Principal Component Analysis

Justin Kim

December 14, 2016

1 Introduction

Many times, a large number of measurements or observations can be made for each sample, thereby leading to a large number of dimensions, sometimes more than the number of samples. This is especially applicable in the life sciences as thousands of mRNA and protein level measurements can be made for every sample. Visualizing such high-dimensional data can be difficult, so a way to reduce the number of dimensions without losing much information would be useful.

1.1 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical method that does just that by identifying specific linear combinations of variables that retain as much variation in the data as possible. These linear combinations, or principal components (PCs), are the new dimensions for the data. It is important to note that, as a result, they are, by definition, orthogonal to one another. It should be noted this method relies on the assumption that some dimensions account for more of the data's variation than others do and that the data actually can be characterized by linear combinations.

The motivation behind maximizing the variation in the data can be demonstrated by measuring the signal-to-noise ratio (SNR), which is the ratio of the variance in signal to the variance in noise. The higher the SNR, the more precise the data, and the lower the SNR, the more contaminated the data. Typically, only the leading principal components are used for analysis since they account for a disproportionate amount of variance. It is assumed that most of the variance is then from signal rather than noise because noise is assumed to be uniformly, randomly distributed over all the principal components. Thus, by selecting a few leading principal components, most of the variance from signal would be retained while the variance from noise would be reduced, thereby keeping the SNR high.

PCA is essentially a change of bases. The original set of bases were the dimensions, and the new set are the principal components. As Singular Value Decomposition (SVD) also represents a change of bases, it is no coincidence the two are closely related. In fact, the two terms are sometimes even used interchangeably. Thus, SVD will be used in both the derivation and implementation of PCA here.

1.2 Intuitive Example

Imagine you were to measure the movement of a ball's bouncing straight down a hallway with one measuring device. The simplest way to do this would be to have the measuring device face straight down the hallway, towards the ball, so it could measure the ball's vertical movement independently from the ball's horizontal movement. However, it is not always clear where to place measuring devices in experiments, and if the measuring device were placed at an angle, the data would be much dirtier. The

resulting data would be a combination of the ball's horizontal and vertical movements. PCA would allow the axes of the data to be realigned so that variance is maximized on orthogonal bases. These bases would presumably be in the ideal orientation described above.

2 The derivation of PCA

First, the equation behind SVD must be derived since it is so similar to that of PCA.
Let matrix X be defined as an mxn matrix in which each row is a vector of measurements of a particular type with zero mean. Thus, each column of X represents a set of measurements from one particular trial. The covariance matrix S_x can be defined as:

$$S_x = \frac{1}{n-1} XX^T$$

A couple interesting properties are that S_x is a symmetric mxm matrix (the product of any matrix and its transpose is a square, symmetric matrix), the diagonal terms of S_x are the variances of each measurement type, and the off-diagonal terms are the covariances between measurement types. It should also be noted that an unbiased estimator for the sample variances is calculated as all terms are divided by (n - 1) rather than n. In order to reduce redundancies as much as possible, it would be useful to be able to transform matrix X into a matrix with zero covariances.

2.1 Describing the Principal Component Matrix More precisely, the goal is to have an orthonormal matrix P such that $Y = PX$, where $S_y = \frac{1}{n-1} YY^T$ is diagonalized. The rows of P will be the principal components of X.
This definition of S_y can be rewritten as

$$S_y = \frac{1}{n-1} YY^T = \frac{1}{n-1} (PX)(PX)^T = \frac{1}{n-1} PXX^T P^T$$

Since XX^T is a symmetric matrix, it can be rewritten as $XX^T = EDE^T$, where E is a matrix of XX^T's eigenvectors as columns, and D is a diagonal matrix. Let matrix P be defined such that each row p_i is an eigenvector of XX^T, in which case, $P = E^T$, making $XX^T = P^T DP$. This can be substituted into the above equation to be

$$S_y = \frac{1}{n-1} PXX^T P^T = \frac{1}{n-1} PP^T DPP^T = \frac{1}{n-1} PP^{-1} DPP^{-1} = \frac{1}{n-1} D$$

since the transpose of an orthogonal matrix is equal to its inverse (meaning $P^T = P^{-1}$). This definition of P diagonalizes S_y, so it satisfies our needs. This value of P also means that the principal components of X are the rows of P or the eigenvectors of XX^T.

