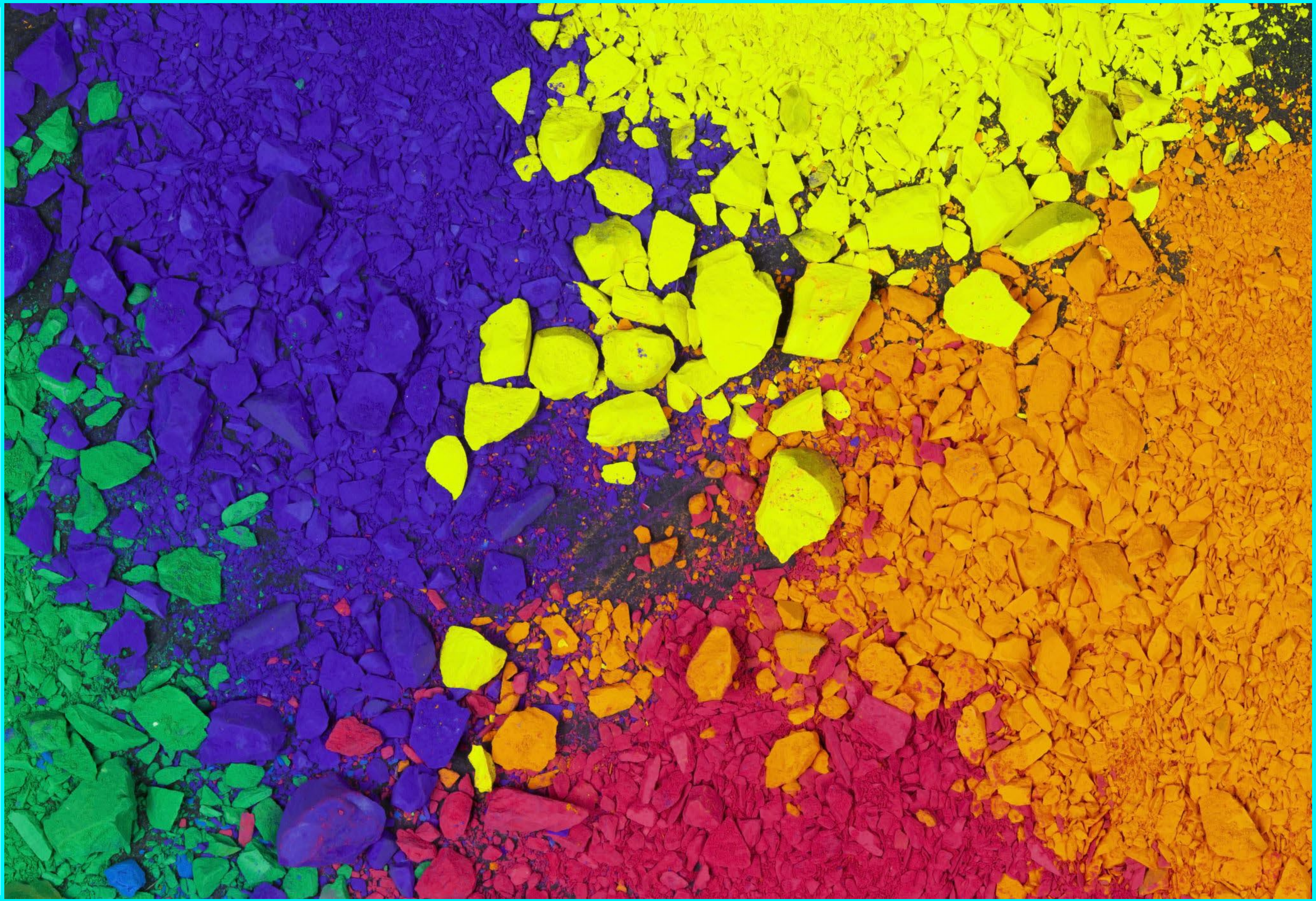


```r
# install.packages("palmerpenguins")
require(ggplot2)
require(palmerpenguins)
ggplot(
  penguins,
  aes(
    x = body_mass_g,
    y = bill_length_mm)) +
  xlab("Body Mass (g)") +
  ylab("Bill Length (mm)") +
  geom_point()
```

Now Let’s Add A Colour Aesthetic!

- That was cool!

- Let’s add some color



Scatterplot 2: Body Mass, Bill Length, Species

1: Data

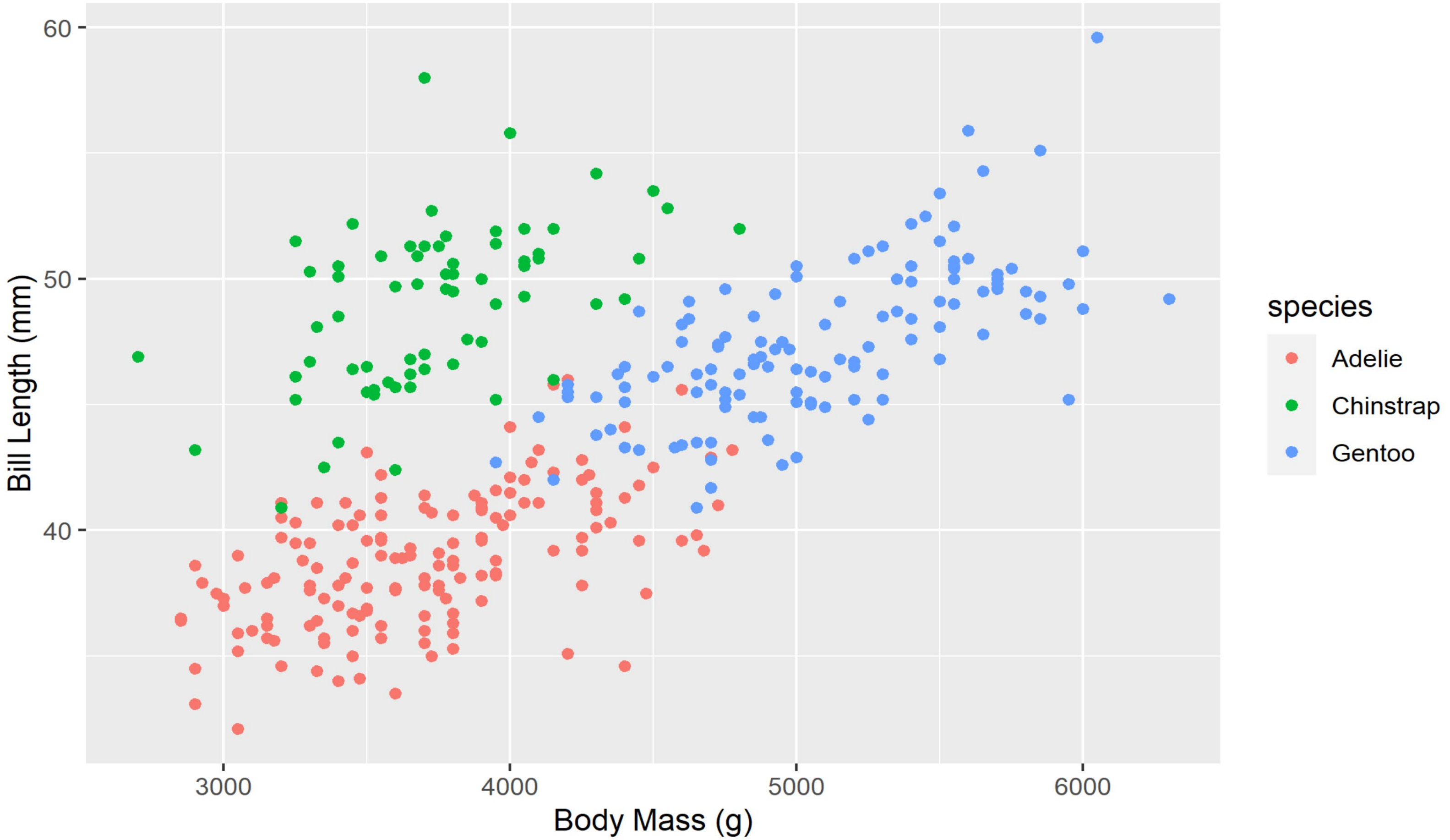
- Penguins

2: Aesthetics

- X: Body Mass
- Y: Bill Length
- Color: Species

3: Geometry

- Points (scatterplot)



Scatterplot 2: Body Mass, Bill Length, Species

1: Data

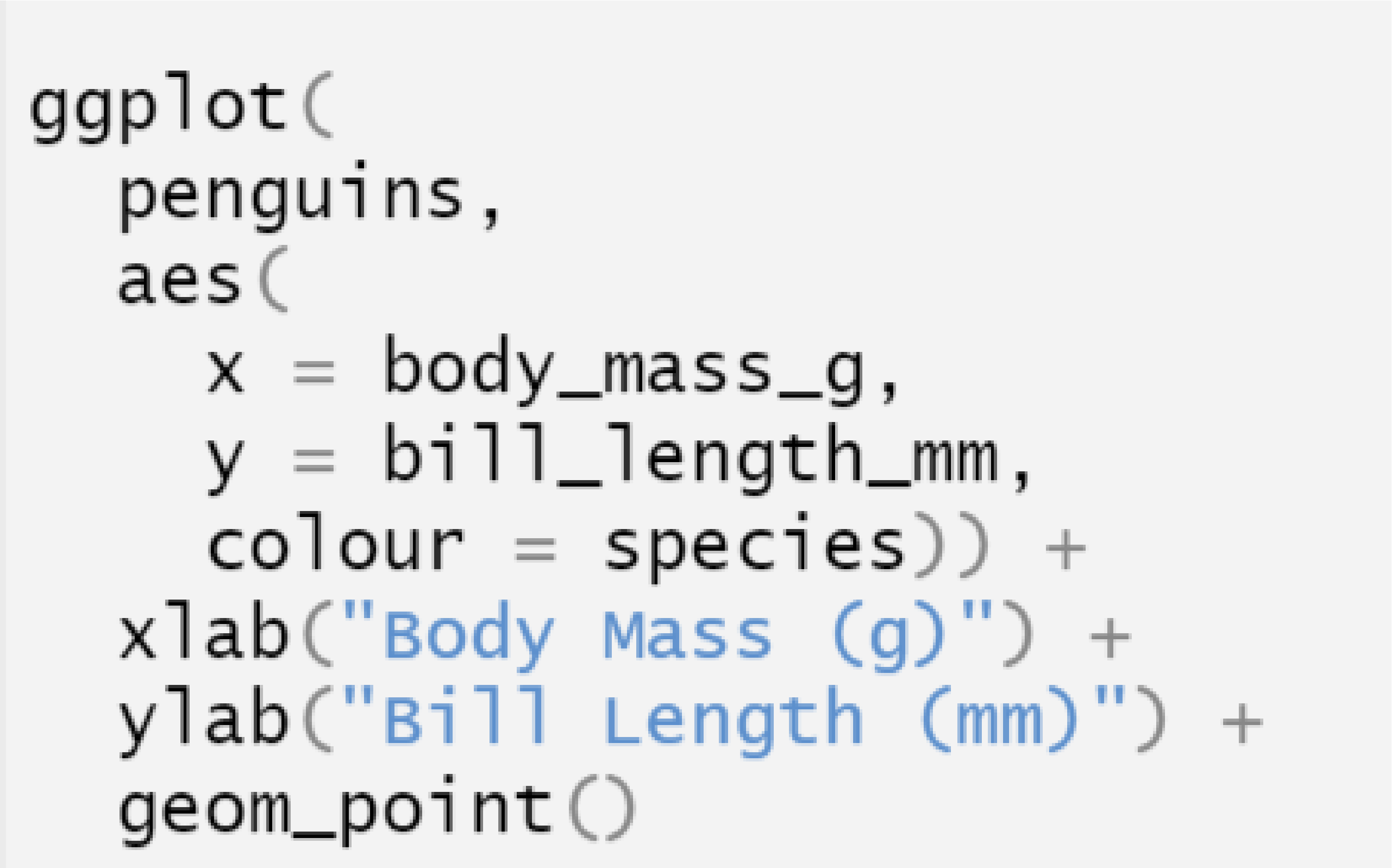
- Penguins

2: Aesthetics

- X: Body Mass
- Y: Bill Length
- Color: Species

3: Geometry

- Points (scatterplot)

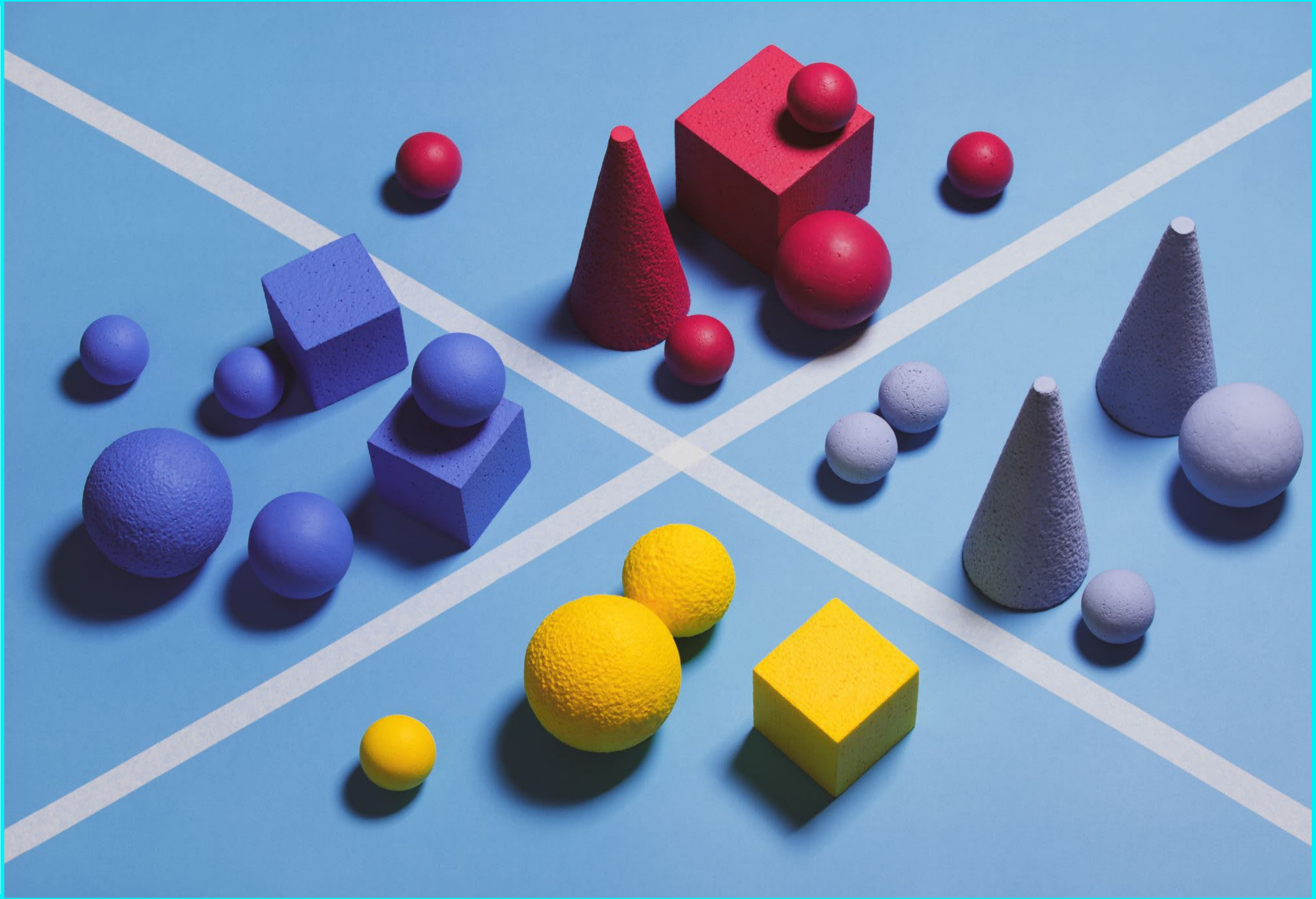


```
ggplot(
  penguins,
  aes(
    x = body_mass_g,
    y = bill_length_mm,
    colour = species)) +
  xlab("Body Mass (g)") +
  ylab("Bill Length (mm)") +
  geom_point()
```

Shape Aesthetic

- Great!

