EE 3025 Spring 2011 Discussion 5:

Part 1:

Let $x = [x_1 \ x_2]'$ and $z = [z_1 \ z_2]'$ be 2-vectors (the $'$ sign denotes transpose).

```matlab
x = randn(2, 100000);   
This generates 100000 samples of the 2-vector $x$. Similarly, we have $z$:

```matlab
z = randn(2, 100000);   
```

The mean of each should be close to $[0 \ 0]'$. Here is how we estimate the mean:

```matlab
mean(x')   
mean(z')   
```

The transpose operations are needed because MATLAB likes to work down columns, there is nothing special about it.

We can estimate the covariance $C_x$ of $x$, and $C_z$ of $z$ as follows:

```matlab
cov(x')  
cov(z')  
```

Since $x_1$ and $x_2$ are independent and of variance 1, we expect $C_x$ to be a 2x2 identity matrix. Similarly, $C_z$ will also be a 2x2 identity matrix. The closer the covariance matrix is to a diagonal matrix, the less correlated the entries of the vector are. In this case, the entries of the vectors $x$ and $z$ are independent, which implies that they are uncorrelated, so the off-diagonal entries of the covariance matrices $C_x$ and $C_z$ are expected to be zero.

Part 2:

Define a matrix $A$:

```matlab
A = [1/2 sqrt(3)/2; 3/5 4/5]   
Define a new vector $w$ as follows:

```matlab
w = A*z;   
```

Analytically, we are saying $w = [w_1 \ w_2]' = A*[z_1 \ z_2]'$. So we can compute that $E[w] = [0 \ 0]'$ using the usual rules of the expectation operator, and compute the covariance matrix as follows:

```matlab
C_w = E[ww'] - E[w]E[w]'  
= E[ww']  
= E[A z(A z)']  
= E[Azz' A'] (because $A B' = B' A'$)  
= AA' (because $E[zz']$ is the identity matrix).  
```

We can confirm this by estimating it in MATLAB:

```matlab
cov(w')  
```

comparing this matrix to $A*A'$ and making sure they are about the same.
Note that since \( \mathbf{w} = \mathbf{A} \mathbf{z} \), we created the entries of \( \mathbf{w} \) by "mixing" the entries of \( \mathbf{z} \). This means that the entries of \( \mathbf{w} \) are correlated, unlike the entries of \( \mathbf{z} \). Sure enough, \( \mathbf{C}_w \) has large off-diagonal entries in accordance with this fact.

Part 3:

Now, if we said
\[
\mathbf{y} = \mathbf{x} + \mathbf{w};
\]
what would we get? We would get an estimate of the covariance matrix of \( \mathbf{y} \), which is \( \mathbf{C}_y = \mathbb{E}[\mathbf{yy}'] - \mathbb{E}[\mathbf{y}]\mathbb{E}[\mathbf{y}]' \). Since \( \mathbf{y} = \mathbf{x} + \mathbf{w} \), so that \( \mathbb{E}[\mathbf{y}] = \mathbb{E}[\mathbf{x}] + \mathbb{E}[\mathbf{w}] \), we have that \( \mathbb{E}[\mathbf{y}] = [0 \ 0]' \). Using this, we have
\[
\mathbf{C}_y = \mathbb{E}[\mathbf{yy}']
\]
\[
= \mathbb{E}[(\mathbf{x} + \mathbf{w})(\mathbf{x} + \mathbf{w})'] \quad \text{(we can use the fact that (A + B)' = A' + B')}
\]
\[
= \mathbb{E}[\mathbf{xx}'] + \mathbb{E}[\mathbf{wx}'] + \mathbb{E}[\mathbf{wx}'] + \mathbb{E}[\mathbf{ww}']
\]
\[
= \mathbb{E}[\mathbf{xx}'] + \mathbb{E}[\mathbf{x}][\mathbb{E}[\mathbf{w}'] + \mathbb{E}[\mathbf{w}]\mathbb{E}[\mathbf{x}'] + \mathbb{E}[\mathbf{ww}']
\]

We can split up \( \mathbb{E}[\mathbf{wx}'] \) into \( \mathbb{E}[\mathbf{x}][\mathbb{E}[\mathbf{w}'] \) because \( \mathbf{x} \) and \( \mathbf{w} \) are independent, but we cannot use this to split up \( \mathbb{E}[\mathbf{xx}'] \) into \( \mathbb{E}[\mathbf{x}][\mathbb{E}[\mathbf{x}'] \) because \( \mathbf{x} \) is not independent of itself, nor can we use it to split up something like \( \mathbb{E}[\mathbf{wz}'] \) into \( \mathbb{E}[\mathbf{w}][\mathbb{E}[\mathbf{z}'] \), because \( \mathbf{w} \) is a function of \( \mathbf{z} \) and is not independent of it.

