***Purpose*:** The purpose of this handout is to take the methods from the GRASS GIS digitization handout and employ them to extract data from raster cells and put them into a two-dimensional view to calculate slope or to better understand the shape of the valley. If you are in need of a refresher, refer back to the digitization workflow, as this picks up where that one left off. The concepts to consider in this lab are best portrayed by the following YouTube videos:

Sinuosity: https://youtu.be/rVs4MSw1e5s
Slope/Longitudinal Profile: https://youtu.be/SK2BFg2PmlQ

In the previous walkthroughs, we went over how to build locations and mapsets, and then how to digitize vector maps, add packages, and use new tools, such as the *v.centerline* tool, using GRASS GIS. The primary purpose of this walkthrough is to teach you how to extract raster cells, along a line, to plot, view, and analyze the data in your preferred plotting software (i.e. Microsoft Excel, Google Sheets, R, Matlab, etc.).

We will start off where the digitization handout concluded. Make sure you have the John Day DEM, Banks_Center vectors, Valley_Center vectors, and CHaMP data vector loaded into the Map Display. Cross sectional transects can be a really useful tool to examine the shape of a river valley as the river advances downstream. The shape of the valley can possibly tell us about the evolutionary history of the river, its potential for flooding, and even the potential threat of natural hazards, such as mass wasting events.

Fortunately, there is a tool in GRASS designed for this exact purpose! In the Modules tab, found in the Layers Manager window, search for the "**v.transects"** tool (Figure 1). For this tool, we will select the valley centerline vector we created. Recall that we created three different valley boundary datasets, so we have three separate valley centerlines. You could combine these, if you would like, and run the **v.centerline** tool again to get an average valley centerline. To do so, you would start a vector editing session, create a new vector, and copy→paste the centerlines from your three valley centerline vectors. Then you would run the v.centerline tool again. While this is not necessary, it may be useful if you have high variability in your centerline vectors.

Back in the v.transects tool, we will input our desired vector map, as well as a name for the output map. Then, we will choose a desired spacing between transects. This tool will generate transects along the line at whatever spacing interval we define, and will depend on the desired output. In this example, I chose a spacing of 50 m. Note that you can enter a unitless number here, as most tools that required distances inputs reference the default map unit if none are provided. Now switch to the optional tab. We need to specify the distance we want the transect lines to extend to either side of the line. In the example, I chose a distance of 250 m to either side. Make sure to also have the "Allow output files to overwrite existing files," as you may want to try different distances to get your preferred coverage. You will likely enter the same value for both sides, but, should you want to finetune these numbers, you should be able to click on your vector in the Map Display window and see arrows pointing in the direction the vector is oriented. Finally, make sure the input feature type, how your transect spacing is measured, and which line is the transect perpendicular to (the dropdown menus at the bottom of the Optional tab) are all set up how I have it in Figure 1. The you can **Run** the tool.