- Now let’s try some different shapes



Scatterplot 3: Body Mass, Bill Length, Species

1: Data

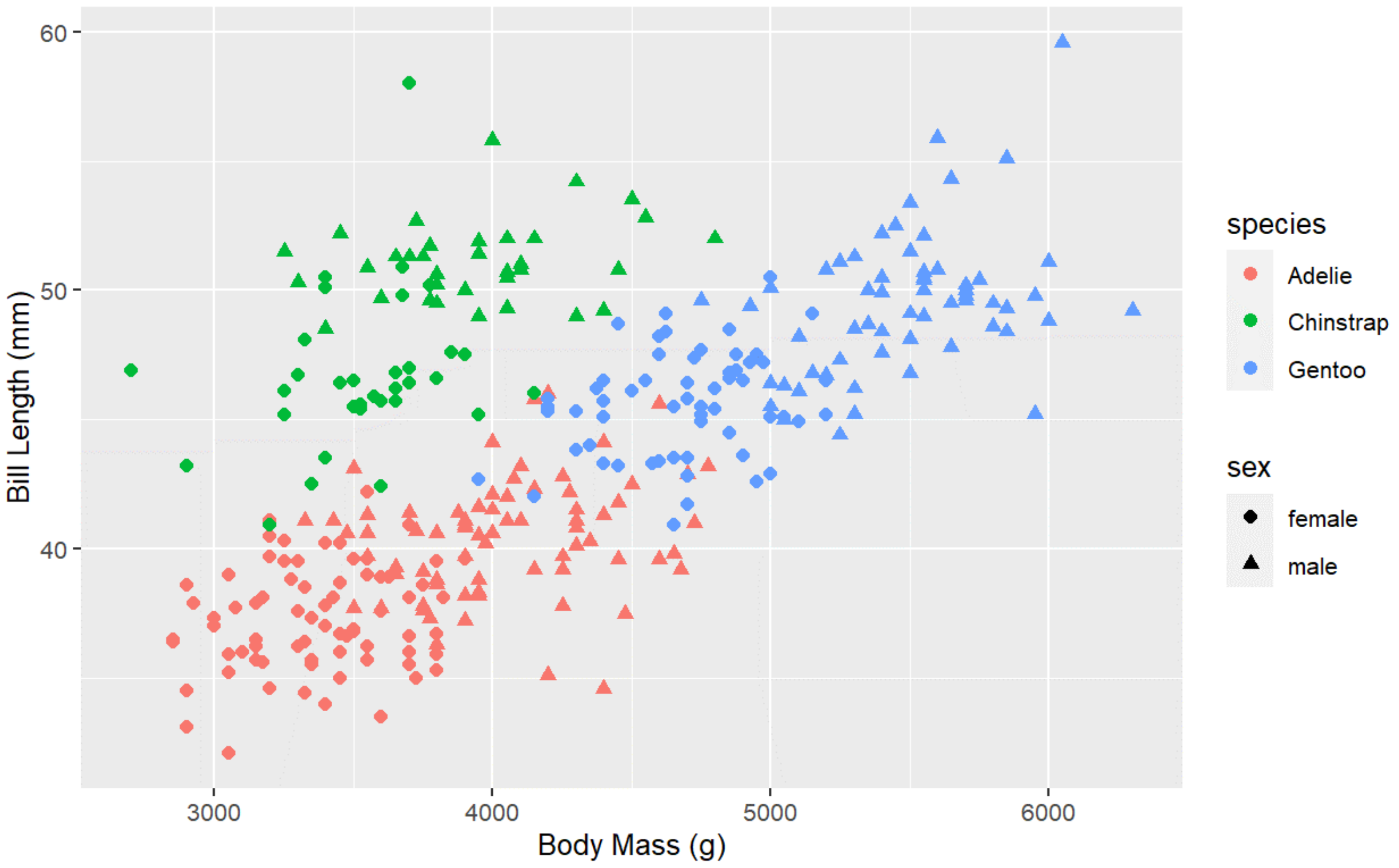
- Penguins

2: Aesthetics

- X: Body Mass
- Y: Bill Length
- Color: Species
- Shape: Sex

3: Geometry

- Points (scatterplot)



Scatterplot 3: Body Mass, Bill Length, Species

1: Data

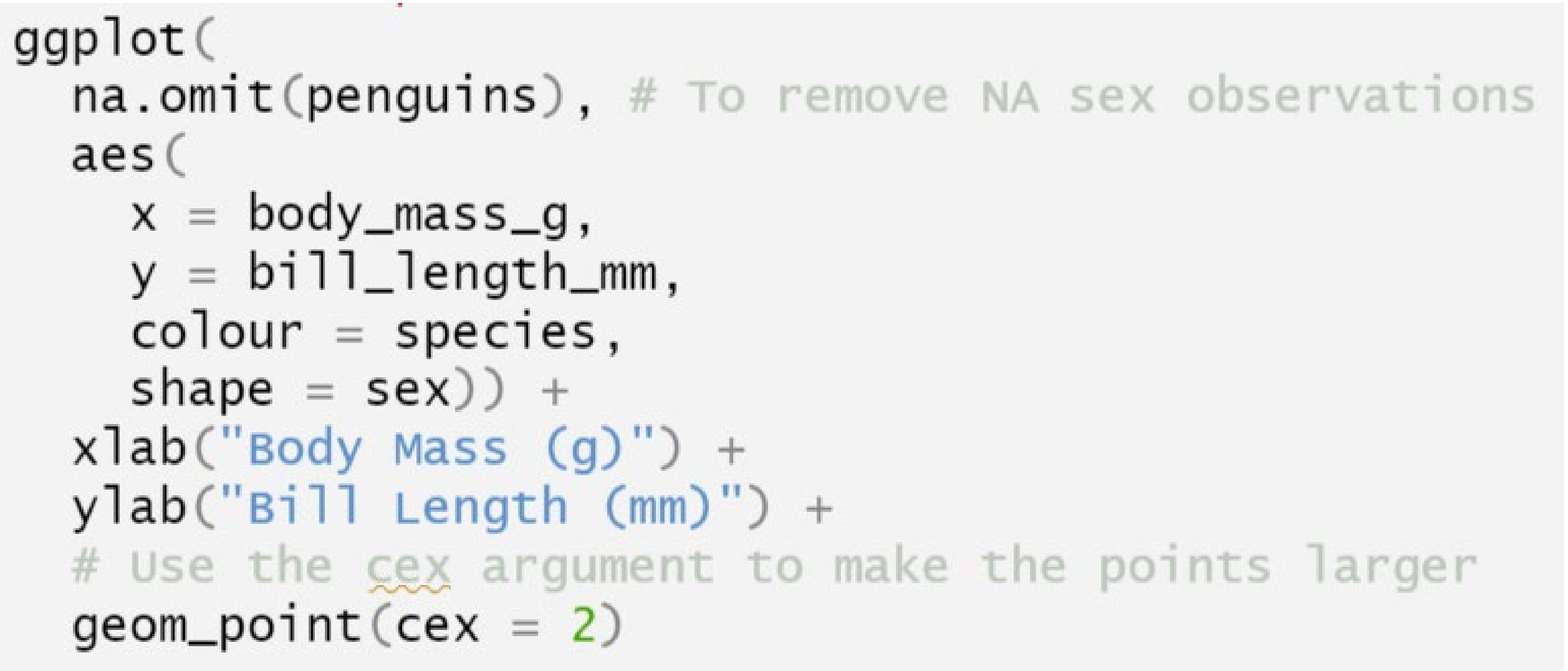
- Penguins

2: Aesthetics

- X: Body Mass
- Y: Bill Length
- Color: Species
- Shape: Sex

3: Geometry

- Points (scatterplot)



```
ggplot(
  na.omit(penguins), # To remove NA sex observations
  aes(
    x = body_mass_g,
    y = bill_length_mm,
    colour = species,
    shape = sex)) +
  xlab("Body Mass (g)") +
  ylab("Bill Length (mm)") +
  # Use the cex argument to make the points larger
  geom_point(cex = 2)
```

Recap: Basic Grammar of Graphics in R

1. Data
2. Aesthetics
3. Geometry

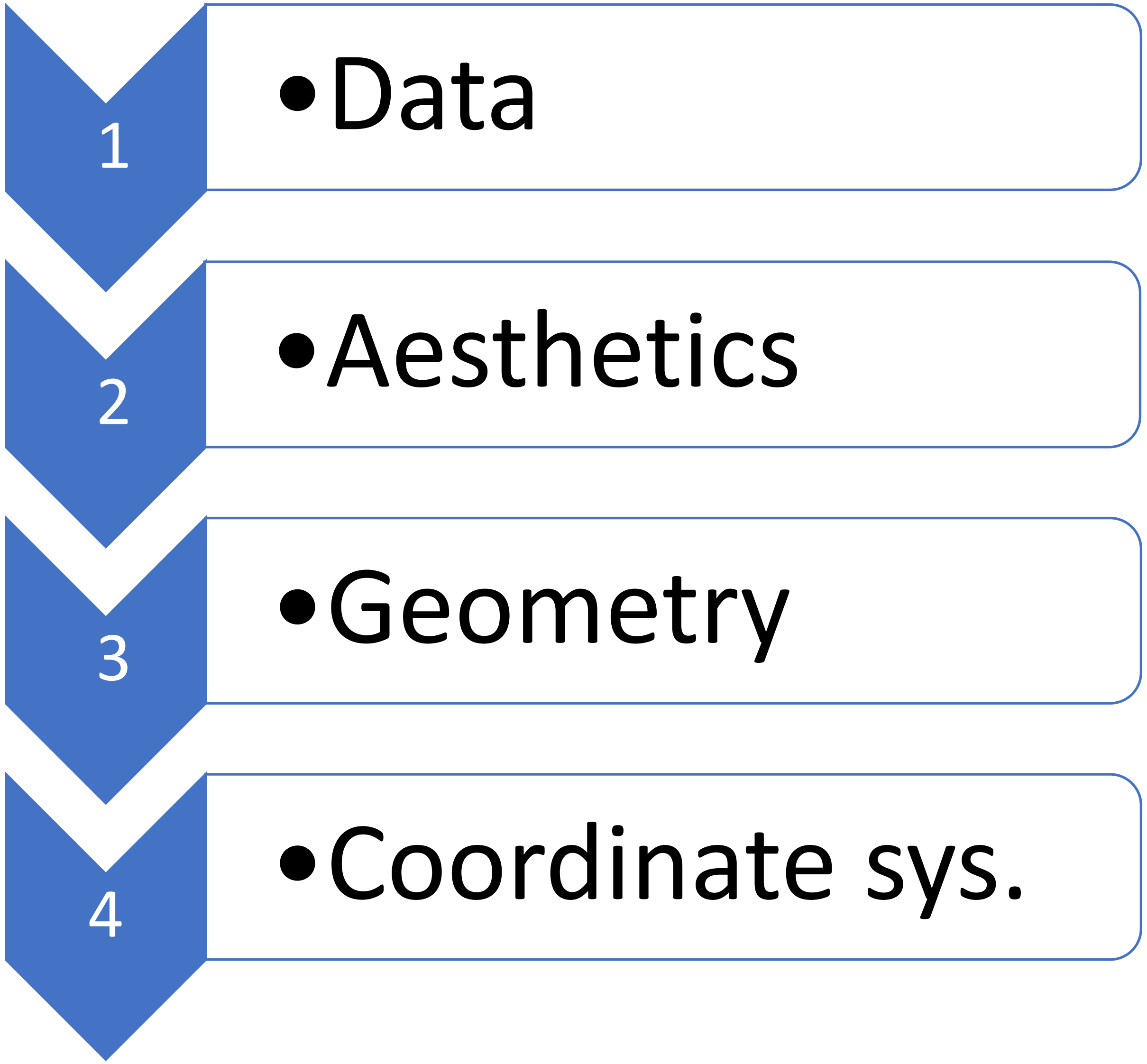
Plotting Simple Features With ggplot

Let’s make some maps!

Grammar of Graphics With Spatial Data

We can follow the familiar ggplot procedure with an additional step:

- Specify a coordinate system



1 • Data

2 • Aesthetics

3 • Geometry

4 • Coordinate sys.

Map 1: State Borders

1: Data

- CONUS

2: Aesthetics

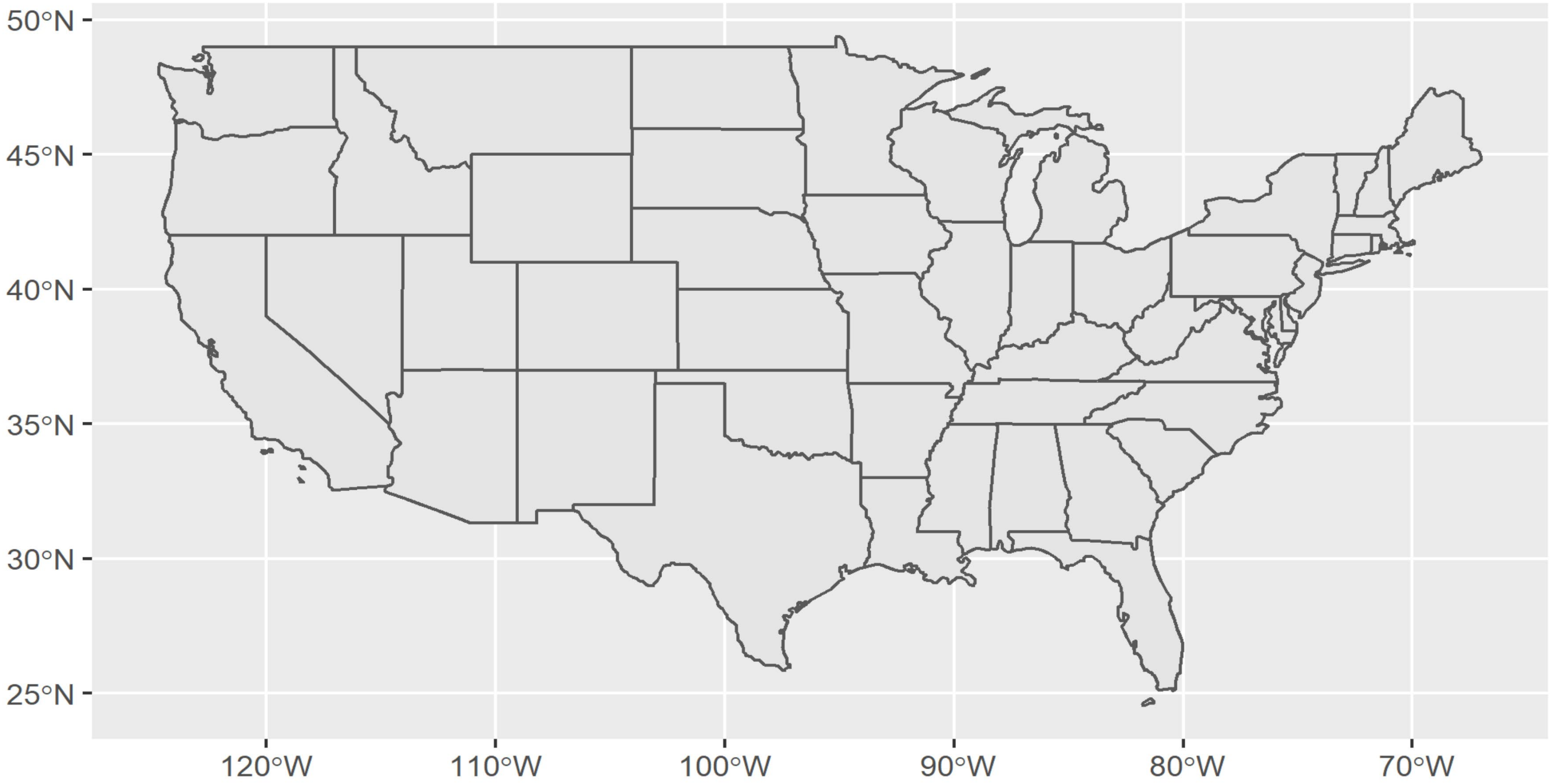
- None

3: Geometry

- Simple Feature: geom_sf

4: Coordinate System

- GCS: NAD83



Map 1: State Borders

1: Data

- CONUS

2: Aesthetics

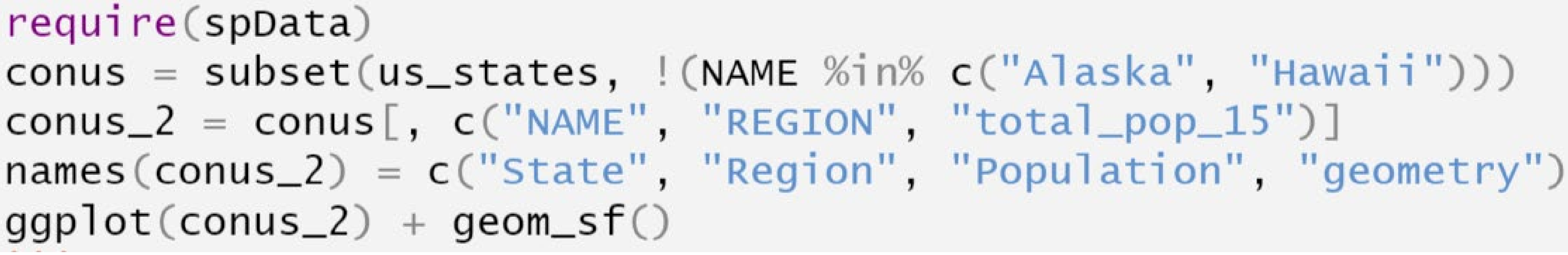
- None

3: Geometry

- Simple Feature: geom_sf

4: Coordinate System

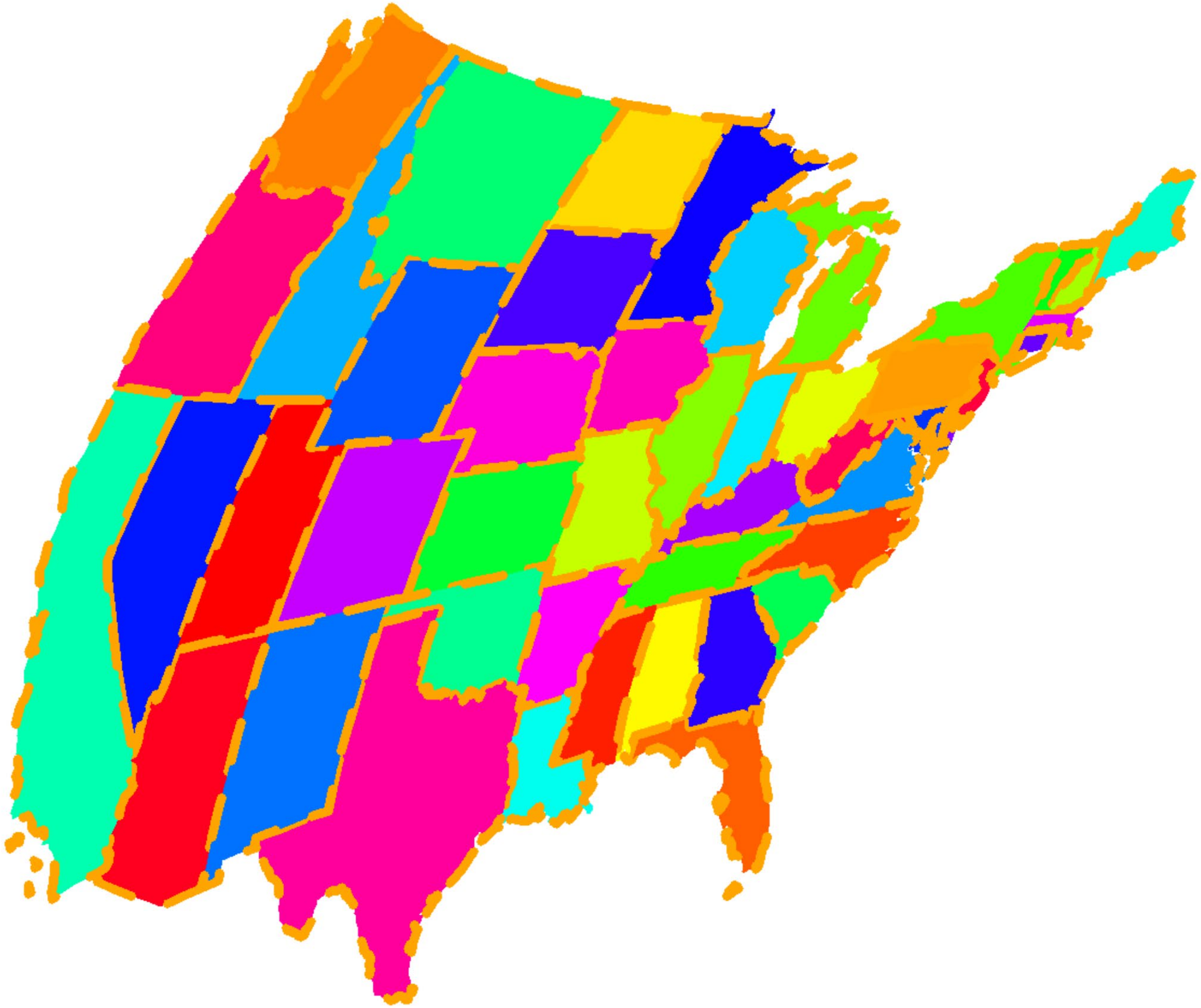
- GCS: NAD83



```
require(spData)
conus = subset(us_states, !(NAME %in% c("Alaska", "Hawaii")))
conus_2 = conus[, c("NAME", "REGION", "total_pop_15")]
names(conus_2) = c("State", "Region", "Population", "geometry")
ggplot(conus_2) + geom_sf()
```

