PreLab 3: Dendrograms and Scaling on Heatmaps, and Matrix Multiplication

Introduction to Data Mathematics 2021

Jared Gridley

Prelab Overview

In this prelab, we will:

- look in more depth at the heatmap command
- learn to do matrix algebra in R

Setting up the Prelab

Copy `PreLab3.Rmd` to your working directory `IDM_work`. Use this for your assignment. Do a practice "knit" to html before you begin.

First, read in `dietary_data_2005_complete.csv` and then convert it to our data matrix, as `heatmap.2()` requires a matrix input.
`heatmap.2()` is very slow for large matrices so we randomly pick 30 data points using the `sample_n` command. We do this as in Lab2 but we'll limit the variables considered to `fruit_raw`, `fruit_juice`, `gender`, and `education_level`

```
# Read in data from the data file; create a dataframe
raw.df <- read.csv('~/MATP-4400/data/dietary_data_2005_complete.csv')


# select just fruit_raw,fruit_juice,gender,education_level and make sure factors are ordered properly a
D.df <- raw.df %>% select(fruit_raw,fruit_juice,gender,education_level) %>%
  mutate(gender=as.numeric(factor(gender,levels=c("male","female")))) %>%
  mutate(education_level = as.numeric(factor(education_level,
                  levels=c("pre_highschool","highschool",
                           "highschool_grad","college","college_grad"))))

# Set the rownames to be 1 to number of data points
rownames(D.df) <- 1:nrow(D.df)

# To make it run fast, we are going to take a random sample of size 30 of the data. Remove this if you 
set.seed(30)

D.df <-slice_sample(D.df, n=30, replace=FALSE)

# add row names 1 to 30
rownames(D.df) <- 1:30

# make a matrix version
```

```
D.matrix<-as.matrix(D.df)
str(D.matrix)
```

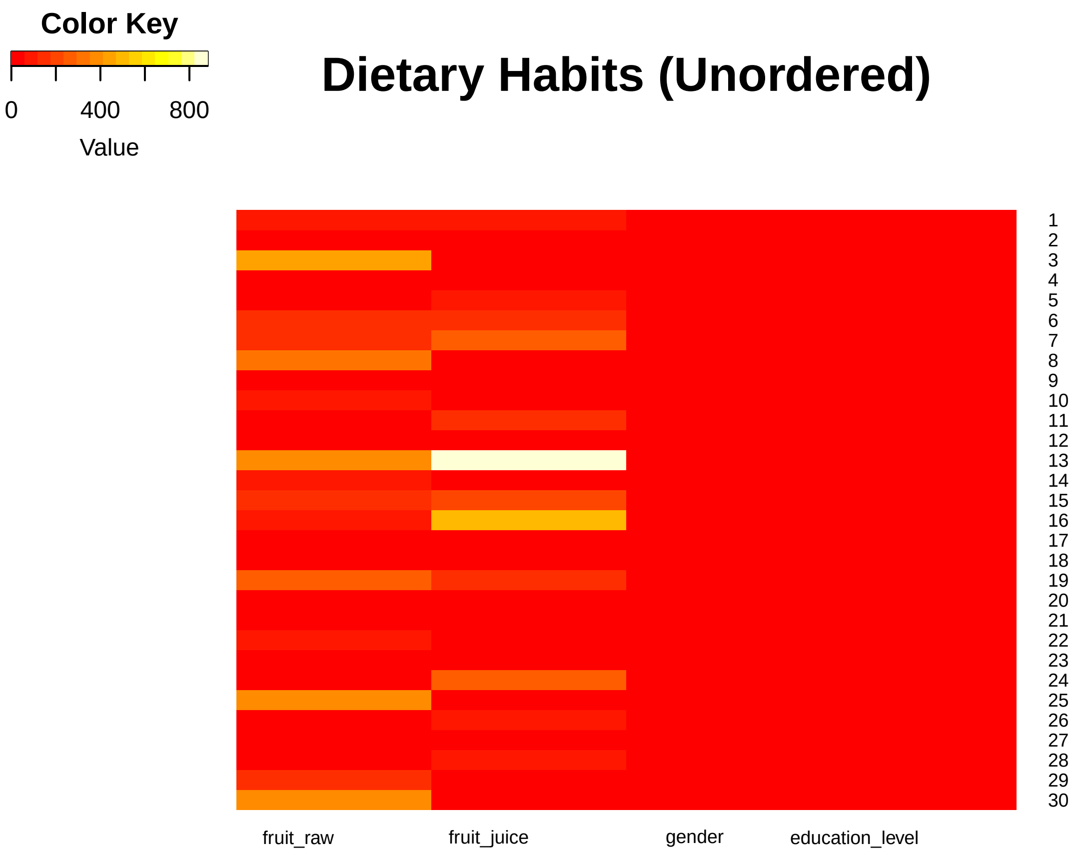
```
##  num [1:30, 1:4] 82.5 0 418.9 0 0 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:30] "1" "2" "3" "4" ...
##   ..$ : chr [1:4] "fruit_raw" "fruit_juice" "gender" "education_level"
```