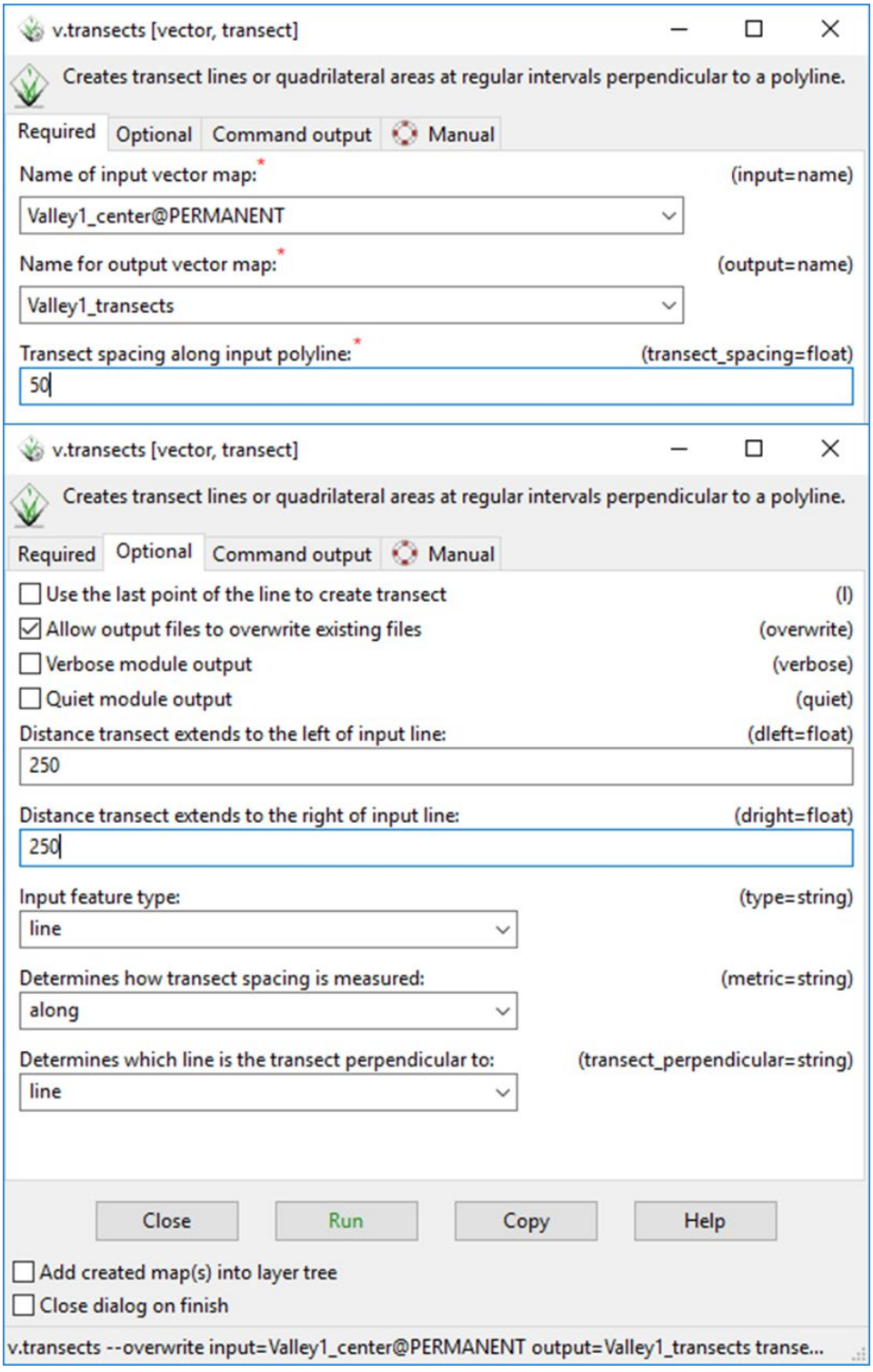

Figure 1. Transect generation tool window. The top window shows the Required tab, while the bottom window shows the Optional tab. Make sure to allow overwrite, so that you can quickly rerun the tool if you do not get the desired output.

The output from the v.transects tool will draw as many lines as will fit along the input line, starting with a transect at the initial vertex of the vector. Fortunately, we have a CHaMP data point close to one of the new transects! We will choose that one for further analyses. In the Map Display window, change to the vector digitizer, make sure the newly created transect feature is selected in the vector menu, and click on the Display/update categories tool, shown by the black arrow in Figure 2. Click on the transect closest to your CHaMP data point to see what the ID of the line is. You can see that the selected line is listed as Category = 7. This will be our reference when we extract that line from the dataset.