Map 1: State Borders

- Not terrible, but pretty basic… and in an ugly CRS.
- Let’s elaborate by changing coordinate systems. We can use an Albers Equal Area Conic centered on CONUS: EPSG 5070



Map 2: State Borders

1: Data

- CONUS

2: Aesthetics

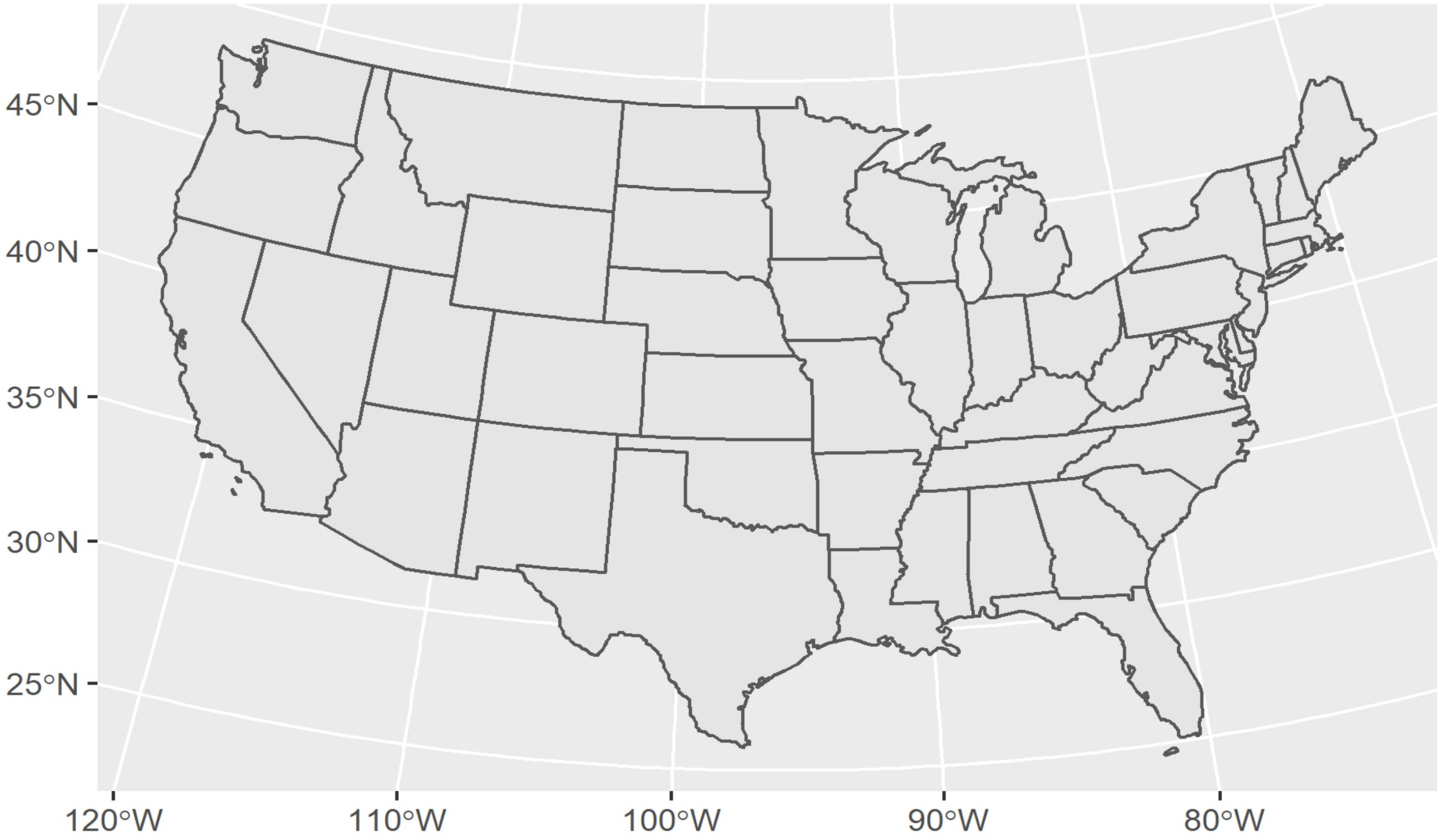
- None

3: Geometry

- Simple Feature: geom_sf

4: Coordinate System

- PCS: Equal Area



Map 2: State Borders

1: Data

- CONUS

2: Aesthetics

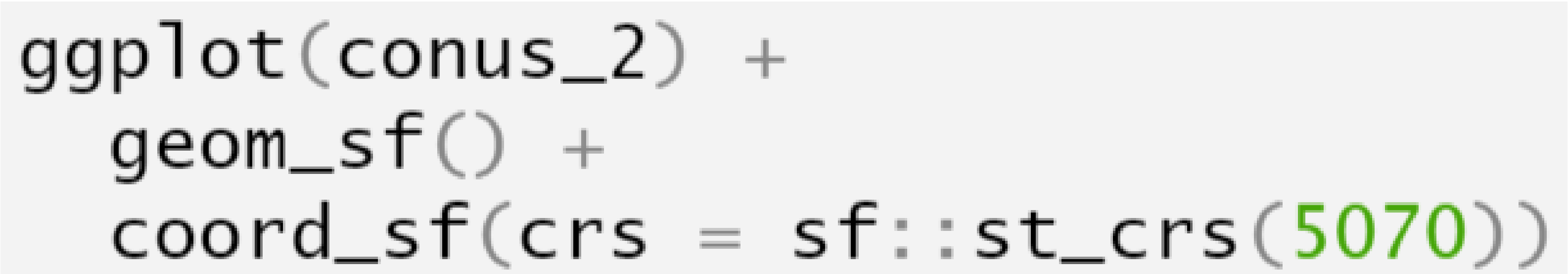
- None

3: Geometry

- Simple Feature: geom_sf

4: Coordinate System

- PCS: Equal Area



```
ggplot(conus_2) +
  geom_sf() +
  coord_sf(crs = sf::st_crs(5070))
```

Map 2: State Borders

What should we do next?

Recall the data attributes:

- State Name
- Region
- Population

Let’s color the borders by region

```
          State   Region Population
1       Alabama    South    4830620
2       Arizona     West    6641928
3      Colorado     West    5278906
4   Connecticut Norteast    3593222
5       Florida    South   19645772
6       Georgia    South   10006693
7         Idaho     West    1616547
8       Indiana  Midwest    6568645
9        Kansas  Midwest    2892987
10    Louisiana    South    4625253
```

Map 3: Regions 1

1: Data

- CONUS

2: Aesthetics

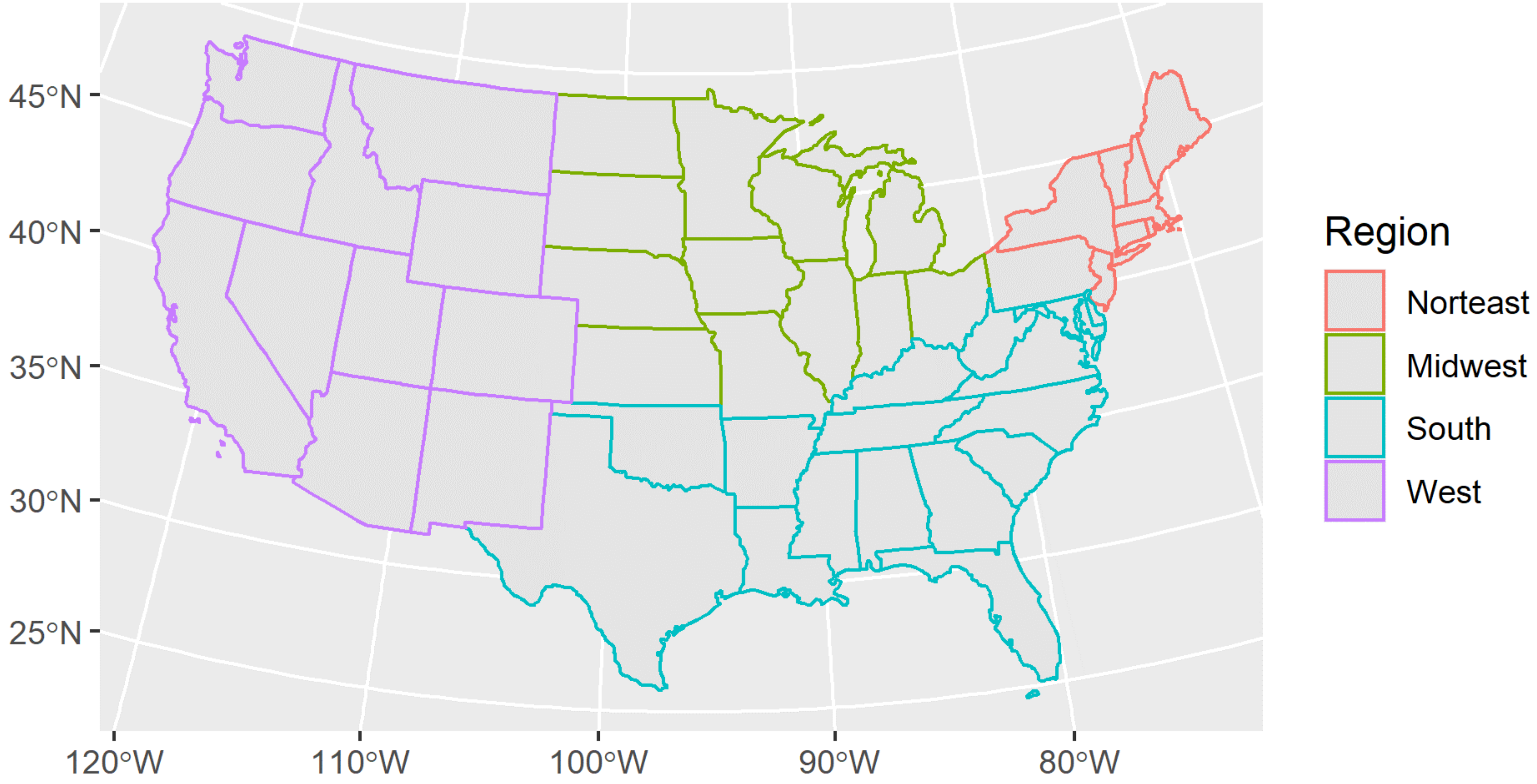
- Color: Region

3: Geometry

- Simple Feature: geom_sf

4: Coordinate System

- PCS: Equal Area



Map 3: Regions 1

1: Data

- CONUS

2: Aesthetics

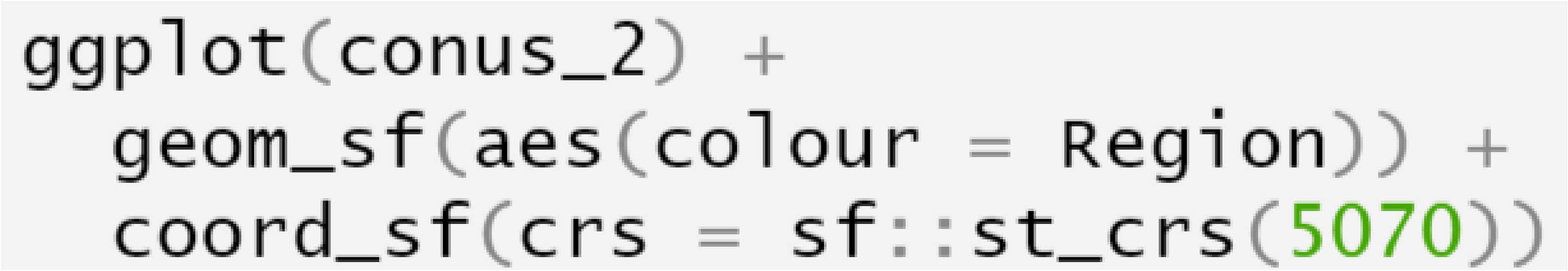
- Color: Region

3: Geometry

- Simple Feature: geom_sf

4: Coordinate System

- PCS: Equal Area



```
ggplot(conus_2) +
  geom_sf(aes(colour = Region)) +
  coord_sf(crs = sf::st_crs(5070))
```

Color and Fill

- Let’s be a fancy and change the fill color by region too.

- We can make the fill semitransparent by adjusting the alpha parameter

α

Map 4: Regions 2

1: Data

- CONUS

2: Aesthetics

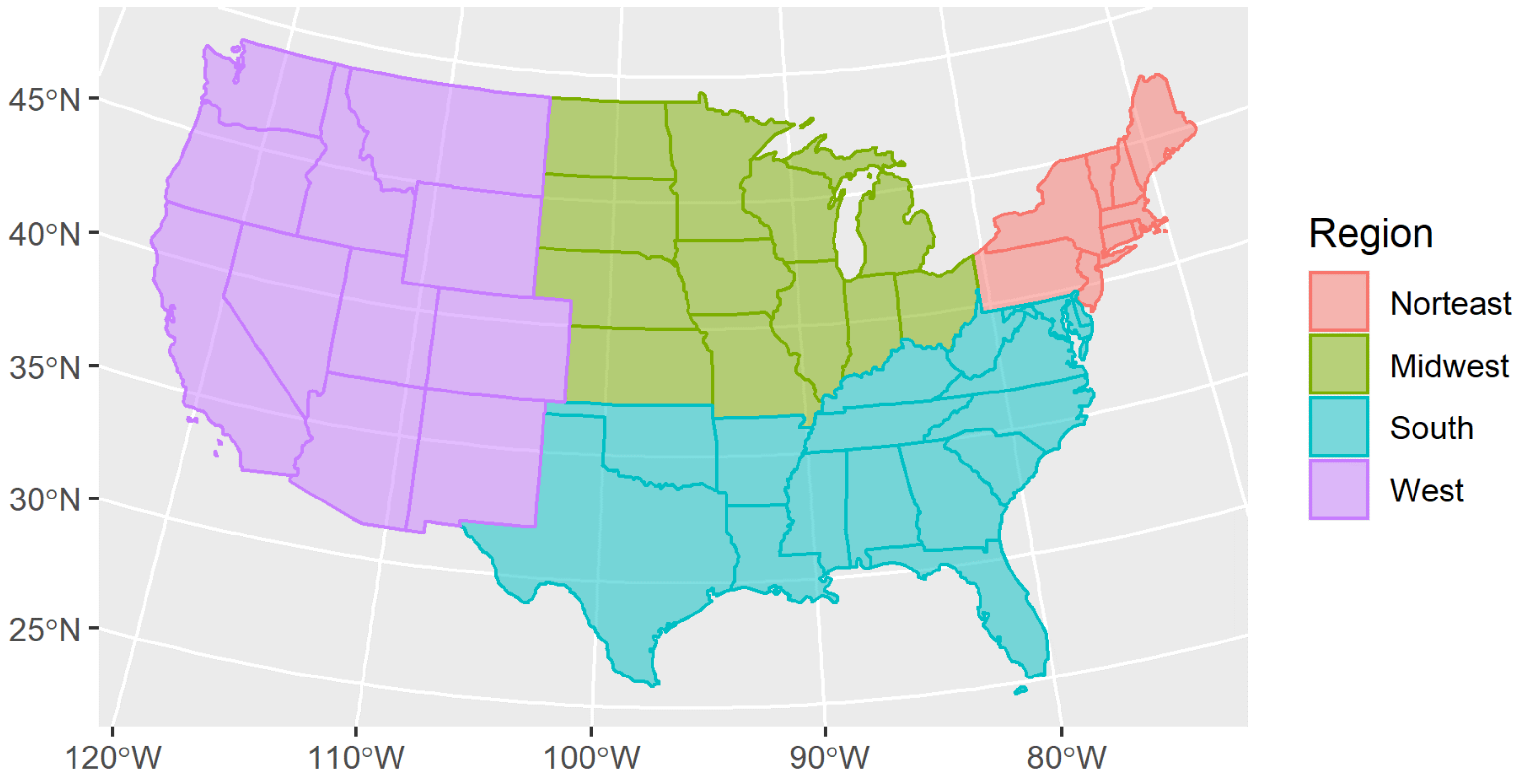
- Color: Region
- Fill: Region

3: Geometry

- Simple Feature: geom_sf

4: Coordinate System

- PCS: Equal Area



Map 4: Regions 2

1: Data

- CONUS

2: Aesthetics

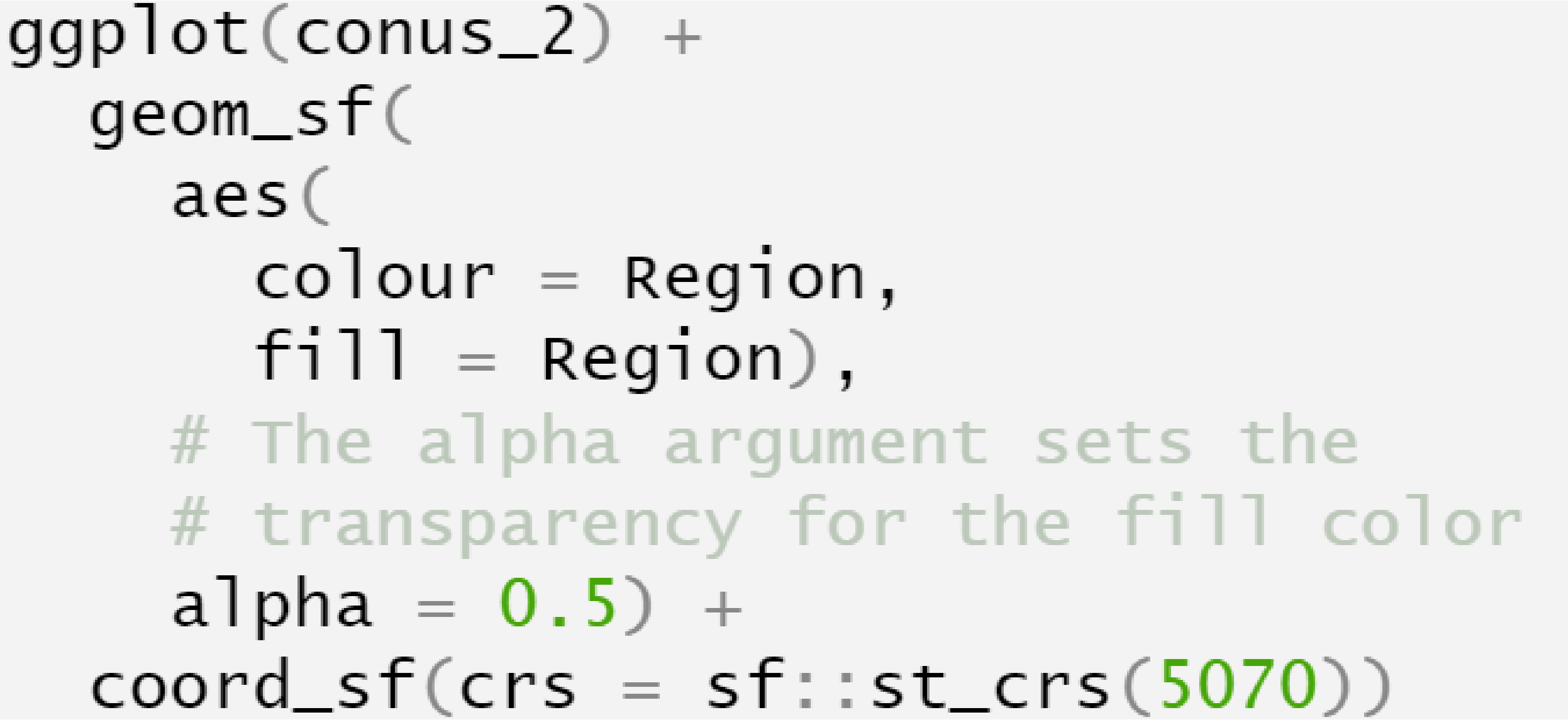
- Color: Region
- Fill: Region

3: Geometry

- Simple Feature: geom_sf

4: Coordinate System

- PCS: Equal Area



```
ggplot(conus_2) +
  geom_sf(
    aes(
      colour = Region,
      fill = Region),
    # The alpha argument sets the
    # transparency for the fill color
    alpha = 0.5) +
  coord_sf(crs = sf::st_crs(5070))
```

What About Data?

