

Importing data in RStudio is fairly straightforward. However, you will need to know the type of file you want to import. Common file types are

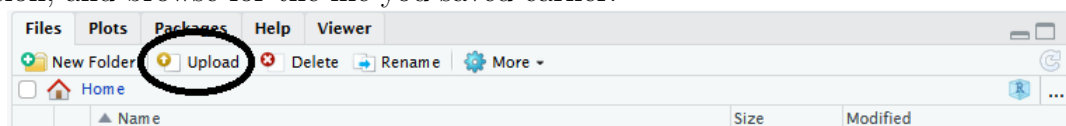
- Comma-separated Values (CSV) files (e.g. “filename.csv”) The data are basically a long list of numbers or other information where the entries in the different columns and rows are separated by commas. Most statistical software packages have the ability to read csv files
- Excel files (“filename.xls”). Microsoft Office’s spreadsheet software is ubiquitous in the business world.
- There are of course other types, but the two above should get you through most of your work in the real world.

Commands/Instructions needed for R:

(1) Once you have opened the RStudio interface, you can import data in two basic ways:

(a) **NOTE: If you are using the online version of RStudio,**

- (i) Download and save the data file to your desktop (or wherever you want).
- (ii) Go to the FILE tab in the bottom right-hand dialogue box, choose the UPLOAD option, and browse for the file you saved earlier.



(iii) Once you have uploaded the file (by clicking OK), the instructions above should work for you.

(b) Under the Environment tab, there is an option to Import Dataset. A dropdown menu will appear with about 6 options (for different file types).

(c) In the File menu, there is the Import Dataset with the same 6 options or so.

(2) **Importing a CSV file:**

(a) Access the dropdown menu for Import Dataset. For CSV files, you should use the option From text (readr)

(i) The first time you run this operation, RStudio may need to download and update some software packages. This took about 5 minutes on the online version of RS.

(b) Once you are in the Import Text Data window, you should see a preview of the data you want to import. There are a number of options at the bottom of the window you will want to review:

- **Name:** Change the name of the data, if desired. Note that when you use this data in the future, you will need to type in this name, so it might be a good idea to change its name if it has a long one.
- **Skip:** You can tell RStudio to skip the first N rows by typing N in this box. For example, if you type “10” in this box, it will omit the first 10 rows in the file. *This would include a row of labels, if there is one.*
- **First Row as Names:** If this is selected, then RS assumes that the first row of the data consists of data labels.
- **Delimiter:** Some file types use other symbols to separate values, like semicolons or simply spaces. If you need to, you can select the appropriate delimiter.

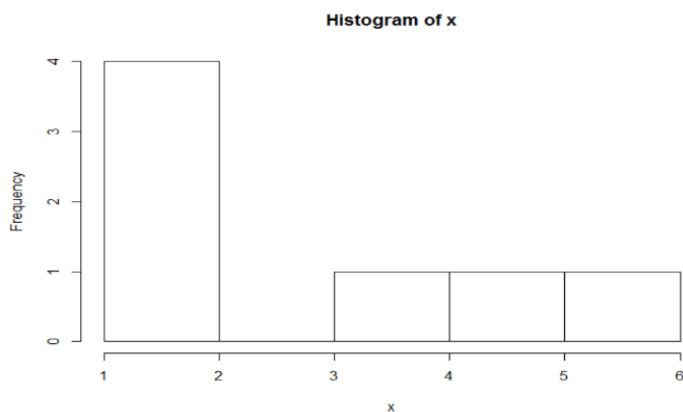
(3) **Importing an Excel file**

- (a) The process is basically the same as above, except that when you import, use the “From Excel” option
- (4) **Histograms:**
 The function `hist()` is used to create histograms. It has many possible arguments. The following are the most important:
- (a) `x` –The data you want to create a histogram for
 - (b) `breaks='<options>'` This allows you to set a method for creating bins. If you put in a number, that will be the width of your bins. You can also input formulas or other algorithms for classification. Leaving this argument blank will force `hist` to use the ‘pretty’ algorithm, which works fine.
 - (c) `freq=TRUE/FALSE` If omitted/TRUE, `hist` will create a proper histogram, with the *y*-axis giving the frequency of the bins. If set to FALSE **and** `breaks=1`, `hist` creates a relative frequency histogram. If bin sizes are larger than 1, you do NOT get a relative frequency histogram.
 - (d) `col='<colorname>'` Allows you to set the color of bars.
 - (e) `border='<colorname>'` Allows you to set the color of the border of the bars.
 - (f) `main, xlab, ylab = 'name'` These allow you to specify a name for the chart (`main`) and the *x* and *y* axes. You can specify none, some, or all of them, separated by commas.
 - (g) `xlim, ylim = c(<lower>,<upper>)` This allows you to fine-tune the range of your axes.

As an example, consider the following:

```
> x=c(1,1,2,2,4,5,6)
> hist(x)
```

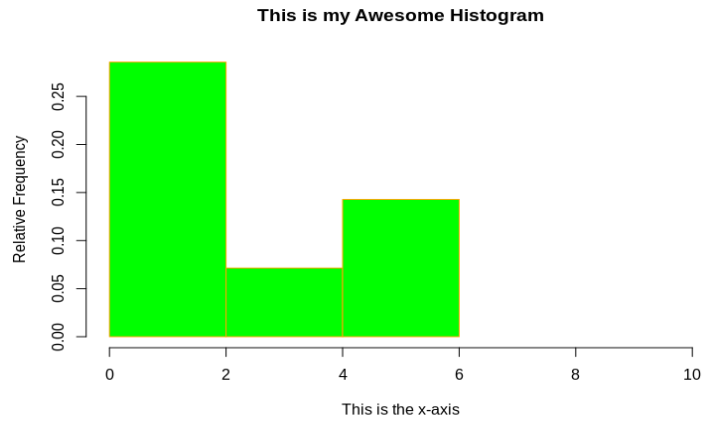
This is just `hist` with all the default settings; it produces the following graph:



However, if I now use the following command:

```
> hist(x,breaks=2,freq=FALSE,col="green",border="orange",
main="This is my Awesome Histogram", xlab="This is the x-axis",
ylab="Relative Frequency", xlim=c(0,10))
```

I get the following graph:



(5) **Calling a single column:**

To create a histogram from a single column of a larger table, your argument for `x` should be `tablename$columnname`

Problems:

Import the csv file “Abies_bals_allom_data_Peart”¹ included with this lab. Change the name of the data set to “Tree” (without the quotation marks).

- (1) Create a default histogram for just the HEIGHT column. Export the chart and insert the image into a word document.
- (2) Now adjust the histogram to have ALL of the following characteristics:
 - Fix the chart name and *x*-axis label. Chart names should be accurate and descriptive, while also being as short as possible. In this case, the data is the recorded heights of around 2000 trees, so a good title would be “Histogram of Tree Heights”
 - Change the colors of both the bars and their borders to colors you like better
 - Adjust the bin width to 5
 - Make the minimum *y*-value 0 and the maximum 600

Export the image and insert it into the Word file.

- (3) Repeat step (1) for CANOPYHEIGHT
- (4) Repeat step (2) for ELEVATION, except make the bin width 25.

¹Wood ZT, Peart DR, Palmiotto PA, Kong L, Peart NV (2015) Asymptotic allometry and transition to the canopy in *Abies balsamea*. *Journal of Ecology* 103(6): 1658-1666. <https://doi.org/10.1111/1365-2745.12465>