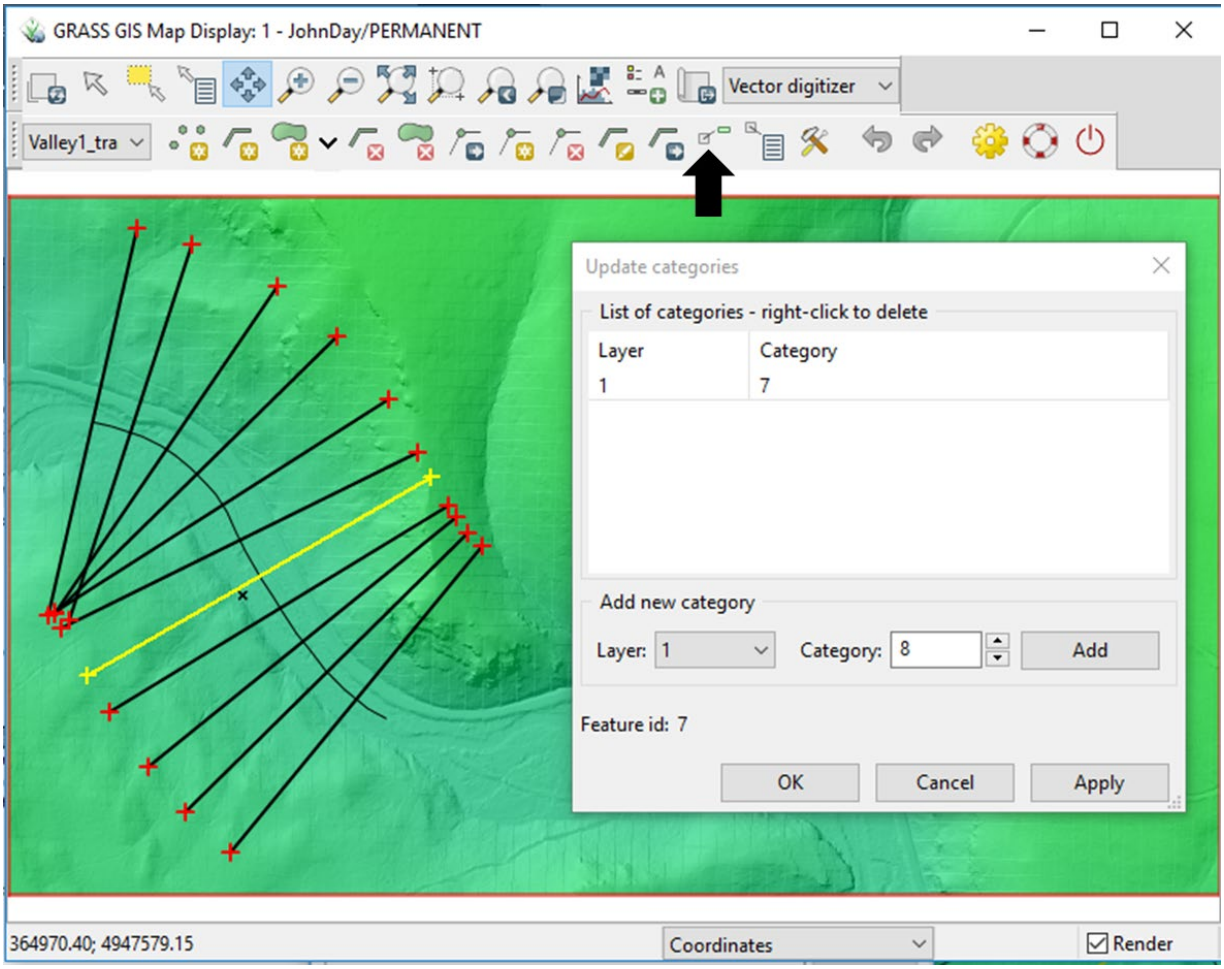

Figure 2. Output from the v.transect tool. Lines are drawn perpendicular to the input line, which is the valley centerline in this example. The black arrow shows the location of the Display/update categories tool. The Update categories window shows the vector feature category as **7**, which is unique to this line.

Next, flip back to the Modules tab and search for the **"v.extract"** tool. This tool lets you extract features from a vector dataset that meet a listed criteria. Follow the specifications in Figure 3 to set up your extraction, personalizing it for your dataset. The Selection tab allows you to list Category values to extract. In our example case, we will input the value 7, which we found with the Update/display categories tool. Run the tool.