- That’s cool, but how can we use R’s data manipulation potential?

- Let’s make a choropleth from population.
- Which aesthetic do we need to use?

Map 5a: Choropleth 1

1: Data

- CONUS

2: Aesthetics

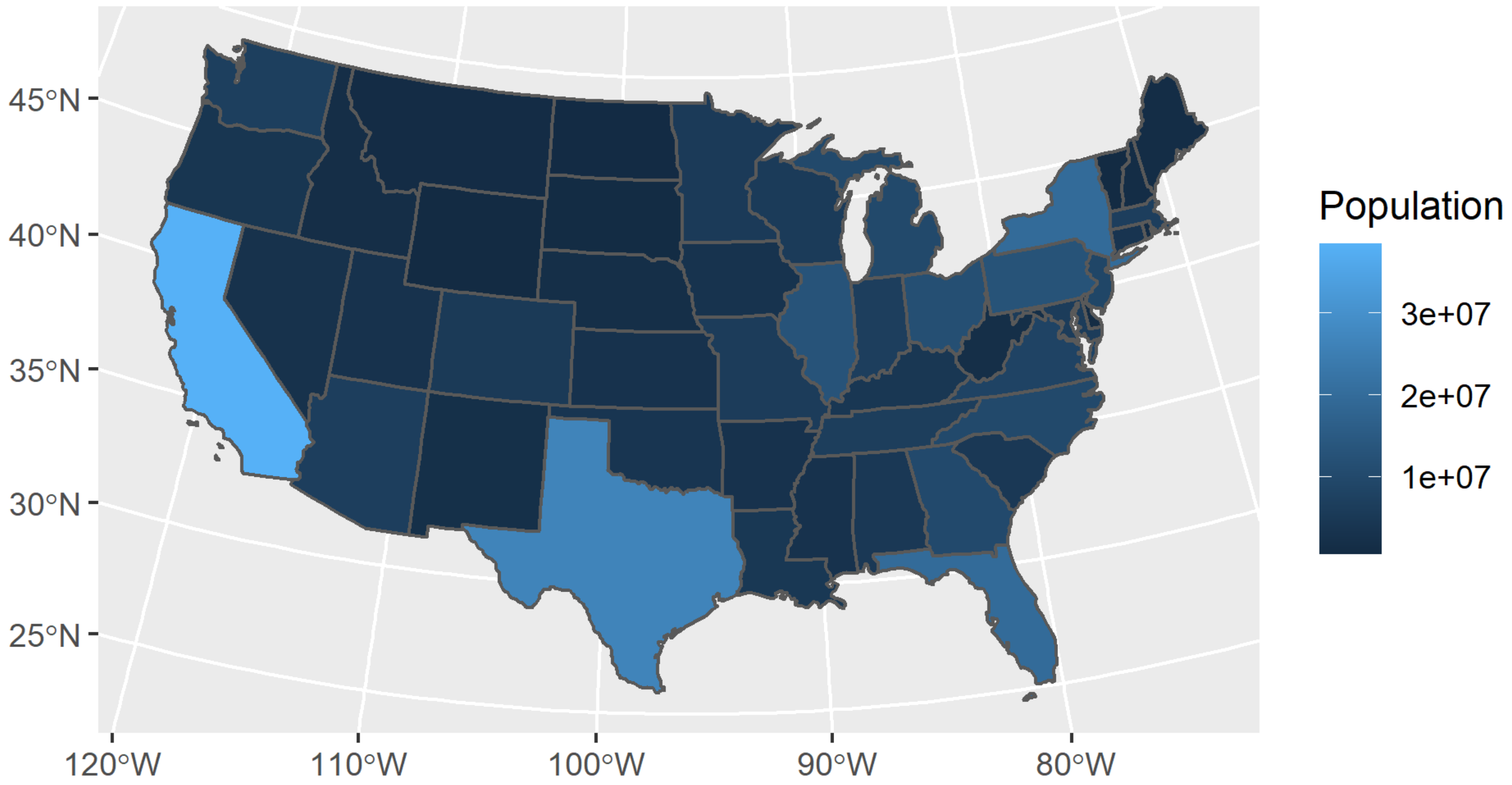
- Fill: Population (blue color scale)

3: Geometry

- Simple Feature: geom_sf

4: Coordinate System

- PCS: Equal Area



Map 5a: Choropleth 1

1: Data

- CONUS

2: Aesthetics

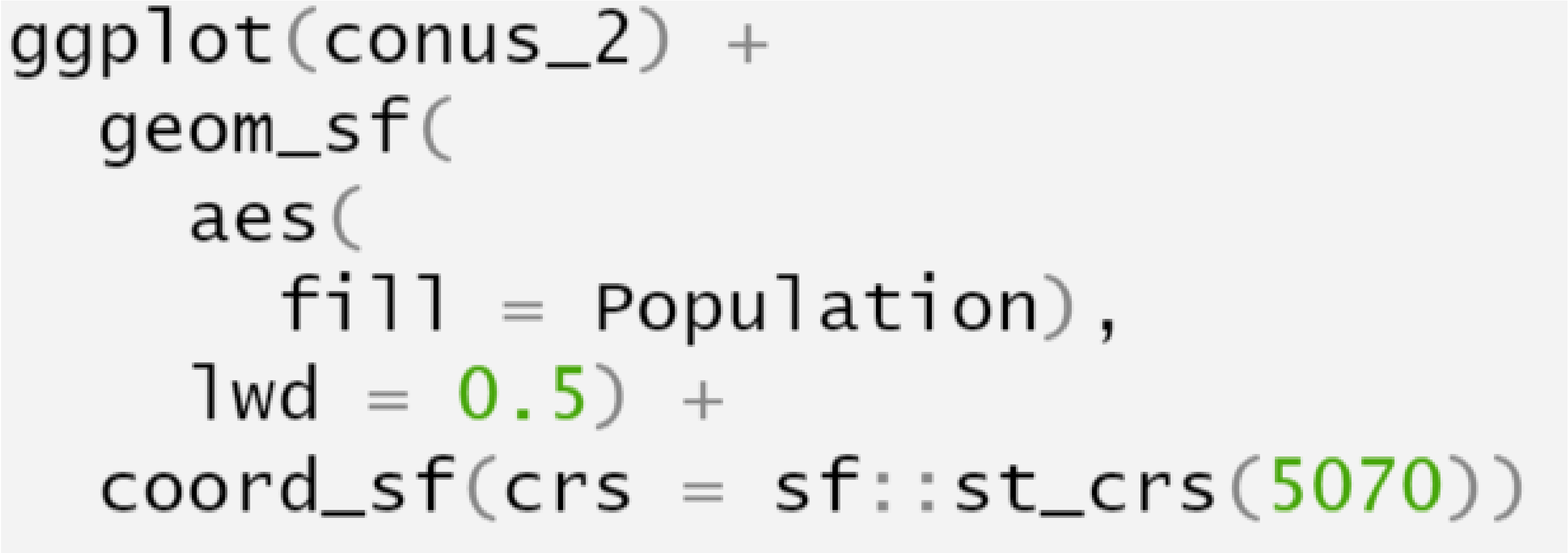
- Fill: Population (blue color scale)

3: Geometry

- Simple Feature: geom_sf

4: Coordinate System

- PCS: Equal Area



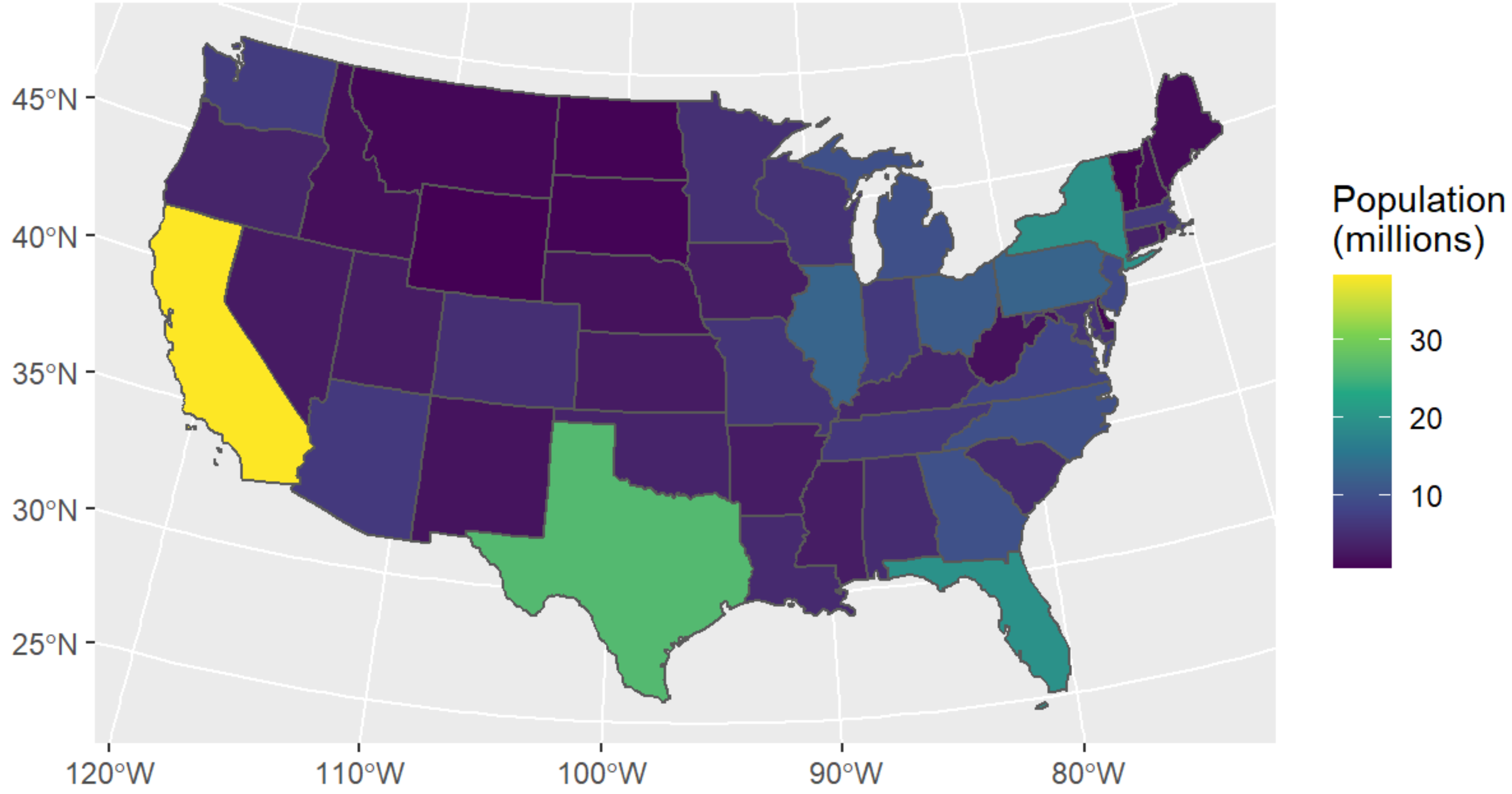
```
ggplot(conus_2) +
  geom_sf(
    aes(
      fill = Population),
    lwd = 0.5) +
  coord_sf(crs = sf::st_crs(5070))
```

Map 5b: Choropleth 1b

That's OK, but the color scale isn't the best.

We can use one of the viridis scales, which are optimized for colorblindness and grayscale.

Now, let's symbolize the region using the state border color…

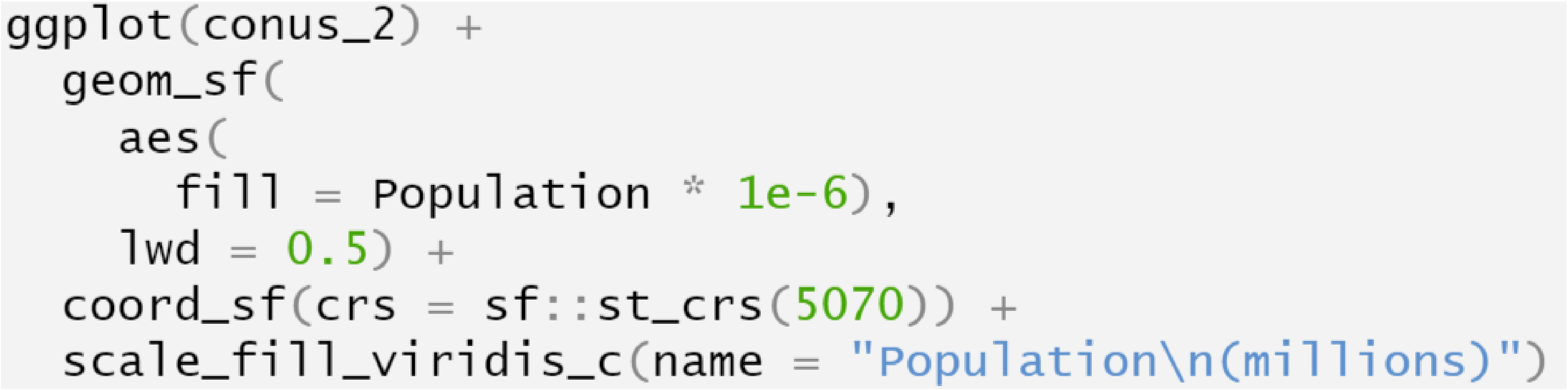


Map 5b: Choropleth 1b

That’s OK, but the color scale isn’t the best.

We can use one of the viridis scales, which are optimized for colorblindness and grayscale.

Now, let’s symbolize the region using the state border color…



```
ggplot(conus_2) +
  geom_sf(
    aes(
      fill = Population * 1e-6),
    lwd = 0.5) +
  coord_sf(crs = sf::st_crs(5070)) +
  scale_fill_viridis_c(name = "Population\n(millions)")
```

Map 6: Choropleth 2

1: Data

- CONUS

2: Aesthetics

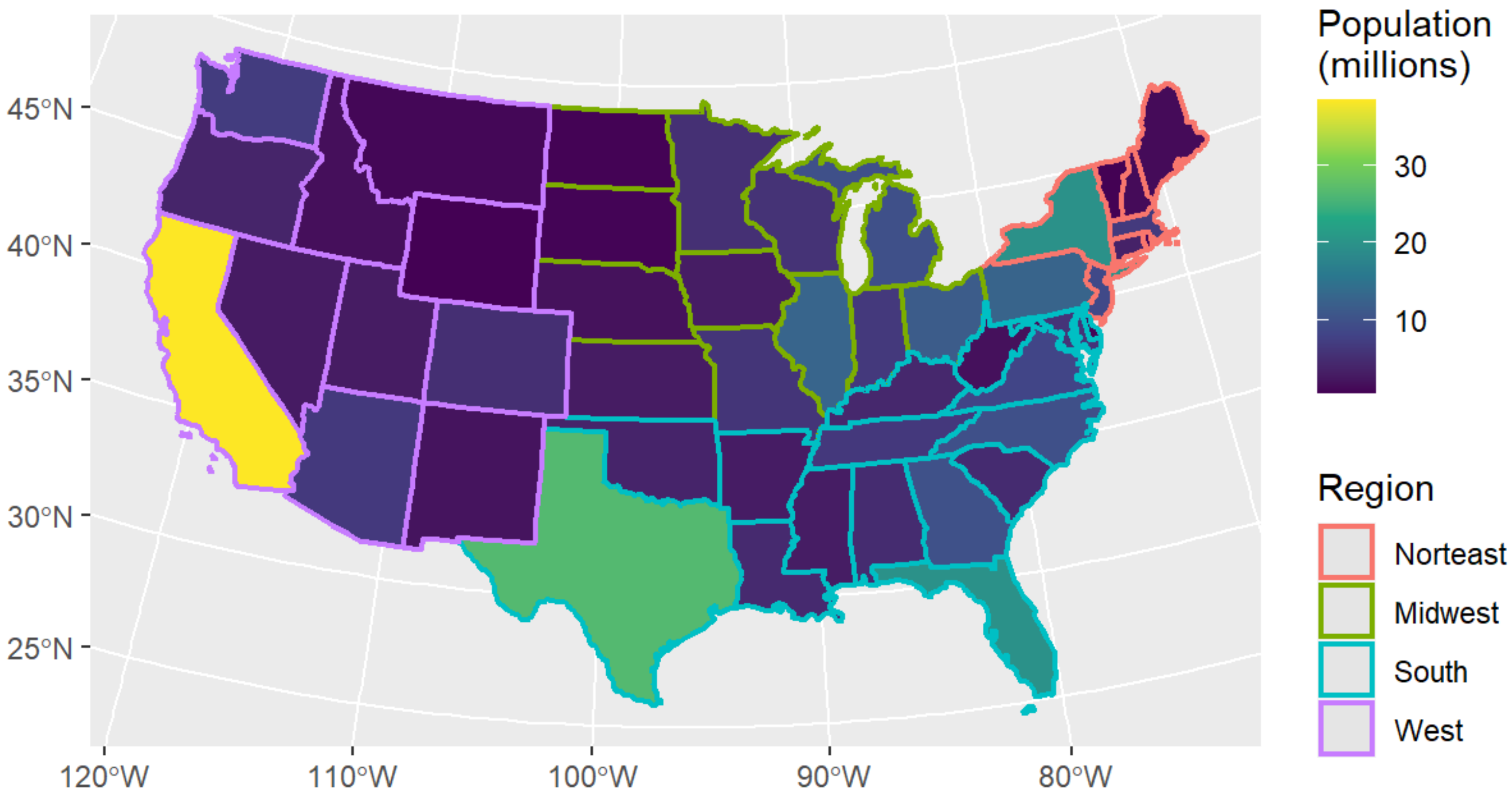
- Fill: Population
  - Viridis scale ‘c’
- Color: Region