Part 1: Dendrograms and Scaling with `heatmap.2`

A. Drawing a heatmap of a matrix with no scaling or reordering.

This command will draw the heatmap of the data as-is. Note there is no reordering of rows and columns and no scaling. If you get an error `Error in plot.new() : figure margins too large` the code is still working, but you may not see the whole graph in Rstudio and you should knit to html or pdf to see the final graph.

```
heatmap.2(D.matrix,
          main='Dietary Habits (Unordered)',
          dendrogram="none",
          Rowv=FALSE,  # Don't reorder rows
          Colv=FALSE,  # Don't reorder columns
          cexRow=0.75, # Make text smaller on rows
          cexCol=0.75, # Make text smaller on columns
          lhei= c(1, 3),  # row heights
          margins = c(1.5, 4),  #plot layout
          scale="none",  # Don't scale anything
          tracecol=NA,  # nonstandard heatmap features turned off
          srtCol = 0,  # This makes the col labels horizontal
          density.info='none') # nonstandard heatmap features turned off
```



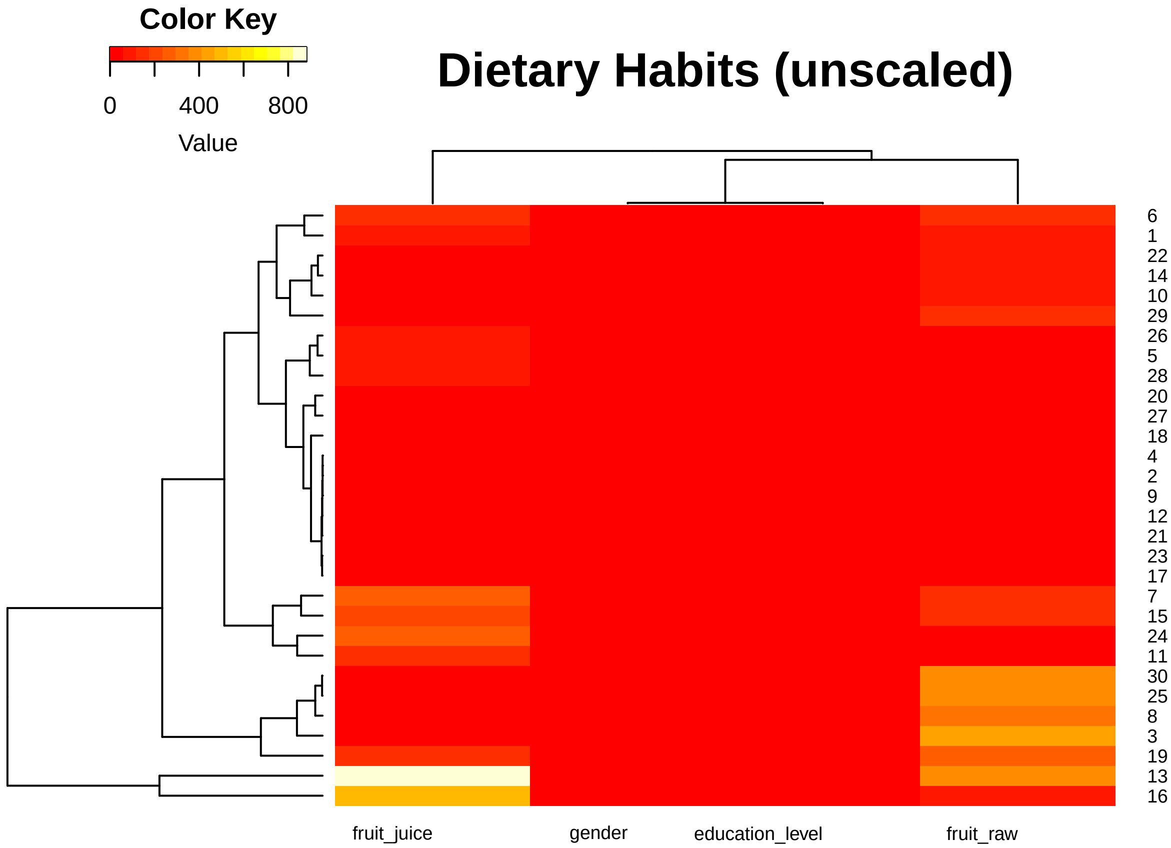
B. Practice generating a heatmap ordered by dendrogram.