2.2 Using SVD As mentioned earlier, SVD will be used. Let $X^T X$ be a symmetric mxm matrix with rank r. Let $\{v_1, v_2, ..., v_r\}$ be the set of orthonormal $nx1$ eigenvectors

with associated eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_r\}$. Thus, $S_x v_i = \lambda_i v_i$ is true. Let σ_i be positive and real and be the singular values such that $\sigma_i = \sqrt{\lambda_i}$. Finally, let $\{u_1, u_2, ..., u_r\}$ be the set of orthonormal $mx1$ vectors defined by $u_i = \frac{1}{\sigma_i} X v_i$. This can be reorganized as $u_i \sigma_i = X v_i$ to result in the value version of SVD.

At this point, a new diagonal matrix Σ can be constructed, in which $\Sigma_{ii} = \sigma_i$, where the elements of σ have been rank-ordered such that $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r$. Likewise, orthogonal matrices V and U can be constructed as $V = \begin{bmatrix} v_1 & v_2 & ... & v_n \end{bmatrix}$ and $U = \begin{bmatrix} u_1 & u_2 & ... & u_m \end{bmatrix}$, where (n - r) and (m - r) orthonormal vectors were appended to V and U respectively. It should be noted that these additional vectors should have no effect on the final solution since variations associated with them should be zero.

Thus, $u_i \sigma_i = X v_i$ can be rewritten as $XV = U\Sigma$ when considering all dimensions at once, and since V is orthogonal, $V^{-1} = V^T$, this can be rewritten to be $X = U\Sigma V^T$. This means that any matrix can be expressed as the product of an orthogonal matrix, a diagonal matrix, and another orthogonal matrix. When calculating the SVD of a matrix, these are the three outputted matrices. One interesting observation to note is that U and V span all possible inputs and outputs.

So, let us take the mxn matrix X and a new nxm matrix Y, such that $Y = \frac{1}{\sqrt{n-1}} X^T$, where each column of Y has a mean of zero. When considering $Y^T Y$,

$$Y^T Y = (\frac{1}{\sqrt{n-1}} X^T)^T (\frac{1}{\sqrt{n-1}} X^T) = \frac{1}{n-1} X^{TT} X^T = \frac{1}{n-1} X X^T = S_x$$

It should be noted that the columns of matrix V from the SVD of Y are the eigenvectors of S_x, and as shown previously, the principal components of X are the eigenvectors of S_x. Thus, the principal components of X are the columns of V.

3 Implementation

```
function [signals, pc, vars] = pca(data)
    % get dimensions of dataset
    [m, n] = size(data);
    % get mean of each row of dataset
    rowMeans = mean(data, 2);
    % subtract mean from each element of dataset to normalize it
    normData = data - repmat(rowMeans, 1, n);

    % construct new matrix of which svd will be calculated
    newMatrix = normData' / sqrt(n - 1);

    % calculate svd of new matrix
    [u, s, pc] = svd(newMatrix);

    % calculate the variances
    s = diag(s);
    vars = s .* s;
```

```
    % project the original data
    signals = pc' * normData;
end
```

3.1 Complexity Analysis Matlab implements SVD with lapack's SGESVD function. More specifically, it is implemented by turning the mxn input matrix into a bidiagonal matrix. This is done through QR decomposition, which has a time complexity of $O(n^3)$ as shown in class, and then Householder reflections are performed to get the bidiagonal matrix. The sum of the time complexities of these two steps is equal to about $O(mn^2) + O(n^3)$. Then, the eigenvalues are calculated by the QR algorithm, which also has a time complexity of $O(n^3)$. Thus, SVD has a total time complexity of about $O(mn^2) + O(n^3)$.

4 Toy Example

Let the following dataset be used: X =

$$\begin{bmatrix} 2.4 & 0.7 & 2.9 & 2.2 & 3.0 & 2.7 & 1.6 & 1.1 & 1.6 & 0.9 \\ 2.5 & 0.5 & 2.2 & 1.9 & 3.1 & 2.3 & 2 & 1 & 1.5 & 1.1 \end{bmatrix}$$

