

# Cartographic Pleasures: Maps Inspired by Joy Division's *Unknown Pleasures* Album Art

Travis M. White  
tmacwhite@gmail.com

## INTRODUCTION

IS THE COVER ART for Joy Division's 1979 album *Unknown Pleasures* (Figure 1) the most famous data visualization in popular culture? I suspect so. The image, based on a plot from an astronomy dissertation (Craft 1970), displays radio intensities from the first known pulsar. As designed by Peter Saville, though, all we see is an unlabeled, mysterious landscape. Forty years later the artwork remains both a sublime representation of the band's sound and an icon of commercial art.

This type of data visualization, known as a "joy plot" (for obvious reasons) or a "ridgeline plot" (the overlapping lines are suggestive of a mountain ridge), is "quite useful for visualizing changes in distributions over time or space" (Wilke 2018). The plot's Cartesian structure certainly lends itself to mapmaking: longitude along the x-axis, latitude along the y-axis, and the "height" of each line can correspond to a variety of spatial phenomena, such as population density, lightning strikes, or—most appealing to me—elevation.

I enjoy sketching transect lines and terrain profiles (Figure 2), and the *Unknown Pleasures* art has always looked to me like the profile of a mid-oceanic ridge. A few years ago I attempted my first *Unknown Pleasures*-inspired transect



**Figure 1.** *Unknown Pleasures* in its natural habitat, the record store. The *References and Further Reading* section contains links to excellent histories of the album art. Photo by author.

maps (Figure 3), but the process was painfully slow and I had to shelve the project. As I was finishing my dissertation, though, Claus Wilke released his R package *ggjoy* (now *ggridges*) and my interest was sparked anew. I

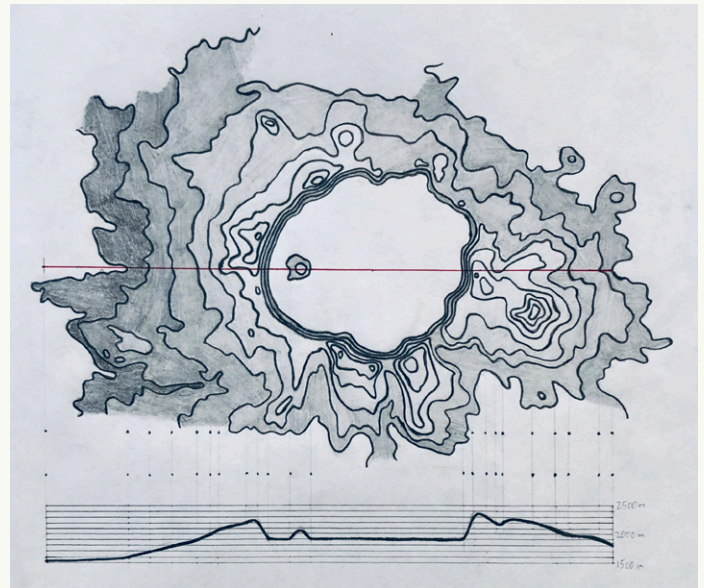


© by the author(s). This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/4.0>.

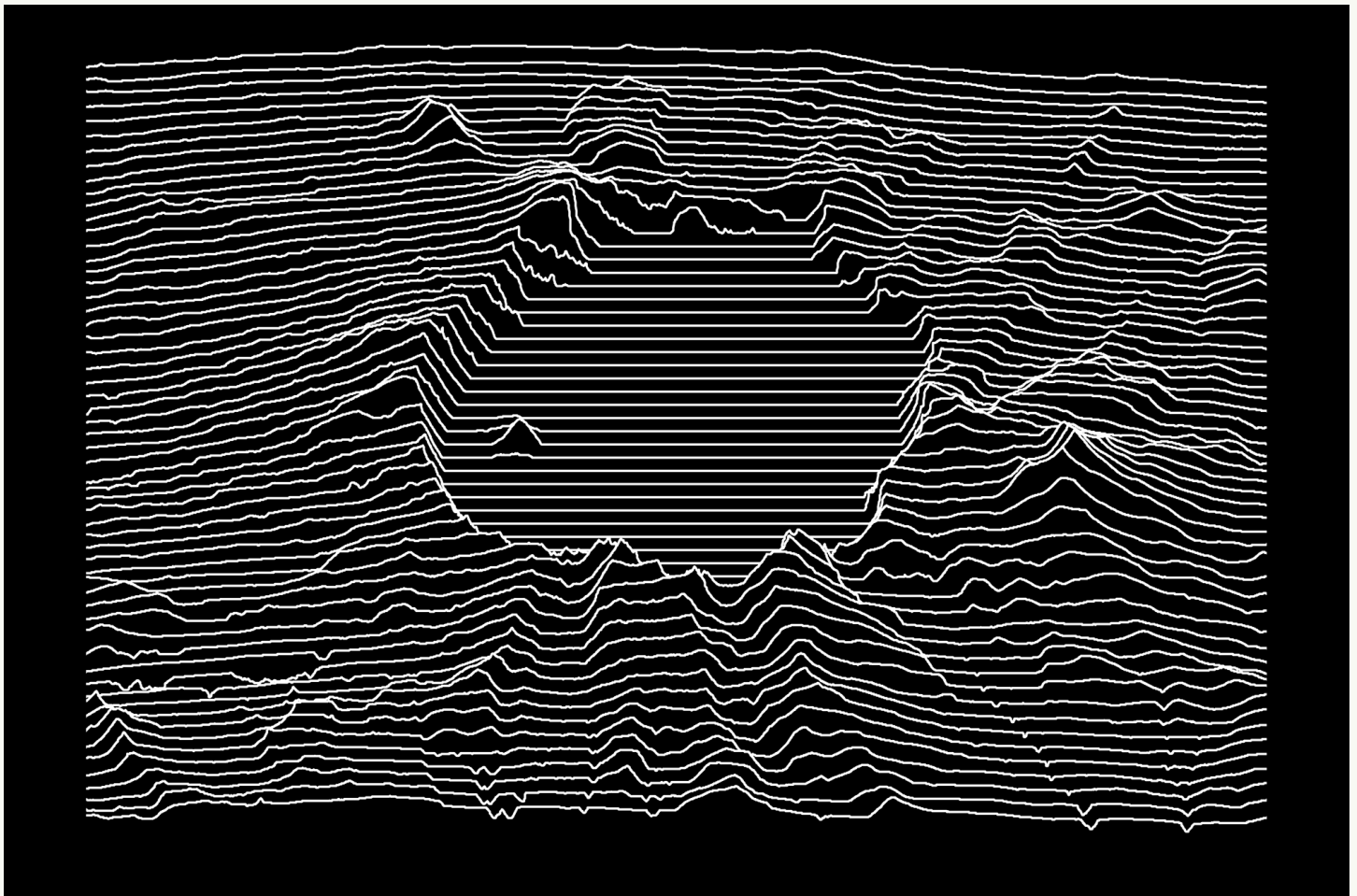
spent the better part of the summer of 2018 refining my transect map workflow, and now I present it for your own mapmaking enjoyment.

## WORKFLOW OVERVIEW

THE WORKFLOW DEMONSTRATED HERE moves through three stages: (1) QGIS to create and prepare the transects, (2) R to plot the transects, and (3) Adobe Illustrator or Photoshop to stylize the transects. In the following sections I walk through each stage of producing an 80-line transect map of Crater Lake National Park, and I encourage you follow along. All files are available for download at [github.com/tmacwhite/PracticalCartoCorner](https://github.com/tmacwhite/PracticalCartoCorner). I presume *Practical Cartographer's Corner* readers have more than a modicum of familiarity with mapping software, so I eschew specific step-by-step instructions such as **Right-click > Save As or Layer > Add Layer > Add Vector Layer** for many basic steps. I also touch on assorted practical and



**Figure 2.** Elevation samples are gathered along a transect and then drawn in profile. Common terms for the completed diagram include terrain profile, terrain diagram, and transect map. Drawing by author.



**Figure 3.** My first attempt at a transect map, 2015. Transect lines created using Python, profiles using the Create Profile Graph tool in ArcGIS 10.3.



