A Notes on the R function solve.QP

This notes is to explain the `solve.QP` function used in `r-codeCAPM.pdf`:

```
solve.QP(Dmat=2*sigma,dvec=rep(0,3),Amat=cbind(rep(1,3),mu),bvec=c(1,muP[i]),meq=2)

solve.QP(Dmat=2*sigma,dvec=rep(0,3),Amat=cbind(rep(1,3),mu,diag(1,3)),
bvec=c(1,muP[i],rep(0,3)),meq=2)
```

The QP in the function name stands for quadratic programming, which is an optimization problem where we want to

$$\text{minimize} \quad \frac{1}{2}\boldsymbol{b}^T\mathbf{D}\boldsymbol{b} - \boldsymbol{d}^T\boldsymbol{b}$$

subject to the constraint

$$\mathbf{A}^T\boldsymbol{b} \geq \boldsymbol{b}_0.$$

Here, I am using the same notation as in the documentation of the `solve.QP` function. The $\boldsymbol{b}$ is our $\boldsymbol{w}$ to be solved. In our problem, we want to solve the Markovitz problem:

$$\text{minimize} \quad \boldsymbol{w}^T\Sigma\boldsymbol{w}$$

subject to

$$\begin{aligned} \sum_{j=1}^{J} w_j &= 1 \\ \sum_{j=1}^{J} w_j\mu_j &= \mu. \end{aligned}$$

By writing the constraints into matrix form, we have

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ \mu_1 & \mu_2 & \dots & \mu_J \end{pmatrix} \boldsymbol{w} = \begin{pmatrix} 1 \\ \mu. \end{pmatrix}.$$

Therefore, we recognize that

(1) `Dmat`: $\mathbf{D} = 2\boldsymbol{\Sigma}$.

(2) `dvec`: $\boldsymbol{d} = \mathbf{0}$.

(3) `Amat`: $\mathbf{A} = \begin{pmatrix} 1 & \mu_1 \\ \vdots & \vdots \\ 1 & \mu_J. \end{pmatrix}.$

(4) `bvec`: $\boldsymbol{b}_\mathbf{0} = \begin{pmatrix} 1 \\ \mu. \end{pmatrix}.$

(5) `meq`: this means the first `meq` constraints are treated as equality constraints, all further as inequality constraints. Here, we have two equality constraints, so we set `meq=2`.

The case with short-sale not allowed means we have the following additional constraints:

$$w_j \geq 0, \quad \text{for j=1, \ldots, J.}$$

These constraints can be written as

$$\begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & 0 \\ 0 & 0 & \ldots & 1. \end{pmatrix} \boldsymbol{w} \geq \mathbf{0}.$$

Hence, `Amat` and `bvec` become, respectively,

$$\mathbf{A} = \begin{pmatrix} 1 & \mu_1 & 1 & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots & \\ 1 & \mu_J & 0 & \ldots & 1. \end{pmatrix} \quad \text{and} \quad \boldsymbol{b}_0 = \begin{pmatrix} 1 \\ \mu \\ 0 \\ \vdots \\ 0. \end{pmatrix}.$$

Finally, `meq` should still be 2, meaning the first two constraints are equality constrains and the rest are inequality constraints.