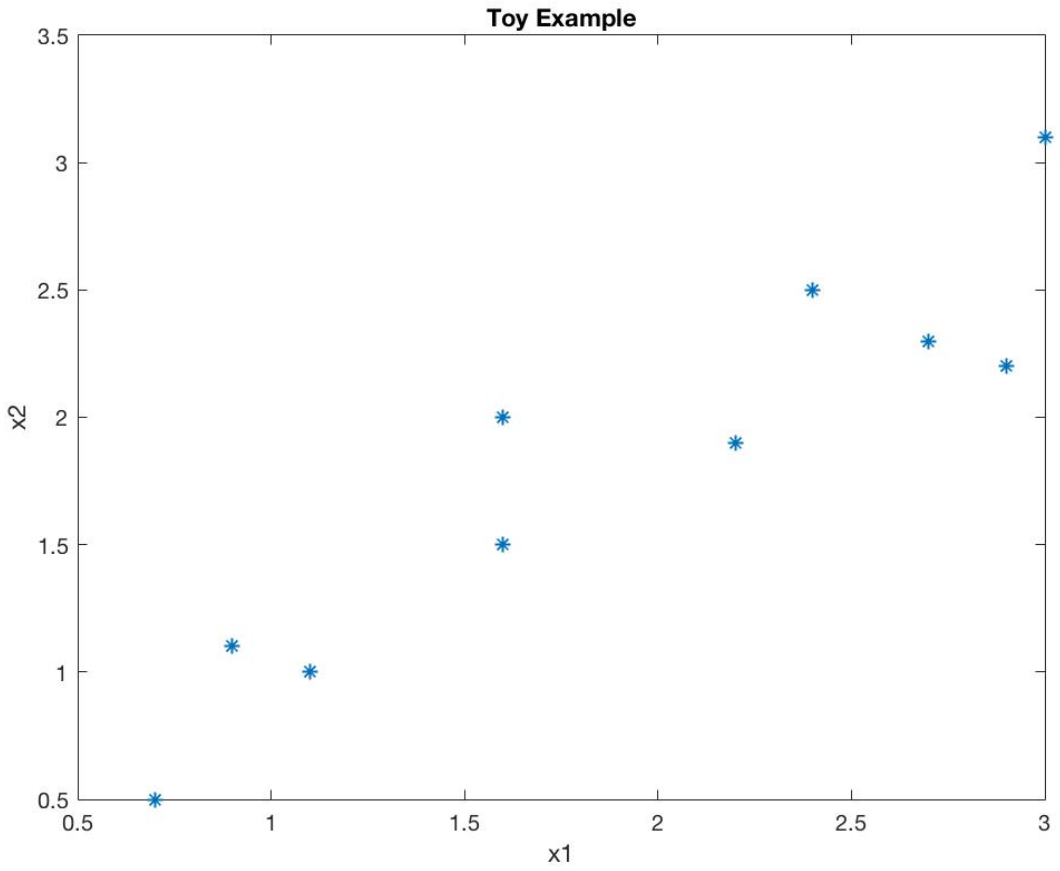

Figure 1: Toy Example Original Data

Then, its covariance matrix S_x will be defined as $\frac{XX^T}{n-1}$, which is equal to

$$\begin{bmatrix} 0.7166 & 0.6154 \\ 0.6154 & 0.6166 \end{bmatrix}$$

The output from the implementation of PCA above results in:

Signal =

$$\begin{bmatrix} 0.828 & 1.7776 & 0.9922 & 0.2742 & 1.6758 & 0.9129 & 0.0991 & 1.1446 & 0.4380 & 1.2238 \\ 0.1751 & 0.1429 & 0.3844 & 0.1304 & 0.2095 & 0.1753 & 0.3498 & 0.0464 & 0.0178 & 0.1627 \end{bmatrix}$$