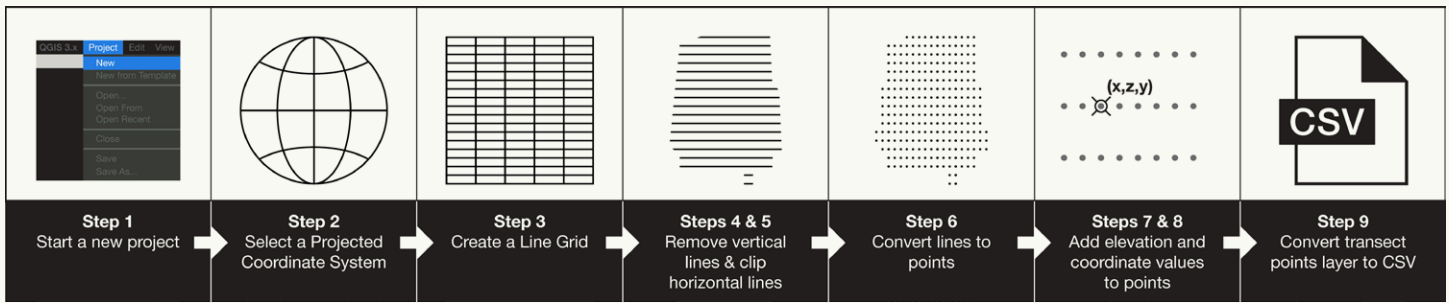


Figure 4. A flowchart illustrating the nine major steps of Stage 1: QGIS. The output from step nine is used in Stage 2: R.

aesthetic issues affecting how I produce transects maps, and how I design them.

Before we begin, know that you can achieve similar results if you substitute ArcGIS for QGIS, Inkscape for

Illustrator, or GIMP for Photoshop. You can also achieve similar results in R alone—and to a lesser extent, using 3D GIS scenes—thus limiting your workflow to a single environment. However, I find that my three-stage method provides the greatest creative control over the final output.

## STAGE 1: QGIS

FIGURE 4 ILLUSTRATES the major steps of my QGIS workflow. I use QGIS for data projection and transect creation. These instructions apply to QGIS 3.4, but are adaptable to older releases of QGIS, as well as ArcGIS Pro and ArcMap. Be aware that QGIS contains multiple tools and plug-ins capable of producing transect data, and as a consequence there are a variety of effective workflows. I conclude this preamble with three recommendations: be consistent with your vector data formats (QGIS defaults to shapefiles or GeoPackages depending

on the tool); include the location and number of transects in every file name (“CraterLake\_80transects.shp”); and use sequential file names for each step in the workflow (“CraterLake\_80transects\_s1.shp”).

### STEP 1: START A NEW PROJECT

Start a new project and add two data layers: *CraterLakeNP\_boundary.shp*, derived from the Natural Earth 1:10m “Parks and Protected Lands” shapefile, and *CraterLake\_DEM*.

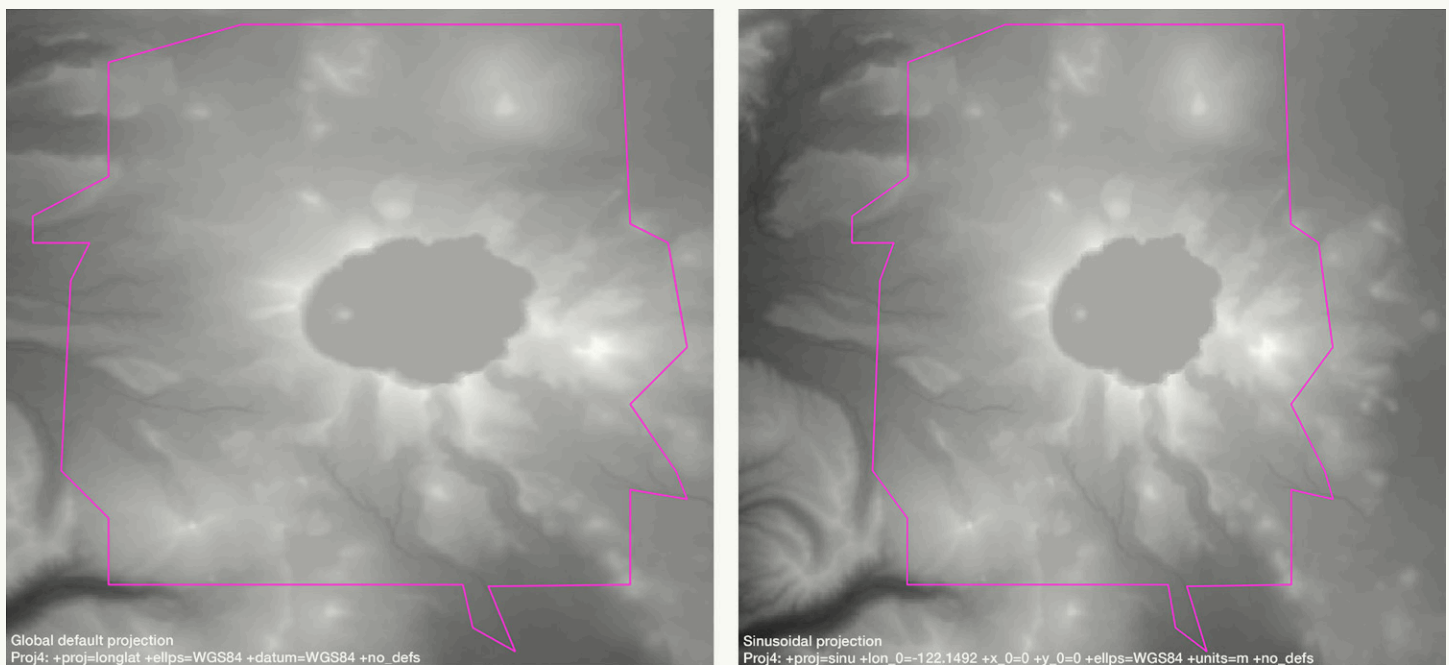


Figure 5. The default equiarectangular projection (left) distorts the shape and size of Crater Lake National Park. A cylindrical or pseudocylindrical projection, such as the Sinusoidal (right), will minimize distortion while maintaining straight, equally-spaced parallels.

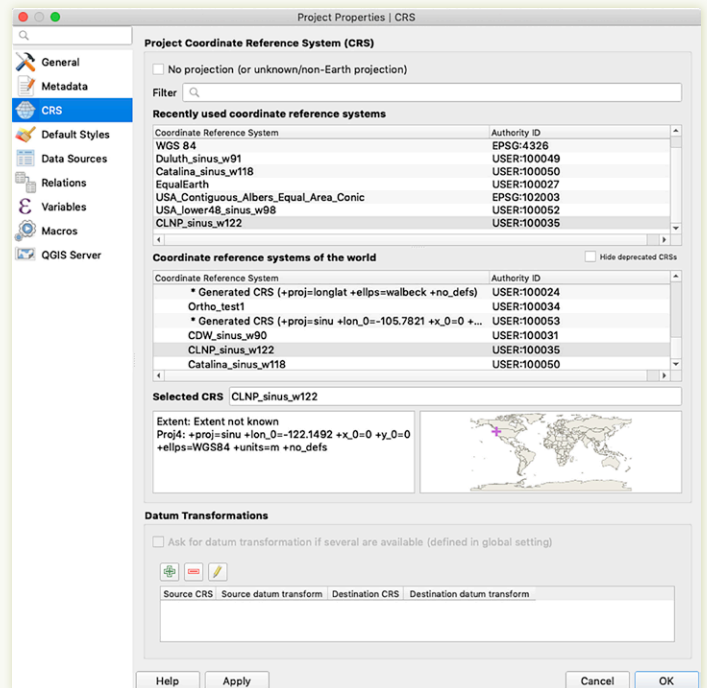
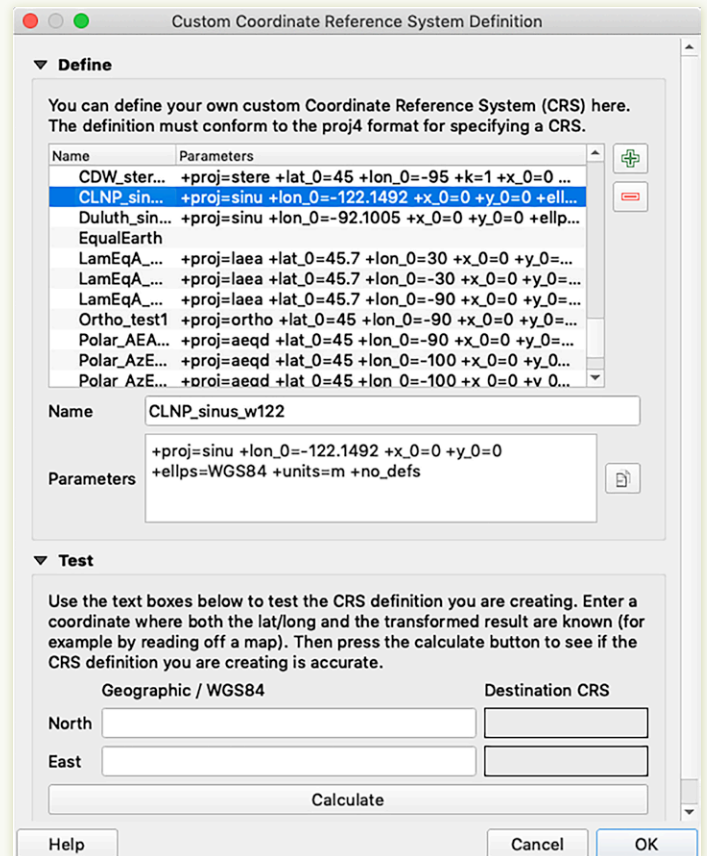
*tif*, derived from 1-arc second Shuttle Radar Topography Mission elevation data ([dwtkns.com/srtm30m](http://dwtkns.com/srtm30m)). Place the park boundary layer on top and change its symbol type to a simple line so you can see the underlying DEM (Figure 5).