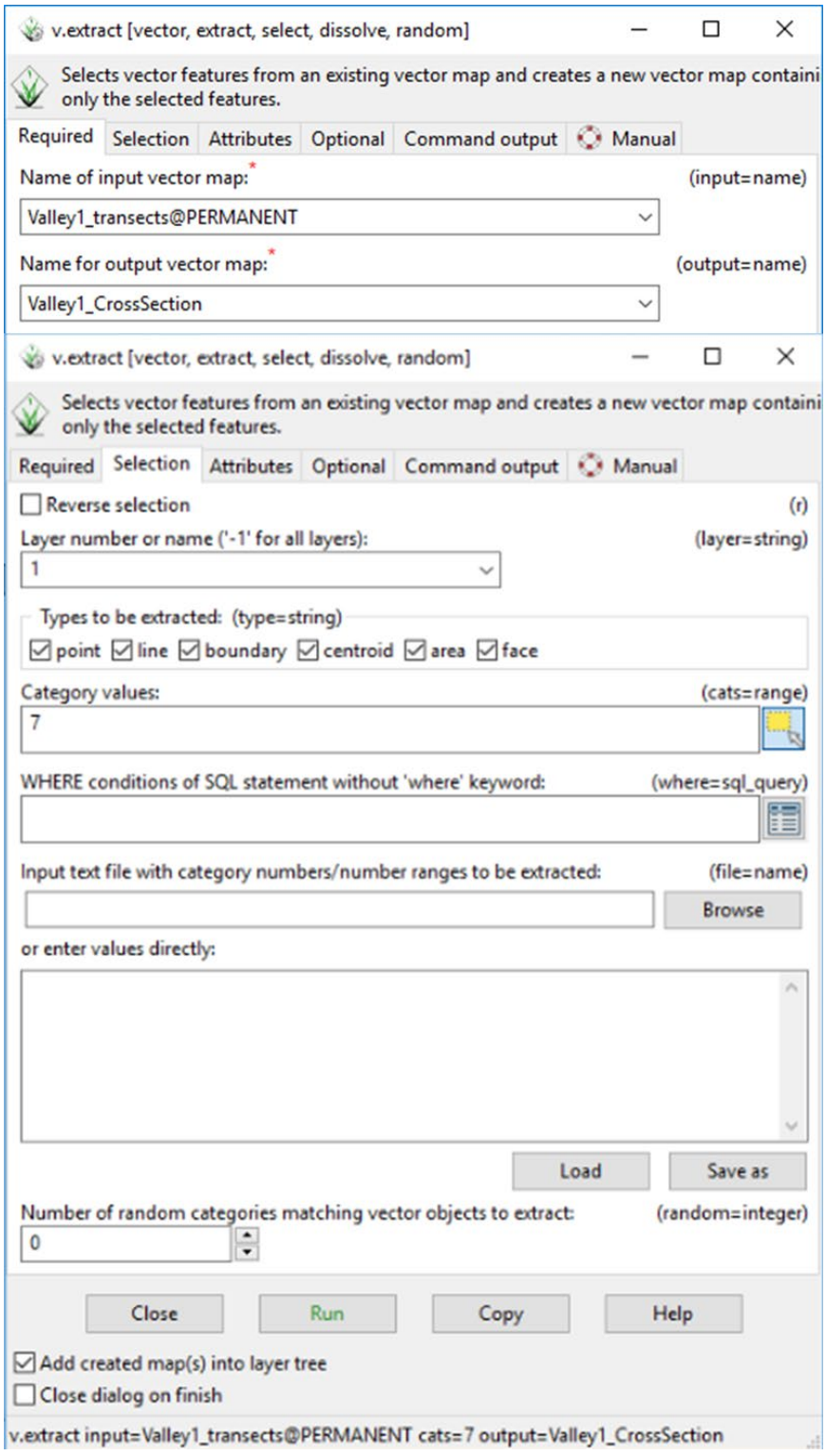

Figure 3. The vector extraction tool window. Like most of the other tools we have discussed, select the input variable and output name in the Required tab. Switching to the Selection tab, we can see just how extensive this tool is and just how much we can do to specify our selections.

Now that we have a single transect line to extract, we need to prepare that line for raster cell extract. This is where the goals of the two desired outputs in the handout converge. Recall that we want to look at the slope of the reach of river we are examining for the classification. We already have the bank centerline(s). With the valley cross sectional transect and the bank centerlines, we want to extract the raster cells that overlay, to plot and analyze in other software. The easiest way to do this is to use the lines to generate points at a specific density. Most GIS software has a way to do this. In GRASS, the **"v.to.points"** is the tool to use for this method (Figure 4). Search that tool in the Modules tab. In the Required tab, input the extracted variable and the desired output name. In the Optional tab, select the Interpolate points between line vertices option and input the maximum distance between points. The value you input here is up to you, but consider what the purpose is going to be. The resolution of our John Day LiDAR DEM is 1 m. You could input 1 as your maximum distance, to try and capture every cell that the lines intersect. This will result in a bulkier, but continuous dataset for future steps.

