Generating correlated random vectors

- Straightforward to implement KF steps in Octave
- To exercise KF code, however, also must simulate system whose state is being estimated
  - Produces system input/output data as input to the KF
  - Provides "true" state value at every point in time
- But, to simulate true system, must be able to create nonzero mean Gaussian noise with covariance $\Sigma_{\widetilde{Y}}$.
- That is, we want $Y \sim \mathcal{N}(\bar{y}, \Sigma_{\widetilde{Y}})$ but `randn.m` returns $X \sim \mathcal{N}(0, I)$.
  - How to convert X to Y?

A needed transformation

- Suppose that we can find matrix A such that $A^T A = \Sigma_{\widetilde{Y}}$
- Then, we generate samples x of random vector $X \sim \mathcal{N}(0, I)$ and compute samples y of random vector $Y = \bar{y} + A^T X$
  - Since X is zero-mean, $\mathbb{E}[Y] = \mathbb{E}[\bar{y} + A^T X] = \bar{y}$, as desired
  - The covariance of Y is

$$\begin{aligned}\mathbb{E}[(Y-\bar{y})(Y-\bar{y})^T] &= \mathbb{E}[(A^T X)(A^T X)^T] \\ &= \mathbb{E}[(A^T X)(X^T A)] \\ &= A^T \underbrace{\mathbb{E}[XX^T]}_{I} A = \Sigma_{\widetilde{Y}},\end{aligned}$$

  also as desired

Important matrix factorizations

- So, if we can find A such that $A^T A = \Sigma_{\widetilde{Y}}$, we can generate samples x of $X \sim \mathcal{N}(0, I)$ and compute $y = \bar{y} + A^T x$
- Three ways to generate A
  - Cholesky factorization of $\Sigma_{\widetilde{Y}}$ computes A such that $A^T A = \Sigma_{\widetilde{Y}}$ as long as $\Sigma_{\widetilde{Y}}$ is positive definite (all eigenvalues are strictly greater than zero)
  - LDL factorization of $\Sigma_{\widetilde{Y}}$ produces L, D such that $LDL^T = \Sigma_{\widetilde{Y}}$ and requires only that $\Sigma_{\widetilde{Y}}$ be positive semi definite: Can then compute $A = \sqrt{D} L^T$
  - LU factorization of $\Sigma_{\widetilde{Y}}$ produces L, U such that $LU = \Sigma_{\widetilde{Y}}$ and requires only that $\Sigma_{\widetilde{Y}}$ be positive semi definite: Can then compute $A = \text{diag}\left(\sqrt{\text{diag}(U)}\right) L^T$
- Cholesky and LU are built into Octave; all are built into MATLAB

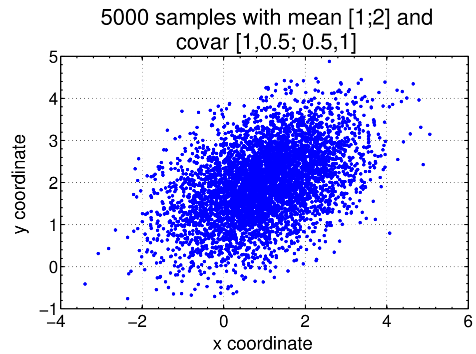
Example Octave code

- Suppose we wish to generate a single sample of y using Cholesky method:

```
ybar = [1; 2];
covar = [1, 0.5; 0.5, 1];
A = chol(covar);
x = randn([2, 1]);
y = ybar + A'*x;
```

- Suppose we wish to generate 5000 samples of y using the LU method:

```
ybar = [1; 2];
covar = [1, 0.5; 0.5, 1];
[L,U] = lu(covar);
A = diag(sqrt(diag(U)))*L';
x = randn([2,5000]);
y = ybar(:,ones([1 5000]))+A'*x;
plot(y(1,:),y(2,:),'.');
```



Summary

- In order to test KF code via simulation, must also simulate system whose state is being estimated
- This requires ability to simulate (possibly) correlated (possibly nonzero mean) noises
- Octave natively produces samples of zero-mean uncorrelated Gaussians only
- But, can transform these to have desired mean and correlation using a matrix determined using either the Cholesky or LU matrix decompositions, both of which are built into Octave