Heatmaps are frequently far more effective for data visualization if we order the rows and columns.

To do this, we set the `dendrogram` argument to `both`. By default, `heatmap.2` uses a **complete linkage agglomerative clustering algorithm** on the columns to determine the order of the columns, and repeats again on the rows. For more information on complete linkage aglomerative clustering, see https://en.wikipedia.org/wiki/Complete-linkage_clustering. You may also watch this video for more information on dendrograms and clustering: https://www.youtube.com/watch?v=2z5wwyv0Zk4.

We also remove the `Rowv=FALSE` and the `Colv=FALSE`. These determine if rows and columns are reordered.

```r
heatmap.2(D.matrix,
          main='Dietary Habits (unscaled)',
          dendrogram="both",
          # This command scale by row or column  or none
          scale="none",
          # these are all plotting formating commands
          cexRow=0.75,
          lhei= c(1, 3),
          margins = c(1.5, 4),
          cexCol=0.75,
          #these are fancy extra plotting features that are a normal part of heatmaps
          tracecol=NA,
          srtCol = 0,
          density.info='none')
```



TRY IT Type `?heatmap.2` and read the documentation to see what the `dendrogram`, `Colv`, and `Rowv` options do. Try re-running heatmaps with different versions of the options. Make sure you understand what they do.

```
heatmap.2(D.matrix,
          main='Dietary Habits',
          dendrogram="column",
          # This command scale by row or column  or none
          Rowv = FALSE,
          Colv = TRUE,
          # these are all plotting formating commands
          cexRow=0.75,
          lhei= c(1, 3),
          margins = c(1.5, 4),
          cexCol=0.75,
          #these are fancy extra plotting features that are a normal part of heatmaps
          tracecol=NA,
          srtCol = 0,
          density.info='none')
```


```
#dendrogram controls the tree diagrams on the top and sides.
#Colv - controls scalling by the column variables, which is more useful for this
#Rowv - controls scalling by the row variables, not specific for the type of data tho.
```

C. Regenerate the heatmap on the scaled data.

Note that the scaled heatmap shows more variation for gender and education_level *because it scales all columns to have mean zero and variance one.*