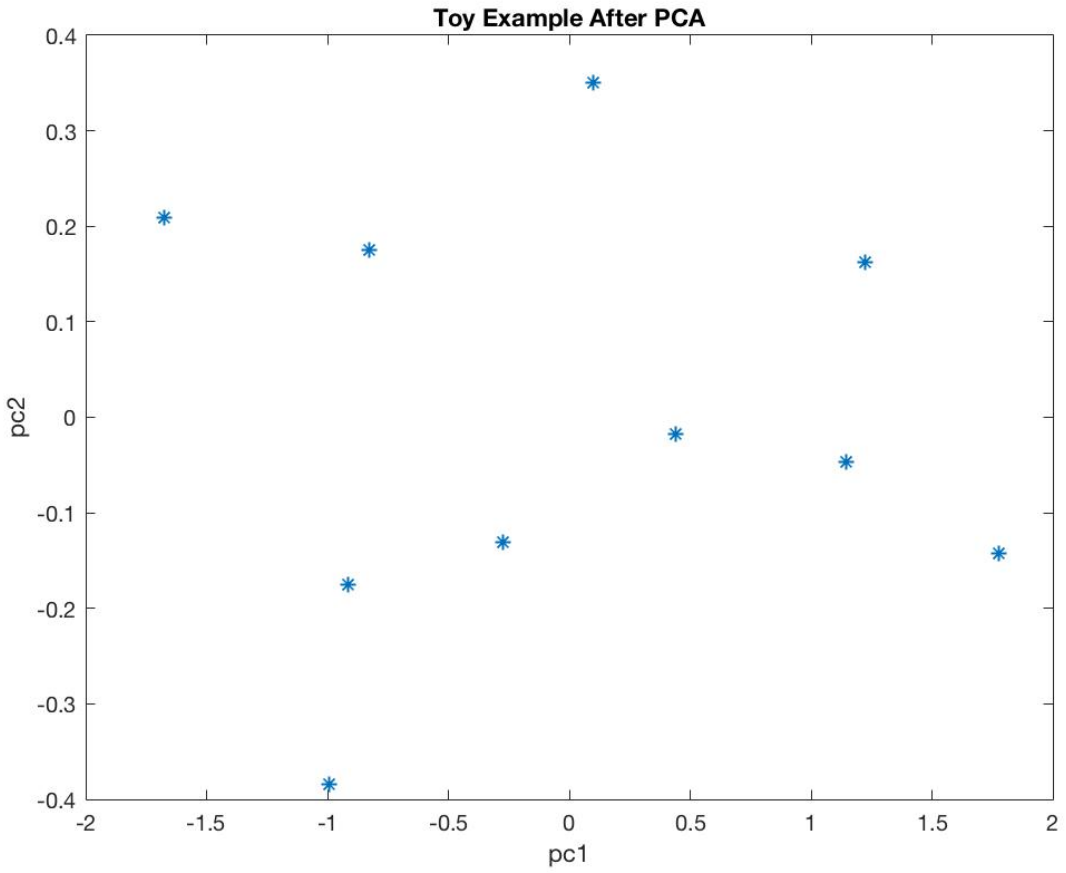

Figure 2: Toy Example After PCA

Note that the data points seem to show the two PC's are independent of each other since they seem to be randomly distributed.

PC =

$$\begin{bmatrix} -0.7352 & -0.6779 \\ -0.6779 & 0.7352 \end{bmatrix}$$

Thus, the first PC is $\begin{bmatrix} -0.7352 \\ -0.6779 \end{bmatrix}$, and the second PC is $\begin{bmatrix} -0.6779 \\ -0.7352 \end{bmatrix}$.

Vars =

$$\begin{bmatrix} 1.2840 \\ 0.0491 \end{bmatrix}$$

This means that the first PC accounts for about 96.3% of the variation, while the second PC accounts for about 3.7% of the variation.

5 Comparison to Least-Squares Regression

The main difference between PCA and Ordinary Least Squares (OLS) regression is that PCA essentially minimizes the orthogonal distance between the data point and the principal component while OLS minimizes the vertical distance. However, it should be noted that these two are different in purpose. The principal component is not necessarily meant to be a model for the data. It is mainly meant to transform the bases of the data to a new one that retains the variation in the data as much as possible while reducing the data's dimensionality. In fact, OLS can be run on the principal component scores once PCA is performed. This process is called Principal Component Regression. This has the advantage of mitigating multicollinearity among variables by excluding principal components with low variances.

It turns out that maximizing the variance of a PC is equivalent to minimizing the distance between the PC and the data points. Let S_x be the covariance matrix, $S_x = \frac{XX^T}{n-1}$, meaning $Var(Xw) = \frac{Xw(Xw)^T}{n-1} = \frac{Xww^TX^T}{n-1}$, where w is the unit vector specifying the new axis in the variable space. Next, let us find the distance between the original points X and the reconstructed points Xww^T with the Frobenius norm, resulting in $||X - Xww^T|| = \sqrt{tr((X - Xww^T)(X - Xww^T)^T)} = \sqrt{tr((X - Xww^T)(X^T - ww^TX^T))}$
$= \sqrt{tr(XX^T) - 2tr(Xww^TX^T) + tr(Xww^Tww^TX^T)}$. Since the trace of a matrix is just a constant, the sum of the trace of the first and last terms will be c, making the previous expression equivalent to $\sqrt{c - tr(Xww^TX^T)}$. Since $Var(Xw) = \frac{Xww^TX^T}{n-1}$, that is equal to $\sqrt{c - \frac{tr(Var(Xw))}{n-1}}$. Since the diagonal elements of $Var(Xw)$ are the variances of the principal components, maximizing them would also maximize the trace of $Var(Xw)$. This in turn would minimize the distance between the PC and the data points since the distance is directly related to the difference between a constant and the trace of $Var(Xw)$.

