

```
In [1]: covar=eye(5)
```

```
Out[1]: 5x5 Array{Float64,2}:  
 1.0  0.0  0.0  0.0  0.0  
 0.0  1.0  0.0  0.0  0.0  
 0.0  0.0  1.0  0.0  0.0  
 0.0  0.0  0.0  1.0  0.0  
 0.0  0.0  0.0  0.0  1.0
```

```
In [2]: for i = 1:5  
        for j=1:5  
            if i==j  
                covar[i,j]=1.0  
            else  
                covar[i,j]=0.5  
            end  
        end  
    end
```

```
In [3]: covar
```

```
Out[3]: 5x5 Array{Float64,2}:  
 1.0  0.5  0.5  0.5  0.5  
 0.5  1.0  0.5  0.5  0.5  
 0.5  0.5  1.0  0.5  0.5  
 0.5  0.5  0.5  1.0  0.5  
 0.5  0.5  0.5  0.5  1.0
```

```
In [4]: precision=inv(covar)
```

```
Out[4]: 5x5 Array{Float64,2}:  
 1.66667  -0.333333  -0.333333  -0.333333  -0.333333  
 -0.333333  1.66667  -0.333333  -0.333333  -0.333333  
 -0.333333  -0.333333  1.66667  -0.333333  -0.333333  
 -0.333333  -0.333333  -0.333333  1.66667  -0.333333  
 -0.333333  -0.333333  -0.333333  -0.333333  1.66667
```

```
In [5]: meanVec=rand(5)
```

```
Out[5]: 5-element Array{Float64,1}:  
 0.0973964  
 0.759355  
 0.0674482  
 0.391715  
 0.91604
```

```
In [6]: \ (covar, meanVec)
```

```
Out[6]: 5-element Array{Float64,1}:  
 -0.549192  
 0.774725  
 -0.609088  
 0.0394449  
 1.08809
```

```
In [7]: covar
```

```
Out[7]: 5x5 Array{Float64,2}:  
 1.0  0.5  0.5  0.5  0.5  
 0.5  1.0  0.5  0.5  0.5  
 0.5  0.5  1.0  0.5  0.5  
 0.5  0.5  0.5  1.0  0.5  
 0.5  0.5  0.5  0.5  1.0
```

```
In [8]: lw=chol(covar,:L)
```

```
Out[8]: 5x5 Triangular{Float64,Array{Float64,2},:L,false}:  
 1.0  0.0  0.0  0.0  0.0  
 0.5  0.866025  0.0  0.0  0.0  
 0.5  0.288675  0.816497  0.0  0.0  
 0.5  0.288675  0.204124  0.790569  0.0  
 0.5  0.288675  0.204124  0.158114  0.774597
```

```
In [9]: lw*transpose(lw)
```

```
Out[9]: 5x5 Array{Float64,2}:  
 1.0  0.5  0.5  0.5  0.5  
 0.5  1.0  0.5  0.5  0.5  
 0.5  0.5  1.0  0.5  0.5  
 0.5  0.5  0.5  1.0  0.5  
 0.5  0.5  0.5  0.5  1.0
```

```
In [10]: covar
```

```
Out[10]: 5x5 Array{Float64,2}:  
 1.0  0.5  0.5  0.5  0.5  
 0.5  1.0  0.5  0.5  0.5  
 0.5  0.5  1.0  0.5  0.5  
 0.5  0.5  0.5  1.0  0.5  
 0.5  0.5  0.5  0.5  1.0
```

```
In [18]: ?Base.LinAlg.BLAS.trsm
```

```
Base.LinAlg.BLAS.trsm(side, ul, tA, dA, alpha, A, B)
```

Returns the solution to "A*X = alpha*B" or one of the other three variants determined by "side" (A on left or right of "X") and "tA" (transpose A). Only the "ul" triangle of "A" is used. "dA" indicates if "A" is unit-triangular (the diagonal is assumed to be all ones).

```
In [19]: Base.LinAlg.BLAS.trsm('L', 'L', 'N', 'N', 1.0, full(lw), covar)
```

```
Out[19]: 5x5 Array{Float64,2}:  
 1.0  0.5  0.5  0.5  0.5  
 0.0  0.866025  0.288675  0.288675  0.288675  
 0.0 -6.7987e-17  0.816497  0.204124  0.204124  
 0.0 -5.26625e-17  3.51083e-17  0.790569  0.158114  
 0.0  0.0  0.0  0.0  0.774597
```

```
In [17]: \(\full(lw), covar)
```

```
Out[17]: 5x5 Array{Float64,2}:
 1.0  0.5  0.5  0.5  0.5
 0.0  0.866025  0.288675  0.288675  0.288675
 0.0 -6.7987e-17  0.816497  0.204124  0.204124
 0.0 -5.26625e-17  3.51083e-17  0.790569  0.158114
 0.0  0.0  0.0  0.0  0.774597
```

```
In [12]: ?Base.LinAlg.BLAS.trsv
```

```
INFO: Loading help data...
```

```
Base.LinAlg.BLAS.trsv(side, ul, tA, dA, alpha, A, b)
```

Returns the solution to "A*X = alpha*b" or one of the other three variants determined by "side" (A on left or right of "X") and "tA" (transpose A). Only the "ul" triangle of "A" is used. "dA" indicates if "A" is unit-triangular (the diagonal is assumed to be all ones).

```
In [20]: Base.LinAlg.BLAS.trsv('L', 'L', 'N', 'N', 1.0, full(lw), meanVec)
```

```
`trsv` has no method matching trsv(::Char, ::Char, ::Char, ::Char, ::Float64, ::Array{Float64,2}, ::Array{Float64,1})
while loading In[20], in expression starting on line 1
```

```
In [21]: Base.LinAlg.BLAS.trsv('L', 'N', 'N', full(lw), meanVec)
```

```
Out[21]: 5-element Array{Float64,1}:
 0.0973964
 0.820596
 -0.26716
 0.203227
 0.842834
```

```
In [22]: \(\full(lw), meanVec)
```

```
Out[22]: 5-element Array{Float64,1}:
 0.0973964
 0.820596
 -0.26716
 0.203227
 0.842834
```

```
In []:
```