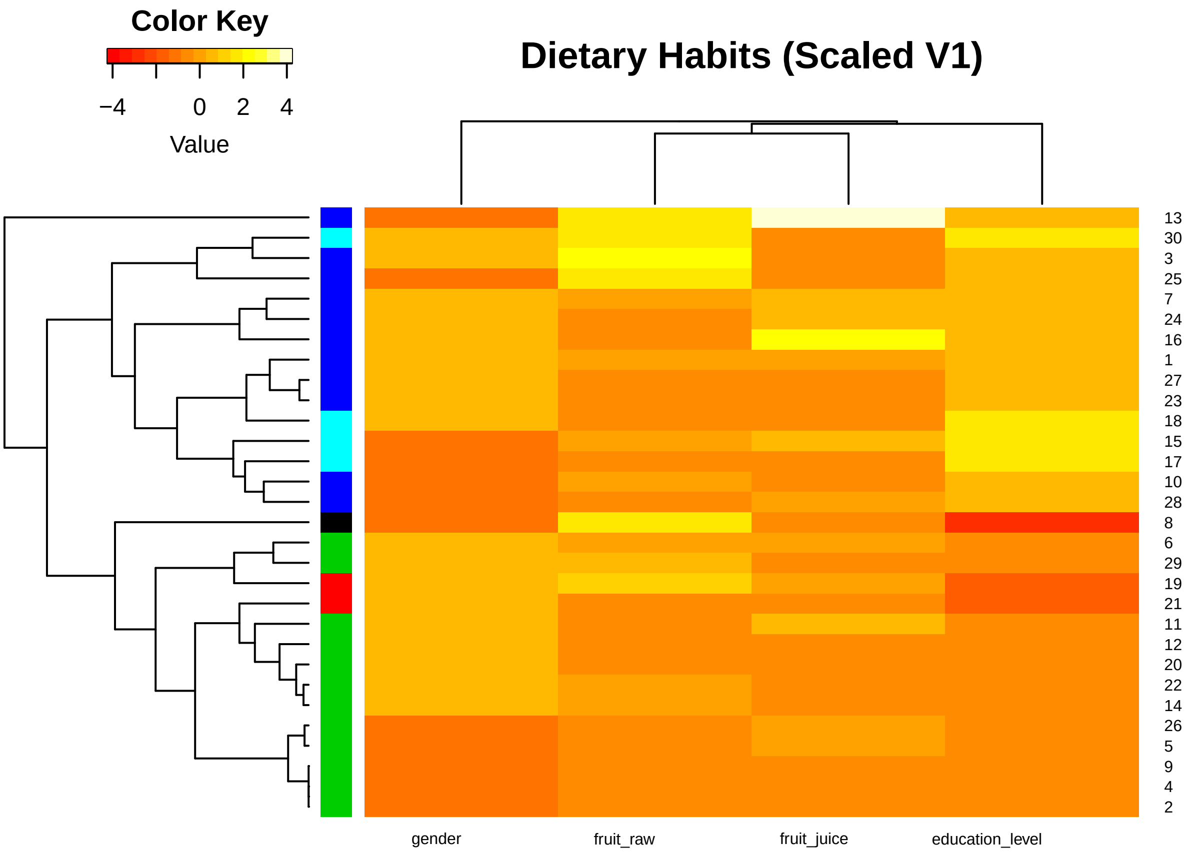
Exercise 1 Scale `D.matrix` using the `scale()` command, and save the results in the matrix `Dscale.matrix`. Use the command from **Part B** to create a heatmap of `Dscale.matrix` with dendograms of both the rows and columns. Title it `Dietary Habits (Scaled V1)`.

```
Dscale.matrix <- scale(D.matrix)

edu_level <- as.character(as.numeric(D.df$education_level))

heatmap.2(Dscale.matrix,
          main='Dietary Habits (Scaled V1)',
          dendrogram="both",
          # these are all plotting formating commands
          cexRow=0.75,
          lhei= c(1, 3),
          margins = c(1.5, 4),
          cexCol=0.75,
          RowSideColors = edu_level,
          #these are fancy extra plotting features that are a normal part of heatmaps
          tracecol=NA,
```

```
        srtCol = 0,
        density.info='none')
```