3: Geometry

- Simple Feature: geom_sf

4: Coordinate System

- PCS: Equal Area



Map 6: Choropleth 2

1: Data

- CONUS

2: Aesthetics

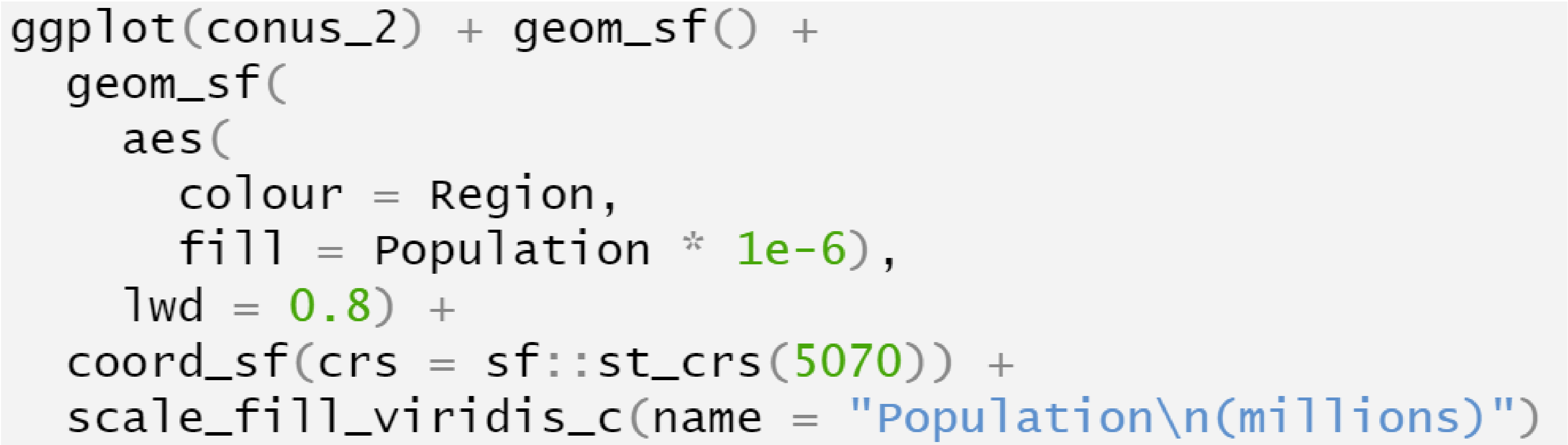
- Fill: Population
  - Viridis scale ‘c’
- Color: Region

3: Geometry

- Simple Feature: geom_sf

4: Coordinate System

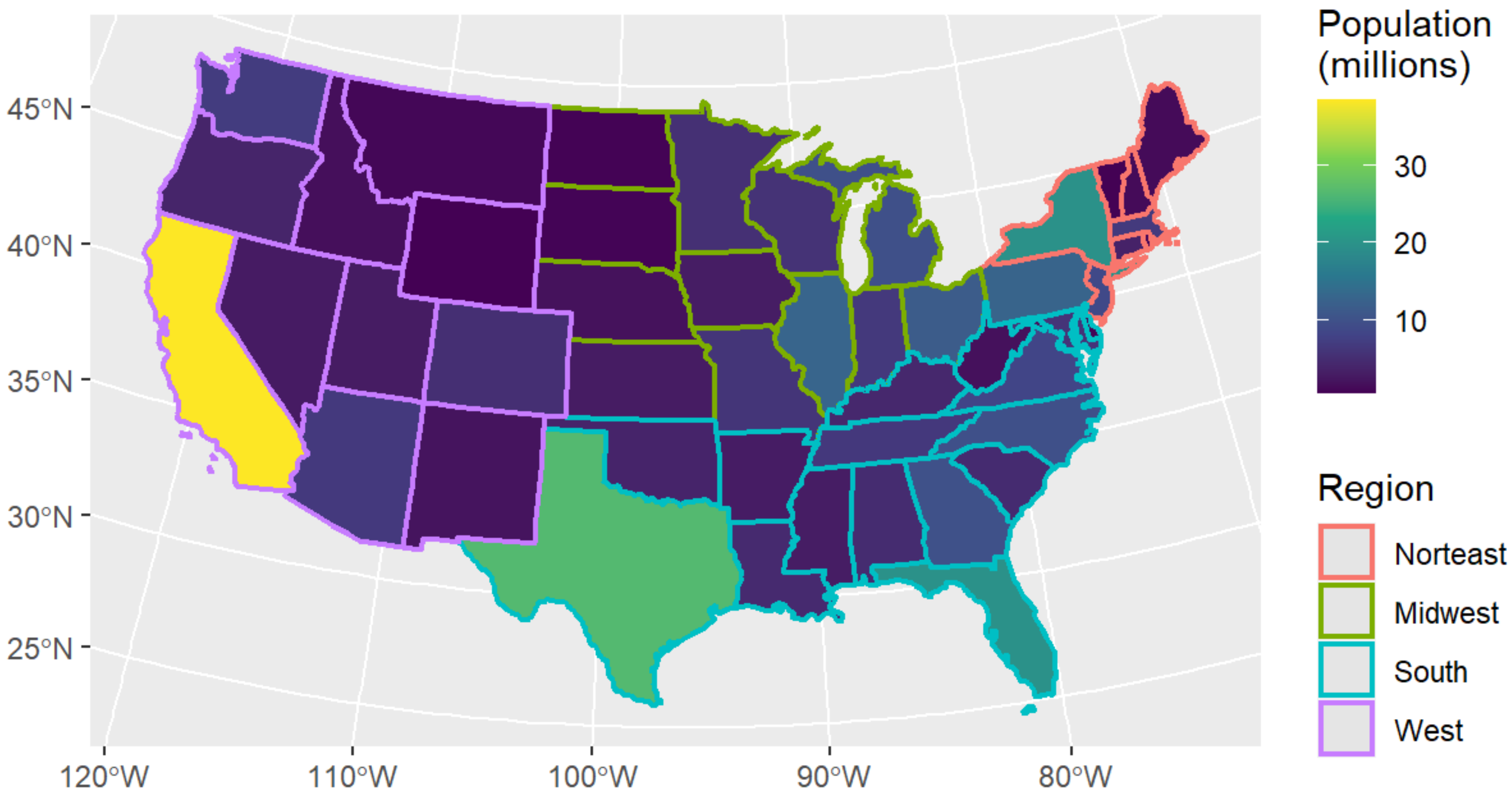
- PCS: Equal Area



```
ggplot(conus_2) + geom_sf() +
  geom_sf(
    aes(
      colour = Region,
      fill = Population * 1e-6),
    lwd = 0.8) +
  coord_sf(crs = sf::st_crs(5070)) +
  scale_fill_viridis_c(name = "Population\n(millions)")
```

Map 6: Choropleth 2

What's the viridict on this map?



Niche Concepts

Niche Concepts

- Abiotic and biotic factors
- Fundamental niche
- Niche breadth
- Niche conservatism
- Realized niche
- Source, sink habitats

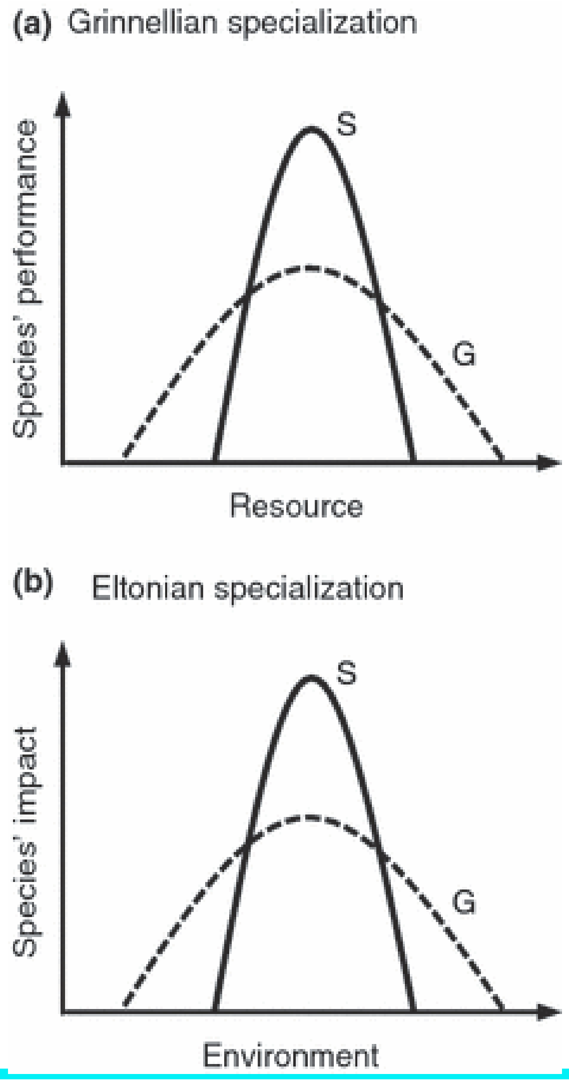


Figure 1 in Devictor et al, 2010

Definition of Grinnellian vs. Eltonian specialization. (a) The Grinnellian specialization of a given species can be described by its variance in performance across a given range of resources. For a given mean performance, the dashed line describes the performance of a generalist species (generalist, G) and the solid line of a more specialist species (specialist, S). (b) Eltonian specialization is defined as the variance in the species' impact (instead of performance) on the environment. For a given mean impact, the species' impact can be distributed through a large part of the environment (G) or be more restricted (S).

Eltonialn and Grinnellian Niches

Grinnellian + Hutchinsonian	**Eltonian**
• “scenopoetic” • Response of species to abiotic factors and biotic resources • Water, food, favorable temperatures, etc. • “…hypervolume in multidimensional space...”	• “bionomic” • Biotic interactions allow persistence • Competition, predator/prey interactions, etc. • Focus on species interactions and impacts

Data for Species Distribution Models

Presence, Absence, and Abundance

- Presence: Species was detected at the site
- Absence: Failed to detect species at the site
  - Can be a true absence or pseudoabsence
  - Detection
- Abundance: X individuals of the specie were observed at the site.
  - Contains more information than presence/absence

Absences

True Absences

- Observations of absences, as part of a planned experiment.
- Subject to detection errors

Pesudoabsences

- Background points
- Randomly generated
  - CSR
  - Grid-based/stratified methods

Species Distribution Modeling Paradigms

What is a Species Distribution Model?

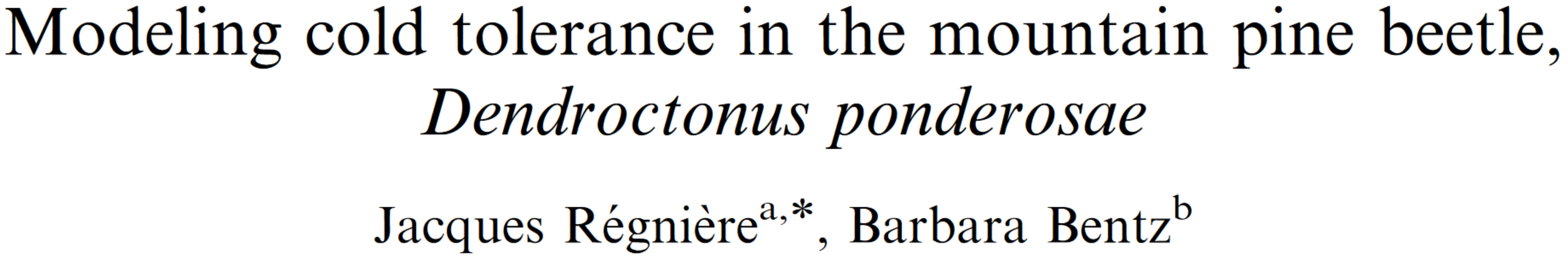
- “The principle of SDM is to **relate known locations of a species with the environmental characteristics** of these locations in order to estimate the response function and contribution of environmental variables [12], and **predict the potential geographical range of a species** [13]. These models estimate the fundamental ecological niche in the environmental space (*i.e.* species response to abiotic environmental factors [14]) and **project it onto the geographical space** to derive the **probability of presence for any given area** or, depending on the method, the likelihood that specific environmental conditions are suitable for the target species [15].”

- From Forcade et al., 2014: Mapping Species Distributions with MAXENT Using a Geographically Biased Sample of Presence Data: A Performance Assessment of Methods for Correcting Sampling Bias

Mechanistic and Phenomenological Models

Mechanistic	**Phenomenological**
• Attempts to describe species distribution based on: • Physiological tolerances • Resource availability • Tries to explicitly model resource use and interactions • Generally more difficult	• Uses characteristics of locations where species is present (or absent). • Does not attempt to explain ‘why’. • Most species distribution models are phenomenological

Mechanistic Model Example: Mountain Pine Beetles

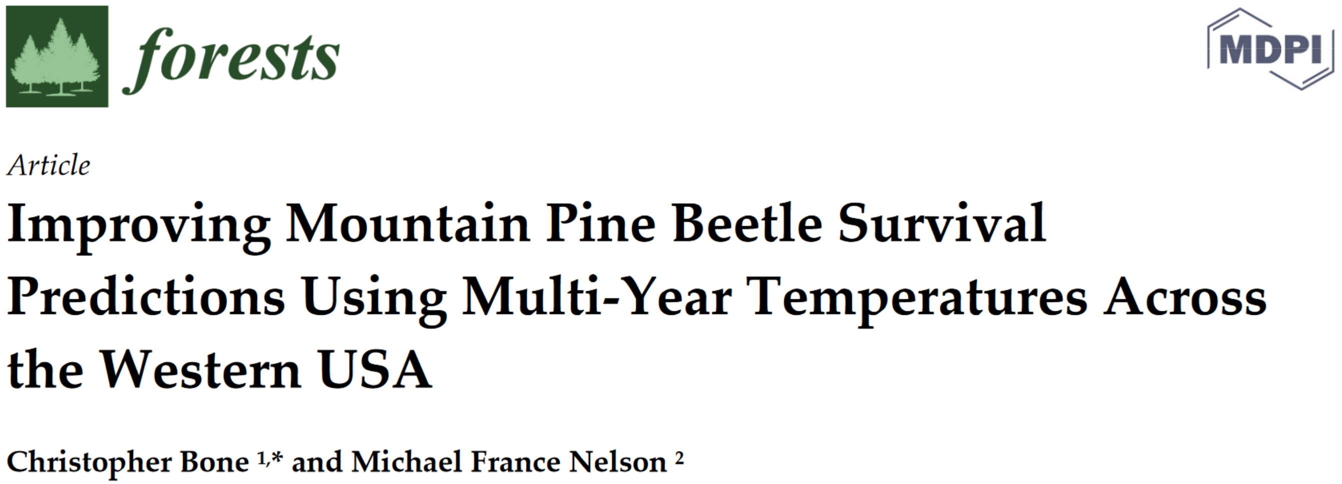


Modeling cold tolerance in the mountain pine beetle, *Dendroctonus ponderosae*

Jacques Régnière[a,*], Barbara Bentz[b]

- Model updates physiological state of overwintering larval beetles based on daily temperature minima and maxima: 1 – summer state, 2 – fall/spring state, 3 – deep winter state.

Physiological Model Example



forests

MDPI

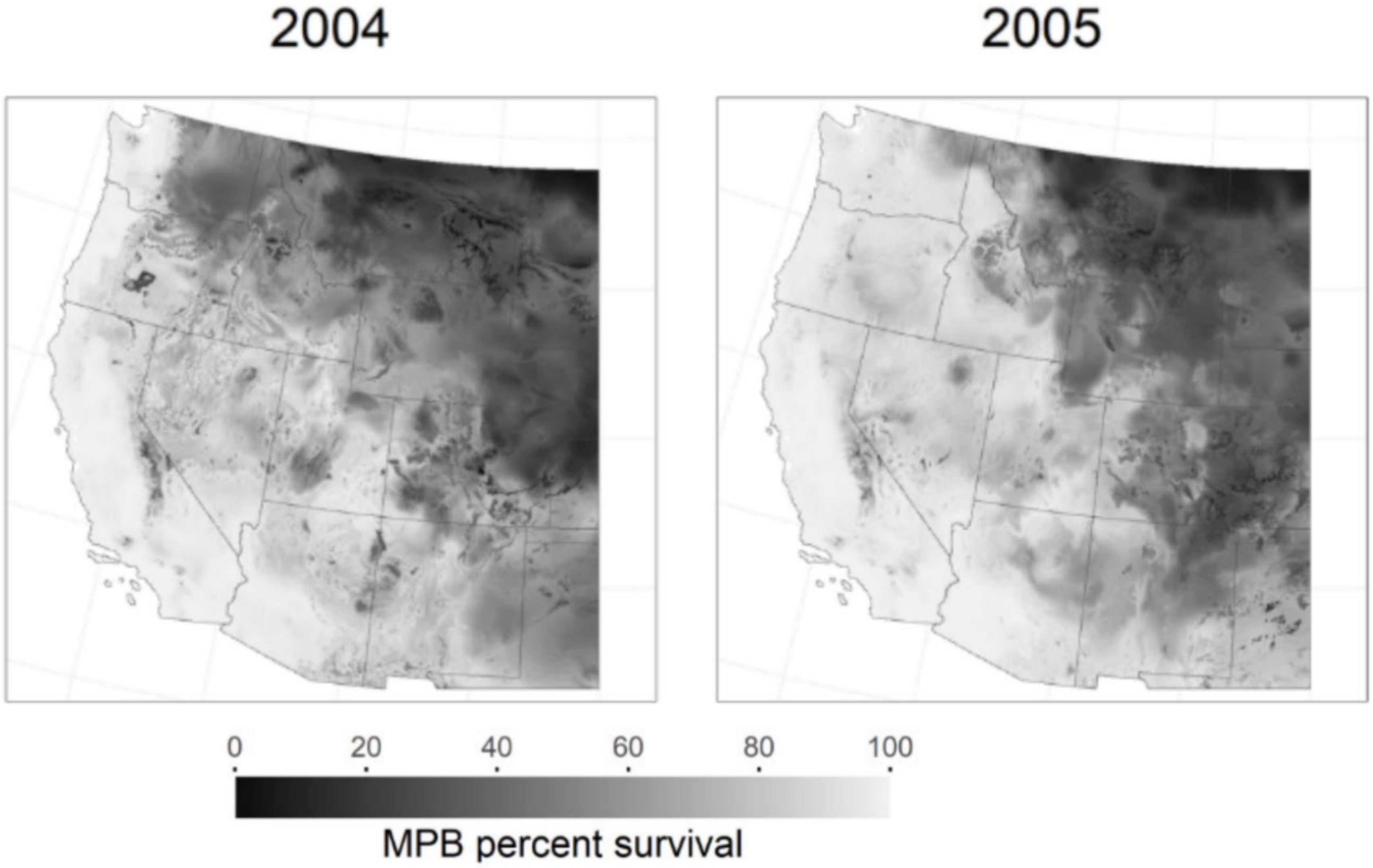
Article

Improving Mountain Pine Beetle Survival Predictions Using Multi-Year Temperatures Across the Western USA