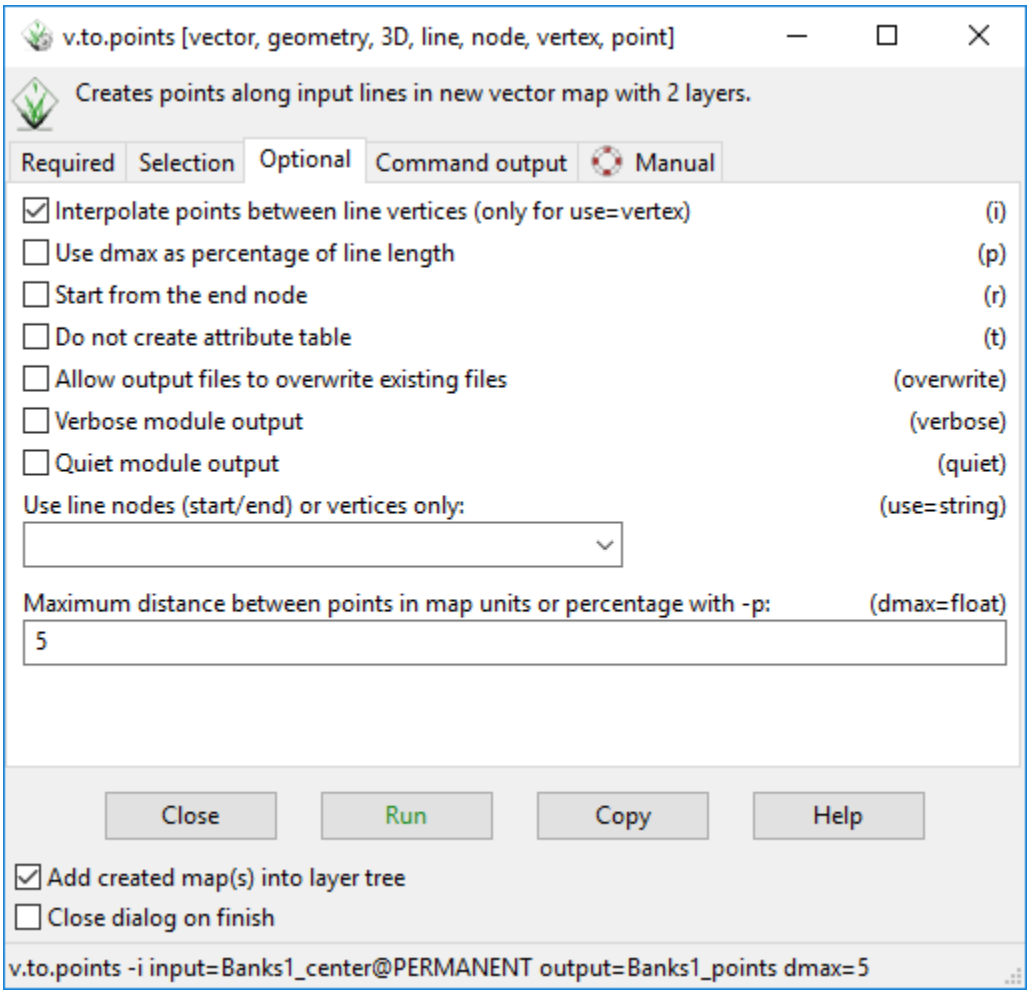

Figure 4. The specification window for the tool that generates points along a vector map. Make sure to create these points for your bank (river) centerlines and valley transect.

We are almost there! Using the **"v.drape"** tool, we can choose an input vector map to extract points from an elevation dataset (Figure 5). This is a pretty straightforward tool, for our purposes. Just input the input variable, output name, and elevation map and run the tool. The output will duplicate the input points, but append the attribute table with the height values for the cell that each point falls on.