### STEP 2: SELECT A PCS

The distortion of Crater Lake’s circular shape should be immediately obvious, so our first task is to assign an appropriate projected coordinate system (PCS). Your choice of PCS will affect the creation and final appearance of the transects—we are making maps, after all—so consider which projection properties you wish to preserve. I find it logical to draw transects along lines of latitude rather than meridians or obliquely, and a cylindrical or pseudocylindrical projection with straight, equally-spaced parallels will ensure equal spacing between the transects. I selected a Sinusoidal projection whose central meridian intersects Wizard Island (Figure 6). Because the sinusoidal projection is distortion-free along its central meridian, the lake shape is no longer distorted (Figure 5).

### STEP 3: CREATE A LINE GRID

Use the Create Grid vector creation tool to produce the transects (**Vector > Research Tools > Create Grid**). The Create Grid tool can be confusing because the number of transect lines cannot simply be assigned. Rather, the grid extent area must be defined, then the desired horizontal and vertical spacing between grid lines within that area must also be defined. Line spacing values use the same units of measurement as the project projection—in this case, meters. Assigning spacing values to create a specific number of grid lines requires a bit of division. For instance, I wanted to use 80 transects in the Crater Lake National Park map, because that is the number of lines used in the *Unknown Pleasures* artwork. To accomplish this, I divided the north-south extent of the park (approximately 35,500 meters) by the desired number of transects (80). Once the quotient is calculated (443.75 meters), enter it into the “Vertical spacing” field of the Create Grid window (Figure 7). Set the grid extent to the park shapefile and the tool will draw the grid within the boundaries of the park. Alternatively, you can draw your own grid extent area; this option is appropriate if you are not mapping a bounded feature such as a national park or country, or you do not need a precise number of transects.



**Figure 6.** To use a custom projection, enter its proj4 string in the Custom Coordinate Reference System Definition parameters field, then assign the projection in the Project Properties menu. For Crater Lake I assigned a Sinusoidal projection centered on -122.1492°, which passes through Wizard Island (Proj4: `+proj=sinu +lon_0=-122.1492 +x_0=0 +y_0=0 +ellps=WGS84 +units=m +no_defs`).

Transect placement is the most important—and often frustrating—step of the entire workflow. Ideally, the transects will overlay the most prominent or recognizable terrain features, such as Wizard Island, otherwise the finished map will appear “off.” How do you get the grid lines to transect the appropriate terrain features? I recommend two approaches. First, incrementally increase or decrease the vertical grid spacing until the tool draws the grid lines in their “ideal” position; in the case of Crater Lake, I want enough lines to capture the shape of the crater, and at least one line to intersect Wizard Island, which is what makes Crater Lake so recognizable.

Alternately, manually adjust the grid position until it overlaps the desired features (**Layer > Toggle Editing**, then **Edit > Move Feature(s)**). While the second approach can be more efficient, it also requires additional steps to edit the feature layer, and can cause the grid to fall outside the boundaries of the park.

Another decision to make is the number of transects to use. While there is no “correct” number for any one map or location, the Goldilocks principle is certainly applicable. Too many transects will overcrowd the image; too few and the underlying terrain becomes unrecognizable. The shape and alignment of the mapped feature will also influence how many transects are appropriate. It is up to you to find that happy medium.

#### STEP 4: REMOVE VERTICAL GRID LINES

The vertical lines in the grid are unnecessary. Select all of the horizontal lines and save them to a new layer (Figure 8). This is a simple step!

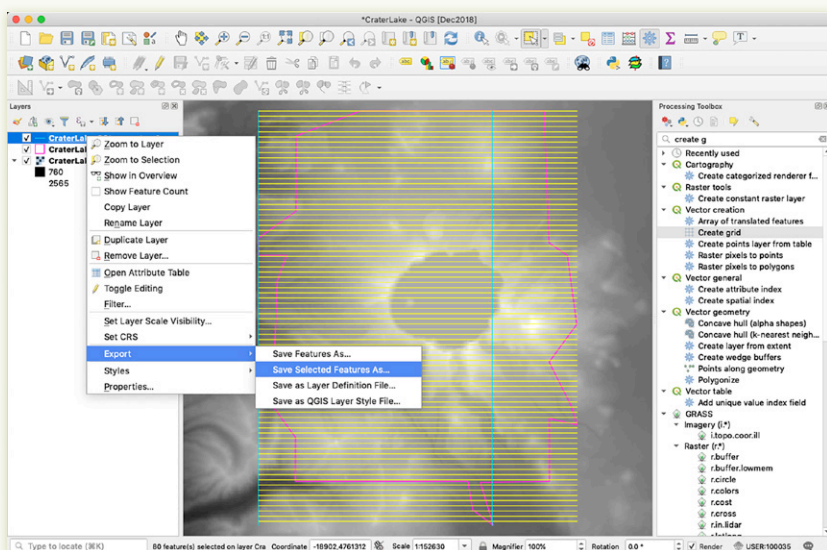


Figure 8. Save the horizontal (latitudinal) lines to a new layer.

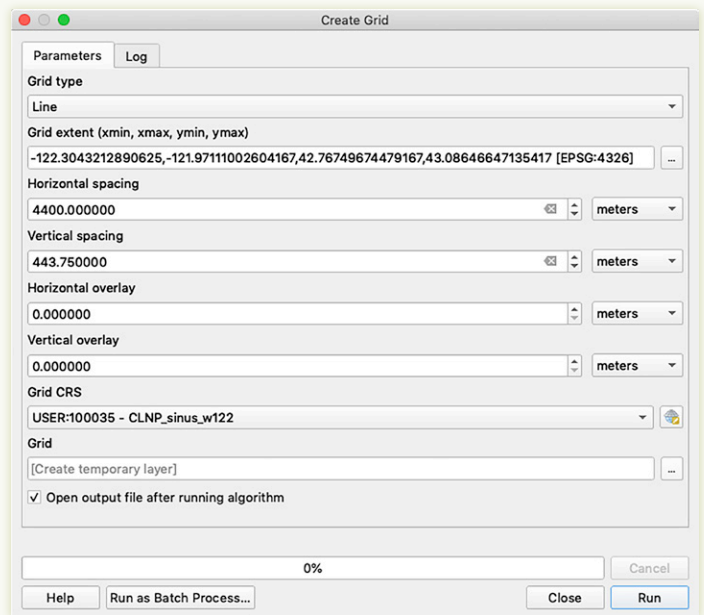
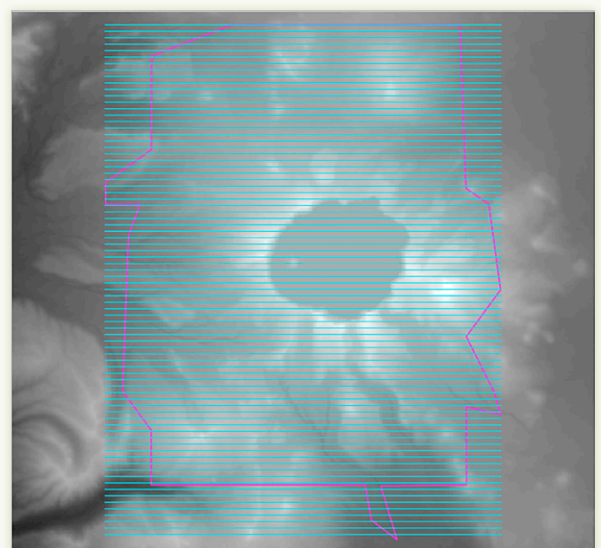


Figure 7. The Create Grid tool. A quirk of the tool is that you cannot enter “0” spacing values, so you must also assign a “Horizontal spacing” value; I typically enter a value one or two orders of magnitude higher than the vertical spacing.