Christopher Bone [1,*] and Michael France Nelson [2]

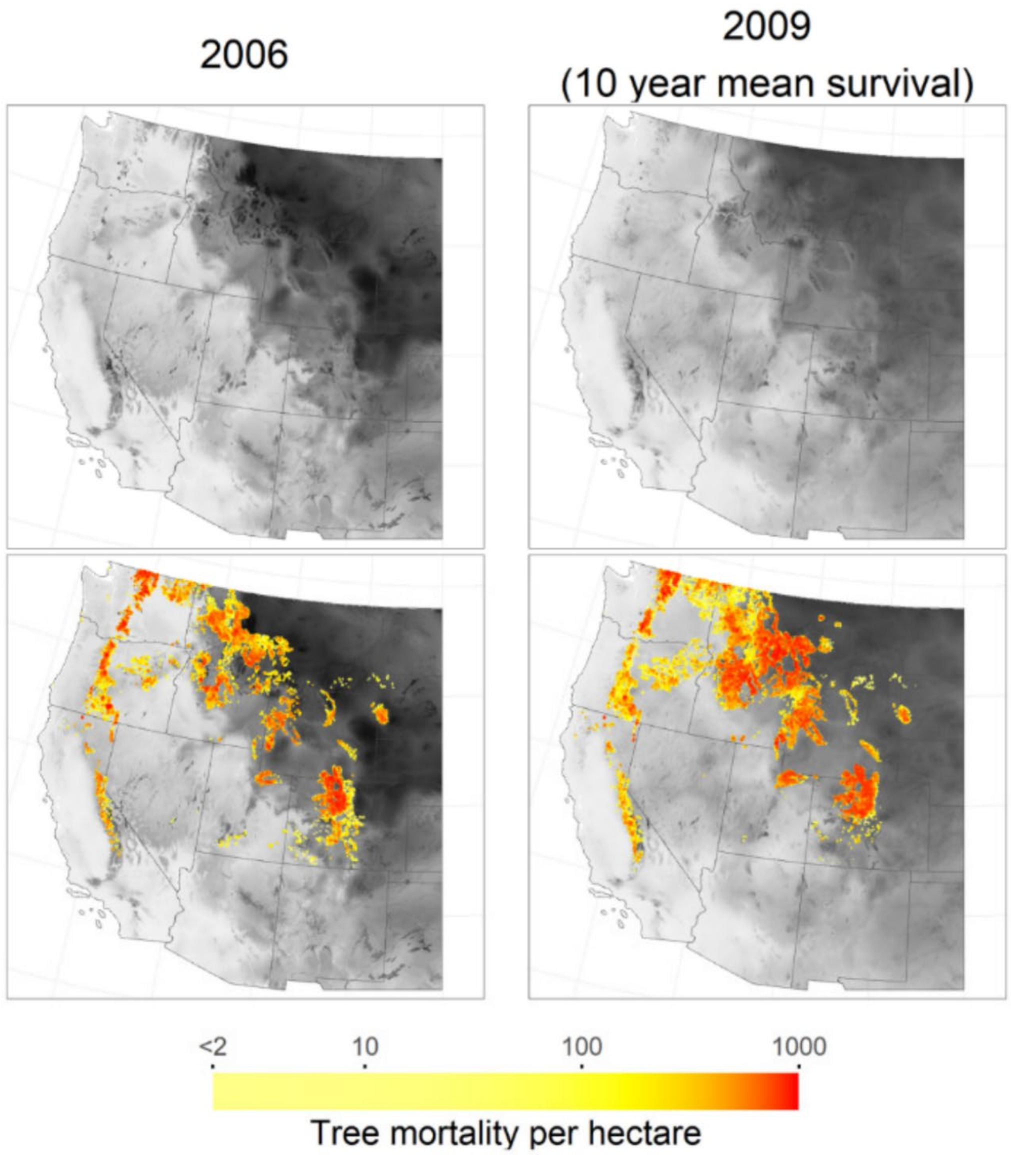
- We used Regniere + Bentz model to create spatial estimates of overwinter MPB survival for specific winters.
- Multi-winter average survivals used to model pine tree death

Physiological Model Example

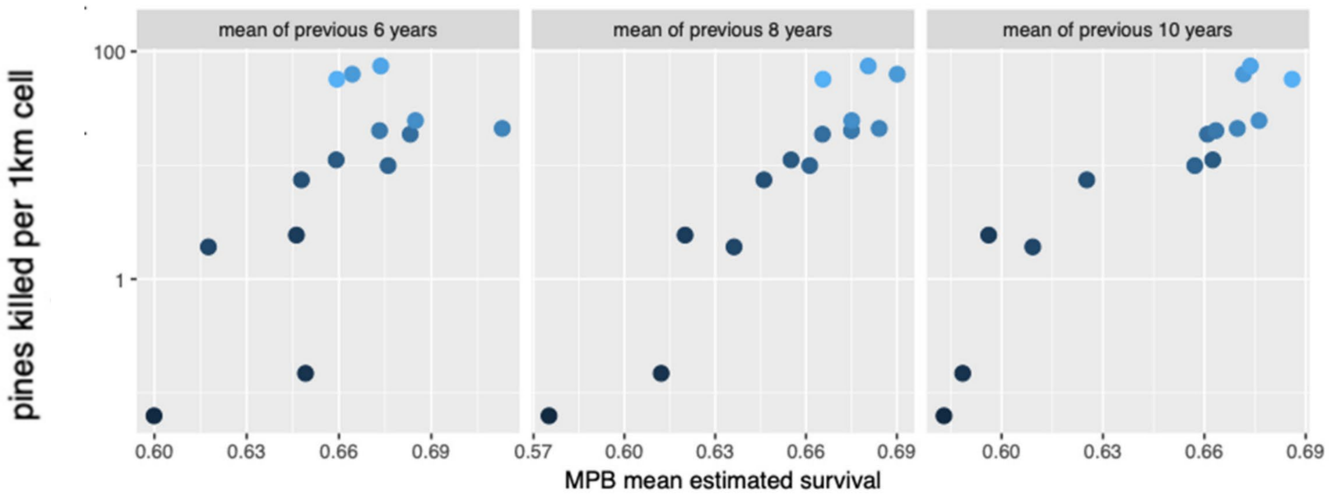


Note the inter-year differences

Physiological Model Example



We used mean multi-year survival as a proxy for beetle population



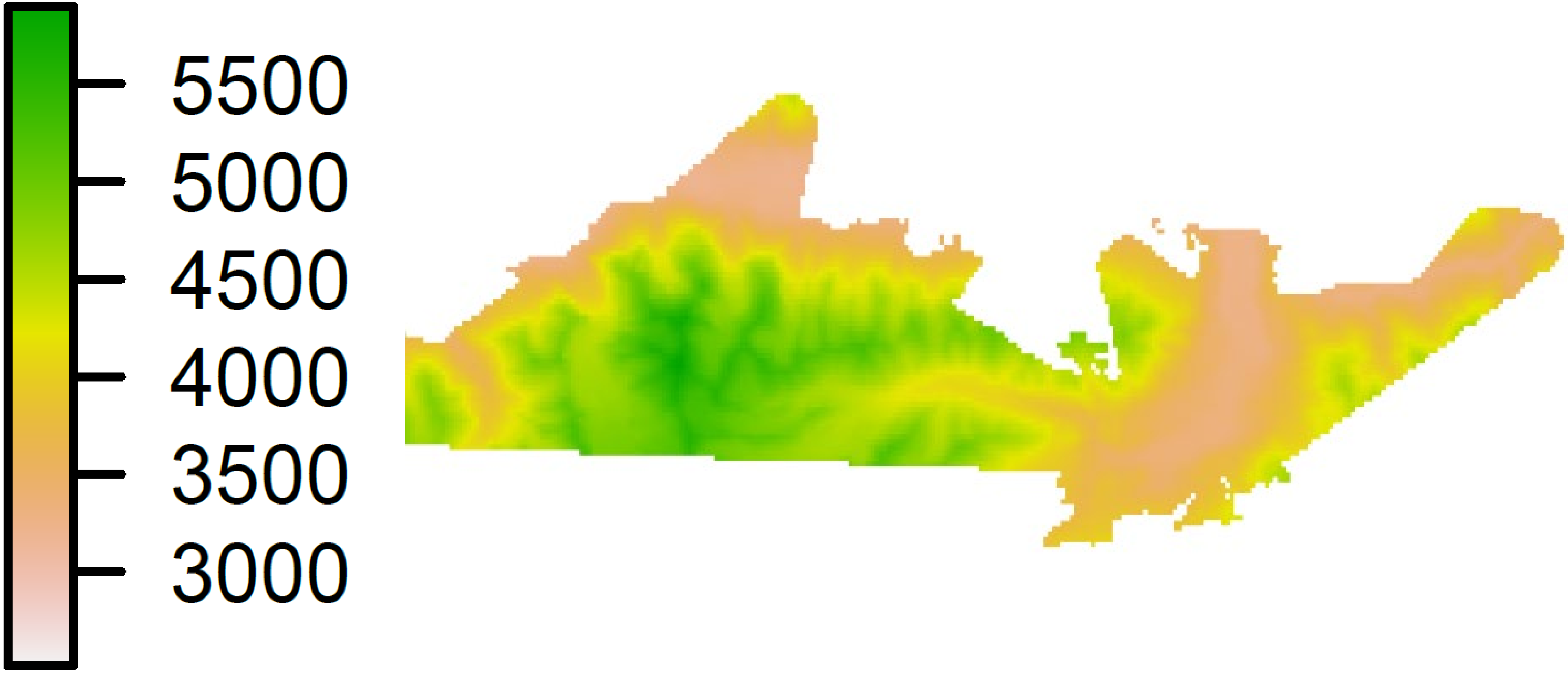
Envelope Approaches

- Profile methods
- Generally constructed from presence-only data
- Not generally mechanistic, but could be informed by physiological limits on temperature, water availability etc.

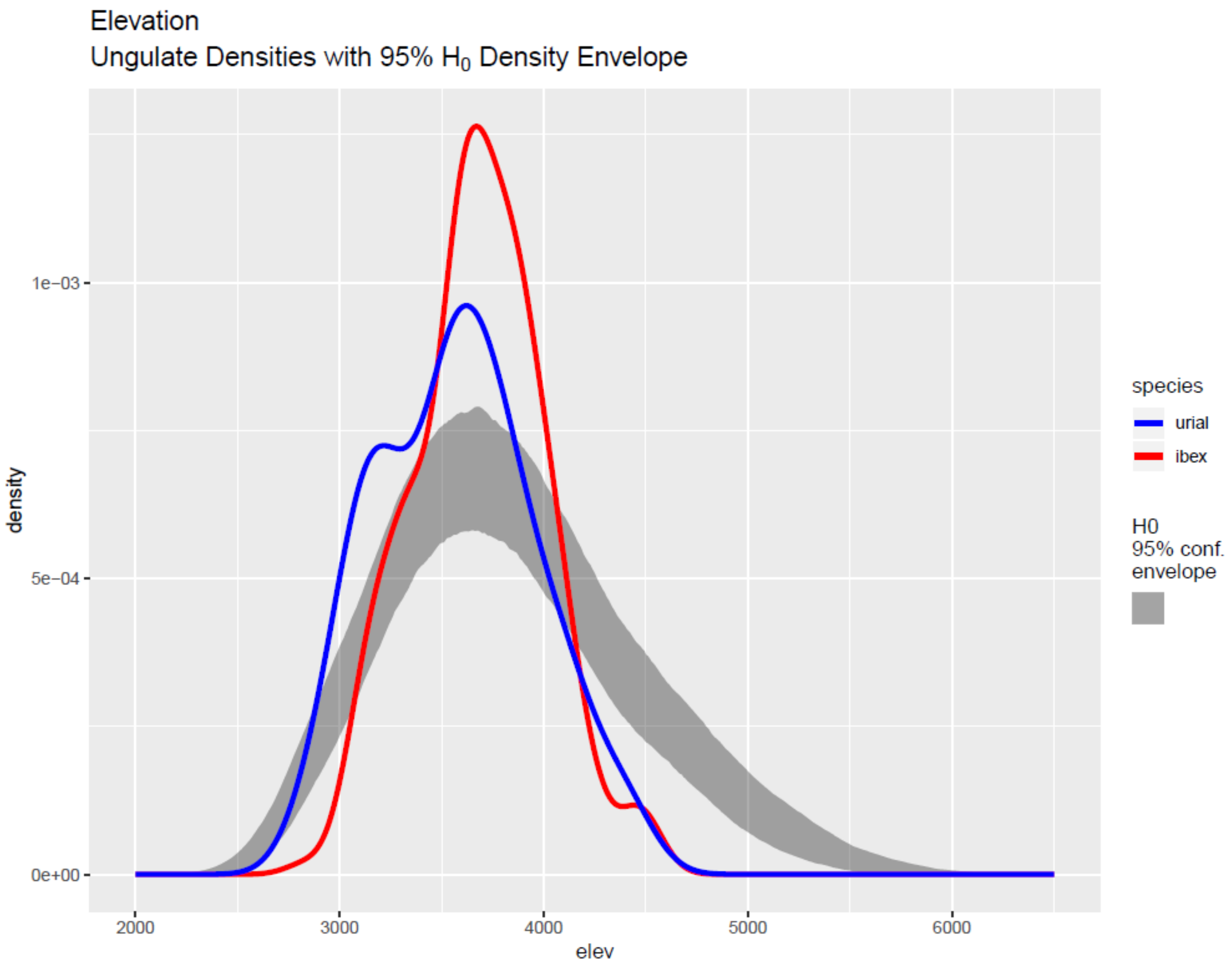
Envelope Example: Ungulate Species

- NOTE: this research is in progress, results are not final.
- Approach:
  - Field surveys of herd locations of 2 species
  - Constructed concave hull around herd locations
  - Calculated density curve for observed locations for variables:
    - Elevation, aspect, slope, roughness
    - Distance to human disturbance
  - Calculated 95% null envelope for randomly selected background points

Elevation

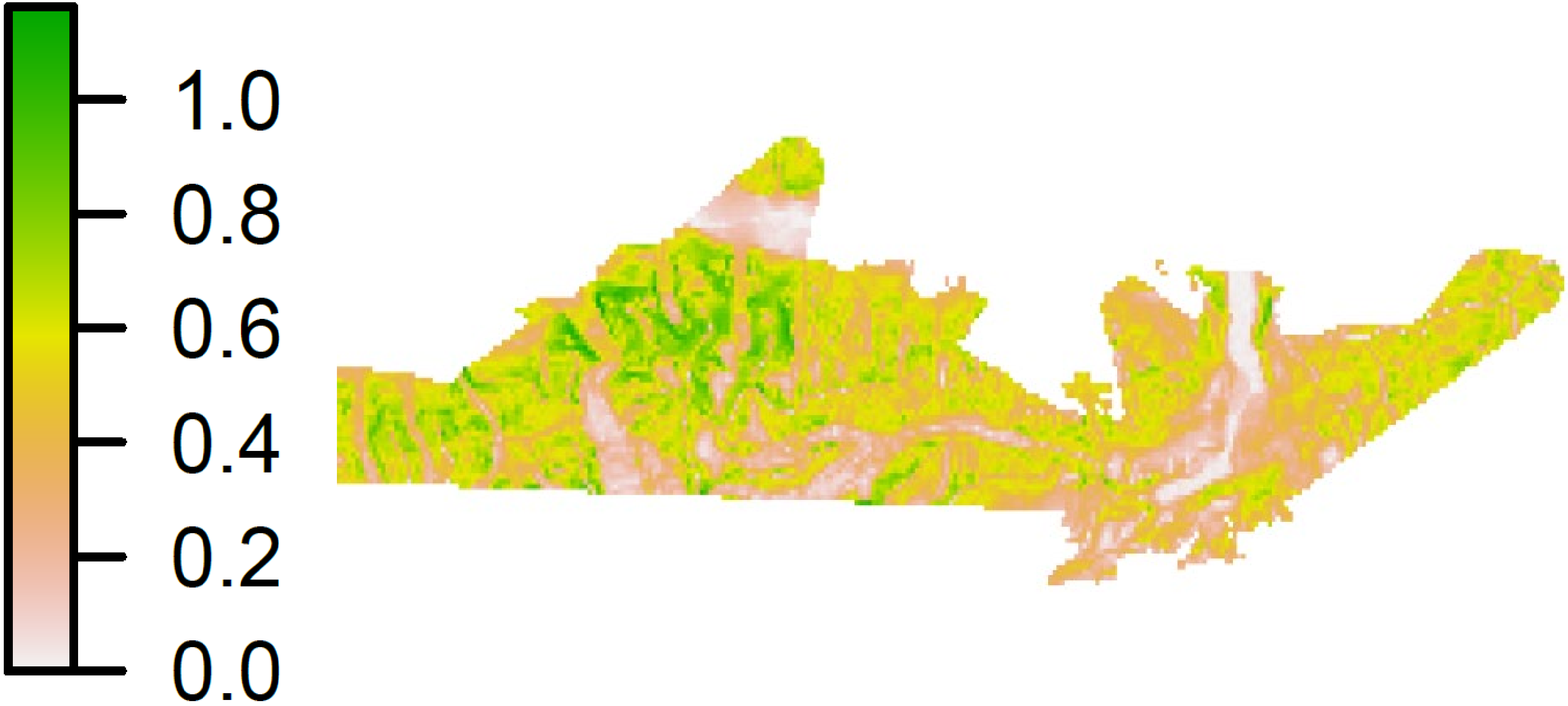


Elevation



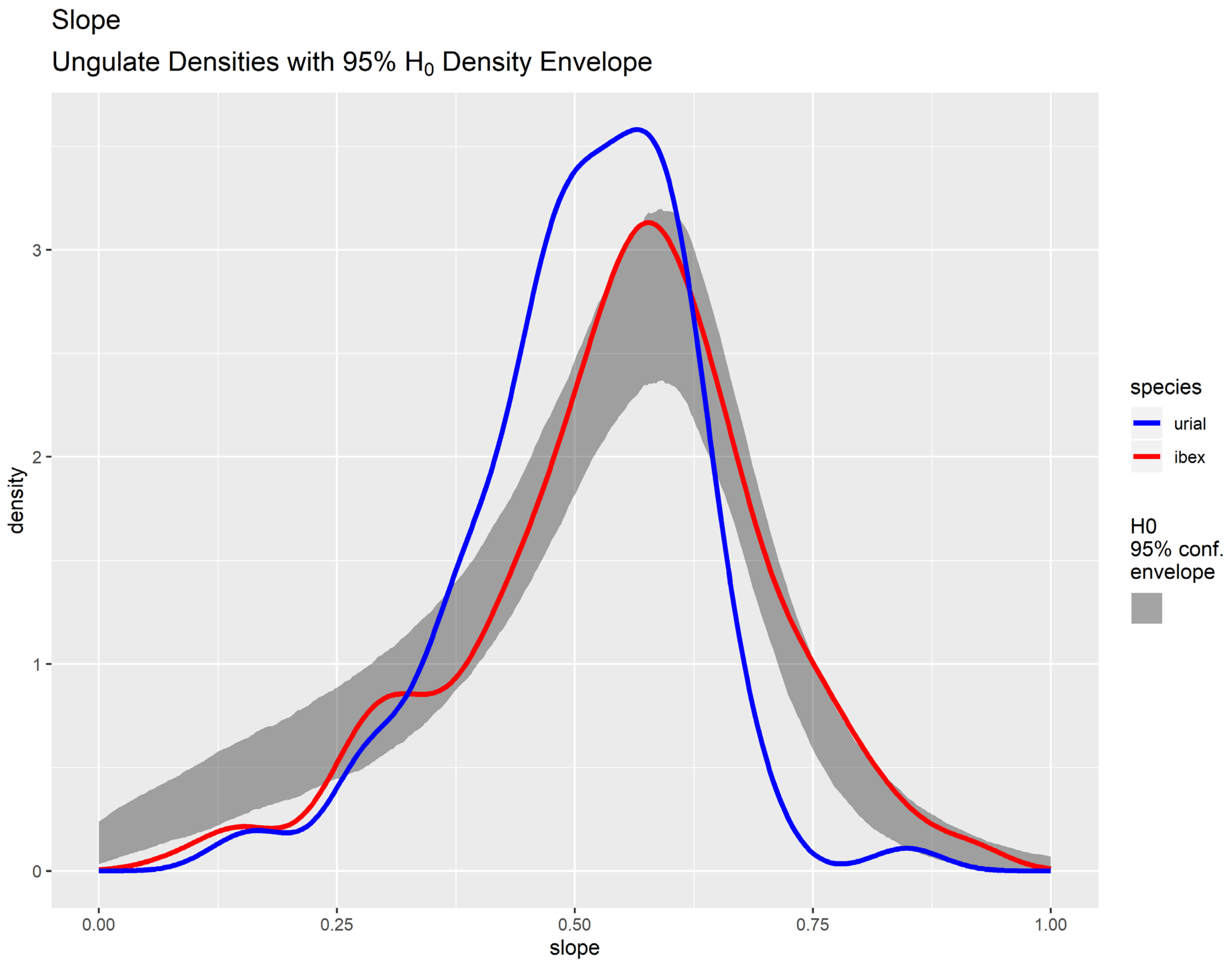
- Both species occur at mid-elevations, Ibex has more restricted range of elevations