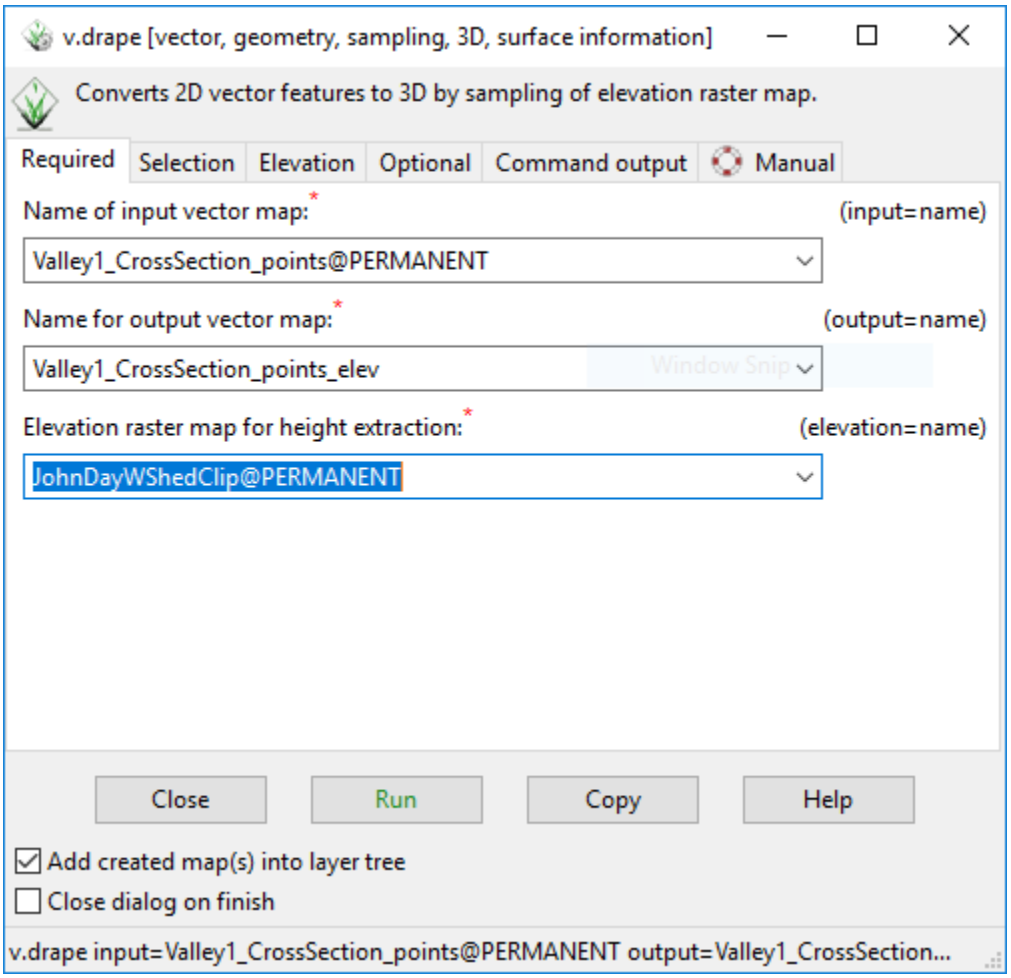

Figure 5. The input window for the v.drape tool. This is an easy tool to run, for our purposes, only requiring the point vector map we previously created and the elevation raster from which to extract the values from.

The final tool we need to use in GRASS is the **"v.out.ascii:** tool, which creates an output list of the attribute table for an input vector map (Figure 6). For this tool, select the points that were created with the v.drape tool (which have the elevation values we want). Run the tool and change to the Command Output tab. Copy the list that was generated. Paste it into a TEXT (.txt) file and save the file. Alternatively, you can go to the output tab and choose an output location and include a ".txt" at the end of your file name to save it as a text file. Now we are ready to import into our favorite plotting software and take a look at the data in a two-dimensional format.

