Going to the Sun Road Avalanches

Zachary Treisman

Overview

In this lab we are going to import a data set from a spreadsheet and use our visualization tools to work on understanding and representing the data. By the completion of this lab, you will be able to:

- Load a real-world csv dataset into R and inspect its structure.
- Use ggplot2, lubridate, and dplyr to explore avalanche timing, size, and type.
- Create a short briefing, with plots and a table, for a nontechnical audience.

As you work through this lab, take notes. Write down answers to the questions that I ask or anything else that you observe; you will use these in your briefing.

There are a couple of packages that we'll use today that you may not have installed.

- **lubridate** for handling dates and times
- **readr** for a more convenient way to load data
- **emdbook** for data and custom functions related to Bolker, Ecological Models and Data in R.

In the lower right hand corner click on the *Packages* tab. You can search for packages to install and load here. Alternatively you can type commands into the console.

```
install.packages("lubridate")
install.packages("readr")
install.packages("emdbook")
```

Remember that you only need to *install* packages once, but you need to *load* them each time you relaunch R. Start a new R script and load the following packages.

```
library(lubridate)
library(readr)
library(emdbook)
library(ggplot2)
library(dplyr)
```

Data

The point of working with R is to use actual data. However you acquire your data, it is easiest to get it in to R if you have it or can save it as a spreadsheet in comma separated variable (.csv) format. More on that in a moment. See https://datacarpentry.org/spreadsheet-ecology-lesson/ for a good lesson on organizing your data in a spreadsheet in preparation for analyzing it in R.

The data that we will use for this lab are here: GTSR Avalanche Occurrences 2003-2020.csv. Download it, find the downloaded file on your computer (it's probably either in Downloads or on your Desktop), and read on.

Project organization

Keeping your work organized will make your life easier. For each project (lab, seminar, thesis, experiment, et cetera) it is good practice to set up a folder on your computer as the project's *working directory*. Make a folder on your computer for this class, and in that folder, make a folder for this lab. Move the data that you downloaded earlier into this folder.

A good first task to always do upon saving a data set to a working directory: immediately save a copy of your data and change the name of the file to include *original_data*. Make a point of not changing that file. Now if you ever need to you can start any analysis over. So make a copy of the data file and name it *avalanches_original_data.csv*.

As your project gets bigger, you might find that it makes sense to create subdirectories of your working directory. Common examples are *data*, *images*, *figures*, *documents, etc.*

R needs to know your working directory. It is displayed at the top of the console window. You can set it with the command `setwd()`, or using the *More* menu on the *Files* tab in the lower right window. For example, I might use the command `setwd("C:/Classes/313/Avalanches")`.

Get in the habit of saving your script periodically. Click on the disk icon in your script window (the upper left) or choose Save from the File menu.

Optional: RStudio has an organizational system that can be handy. In the upper right there is a menu that says "Project: (None)". From this menu choose *New Project* and associate the

project with your working directory. This way, if you have multiple projects in your life, they don't clutter each other and you have some ability to customize how you work on each. You will have to reopen the script you just saved after you move to your new project.

Comma separated variable format

Often (ideally?) data are provided to us in spreadsheets. People often use programs like Excel to record data in a lab or fieldwork. One of the skills of a good Data Engineer is the ability to effectively query a database and return a useful "flat" dataset, where rows are observations and columns are variables. The value encoded in the cell (observation, variable) is the value of the specified variable during the specified observation. A great format for saving spreadsheets is *comma separated variable* or *csv*. When saving a spreadsheet in Excel, you can create a csv by choosing "Save As" from the File menu and then selecting csv as the format. This is probably the most common way that data is recorded and loaded into R. Using a simple format like csv instead of xlsx or another spreadsheet file format makes it easier to load the data into R, and less likely that future changes in technology will make the data unreadable.

When saving a multiple sheet Excel spreadsheet to csv, you have to save each sheet individually, and other than column headings and possibly metadata at the top of the sheet, there should be nothing other than the data, and no formatting such as dollar signs, units or commas in large numbers.

The Data: Avalanche activity along the Going to the Sun Road in Glacier National Park

Avalanche activity is actively monitored along the Going to the Sun Road in Glacier National Park. Here is the website of the program that does the monitoring. Understanding the science of avalanches is important for people responsible for protecting the safety of people, such as Park visitors and employees, and infrastructure, such as the road and buildings, that might be directly or indirectly affected by an avalanche. For a more in-depth and very accessible discussion of this story, see this site.

The observations in this data set are individual avalanches. Variables include the (sometimes approximate) date and time of the avalanche; its size and destructive force; and terrain features like the elevation, compass aspect and slope angle of the location where the avalanche started. Additionally the avalanches are classified by type - slab avalanches involve a cohesive slab of snow moving down the mountain versus loose avalanches where the snow does not have this cohesion. Avalanche type also distinguishes between wet and dry avalanches, describing the water content of the snow. Snow with higher water content - from snow that formed and fell at warmer temperatures, or because warmer temperatures partially melte the snow - is heavier than colder, drier snow.