#### STEP 5: CLIP THE HORIZONTAL LINES

Next, use the Clip tool (**Vector > Geoprocessing Tools > Clip**) to clip the horizontal line layer to the boundary of Crater Lake National Park (Figure 9). Skip this step if you are not clipping your transects to match the shape of a particular feature, such as if we were mapping Crater Lake itself and not the entire national park. Also, I do not recommend clipping transects to a vector coastline, since coastline vectors are often misaligned with elevation rasters; these situations are better handled in R, as demonstrated below.





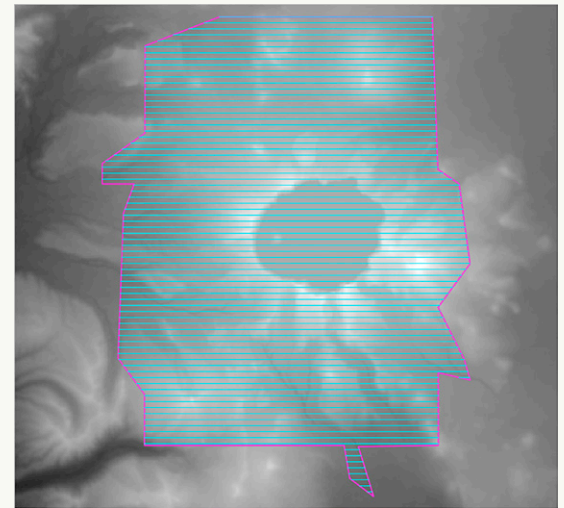
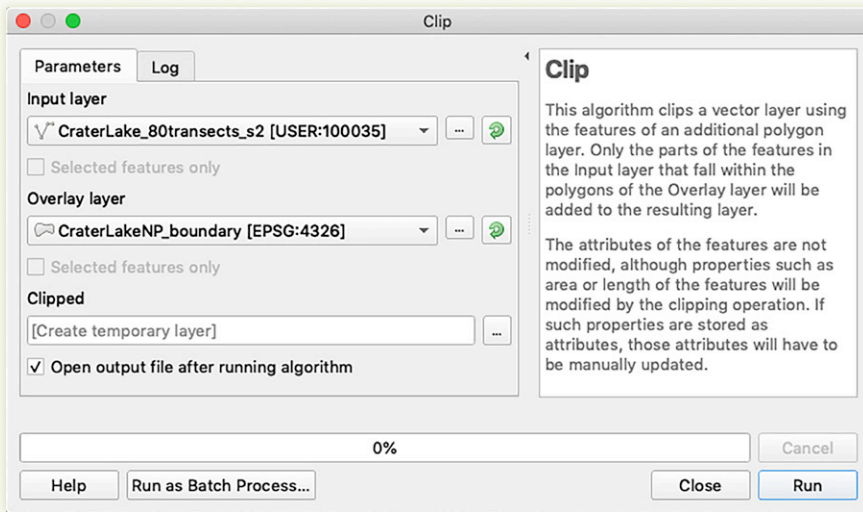


Figure 9. Clip the horizontal transect lines to the shape of the park boundary before converting them to points.

### STEP 6: CONVERT LINES TO POINTS

In order to plot the transects in R you must first convert the lines to points, then sample coordinate and elevation values for each point. QGIS contains multiple line-to-point conversion tools; in this workflow I use the vector creation tool “Generate Points (Pixel Centroids) Along Line,” which produces 24,654 points (Figure 10). For large areas such as the United States, I recommend the SAGA vector point tool “Convert Lines to Points,” which allows you to increase the spacing of your sample points,

and thus avoid generating thousands or even millions of unnecessary points.

### STEP 7: ADD ELEVATION VALUES TO POINTS

This step requires the Point Sampling Tool plug-in ([github.com/borysiasty/pointsamplingtool](https://github.com/borysiasty/pointsamplingtool)). Download the plug-in, then use it (**Plugin > Analyses > Point Sampling Tool**) to extract elevation values from the Crater Lake DEM at each sample point.

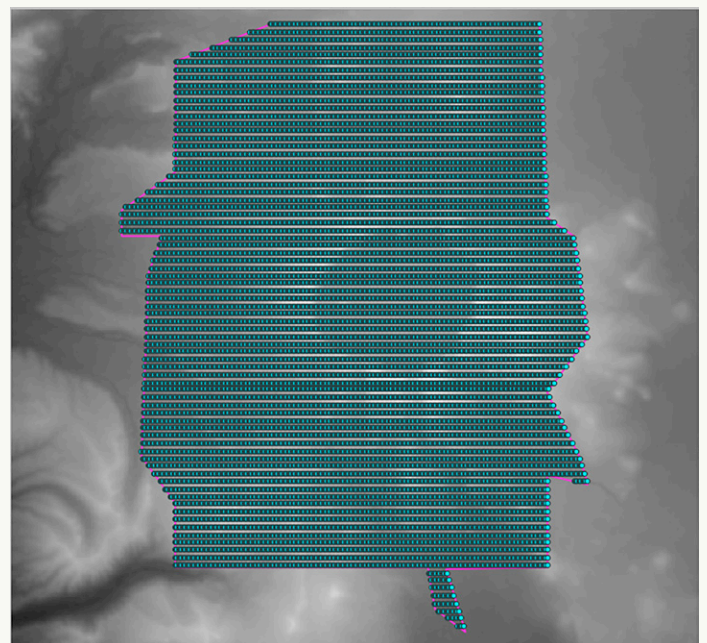
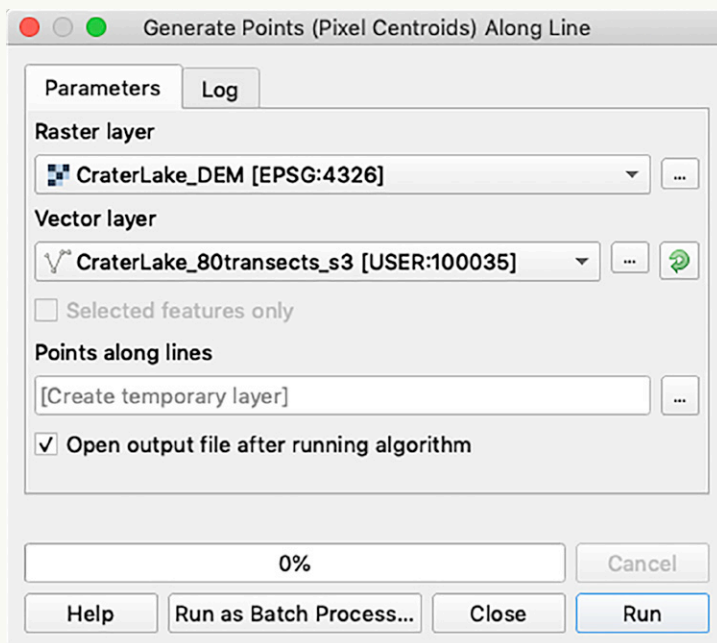


Figure 10. This specific tool generates one sample point per raster grid cell, which is often overkill and can produce an unnecessarily large file size. Proceed with caution.

## STEP 8: ADD COORDINATE DATA TO POINTS

Open the Processing Toolbox (**Processing > Toolbox**) and select the SAGA GIS vector point tool “Add coordinates to points” to add a coordinate pair to each sample point. The resulting vector layer contains our transect data.