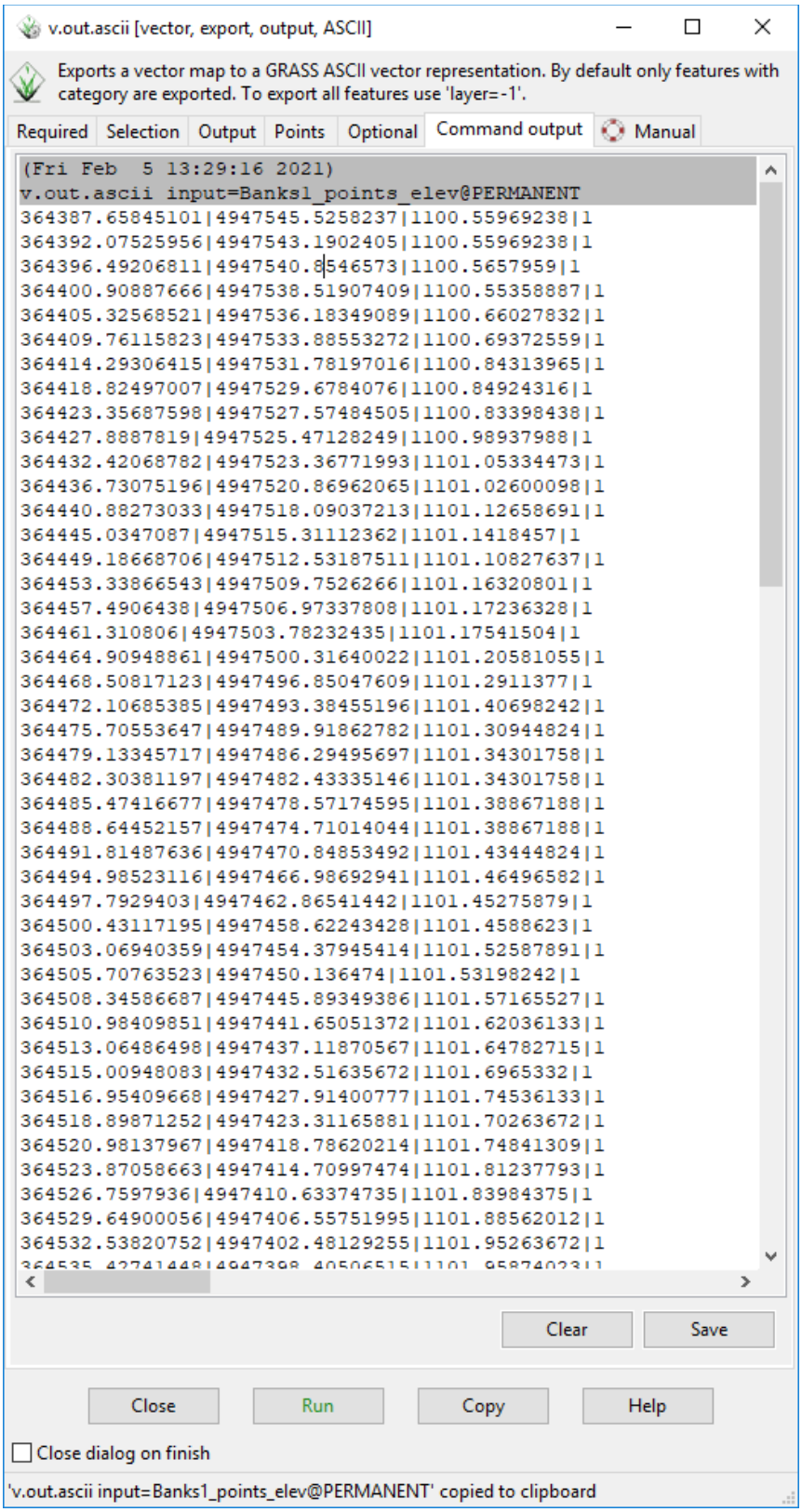

Figure 6. The list created from the attribute table of the point vector map with elevation values.

The rest of the viewing and analyzing will take place in the associated RStudio notebook, provided.

In this handout, we learned how to generate transects and convert lines into points, that we then used to extract values from a raster for further assessment. With this workflow, we are able to get the slope and sinuosity values needed for the Rosgen classification! The final workflow will walk you through how to use our dataset to classify your datapoint.