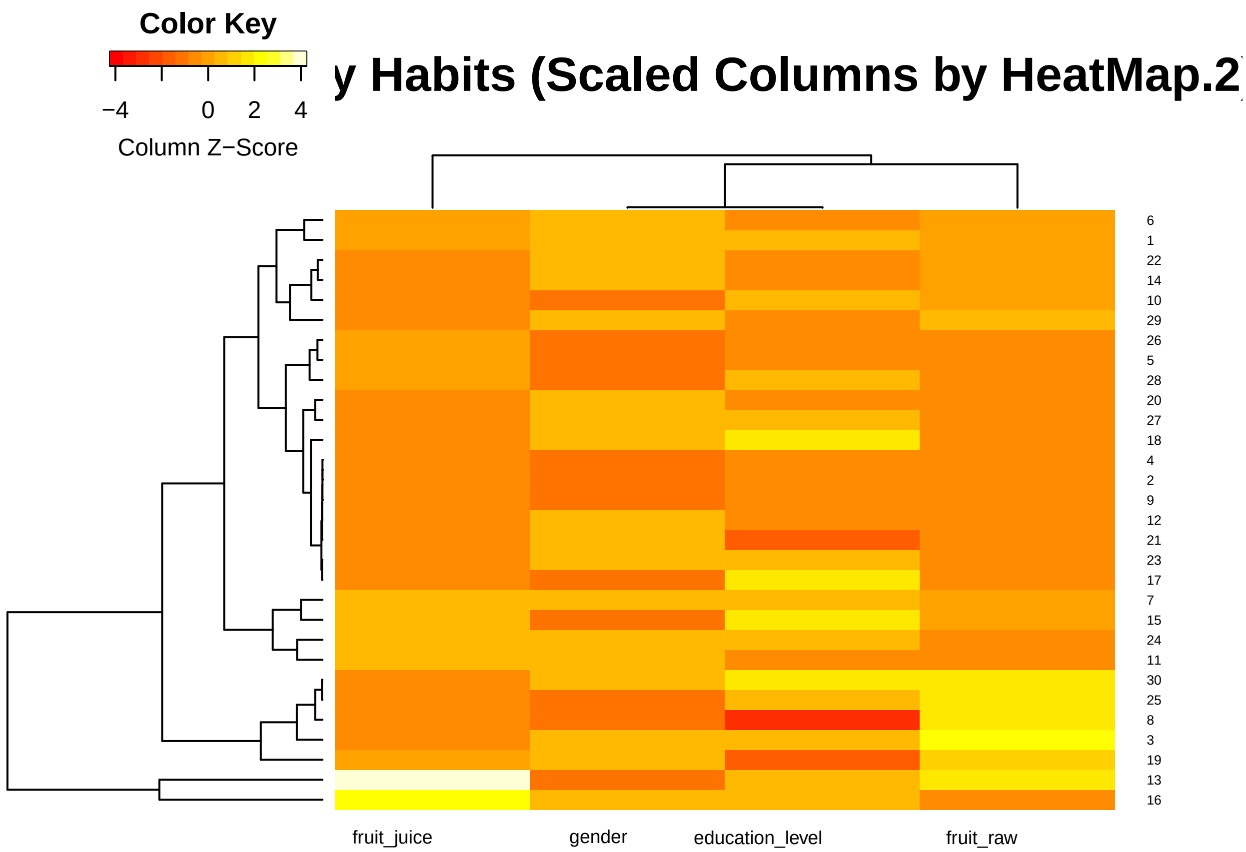


Be prepared to answer questions about this visualization for your Prelab4

D. Using the Heatmap.2 scale function.

WARNING: USING heatmap.2 scale function to columns is not the same thing as you scaling data and then applying heatmap.2. In theory, one might expect this to be true.

```
heatmap.2(D.matrix,
        main='Dietary Habits (Scaled Columns by HeatMap.2)',
        dendrogram="both",
        cexRow=0.5,
        cexCol=0.75,
        lhei= c(1, 3),
        margins = c(1.5, 4),
        scale="column",
        tracecol=NA,
        srtCol = 0,
        density.info='none')
```



Compare the clustering of the rows and columns to the heatmap created in **Part B?** This is better than the scaling that we saw in Part B, this more accurately depicts the differences between the column values. As can be seen by the larger range of colors compared to primarily red.

Compare the clustering of the rows and columns to the heatmap created in **Part C?** Part C is better than the clustering we saw done automatically by the heatmap function. In part C the rows are more distinct where in the heatmap scaling, some of the differences are indistinguishable.

In theory, we might expect this heatmap to be the same as the one in Part C, but it is not. This is because the dendogram is generated using the unscaled data by default and the scaling is only used for drawing the heatmap. Dr. Bennett thinks this is a very goofy option (or maybe even a bug). Dr. Bennett recommends ** always scale data yourself **. Then you know exactly what you are getting.

Part 2: Matrix Operations in R

This part of the lab will guide you through how to perform matrix operations in R. First we create some matrices:

```
# Make some matrices
U <- matrix(c(1,2,3,4,5,6),nrow=2)
U
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
V <- matrix(c(-1,4,7,0,-1,3),nrow=2)
V
```

```
##      [,1] [,2] [,3]
## [1,]   -1    7   -1
## [2,]    4    0    3
```

```r
Q <- matrix(c(1,2,3),ncol=1)
Q
```

```
##      [,1]
## [1,]    1
## [2,]    2
## [3,]    3
```

```r
S <- matrix(c(6,5),nrow=1)
S
```

```
##      [,1] [,2]
## [1,]    6    5
```

- `U` and `V` are matrices.
- Note that `Q` is a matrix *and also a column vector*.
- Observe that `S` is a matrix *and also a row vector*

R uses `%*%` to perform matrix multiplication and `*` to perform element-wise scalar multiplication. Practice matrix algbera in R by running each of the following commands and examining the results.

