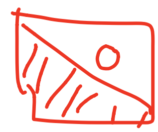
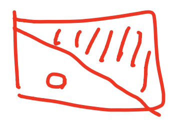
Numerical Linear Algebra

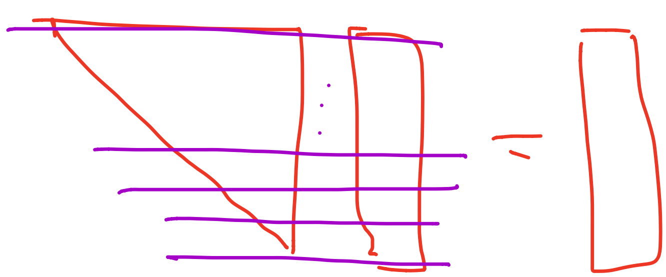
A square matrix A of order n is upper / lower triangular if

$A_{ij} = 0 \quad \forall i < j$ 

$A_{ij} = 0 \quad \forall i > j$ 

We want to compute for a given upper triangular matrix $U \in \mathbb{R}^{n \times n}$ and vector $v \in \mathbb{R}^n$, $U^{-1} v$

Want: $x = U^{-1} v \implies Ux = v$



$U \quad x \quad\quad v$

Well $u_{nn} x_n = v_n \implies x_n = \frac{v_n}{u_{nn}}$.

and the second to last row gives

$$u_{n-1,n-1} x_{n-1} + u_{n-1,n} x_n = v_{n-1}$$

$$\implies x_{n-1} = \frac{1}{u_{n-1,n-1}} \left(v_{n-1} - u_{n-1,n} x_n \right)$$

So in general

$$x_i = \frac{1}{u_{ii}} \left(v_i - \sum_{j=i+1}^{n} u_{ij} x_j \right).$$

This process is called back solving

as we solve $n, n-1, \ldots, 2, 1$

<u>Rmk:</u> Similar algorithm for lower triangular matrix exists called

forward solving

Rmk: If we wanted $U^{-T}v$?

Well $U^T x = v$ and we could do forward solve with the columns dotted with the rows.

Rmk: In R

$x = U^{-1}v$: x <- backsolve(U, v)

$x = U^{-T}v$: x <- backsolve(U, v, trans = TRUE)

math	R
AB	A %*% B
$A^T B$	crossprod(A, B)
$A B^T$	$\boxed{\text{tcrossprod(A, B)}}$

Some Useful Decompositions

(1) (Cholesky)

If A is a square symmetric positive definite matrix then we can find an upper triangular matrix C such that

$$A = C^T \underbrace{C}_{\text{Cholesky factor.}}$$

<u>Ex</u>: How to compute $A^{-1}v = x$

$$A^{-1}v = \underbrace{C^{-1}\underbrace{C^{-T}v}_{\text{forward solve}}}_{\text{backward solve.}}$$

$\boxed{R}$ $C \leftarrow \text{chol}(A)$

$y \leftarrow \text{backsolve}(C, v, \text{trans=TRUE})$

$x \leftarrow \text{backsolve}(C, y)$

E[illegible] Compute the [illegible]

<u>Ex:</u> Compute the log density $N(\vec{\mu}, \Sigma)$ at x.

$$\log f(x) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)$$

<u>Define:</u> $\Sigma = C^T C$ then.

$$\log f(x) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|C^T C| - \frac{1}{2}(x-\mu)^T C^{-1} C^{-T}(x-\mu)$$

$$= -\frac{n}{2}\log 2\pi - \log|C| - \frac{1}{2}\underbrace{\left[C^{-T}(x-\mu)\right]^T \left[\underbrace{C^{-T}(x-\mu)}_{y}\right]}_{y^T y}$$

$$= -\frac{n}{2}\log 2\pi - \sum_{i=1}^{n}\log c_{ii} - \frac{1}{2}y^T y$$

$\boxed{R}$ `C <- Chol(Σ)`

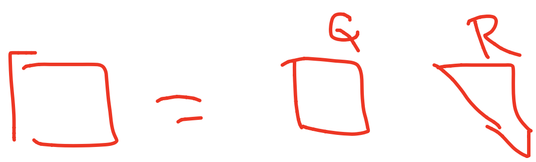
y <- backsolve(C, X-μ, trans=TRUE)

$-\frac{n}{2}$ * log(2*pi) - sum(log(diag(C)))

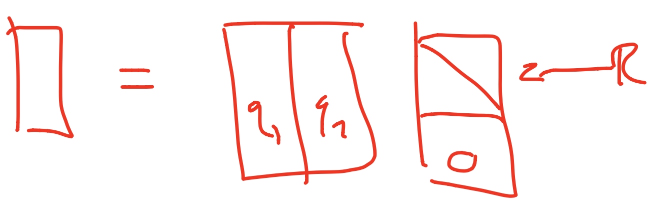
- sum(y^2)

(ii) If X is an $n \times p$ matrix then we can find an $n \times n$ orthogonal matrix Q, $(Q^T Q = Q Q^T = I)$ and $n \times p$ trapezoidal matrix R.

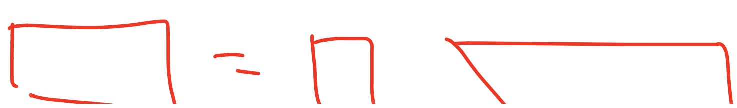
$n = p$



$n > p$



$n < p$



$\underbrace{\qquad}_{} \quad \underbrace{\quad}_{Q} \quad \underbrace{\qquad\qquad}_{R}$

Take $n>p$ and minimize

$$SS(\beta) = (Y - X\beta)^T (Y - X\beta)$$

$$X = QR = [Q_1, Q_2]\begin{bmatrix} R_1 \\ 0 \end{bmatrix} = \underbrace{Q_1 R_1}_{\text{Thin QR}}$$

$$SS(\beta) = \left(\begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} Y - \begin{bmatrix} R_1\beta \\ 0 \end{bmatrix}\right)^T \left(\begin{bmatrix} Q_1^T Y - R_1\beta \\ Q_2^T Y \end{bmatrix}\right)$$

$$= \underbrace{(Q_1^T Y - R_1\beta)^T (Q_1^T Y - R_1\beta)}_{\text{Minimizer}} + Y^T Q_2 Q_2^T Y$$

$$Q_1^T Y - R_1\hat\beta = 0 \quad \text{or}$$

$\hat\beta = R_1^{-1} Q_1^T Y$ is the problem

$$\hat{\beta} = R_1^{-1} Q_1^T y$$

we solve "easy"

So

$$SS(\hat{\beta}) = y^T Q_2 Q_2^T y$$

$$\hat{y} = X\hat{\beta} = \underbrace{Q_1 Q_1^T}_{H} y$$

$$\hat{e} = \underbrace{Q_2 Q_2^T}_{I-H} y$$

- Q partitions the X into the covariate & residual space.