Ordinary least squares regression was performed on the data via $fitlm(x(1,:), x(2,:))$, and it resulted in the linear model: $y = 0.16951 + 0.85889 * x$ with an R^2 value of 0.857. The following plot resulted:

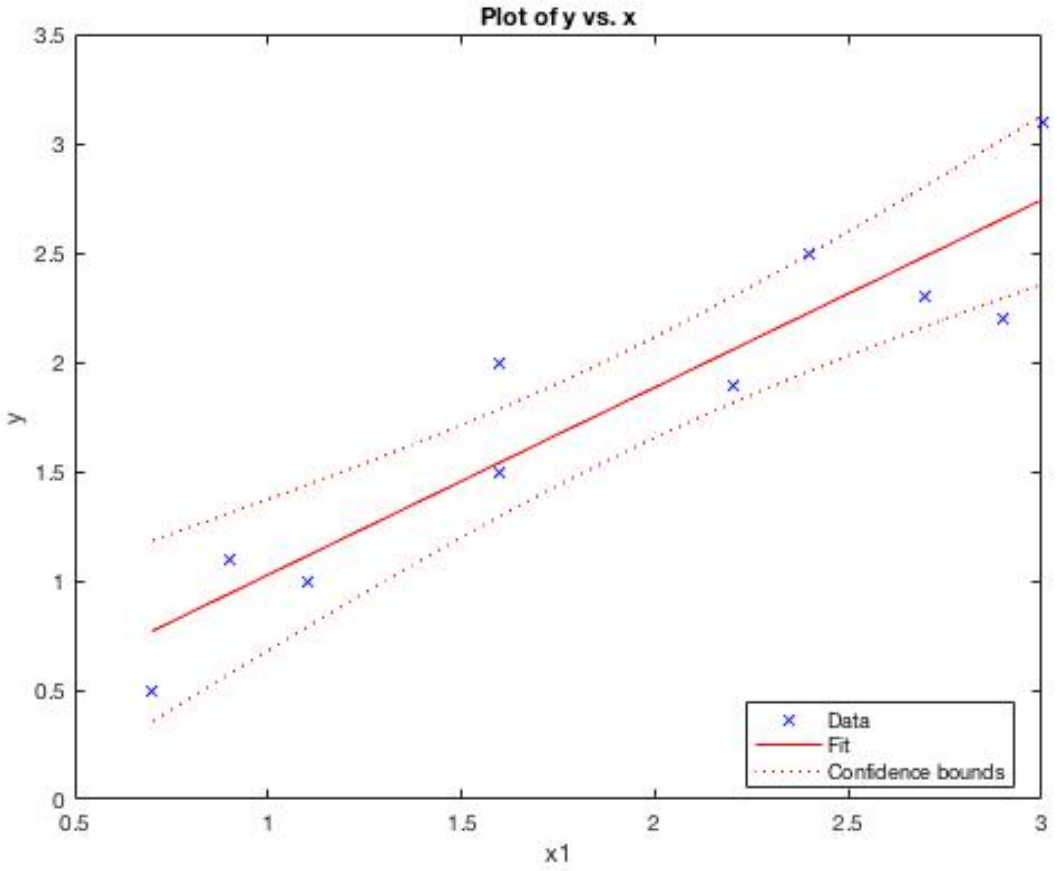

Figure 3: OLS of toy example

Principal Component Regression was also performed on the data, as shown below:

```
x = [2.4 0.7 2.9 2.2 3.0 2.7 1.6 1.1 1.6 0.9;
     2.5 0.5 2.2 1.9 3.1 2.3 2 1 1.5 1.1];
% Name variables
y = x(2, :)';
[m, n] = size(x);
% perform PCA
[toySignals, toyPc, toyVars] = pca(x);

% calculate coefficients for multilinear regression with scores from the
% first principal component
betaPCR = regress((y - mean(y)), toySignals(1,:)');
% center coefficients on original data
betaPCR = toyPc(:,1) * betaPCR;
betaPCR = [mean(y) - mean(x') * betaPCR; betaPCR];
% Take the sum of the intercept (ones) and the predictors (x) for each
% point
yfitPCR = [ones(n, 1) x'] * betaPCR;
```

```
pcrLM = fitlm(yfitPCR, y, 'linear');
plot(pcrLM)
title('Principal Component Regression')
xlabel('PCR fitted')
ylabel('y')
```

It resulted in the linear model: $y = -1.8801e^{-16} + x$ with an R^2 value of 0.957. This means that the linear model generated by principal component regression explains more of the variance of the data.