Take a moment to think about the questions that you might want to ask these data. It might help to put yourself in the shoes of an employee of Glacier National Park who relies on this information to plan for activity in parts of the park accessed by this road. Imagine that your job is to help prepare a briefing that will influence when the road clearing crews get to work, how many resources they will have, and highlight areas of particular concern.

A very general question is: In what way is the destructive force of an avalanche a function of the other recorded variables? A more specific version of this might focus on the seasonal differences exhibited by slab and loose avalanches - for example can we show that large wet slab avalanches in the spring are potentially particularly destructive? A more specific question is: What sections of road are most prone to avalanches that require extensive clearing and repair? Knowing answers to these sorts of questions can be useful in planning for mitigation and maintenance work.

Loading the data

In order to answer these questions, we look at the data. In this case, it isn't necessary to examine the data in Excel first - the data are delivered to us in a clean format, ready to be loaded into R. Often, this this is not the case. It is sometimes easier to clean your spreadsheet in Excel. Be careful at this point in your process not to let Excel save the file as an xlsx file - depending on your settings Excel might try really hard to get you out of the csv format.

We will load the data using the command `read_csv`. In the *Files* tab on the lower right you can navigate to and click on the file you downloaded and it will open a dialog to import the data. You will see a preview of the data.

Now is a good time to tell R what sorts of variables your data have. Click on the drop-down arrow in the **Date** column header and select *Date*. The format '%m/%d/%Y' should match what you see in the preview. Similarly, change the type of the **Time** column to *Time*.

In the lower left of the dialog, you can change the name of the data to `avalanches`. In the lower right of the dialog, you should now see the command below. Paste it into your script, or run it with the *Import* button at the bottom of the dialog.

```
avalanches <- read_csv("GTSR Avalanche Occurrences 2003-2020.csv",
    col_types = cols(Date = col_date(format = "%m/%d/%Y"),
                     Time = col_time(format = "%H:%M:%S")))
```

Exploration

Now you have `avalanches` in your environment, and you can examine it using the `View` command, or by clicking on it in the upper right pane of RStudio. Note that there are 875 observations of 34 variables. Scroll through the and see if you can find some variables that

seem interesting or potentially useful. Ask a domain expert or the internet for some ideas if you are so inclined.

Now let's take a look at the data using some of the tools in R.

```
summary(avalanches) # always a good first step
```

Note that for many of the variables, there are a lot of `NA`s. These are missing data that for whatever reason was not recorded. For some variables, almost all of the observations are `NA`. You will want to be hesitant to use these variables, because for the most part, the information is not there.

Categorical variables

R has read all of the categorical variables in the data as `character` type. There is also a `factor` variable type that has some advantages over `character` for categorical variables. Which is preferable is a matter of debate. We can leave things as `character` for now.

Avalanche Type

One of the categorical variables is `AvalancheType`. Let's take a look at this variable, by making a table and a barplot.

```
table(avalanches$AvalancheType)
ggplot(avalanches, aes(AvalancheType))+geom_bar()
```

- Which two avalanche types are most common?

Here are the meanings of the abbreviations.

- GS: Glide Slab - an avalanche that breaks all the way to the ground and glides on a bed of rock.
- HS: Hard Slab - a very cohesive slab, often formed by wind-affected snow.
- LL: Loose - unconsolidated snow, like sluff from a cliff.
- SS: Soft Slab - a less cohesive avalanche, but one that still moves as a unit.
- WL: Wet loose - unconsolidated snow with high water content.
- WS: Wet Slab - a slab with high water content, more often an issue in the spring when temperatures are higher.

We see that the vast majority of the avalanches are either Glide Slab or Wet Loose avalanches. There are only 5 observations where this variable was not recorded - let's filter them out of our data.

```
sum(is.na(avalanches$AvalancheType))
avalanches <- avalanches %>%
  filter(!is.na(AvalancheType))
```

When do avalanches happen?

These data suffer from being a *convenience sample* - there are significant parts of the year when nobody is in the park observing avalanches. We can see that by drawing a histogram. We'll use the function `yday()` to get the day of the year out of the `Date` variable. For example, January 1 is the first day of the year, and February 10 is the 41st day of the year.

```
avalanches$yday <- yday(avalanches$Date)
ggplot(avalanches, aes(yday))+geom_histogram()
```

Just for fun, we can also add in the type of avalanche. Note that for barplots and boxplots and other geometries that cover an area on the screen, the aesthetic that you use for coloring in the area is `fill`, not `color`.

```
ggplot(avalanches, aes(yday, fill=AvalancheType))+geom_histogram()
```

This graphic becomes more intelligible if instead of `yday` we use `month`. Since the `month` variable is categorical, we use `geom_bar()` instead of `geom_histogram()`.

```
avalanches$month <- month(avalanches$Date, label = TRUE)
ggplot(avalanches, aes(x = month, fill = AvalancheType)) +
  geom_bar(position = "stack")
```

- What is the month with the highest number of recorded avalanches?