Slope

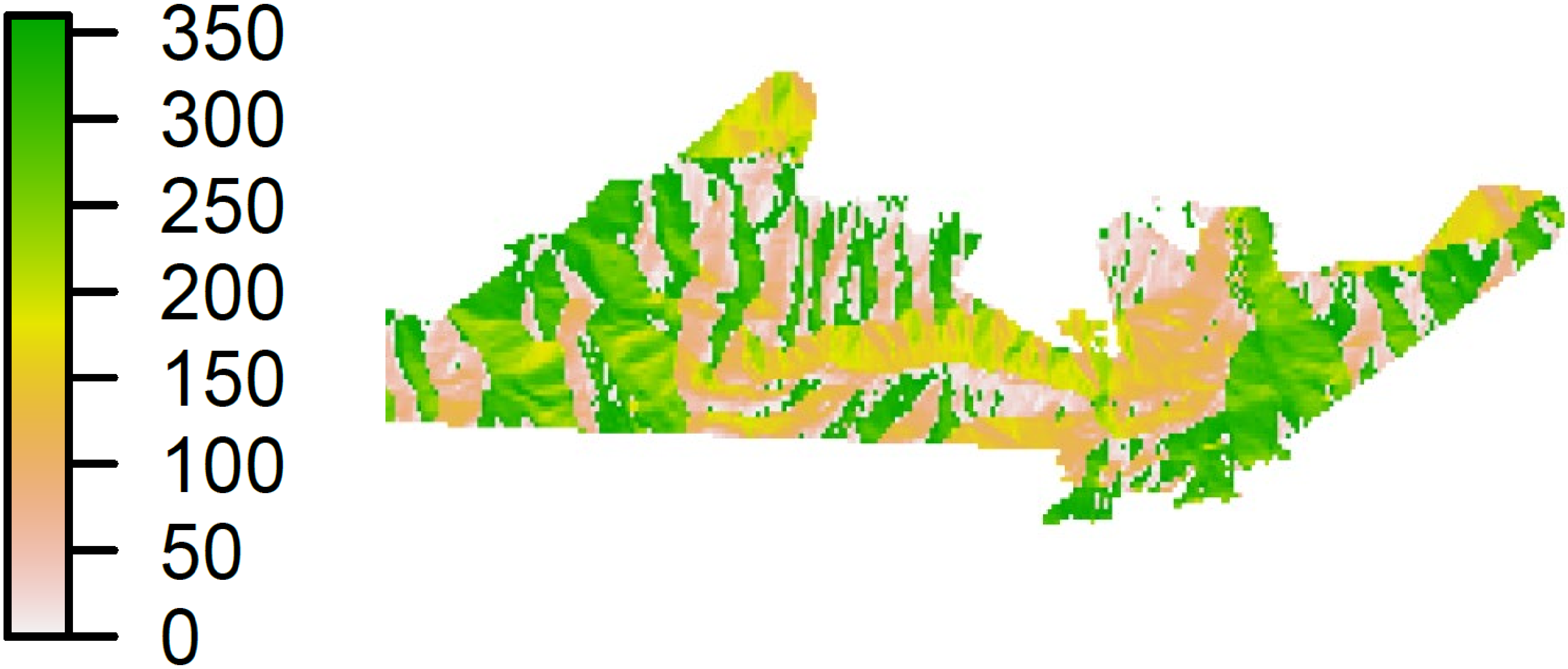


Slope

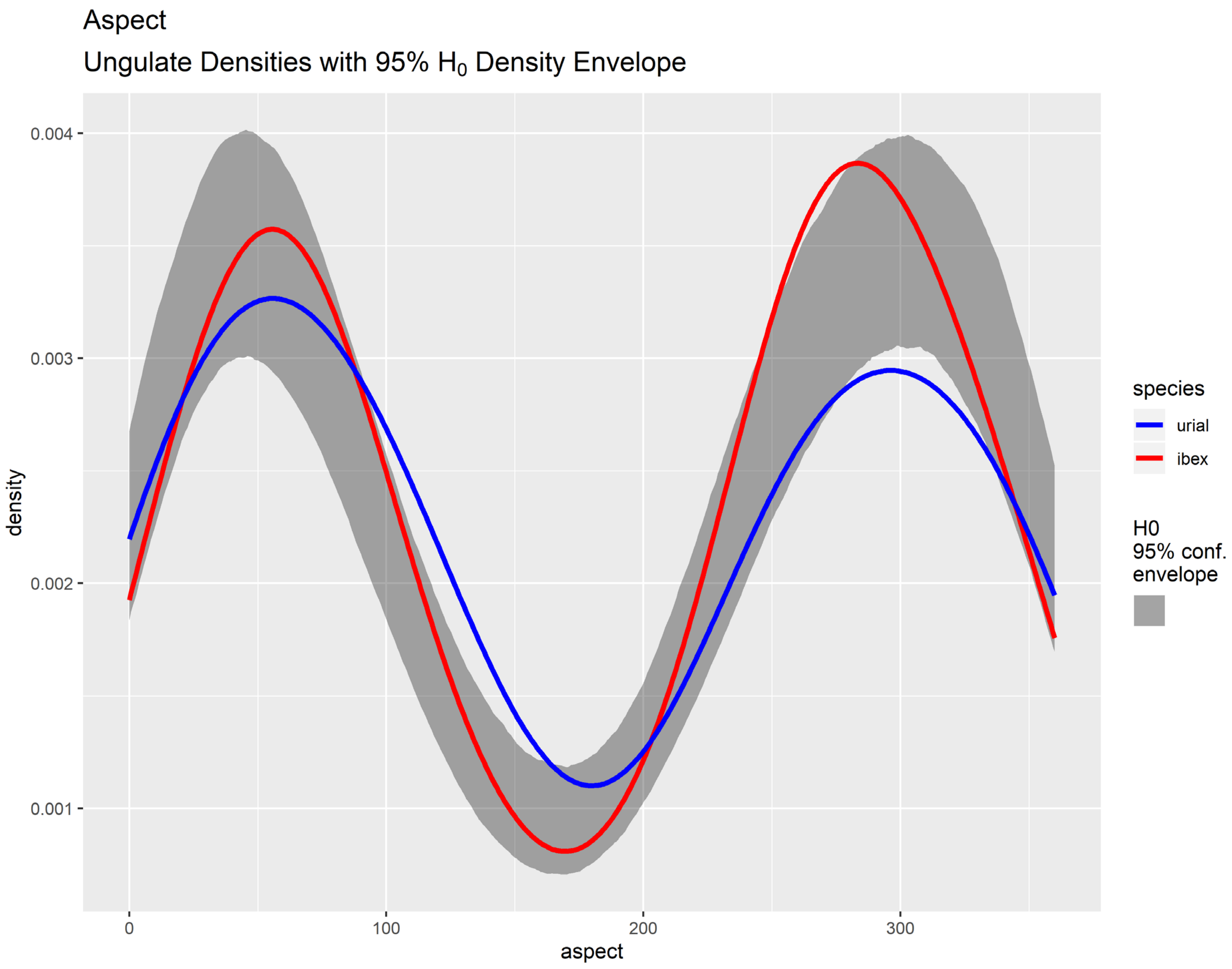
- Urial 'prefers' lower slopes
- Ibex distributes itself randomly with respect to slope.



Aspect

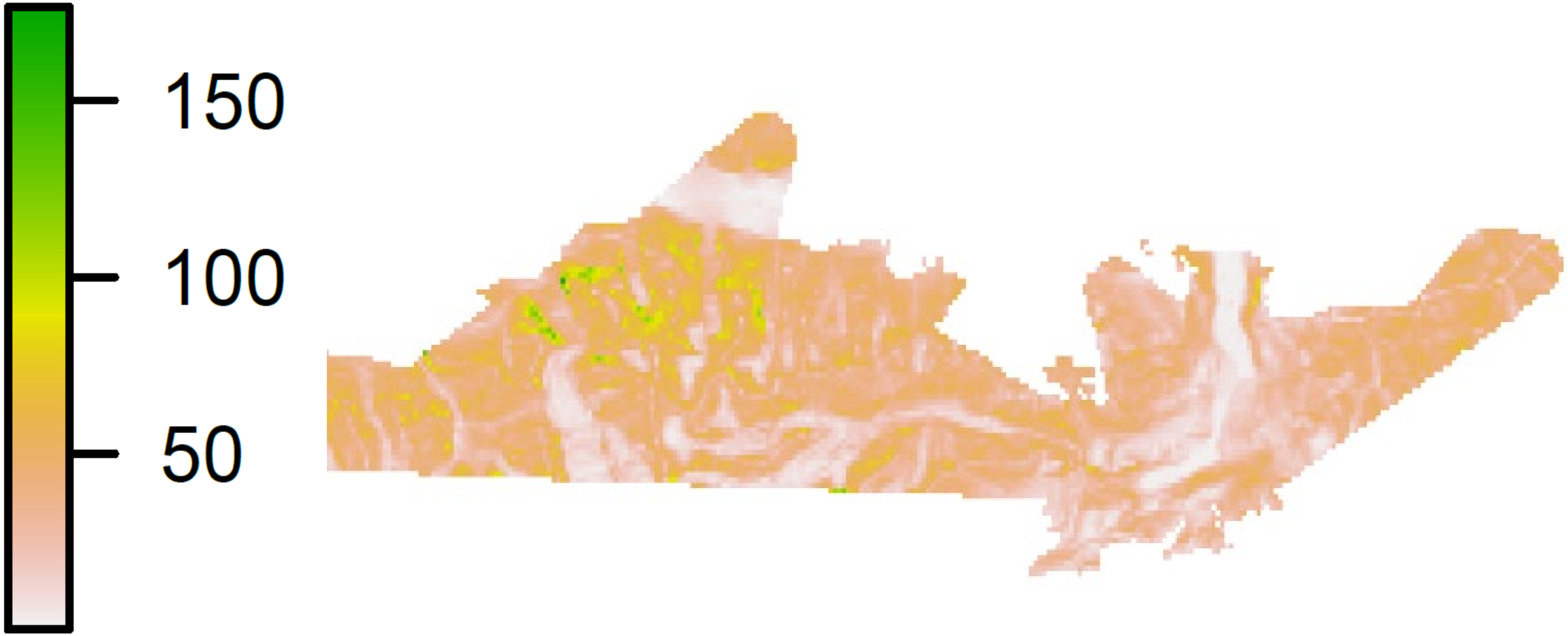


Aspect



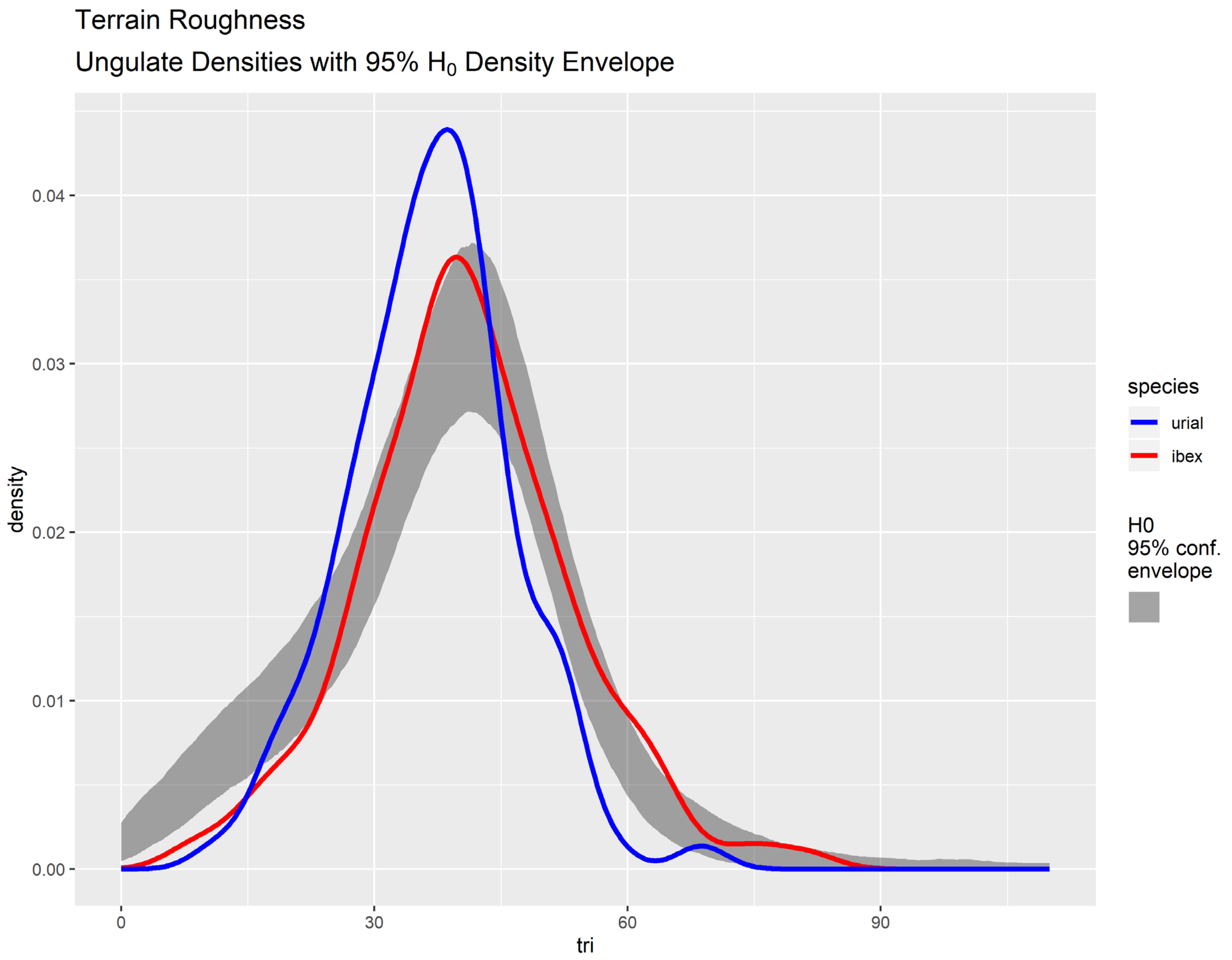
- Ibex appears randomly distributed
- Urial seems to slightly prefer NE slopes
- Neither animal strays very far from random