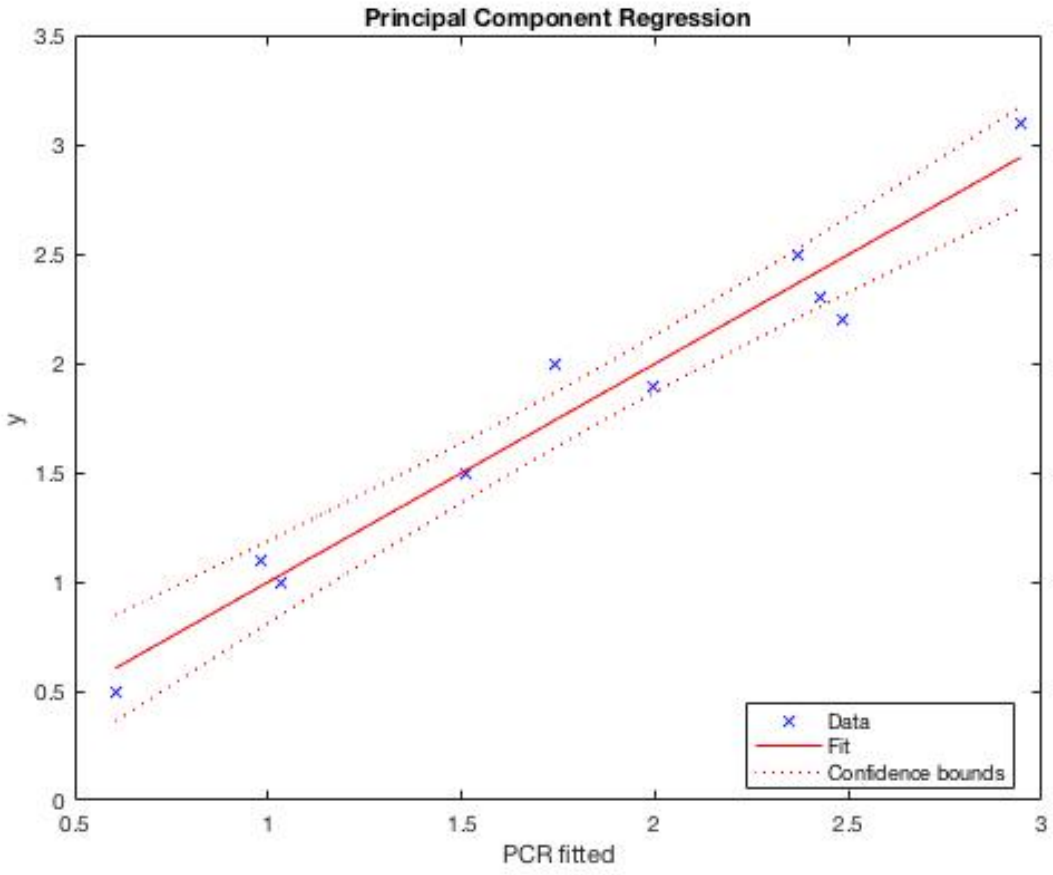

Figure 4: Principal Component Regression

6 Analysis of Dataset 1

This dataset can be found here[1].

There are 27,648 dimensions with 105 samples. More specifically, expression levels of 27,648

genes are measured in 105 breast tumor samples. A paper describing PCA uses this dataset as an example [2].

There were a lot of NaN values throughout the dataset, and since SVD does not work with those invalid values, the data had to be cleaned. While these NaN values would have ideally been evaluated on a row by row basis (i.e. for each gene), all values of NaN just set to 0. One area for improvement for this analysis would be to consider other default values for each gene.