## STEP 9: CONVERT VECTOR LAYER TO CSV

In the final step, convert the transect dataset to a comma-separated values (CSV) file (**Layer > Save As...**). This ends the QGIS stage of the workflow.

## STAGE 2: R

THESE INSTRUCTIONS WERE written using RStudio Desktop version 1.1.463 running R version 3.5.1. In this section I discuss each line of code from Example

1, below, which is also available in the R script file “CraterLakeTransectMap.R” in the Git repository for this project.

```
1 # load requisite packages
2 library(ggplot2)
3 library(ggribes)
4 library(mapproj)
5
6 # set your working directory
7 setwd("~/CraterLakeNP")
8
9 # Import the Crater Lake transect data
10 CraterLake_80transects <- read.csv(file="CraterLake_80transects.csv",
11   header=TRUE, sep=",")
12
13 # view data frame and change column headers
14 head(CraterLake_80transects)
15 names(CraterLake_80transects)[1] <- "Elev"
16 names(CraterLake_80transects)[2] <- "Lon"
17 names(CraterLake_80transects)[3] <- "Lat"
18
19 # plot the transects with ggplot2 & ggribes
20 CraterLake_basic <- ggplot(CraterLake_80transects,
21   aes(x = Lon, y = Lat, group = Lat, height = Elev)) +
22   geom_density_ridges(stat = "identity")
23
24 # Call the default plot variable
25 CraterLake_basic
26
27 # customize the appearance to mimic the Unknown Pleasures artwork
28 CraterLake_Joy <- ggplot(CraterLake_80transects,
29   aes(x = Lon, y = Lat, group = Lat, height = Elev)) +
30   geom_density_ridges(stat = "identity", scale = 15,
31   fill="black", color = "white") +
32
```

**Example 1.** Full R script referenced in Stage 2, which is also available in the R script file “CraterLakeTransectMap.R” in the Git repository for this project.

```

33 # set the upper and lower y-axis limits
34 ylim(42.77, 43.15) +
35
36 # add a title to the bottom of the plot frame
37 scale_x_continuous(name = "CRATER LAKE NATIONAL PARK") +
38
39 # use theme() to customize the background, axis labels, titles, etc.
40 theme(panel.grid.major = element_blank(),
41       panel.grid.minor = element_blank(),
42       panel.background = element_rect(fill = "black"),
43       axis.line = element_blank(),
44       axis.text.x=element_blank(),
45       plot.background = element_rect(fill = "black"),
46       axis.ticks.x=element_blank(),
47       axis.title.y=element_blank(),
48       axis.text.y=element_blank(),
49       axis.ticks.y=element_blank(),
50       axis.title.x = element_text(colour = 'white', size = 18)) +
51
52 # projects the transect data to a specified PCS
53 coord_map()
54
55 # Call the stylized plot variable
56 CraterLake_Joy
57
58 # Save the plot as a PNG or PDF
59 ggsave("CraterLake_Joy.png", dpi=300)
60 ggsave("CraterLake_Joy.pdf")
61
62 # Customized plot:
63 ggplot(CraterLake_transects,
64       aes(x = Lon, y = Lat, group = Lat, height = Elev)) +
65       geom_density_ridges(stat = "identity", scale = 15,
66       fill="pink", color = "violetred4", size = 1, linetype = "12") +
67       scale_x_continuous(name = "CRATER LAKE NATIONAL PARK") +
68       ylim(42.77, 43.15) +
69       theme(panel.grid.major = element_blank(),
70           panel.grid.minor = element_blank(),
71           panel.background = element_rect(fill = "pink"),
72           axis.line = element_blank(),
73           axis.text.x=element_blank(),
74           plot.background = element_rect(fill = "pink"),
75           axis.ticks.x=element_blank(),
76           axis.title.y=element_blank(),
77           axis.text.y=element_blank(),
78           axis.ticks.y=element_blank(),
79           axis.title.x = element_text(colour = 'violetred4', size = 18)) +
80       coord_map()

```

**Example 1**, continued.



```

1 # load requisite packages
2 library(ggplot2)
3 library(ggribes)
4 library(mapproj)

```

My method of plotting transect maps in R requires three packages: `ggplot2`, `ggribes`, and `mapproj`. `ggplot2` is a graphics and data visualization creation package based on *The Grammar of Graphics* (Wilkinson 2006). `ggribes` contains functions that plot data in the *Unknown Pleasures* style; this package is what inspired me to revisit transect mapping (and ultimately write this article), and is rightly the star of this workflow. The third package, `mapproj`, assigns a projected coordinate system to the plotted data.

```

6 # set your working directory
7 setwd("~/CraterLakeNP")
8
9 # Import the Crater Lake transect data
10 CraterLake_80transects <- read.csv(file=
    "CraterLake_80transects.csv",
11   header=TRUE, sep=",")

```

Set the working directory to the folder containing the transect data, then import the Crater Lake transect data. The `read.csv` function imports data from a CSV file, reads it as a data frame, and assigns the data frame to a new R variable. In this example, we will also name the variable “CraterLake\_80transects.”

```

13 # view data frame and change column headers
14 head(CraterLake_80transects)
15 names(CraterLake_80transects)[1] <- "Elev"
16 names(CraterLake_80transects)[2] <- "Lon"
17 names(CraterLake_80transects)[3] <- "Lat"

```

Call the `head()` function to preview the first six rows of your transect data frame. Notice the data frame uses the same column headers as the CSV file: “srtm\_12\_04” for elevation values, “X” for longitude, and “Y” for latitude. I prefer to work with descriptive column headers, so I use the `names()` function to rename each header. These lines in the code above are completely optional and skipping them will not affect the operation of the plotting functions.

```

19 # plot the transects with ggplot2 & ggribes
20 CraterLake_basic <- ggplot(CraterLake_80transects,
21   aes(x = Lon, y = Lat, group = Lat, height = Elev)) +
22   geom_density_ridges(stat = "identity")

```

The lines above create a basic plot of the transect data. `CraterLake_basic <-` assigns the plot to its own R

variable. Calling `ggplot()` initializes the plotting process. `aes()` refers to “aesthetic mappings” and defines how the variables in the data frame—Lon, Lat, Elev—will be mapped in the plot. Pass `aes()` arguments to assign longitude to the x-axis, latitude to the y-axis, and elevation to the height of each ridgeline. A fourth argument, `group = Lat`, instructs the plot to draw ridgelines by connecting points of equal latitude.

The `ggribes` function `geom_density_ridges()` is what draws the transects in a ridgeline plot. The argument `stat = “identity”` sets the height of the ridgelines to the elevation values in the data.

```

24 # Call the default plot variable
25 CraterLake_basic

```

Calling `CraterLake_basic` displays the plotted transect lines (Figure 11). Now, the default plot is not particularly attractive, so pass additional arguments to customize the plot’s appearance.

```

27 # customize the appearance to mimic the Unknown
    Pleasures artwork
28 CraterLake_Joy <- ggplot(CraterLake_80transects,
29   aes(x = Lon, y = Lat, group = Lat, height = Elev)) +
30   geom_density_ridges(stat = "identity", scale = 15,
31     fill="black", color = "white") +

```

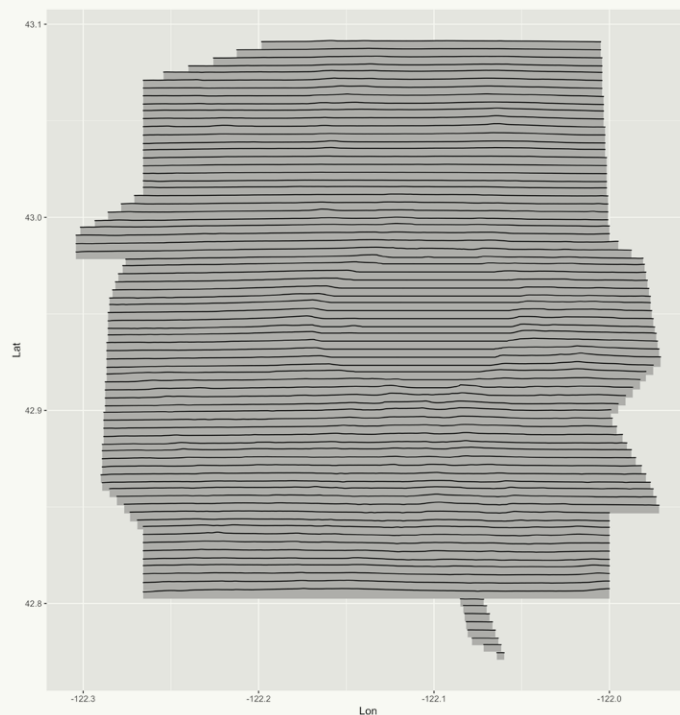


Figure 11. The default ridgeline plot of Crater Lake National Park.