- Add matrices: `U+V`
- Subtract matrices: `U-V`
- Multiply by scalar: `3*U`
- Divide by scalar: `V/2`
- Multiply matrices: `U%*%Q`

- Multiply elements of matrix: `U*V`
- Transpose elements of matrix: `t(U)`
- Multiply S two different ways S'S: `t(S)%*%S`
- Multiple S two different ways SS': `S%*%t(S)`

```r
# Do the operations given above here
#Add matrices
U+V
```

```
##      [,1] [,2] [,3]
## [1,]    0   10    4
## [2,]    6    4    9
```

```r
#Subtract matrices
U-V
```

```
##      [,1] [,2] [,3]
## [1,]    2   -4    6
## [2,]   -2    4    3
```

```r
#Multiply by scalar
3*U
```

```
##      [,1] [,2] [,3]
## [1,]    3    9   15
## [2,]    6   12   18
```

```r
#Divide by scalar
V/2
```

```
##      [,1] [,2] [,3]
## [1,] -0.5  3.5 -0.5
## [2,]  2.0  0.0  1.5
```

```
#Multiply matrices
U%*%Q
```

```
##      [,1]
## [1,]   22
## [2,]   28
```

```
#Multiply elements of matrix
U*V
```

```
##      [,1] [,2] [,3]
## [1,]   -1   21   -5
## [2,]    8    0   18
```

```
# Transpose elements of matrix
t(U)
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
## [3,]    5    6
```

```
# Multiply S two different ways S'S
t(S)%*%S
```

```
##      [,1] [,2]
## [1,]   36   30
## [2,]   30   25
```

```
# Multiple S two different ways SS'
S%*%t(S)
```

```
##      [,1]
## [1,]   61
```

TRY IT Make up some matrices and experiment with linear algebra in R.

```
J <- matrix(c(86,95,8,5,4,7,14,25,3,6,2,5,48,15,24,86,7,58), ncol = 6)
J
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]   86    5   14    6   48   86
## [2,]   95    4   25    2   15    7
## [3,]    8    7    3    5   24   58
```

```
num_col <- matrix(c(1,1,1), ncol = 3)
sum_col <- num_col%*%J
sum_col
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]  189   16   42   13   87  151
```

```
avg_col <- sum_col *(0.33)
avg_col
```

```
##       [,1] [,2]  [,3] [,4]  [,5]  [,6]
## [1,] 62.37 5.28 13.86 4.29 28.71 49.83
```

In R, when accessing a single column of a matrix, for instance `Q[,1]`, R automatically coerces the result to a vector of class `numeric` so that it is no longer a matrix. This may be undesirable in certain contexts when we would prefer for the `Q[,1]` to remain a (single-column) matrix. To work around this, you may use the syntax `Q[,1,drop=FALSE]`.

TRY IT Get the second column of `V`, but force the result to remain as a single-column matrix.

```
#Doesn't convert to a vector
V[,2, drop=FALSE]
```

```
##      [,1]
## [1,]    7
## [2,]    0
```

Exercise 2 Find the product QQ^T using R code. What is the dimension of the result?

```
Q%*%t(Q)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    2    4    6
## [3,]    3    6    9
```

```
#Result is a 3x3 matrix
```

Save PreLab3.Rmd to your account directory and knit it to pdf.

** You've now completed Prelab3! Go to LMS and complete the online quiz **

Appendix

Summary of useful R functions:

- `heatmap.2()` - Makes a heat map of a matrix. Use the dendrogram argument to create a dendrogram via clustering by either row, column, both, or neither. The scale argument is used to scale either by column, row, or neither. Below, dendrogram and scale are shown for column.
  - `heatmap.2(A,scale='column', dendrogram='column')`
- `t()` - transpose a matrix
  - `t(A)`
- `%*%` - matrix multiplication
  - `A %*% B` - Matrix product of `A` and `B`.
- `+` - (elementwise) addition and `-` - (elementwise) subtraction
  - `A + B` - subtract two matrices elementwise
  - `u - v` - subtract two vectors elementwise
- `*` - (elementiwise) scalar multiplication. NOTE: Does not do matrix multiplication.
  - `s*t` - multiply two vectors elementwise.
  - `A*S` - multiply two matrices elementwise.
- `/` - (elementwise) scalar division.
  - `s/t` - divide vector `s` by vector `t` elementwise
  - `A/a` - divide each element of matrix `A` by a scalar `a`.
  - `s/a` - divide each element of vector `s` by a scalar `a`.