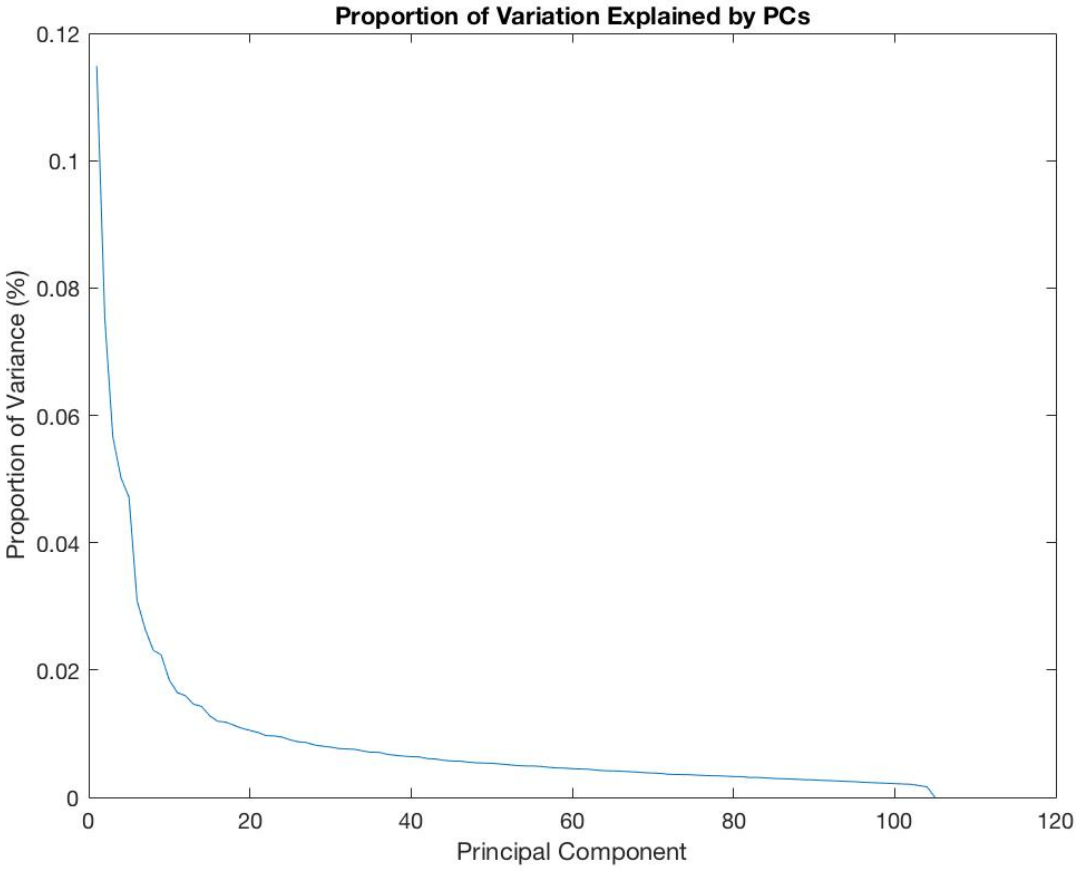

Figure 5: Proportion of Variation Explained by Principal Components of Cancer Data