I have made four changes to the code seen in lines 19–22. First, I assigned the plot to a new R variable `CraterLake_Joy`. Second, I set the `scale` parameter, which controls the height of each ridgeline, to add vertical exaggeration to the transects and increase the amount of overlap. Lower scale values will minimize exaggeration and overlap, and may obscure the relative relief of the plotted transects; setting too high a scale value may result in over-exaggerated terrain and overlap. Third, I assigned two colors, white for the ridgelines and black for their fills. Fourth, I added a plus sign at the end (+) to indicate that subsequent lines contain additional plotting arguments.

```
33 # set the upper and lower y-axis limits
34 ylim(42.77, 43.15) +
```

`ylim()` sets the upper and lower latitudinal limits of the y-axis. This command is useful when mapping the same dataset to multiple scale factors because it ensures each plot uses the same y-axis interval. Be sure to assign latitudes that exceed the upper and lower boundaries of the mapped feature, otherwise the plot may cut off the transects or fail to plot them entirely.

```
36 # add a title to the bottom of the plot frame
37 scale_x_continuous(name = "CRATER LAKE NATIONAL PARK") +
```

This command adds a title underneath the x-axis.

```
39 # use theme() to customize the background, axis
  labels, axis titles, tick marks, etc.
40 theme(panel.grid.major = element_blank(),
41 panel.grid.minor = element_blank(),
42 panel.background = element_rect(fill = "black"),
43 axis.line = element_blank(),
44 axis.text.x=element_blank(),
45 plot.background = element_rect(fill = "black"),
46 axis.ticks.x=element_blank(),
47 axis.title.y=element_blank(),
48 axis.text.y=element_blank(),
49 axis.ticks.y=element_blank(),
50 axis.title.x = element_text(colour = 'white',
  size = 18)) +
```

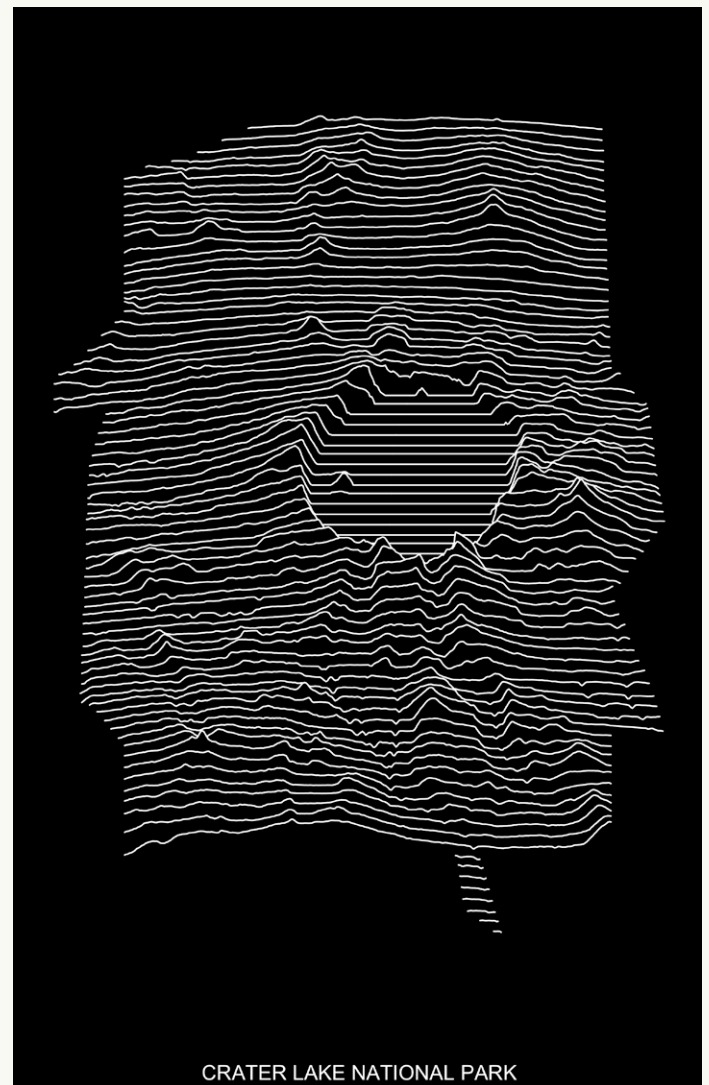
Use `theme()` to customize the appearance of the plot. These lines remove all typical plot elements, such as grids, tick marks, axis labels, and adjust the background colors to match the ridgeline fill colors. I entered values to mimic the appearance of the *Unknown Pleasures* artwork—white lines and text on black background.

```
52 # projects the transect data to a specified PCS
53 coord_map()
```

The `coord_map()` function from the `mapproj` package defines a projection for the plotted data. Neither the CSV file nor the data frame contain any projection information, and as a result the plotted transects exhibit the sort of spatial distortion illustrated at the beginning of Stage One. By calling this function you can project the transect data to a specific projection, although I have found that leaving the field blank produces acceptable results for medium- and large-scale maps. Note that lines 28–53 of the R script will all execute at once.

```
55 # Call the stylized plot variable
56 CraterLake_Joy
```

Call `CraterLake_Joy` to view the *Unknown Pleasures*-styled transect map. As you can see, the results are pretty spot on (Figure 12)!



**Figure 12.** Crater Lake National Park in the iconic style of *Unknown Pleasures*.

```

58 # Save the plot as a PNG or PDF
59 ggsave("CraterLake_Joy.png", dpi=300)
60 ggsave("CraterLake_Joy.pdf")

```

As a final step, use `ggsave()` to save the plot in a format of your choice. In this example I exported both PNG and PDF formats to use in Stage Three of this tutorial. The exported maps will save to your working directory.

Concluding the R stage of this workflow are four practical and aesthetic suggestions. First, the graphic capabilities of `ggplot2` are extensive and I encourage readers to play around with their own stylizations beyond the *Unknown Pleasures* aesthetic. For instance, changing a few aesthetic arguments such as color, line weight, and line type can produce considerably different map styles (Figure 13).

Second, although the `coord_map()` argument can remain empty when mapping small areas, the default projection is inappropriate for large areas. The transect maps in Figure 14 illustrate this issue: A omits the `coord_map()` argument, B passes a blank argument, and C assigns an Azimuthal Equal Area projection. Clearly, C produces the most aesthetically pleasing representation of the continental United States.

Third, converting water elevation values to `NA` is an efficient way to remove those water features from the plot. For instance, Figure 15 displays Catalina Island before and after all sea level values were converted to `NA`. This method of “clipping” transects along a coastline is more effective than pairing elevation rasters and coastline vectors, which are often misaligned.