Does the answer change if we look at one type of avalanche or another? For that question, one thing we can do is create a table.

```
table(avalanches$AvalancheType, avalanches$month)
```

We can make a similar plot using time of day instead of the day of the year. We'll facet by `TimeAccuracy`. Since there are a lot of observations without `Time` recorded, we'll filter and create a copy of the data where the time is recorded.

```
sum(!is.na(avalanches$TimeAccuracy))
have_time <- avalanches %>%
  filter(!is.na(TimeAccuracy))
ggplot(have_time, aes(Time, fill=AvalancheType))+
  geom_histogram()+
  facet_wrap(~TimeAccuracy)
```

- Are avalanches more likely at any particular time of day?

Avalanche size

Let's look at `VerticalRunFeet` and `SizeDestructiveForce`. Not surprisingly, these are positively correlated. We can add regression lines, sorted by `AvalancheType`. The grey areas around the regression lines indicate the uncertainty in the line.

```
ggplot(avalanches, aes(VerticalRunFeet, SizeDestructiveForce,
                       color=AvalancheType))+
  geom_point()+
  geom_smooth(method = "lm")
```

That plot looks a bit crowded, maybe faceting would help. Also, try rotating the labels on the x-axis for readability.

```
ggplot(avalanches, aes(VerticalRunFeet, SizeDestructiveForce,
                       color=AvalancheType))+
  geom_point()+
  geom_smooth(method = "lm")+
  facet_wrap(~AvalancheType)+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

- What would the slope mean for these linear regressions?

Circular mappings

Sometimes we encounter variables that represent quantities measured by a circle instead of a line. For example, `StartZoneAspect` is the compass direction that the slope where an avalanche starts is facing. We can use `coord_polar()` to make plots in polar coordinates. We could have done this with `yday` if there were actually observations spread out through the year. This is also as good a time as any to introduce the `scale_x_continuous()` function for manipulating the axis.

```
ggplot(avalanches, aes(StartZoneAspect, SizeDestructiveForce))+
  geom_point()+
  facet_wrap(~AvalancheType)+
  coord_polar(start = 0, direction = 1) +  # Circular mapping
  scale_x_continuous(limits = c(0, 360), breaks = seq(0, 360, 45))
```

- Do you notice anything about the type or size of avalanches starting on different-facing slopes?

Analysis

Now, let's work on reshaping the data to answer the question about what parts of the road require the most attention.

Key avalanche paths

Often, you will want to use `group_by()` and `summarize()` to reshape the data to focus on particular variables. The following code creates a data set that groups the observations by path, and then computes the total time spent clearing the debris from each path, the average depth and length of the road burial, and an index that averages the `SizeDestructiveForce` for avalanches that hit the road. Calculating the index uses `(HitRoad=="Yes")*SizeDestructiveForce`. This works because logical values become 0 for `FALSE` and 1 for `TRUE` when you use them in an arithmetic expression. The `na.rm=T` are needed because the default behavior is for functions like `mean()` and `sum()` to return `NA` if any of the data are missing.

```
key_paths <- avalanches %>%
  group_by(PathName) %>%
  summarise(clearing_time = sum(TimeToClear, na.rm=T),
            mean_depth = mean(RoadBurialDepth, na.rm=T),
            mean_length = mean(RoadBurialLength, na.rm=T),
            road_index = mean((HitRoad=="Yes")*SizeDestructiveForce, na.rm=T),
            frequency = n()
            )
View(key_paths)
```

Using `key_paths`:

- Identify 2–3 paths with the largest `clearing_time`.
- Identify 2–3 with the largest `road_index`.
- Identify any path that is both frequent and highly destructive.

Assignment

Write a short informal briefing to a hypothetical road-crew supervisor, structured as:

- 2–3 bullet points naming specific paths and why they're high-priority (referencing clearing_time, road_index, frequency).
- 1–2 bullet points about when avalanches are most commonly recorded (by month and, if they got to it, time of day).
- 1–2 bullet points connecting avalanche type to destructive force or timing (e.g., "Wet slab avalanches in late spring tend to have high destructive force")

For example, you might say something like the following:

- The paths that appear to require the most attention are _____ and _____ because they have [highest clearing times / highest road_index / frequent road-hitting avalanches].
- Most recorded avalanches occur in the months of _____; crews might prioritize staffing in these months.
- Avalanches that hit the road tend to be of type _____ in season _____ and have average destructive force of about ___ (on the SizeDestructiveForce scale).

Aim for about 1 page of text plus 1–2 plots and one small summary table. As usual, if you want to work in groups of 2-3, this is encouraged. Everyone please turn in their own assignment on Canvas.