Further, since \( \mathbf{x} \) and \( \mathbf{w} \) have zero mean,
\[
\mathbf{C}_y = \mathbb{E}[\mathbf{xx}'] + \mathbb{E}[\mathbf{ww}']
\]
\[
= \mathbf{I} + \mathbf{AA}'
\]
where \( \mathbf{I} \) is the 2x2 identity matrix. We can confirm that this is indeed the covariance estimate in MATLAB by evaluating:
\[
[1 \ 0; \ 0 \ 1] + \mathbf{A}^\top \mathbf{A}
\]
and checking that it is the nearly same as \( \mathbb{cov}(\mathbf{y}') \).

Part 4:

Now, consider only the vector \( \mathbf{w} \), which is a function of the vector \( \mathbf{z} \). We will plot 100 samples of the first component the vector \( \mathbf{w} \), and color it blue:
\[
\text{plot}(\mathbf{w}(1,1:100), 'b')
\]
Next, we will plot 100 samples of the second component on the same graph, and color it red.
\[
\text{hold on;}
\]
\[
\text{plot}(\mathbf{w}(2,1:100), 'r')
\]
You will see that they are very similar. This is because \( \mathbf{w}_1 \) and \( \mathbf{w}_2 \) are highly correlated, and the covariance matrix \( \mathbf{C}_w \) has large off-diagonal entries.

Now, we will plot 100 samples of the first component the vector \( \mathbf{z} \) in a new graph, and color it blue:
\[
\text{figure;}
\]
\[
\text{plot}(\mathbf{z}(1,1:100), 'b')
\]
Next, we will plot 100 samples of the second component on the same graph, and color it red.
\[
\text{hold on;}
\]
\[
\text{plot}(\mathbf{z}(2,1:100), 'r')
\]
Notice that the components z1 and z2 are not very similar at all. This is because they vary independently, and the covariance matrix \( C_z \) has zero off-diagonal entries.

**Part 5:**

Finally, we would like to "decorrelate" w1 and w2, so that they look like z1 and z2. Specifically, we would like to figure out a 2x2 matrix \( B \), such that \( u = B w \), and \( u \) has independent components (and the covariance matrix \( C_u \) will be identity). The matrix \( B \) is also called a "whitening filter", and this operation is sometimes called "whitening" or conversion to a standard Gaussian.

How do we find this \( B \)? Since \( w = A z \), if we knew \( A \), we could easily say \( B = A^{-1} \). Clearly, then \( u = B w = A^{-1} A z = z \), and since z1 and z2 are independent, u1 and u2 will also be independent. Sometimes, we might not know the matrix \( A \), so here are 2 ways to find a matrix \( B \) without inverting \( A \).

Any matrix can be written as follows:
\[
A = U S V' \]
where \( U \), \( S \), and \( V \) are matrices, with \( S \) being a diagonal matrix. This is called the singular value decomposition or SVD, and it has many interesting applications and properties. The \( U \) and \( V \) decompose any matrix \( A \) into component vectors, and the \( S \) matrix tells you "how much" of each component is needed to construct the matrix \( A \). In MATLAB, we can do this as follows:
\[
[U S V] = svd(A) \]
Then, we can compute \( B \) as follows:
\[
B = S^{(-1)} * U' \]

Another way to compute \( B \) is using the eigen decomposition (ED) of \( A \). Recall in Part 2 we were able to estimate the covariance matrix of \( w \), given just samples of \( w \). Since \( C_w = AA' \), we can estimate \( AA' \) even if we don't know \( A \). The eigen decomposition says that we can decompose \( AA' \) into:
\[
AA' = V D V' \]
where \( D \) is a diagonal matrix containing the eigenvalues, and \( V \) is a collection of vectors called eigenvectors. In MATLAB, we can do this as follows:
\[
[V D] = eig(A*A') \]
and compute \( B \) as follows:
\[
B = D^{(-1/2)} * V' \]

Whichever method you used to compute \( B \), you can now say
\[
u = B*w \]
\[
cov(u') \]
which should give you a covariance matrix close to identity. In other words, we have successfully decorrelated the components of \( w \). Remember that w1 and w2 were formed by "mixing" z1 and z2. Now we have constructed a u1 and u2 by "unmixing" w1 and w2, so that u1 and u2 are independent.

Note that u1 and u2 are NOT exactly equal to z1 and z2 for the SVD or ED construction of \( B \) (unless you used \( B = A^{-1} \)), but \([u1 u2]' \) has the same *joint distribution* as \([z1 z2]' \). Random variables having the same distribution is not the same as them being exactly the same, but they will have the same means, covariance matrix, etc.