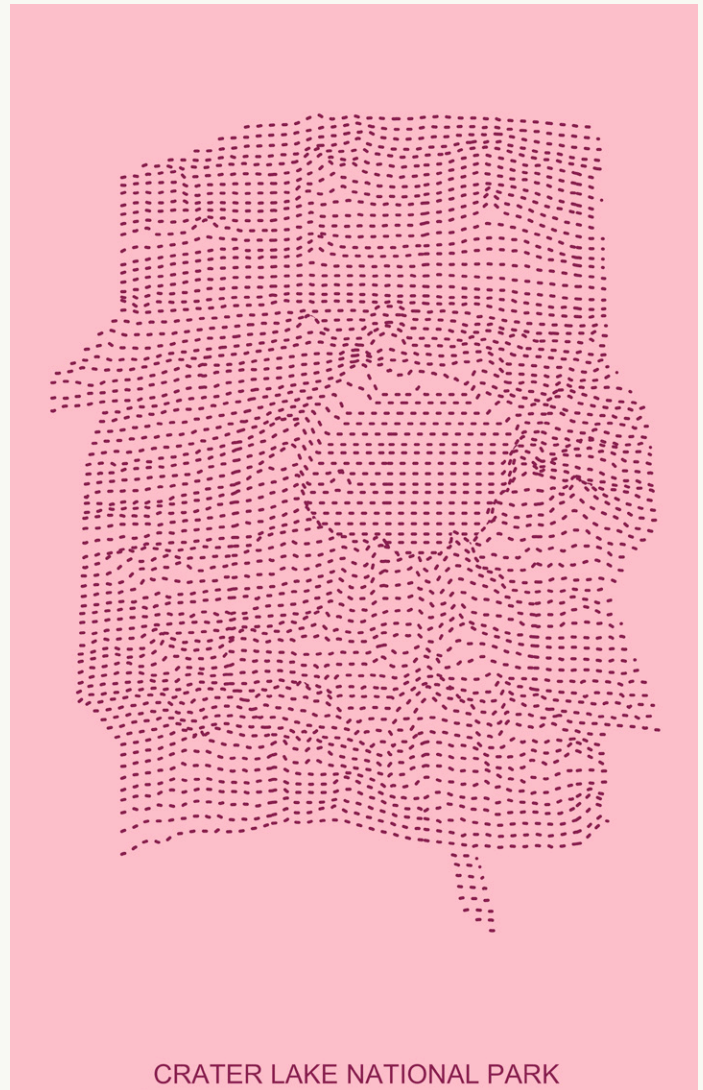


Figure 13. A different aesthetic spin on the same data file.

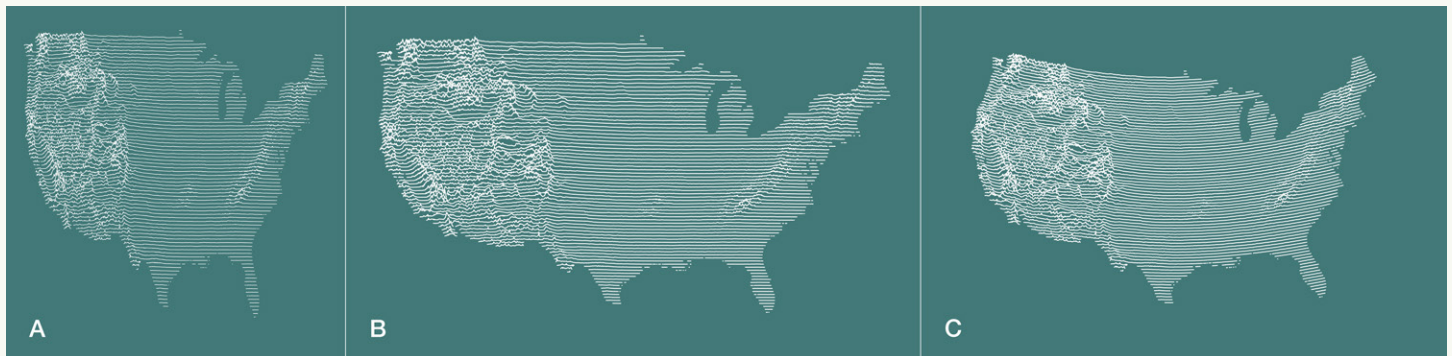


Figure 14. A ridgeline plot of the continental United States. Omitting `coord_map()` entirely (A), passing a blank `coord_map()` (B), and passing `coord_map("azequalarea", orientation = c(39.8283, -98.5795, 0))` (C).



Fourth and finally, look at the work of others! Many members of the cartography and R communities have developed their own transect map packages and workflows. As the GIS capabilities of R continue to expand, I expect to migrate the entirety of stage one into the R environment.

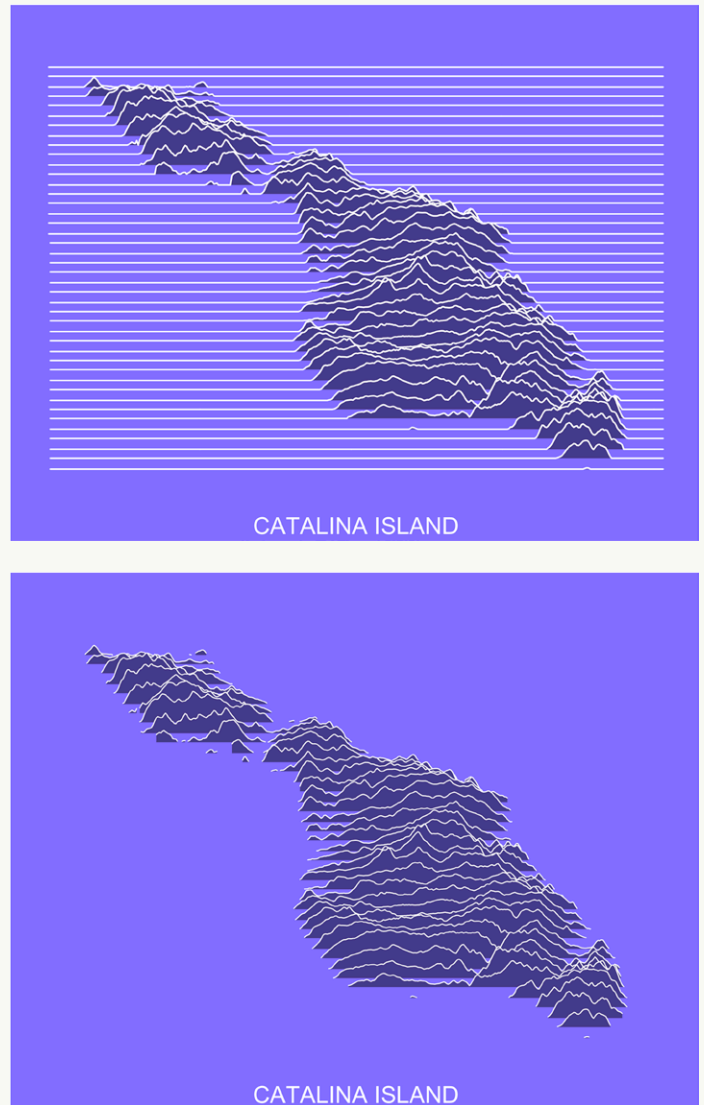
## STAGE 3: ADOBE ILLUSTRATOR OR PHOTOSHOP

THESE INSTRUCTIONS WERE written using Adobe Illustrator CC 2019 (version 23.0.1) and Photoshop CC 2018 (version 20.0.1). First, a disclaimer: this stage is not required to produce an attractive transect map. Feel free to stop here! The R packages `ggplot2` and `ggridges` (and many others) are capable of beautiful data visualization, and I create many of my transect maps in R with no external modifications. When I feel a bit more visual oomph is necessary, however, I turn to Adobe Illustrator or Photoshop to apply subtle modifications that I currently cannot perform in R. In this section I present the two most common “tricks” I use to enhance the final appearance of my transect maps: line gradients in Illustrator and background gradients in both Illustrator and Photoshop. In both cases I encourage the reader to explore and improve upon my examples. While the directions below are for Adobe products, other vector or raster graphics programs should be able to achieve similar results.

### LINE GRADIENTS IN ADOBE ILLUSTRATOR

Apply a subtle gradient to transect lines to emphasize relative relief, create a faint impression of shaded relief, and add a touch of depth to the otherwise flat image (Figure 16). This technique requires the PDF created in R. Be aware the PDF, as exported, is composed of a single layer containing one text path and two clipping masks; the foreground clipping mask holds all of the individual transect line and fill pairs, and the background mask contains the panel and plot fills.

Open the PDF in Illustrator, then expand the layer and lock the bottom (background) clipping mask. Activate the Stroke (keyboard shortcut: **X**). Use the Direct Selection Tool (white selection arrow, keyboard shortcut: **A**) to select a single transect line, then select all other transect lines (**Select > Same > Stroke Color**); *do not* group the transect lines, otherwise you will ruin the overlapping pattern of lines and fills. Next, open the gradient tool



**Figure 15.** Sea-level sample points removed with the line `Catalina_NoWater$Elev[Catalina_NoWater$Elev <= 0] <- NA`

(**Window > Gradient**) and select the Linear Gradient option (this should be the default). Change the angle to  $-90^\circ$  and set the Midpoint location to 75%. Click on the right color stop (the square at the end of the gradient slider) and assign the desired color. That’s the whole procedure! It can take quite a bit of testing to find the appropriate gradient colors. Because I prefer a *subtle* gradient effect, I typically begin with a lighter tint of the fill color then make it progressively darker until I find the desired color. Assign too light a color and the gradient is not perceptible; assign too dark a color and the effect can be overwhelming. Note this does not produce a quantitatively consistent scale—that is, Illustrator does not assign a distinct color for each elevation value. Instead, the full gradient is stretched across the vertical range of each individual transect line.