Terrain Roughness



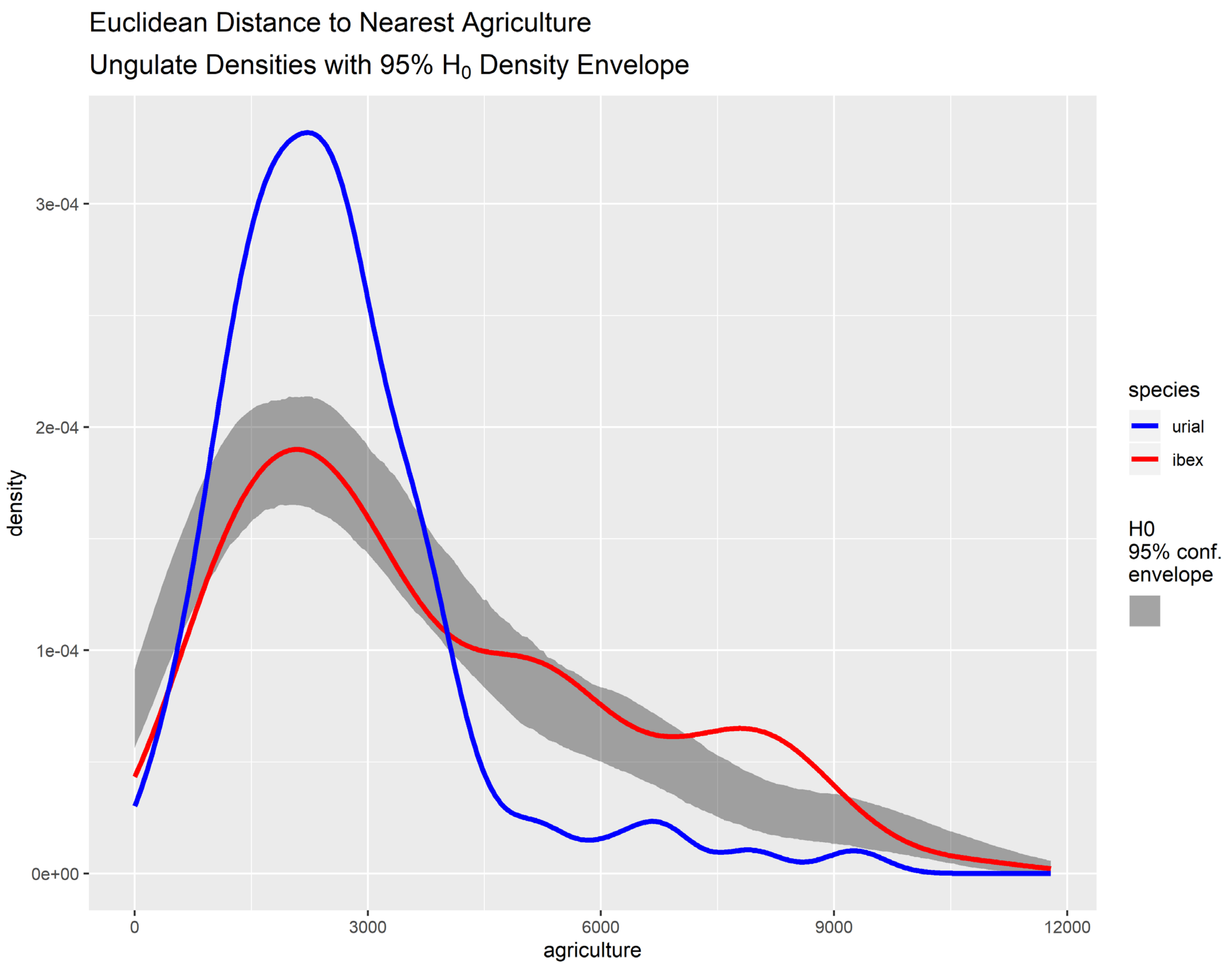
Terrain Roughness

- Ibex is more of a generalist
- Urial prefers less rough terrain

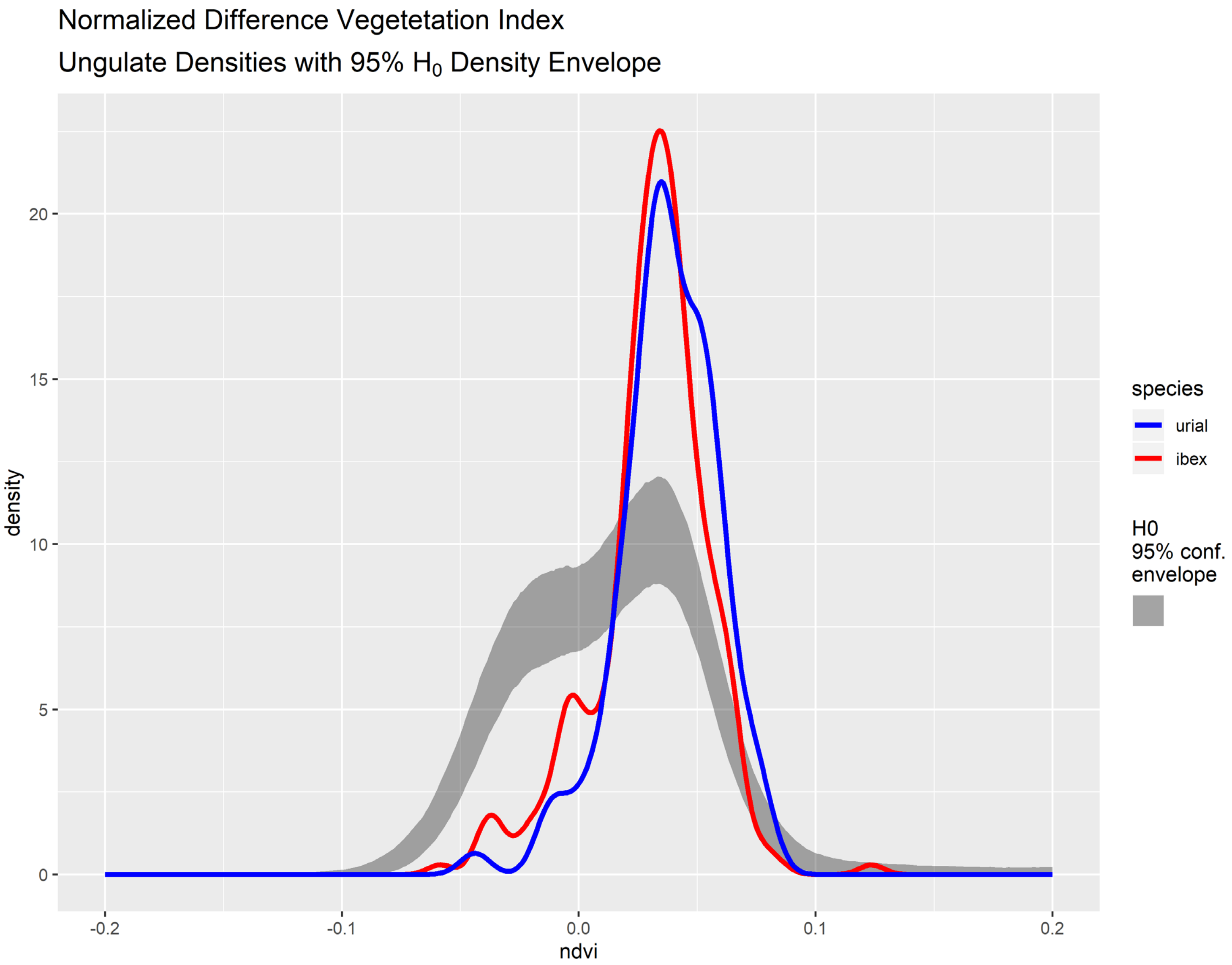


Distance to Disturbance

- Urial prefers closer distances



Normalized Difference Vegetation Index



- Not surprising, both animals concentrate on areas with high NDVI!
- Ndvi measures plant health (approximately)
- Higher NDVI = More abundant food

Statistical and Machine Learning Methods

Statistical and Machine Learning Methods

“The statistical approach focuses on questions such as what model will be postulated (e.g. are the effects additive, or are there interactions?), how the response is distributed, and whether observations are independent. By contrast, the ML approach assumes that the data-generating process (in the case of ecology, nature) is complex and unknown, and tries to learn the response by observing inputs and responses and finding dominant patterns. This places the emphasis on a model's ability to predict well, and focuses on what is being predicted and how prediction success should be measured.”

- From Elith et al. 2008, A working guide to boosted regression trees.

Statistical and Machine Learning Methods

Statistical	**Machine Learning**
• May be mechanistic or phenomenological. • We like to think our modeling is based on our expert understanding (or informed hypotheses) of the system, and therefore mechanistic: • We think that thrushes will prefer mesic forest because of more abundant food sources, for example.	• Phenomenological: learning patterns from the data • Does not require previous understanding of the system • Nonlinearity? Dependence? Who cares?

Maximum Entropy Principle

What is the Maximum Entropy principle? Per Wikipedia:

- "…Another way of stating this: Take precisely stated prior data or testable information about a probability distribution function. Consider the set of all trial probability distributions that would encode the prior data. According to this principle, the distribution with maximal information entropy is the best choice.
- Since the distribution with the maximum entropy is the one that makes the fewest assumptions about the true distribution of data, the principle of maximum entropy can be seen as an application of Occam's razor.

Information Entropy

That was a lot to take in all at once. Let's start with something simpler: Information entropy.

Informally, information entropy is the potential amount of information that can be conveyed by a set of characters. It is related to uncertainty:

- High information entropy: an alphabet of A, B, and C, each with probability 1/3.
- Low information entropy: an alphabet of A, B, and C with probabilities 98/100, 1/100, and 1/100.

Shannon Entropy/Diversity

- Measures diversity and evenness
- Maximized when:
  - There are lots of letters, each letter in the bucket occurs with equal frequency
  - There are many species, each with equal abundance
- In a certain sense, maximum entropy/diversity is equivalent to maximum uncertainty.

- Pi shrinks more quickly than ln(pi)

$$H' = -\sum_{i=1}^{R} p_i \ln p_i$$

Modeling Species Distributions With Maxent

- Maxent was designed for presence-only data
- In R:
  - maxent package: easy to implement, but requires rJava which can be moody.
  - maxnet package: also easy and doesn’t require Java
  - Examples of both in F+F

Trees and Forests

Decision Trees	**Forest**
Individual tree is like a dichotomous key. • Recursive binary splits: sequential set of questions (branches) that arrives at a classification. Trees and forests are kinds of Machine Learning algorithms Output is usually binary	Set of many trees Data is fed to all the trees, an average (or consensus) taken from the results of all the trees. Output is continuous.

Regression Approaches

- All the standard regression paradigms we know about can be used for SDMs:
- General linear models (usually too simple)
- Generalized linear models: can accommodate different response types, certain kinds of nonlinearity, different error distributions
- Additive models: good for phenomenological modeling of nonlinear relationships
- Hierarchical models: helpful for data with complicated structures

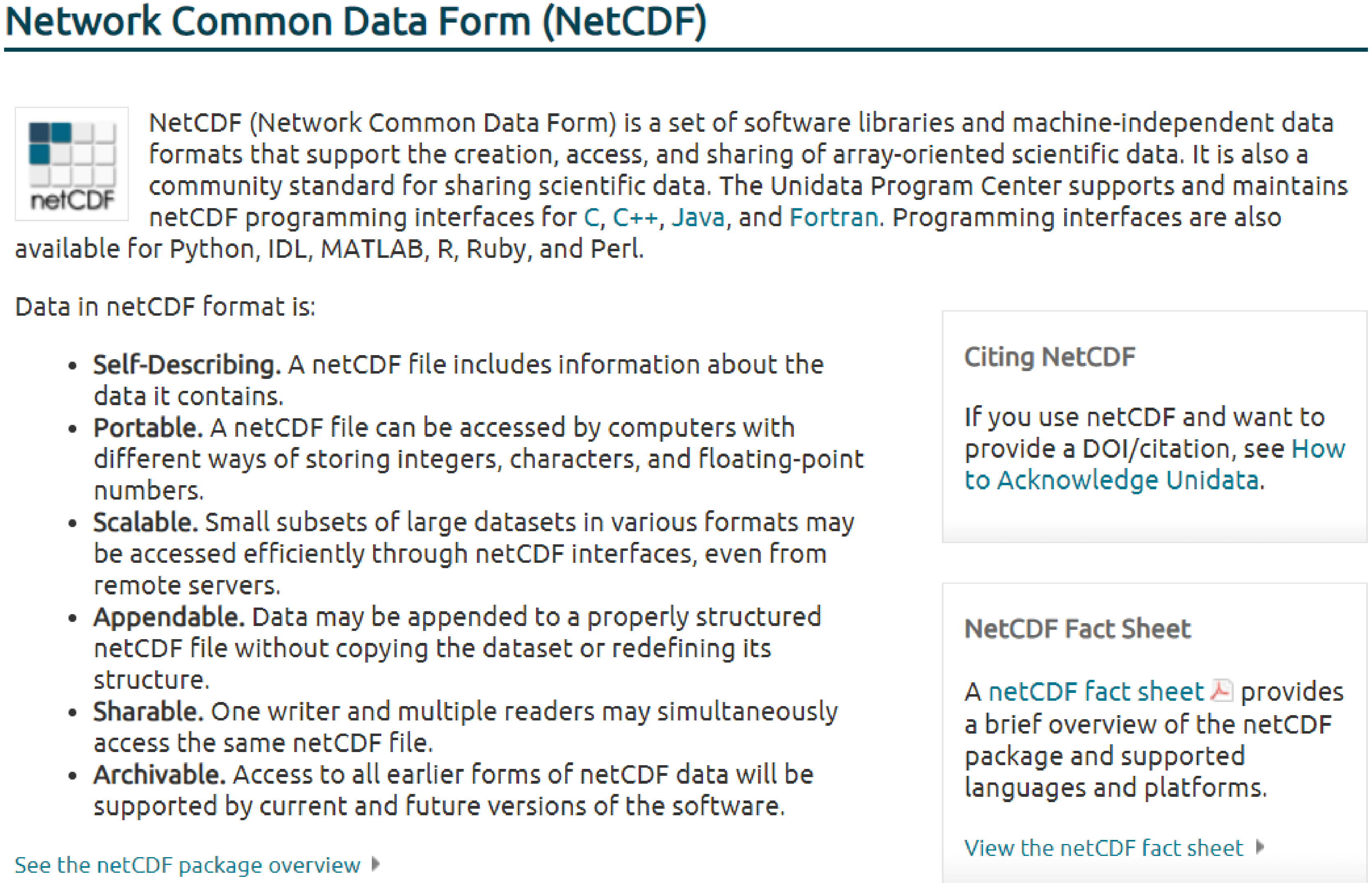
NetCDF Files

Common Data Format

The Common Data Format is a format that aims to:

- Combine data and metadata
  - Can store information on projections, etc.
  - Attempts to enforce meaningful naming conventions
- Provide efficient storage for array-based data.
  - What’s an array? A regular, rectangular, grid of data
  - Can store multidimensional arrays: 2D and 3D are most common
  - Provides functionality for compressing data

Common Data Format



Network Common Data Form (NetCDF)

NetCDF (Network Common Data Form) is a set of software libraries and machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data. It is also a community standard for sharing scientific data. The Unidata Program Center supports and maintains netCDF programming interfaces for C, C++, Java, and Fortran. Programming interfaces are also available for Python, IDL, MATLAB, R, Ruby, and Perl.

Data in netCDF format is:

- **Self-Describing.** A netCDF file includes information about the data it contains.
- **Portable.** A netCDF file can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.
- **Scalable.** Small subsets of large datasets in various formats may be accessed efficiently through netCDF interfaces, even from remote servers.
- **Appendable.** Data may be appended to a properly structured netCDF file without copying the dataset or redefining its structure.
- **Sharable.** One writer and multiple readers may simultaneously access the same netCDF file.
- **Archivable.** Access to all earlier forms of netCDF data will be supported by current and future versions of the software.

See the netCDF package overview ▸

Citing NetCDF

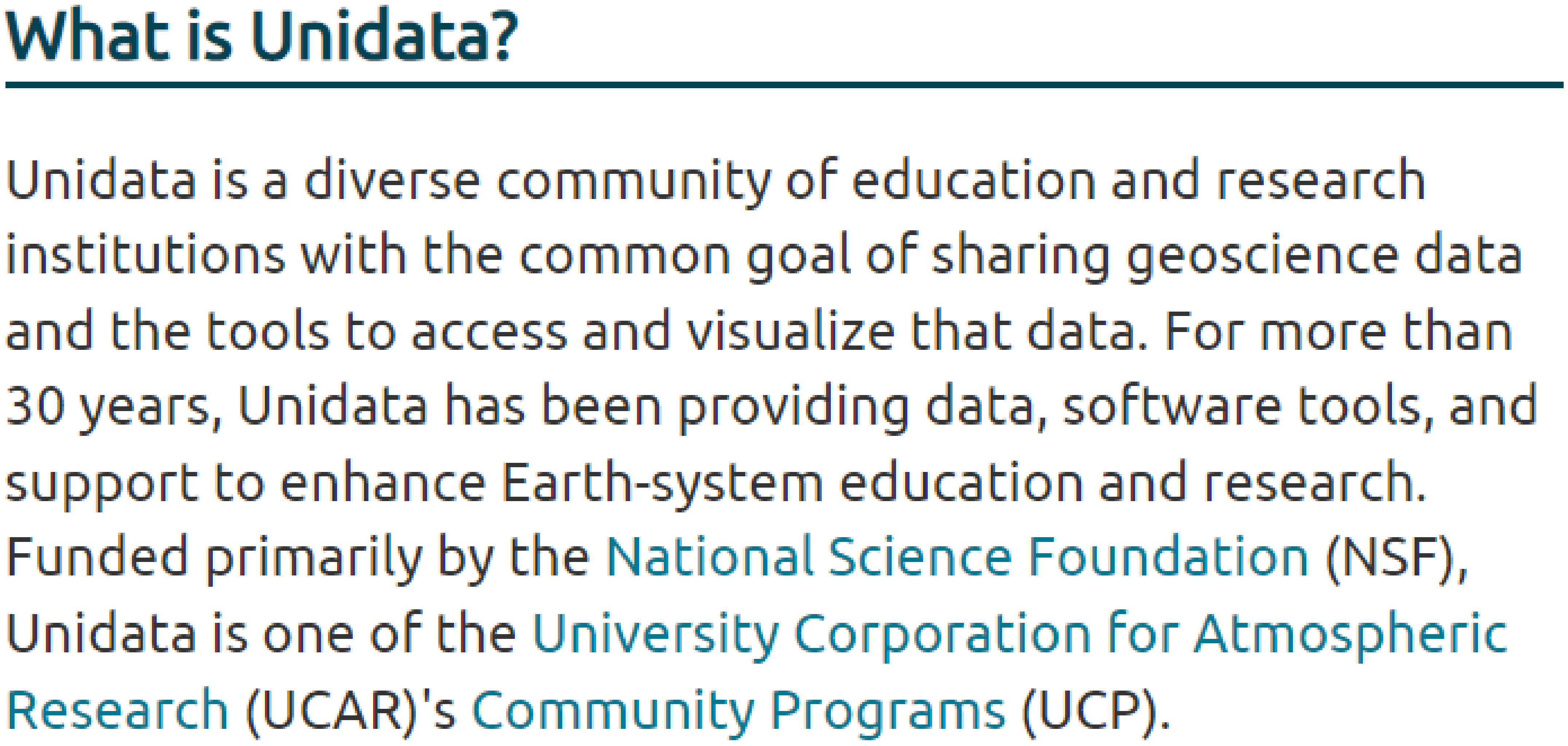
If you use netCDF and want to provide a DOI/citation, see How to Acknowledge Unidata.

NetCDF Fact Sheet

A netCDF fact sheet provides a brief overview of the netCDF package and supported languages and platforms.

View the netCDF fact sheet ▸

UNIDATA



What is Unidata?

Unidata is a diverse community of education and research institutions with the common goal of sharing geoscience data and the tools to access and visualize that data. For more than 30 years, Unidata has been providing data, software tools, and support to enhance Earth-system education and research. Funded primarily by the National Science Foundation (NSF), Unidata is one of the University Corporation for Atmospheric Research (UCAR)'s Community Programs (UCP).

Working With NetCDF Files

- It’s pretty easy in R, you’ve already done it!
  - Remember lab 0? You read a ‘grid’ file and save it as a .nc file using the writeRaster function.
- R has several packages that work with NetCDF files:
  - Raster (and terra): These can read and write NetCDF files. Do not provide direct access to advanced functionality. It can be a pain to add projection info to a raster before writing it.
  - ncdf4: Allows you to manipulate the ‘guts’ of a NetCDF file.

Viewing NetCDF Data

- You can read a NetCDF file into R and plot it.
- It’s easier to use Panoply to visualize nc files.
  - Panoply requires Java. Usually this is not a problem, but sometimes it’s tricky to set up Java on your computer.
  - Let’s look at an example file!