test

Me

Write out the SVD using `Eqn()` and `symbolicMatrix()`. In Rmd, `Eqn()` can be given an equation label. Both of these equations are numbered. (In this example, using `echo = TRUE` gives output that intersperses the code with the equation output for some reason.)

```
Eqn("X=U \\lambda V", label='eqn:svd')
Eqn(symbolicMatrix("u", "n", "k", lhs = ''),
    symbolicMatrix("\\lambda", "k", "k", diag=TRUE),
    symbolicMatrix("v", "k", "p", transpose = TRUE))
```

produces:

$$X = U\lambda V \tag{1}$$

$$= \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1k} \\ u_{21} & u_{22} & \cdots & u_{2k} \\ \vdots & \vdots & & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{nk} \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_k \end{pmatrix} \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1p} \\ v_{21} & v_{22} & \cdots & v_{2p} \\ \vdots & \vdots & & \vdots \\ v_{k1} & v_{k2} & \cdots & v_{kp} \end{pmatrix}^\top \tag{2}$$

This can be referenced: As seen in Equation 1 ...

aligned

You can also align the equations. Note the locations of the `&` for alignment.

Show the singular value decomposition:

```
Eqn("\\mathbf{X} & = \\mathbf{U} \\mathbf{\\Lambda} \\mathbf{V}",
    Eqn_newline(),
    symbolicMatrix("u", "n", "k", lhs = '&'),
    symbolicMatrix("\\lambda", "k", "k", diag=TRUE),
    symbolicMatrix("v", "k", "p", transpose = TRUE),
    align=TRUE)
```

$$\mathbf{X} = \mathbf{U}\,\mathbf{V} \tag{3}$$

$$= \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1k} \\ u_{21} & u_{22} & \cdots & u_{2k} \\ \vdots & \vdots & & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{nk} \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_k \end{pmatrix} \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1p} \\ v_{21} & v_{22} & \cdots & v_{2p} \\ \vdots & \vdots & & \vdots \\ v_{k1} & v_{k2} & \cdots & v_{kp} \end{pmatrix}^\top \tag{4}$$

matrix2latex

`matrix2latex()` can also generate symbolic equations from numeric or character matrices.

Create character matrix:

```
A <- matrix(paste0('a_', 1:9), 3, 3)
b <- paste0("\\beta_", 1:3)
```

Show $[\mathbf{A}|\mathbf{b}]$:

```
matrix2latex(cbind(A,b)) |> Eqn(number=FALSE)
```

$$\left[\begin{array}{cccc} a_1 & a_4 & a_7 & \beta_1 \\ a_2 & a_5 & a_8 & \beta_2 \\ a_3 & a_6 & a_9 & \beta_3 \end{array}\right]$$

showEqn

`showEqn()` can also write LaTeX, but writes out the equations. However, but the `array` environment needs to be included inside `$$ ... $$` to be evaluated in a chunk, so this chunk is not evaluated (it causes a LaTeX error).

Perhaps look at `showEqn.R` to see if this can be fixed.

```
A <- matrix(paste0("a_{", outer(1:3, 1:3, FUN  = paste0), "}"),
            nrow=3)
b <- paste0("b_", 1:3)
x <- paste0("x", 1:3)
```

```
showEqn(A, b, vars = x, latex=TRUE)
```

It would produce:

```
\begin{array}{lllllll}
a_{11} \cdot x_1 &+& a_{12} \cdot x_2 &+& a_{13} \cdot x_3  &=&  b_1 \\
a_{21} \cdot x_1 &+& a_{22} \cdot x_2 &+& a_{23} \cdot x_3  &=&  b_2 \\
a_{31} \cdot x_1 &+& a_{32} \cdot x_2 &+& a_{33} \cdot x_3  &=&  b_3 \\
\end{array}
```

Evaluating the above code in an unnumbered LaTeX math environment via `Eqn()`:

```
showEqn(A, b, vars = x, latex=TRUE) |> Eqn(number=FALSE)
```

$$
\begin{array}{lllllll}
a_{11} \cdot x_1 &+& a_{12} \cdot x_2 &+& a_{13} \cdot x_3 &=& b_1 \\
a_{21} \cdot x_1 &+& a_{22} \cdot x_2 &+& a_{23} \cdot x_3 &=& b_2 \\
a_{31} \cdot x_1 &+& a_{32} \cdot x_2 &+& a_{33} \cdot x_3 &=& b_3 \\
\end{array}
$$