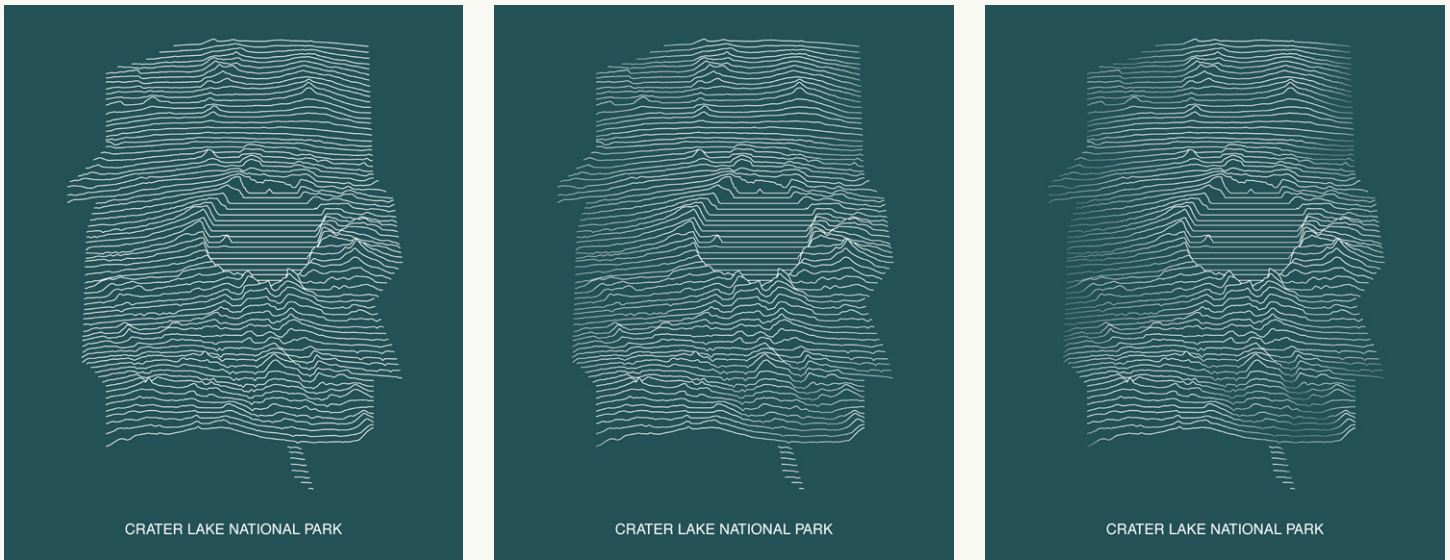


Figure 16. Line gradients added in Illustrator: no gradient (left), subtle gradient (middle), and strong gradient (right).

### BACKGROUND GRADIENTS IN ADOBE ILLUSTRATOR OR PHOTOSHOP

Background gradients can give transect maps a nifty screen print effect (Figure 17). Unlike line gradients, you *can* apply a background gradient to a ridgeline plot in R. However, I am usually unsatisfied by the results and thus use either Illustrator or Photoshop to achieve the desired effect. The process, which is nearly identical in both software environments, utilizes the black-and-white Crater Lake transect map and the Screen blending mode. A layer's blending mode determines how its colors (the blend colors) interact with colors in the underlying layers (the base colors). The Screen blending mode combines blend colors and base colors to produce brighter colors, except for black, which is replaced entirely, and white, which is left intact. Screening works particularly well with the transect map, because the gradient replaces the black fill while leaving the white transects lines untouched.

In Photoshop, open the Crater Lake PNG (created in R) and add a new gradient fill layer on top (**Layer > New Fill Layer > Gradient**). Accept all defaults and click OK. In the Gradient Fill window click on the default gradient to open the Gradient Editor. Select the **Blue, Red, Yellow** preset. Click OK to close the Gradient Editor, then click OK to close the Gradient Fill window. Finally, change the gradient fill layer's blending mode to **Screen**.

In Illustrator, open the Crater Lake PDF and add a new layer on top. Create a rectangle (keyboard shortcut: **M**) with the same dimensions as the map. Change

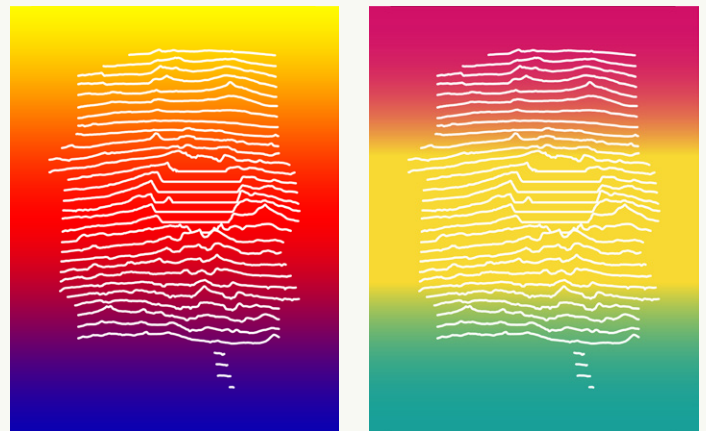


Figure 17. Background gradients created in Photoshop (left) and Illustrator (right).

the rectangle blending mode to Screen (**Window > Transparency > Blending Mode > Screen**). Set the rectangle stroke color to **none**. Activate the rectangle fill color. Open the Gradient tool (**Window > Gradient**) and select the **Linear Gradient** option. Click on the right and left color stops (the circles squares at either end of the gradient slider) and assign the desired colors.

### MIMICRY, INSPIRATION, AND CREATION

There is a lot to be said for mimicry in mapmaking. I often find that I learn the most about cartographic forms and content through deliberate deconstruction and mimicry. This applies to everything from iconic works such as

Harry Beck's Tube map, to popular techniques like the illuminated (Tanaka) contour method, to singular data visualizations like the *Unknown Pleasures* album art. The challenge of this approach, as I have also found, is not limiting myself to simple mimicry or derivatives, but using them as sources of inspiration for my own creativity.

To this end, I encourage you to map your favorite locations in the *Unknown Pleasures* style. Go ahead, it's fun! But do

not stop there—continue to explore the potential of this form, map unconventional data, develop your own workflow, and continue to improve your cartographic skills. There is no final design, and there is no correct number of transects. Does it matter if, regardless of how far you push the form, there will always be a shadow of a resemblance to the *Unknown Pleasures* album art? Perhaps, but who cares? Happy mapping!

## RESOURCES

---

- ggplot2: [cran.r-project.org/web/packages/ggplot2](https://cran.r-project.org/web/packages/ggplot2)
- ggribbles: [cran.r-project.org/web/packages/ggribbles](https://cran.r-project.org/web/packages/ggribbles)
- mapproj: [cran.r-project.org/web/packages/mapproj](https://cran.r-project.org/web/packages/mapproj)

## REFERENCES & FURTHER READING

---

Christiansen, Jen. 2015. "Pop Culture Pulsar: Origin Story of Joy Division's Unknown Pleasures Album Cover." *Scientific American* February 18, 2015. <https://blogs.scientificamerican.com/sa-visual/pop-culture-pulsar-origin-story-of-joy-division-s-unknown-pleasures-album-cover-video>.

Craft Jr., Harold. 1970. "Radio Observations of the Pulse Profiles and Dispersion Measures of Twelve Pulsars." Ph.D. diss., Cornell University

Field, Kenneth. 2018. "Joy Plots in ArcGIS Pro." *ArcGIS Blog* July 18, 2018. <https://esri.com/arcgis-blog/products/arcgis-pro/mapping/joy-plots-in-arcgis-pro>.

Klotz, Eric. 2012. "Data Visualization Reinterpreted: The Story of Joy Division's Unknown Pleasures Album." <http://youtu.be/reEQye0EOAw>.

Wilkinson, Leland. 2006. *The Grammar of Graphics*. New York: Springer Science & Business Media

