center():

This module uses the linear constraints to generate the transformation matrix, $\mathbf{Z}$.

Typically, the linear constraints can be written in the form of $\mathbf{A}\boldsymbol{\beta} = \mathbf{0}$, where $\boldsymbol{\beta}$ is a $K \times 1$ vector of parameters to be estimated, and $\mathbf{A}$ is a $L \times K$ matrix of linear constraints. So there are L linear constraints and each constraint is a linear combination of the K parameters.

With the L constraints in effect, we are only left with $K - L$ parameters to estimate. So it is desirable to come up with a $K \times (K - L)$ transformation matrix $\mathbf{Z}$ that lets $\boldsymbol{\beta} = \mathbf{Z}\boldsymbol{\beta}_z$, where $\boldsymbol{\beta}_z$ is a $(K - L) \times 1$ vector of parameters and $\mathbf{A}\boldsymbol{\beta} = 0$ for any $\boldsymbol{\beta}_z$.

In this way, we can simply replace $\boldsymbol{\beta}$ with $\mathbf{Z}\boldsymbol{\beta}_z$ in the estimating equation and then solve $\boldsymbol{\beta}_z$. Usually $\mathbf{Z}$ is incorporated into the design matrix and the form of solution is preserved.

The $\mathbf{Z}$ matrix can be obtained from QR decomposition of $\mathbf{A}^T$, which gives a $K \times K$ orthogonal matrix $\mathbf{Q}$. The last $K - L$ columns provide an orthonormal basis for the null space of $\mathbf{A}^T$, and we can use them as the $\mathbf{Z}$ matrix, because $\mathbf{Z}^T\mathbf{A}^T = \mathbf{0}$. In this way, with any $\boldsymbol{\beta}_z$, $\mathbf{A}\boldsymbol{\beta} = \mathbf{AZ}\boldsymbol{\beta}_z = \mathbf{0}\boldsymbol{\beta}_z = \mathbf{0}$.

The design matrix under the reduced parameter vector $\boldsymbol{\beta}_z$ is $\mathbf{XZ}$, where $\mathbf{X}$ is the original design matrix.

Arguments:

(1) **c**: input to the QR decomposition, $\mathbf{A}^T$.

(2) nr, nc: K and L, respectively.

(3) **z**: the matrix $\mathbf{Z}$, as the output.

penalty():

This module constructs the 'half' penalty matrix, $\mathbf{B}$, of a first order difference penalty matrix.

It has the following form,

$$\mathbf{B} = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -1 \end{pmatrix}.$$

In this way, $\mathbf{B}\boldsymbol{\beta}$ is a column vector of the first order difference between adjacent parameter values in $\boldsymbol{\beta}$, and $\boldsymbol{\beta}^T\mathbf{B}^T\mathbf{B}\boldsymbol{\beta}$ is the sum of squared differences. **Note that B is not a square matrix, but adding a row of zeros at the end will make it a square matrix. However, this does not change the calculation. So we do not add the row of zeros here.**

The first order difference penalty matrix sums up the squared difference between adjacent parameter values in a spline coefficient vector.

Arguments:

(1) **d**: the dimension of $\boldsymbol{\beta}$.