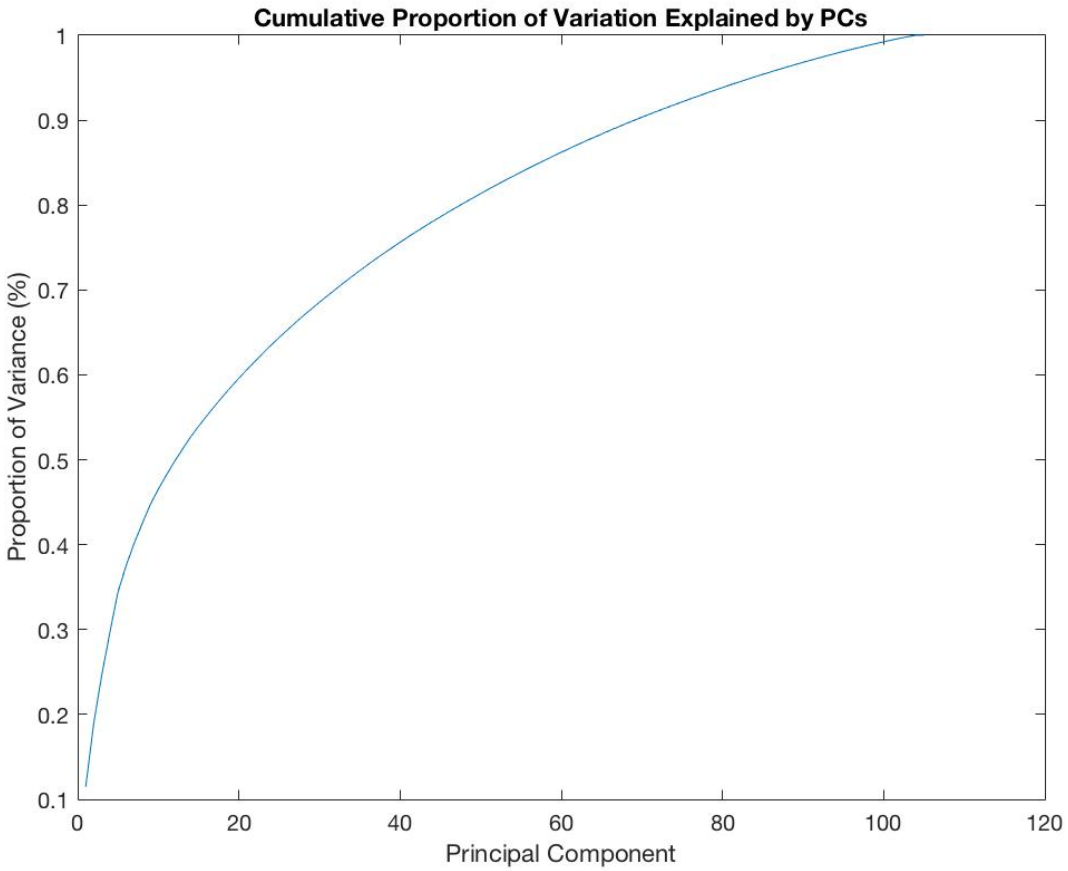

Figure 6: Cumulative Proportion of Variation Explained by Principal Components of Cancer Data

This distribution of the proportion of variation explained by each principal component seems reasonable since the first few principal components typically disproportionately represent the most variation. This is motivated by the idea each subsequent principal component has the extra constraint of being orthogonal to every preceding principal component, so it would not be able to capture as much variation. It should be noted that this distribution is similar to that shown in the paper. A graph of the second PC vs the first PC is shown below.

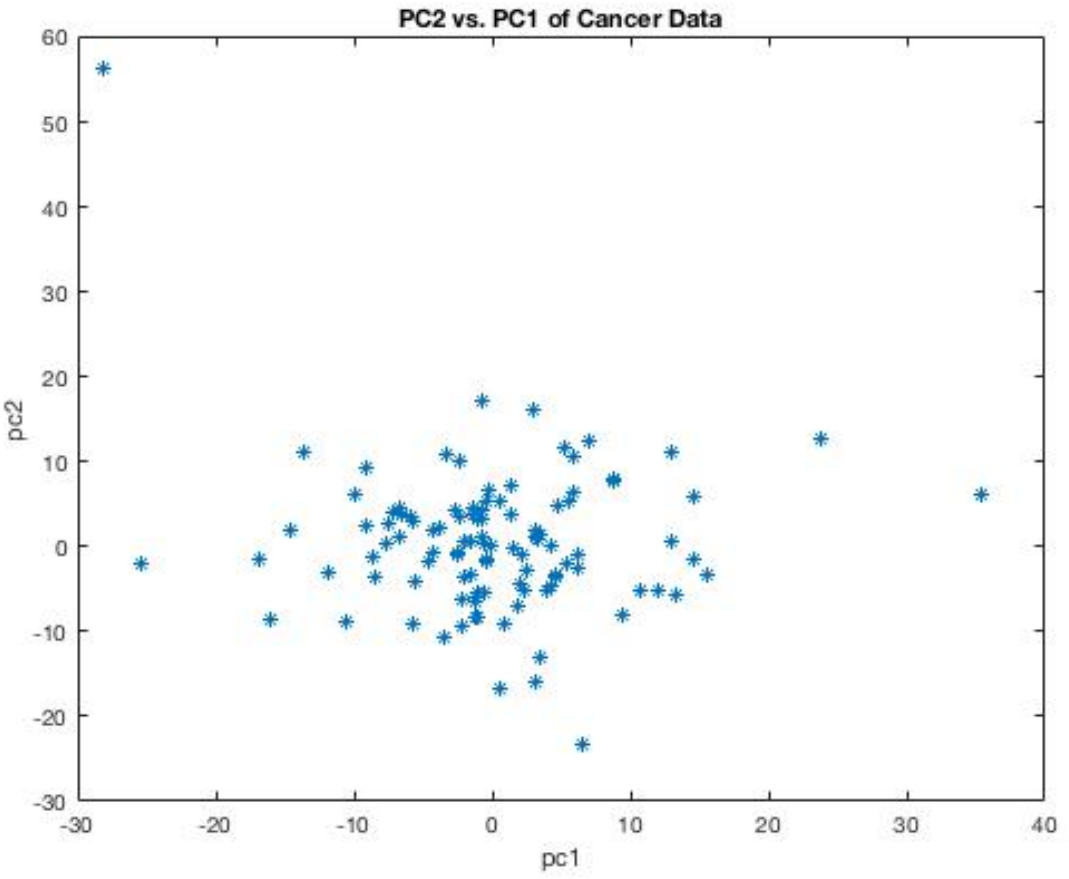

Figure 7: PC2 vs PC1 of Cancer Data

This shows that the two principal components are independent of each other. Furthermore, the biplot of the two PCs is shown below.

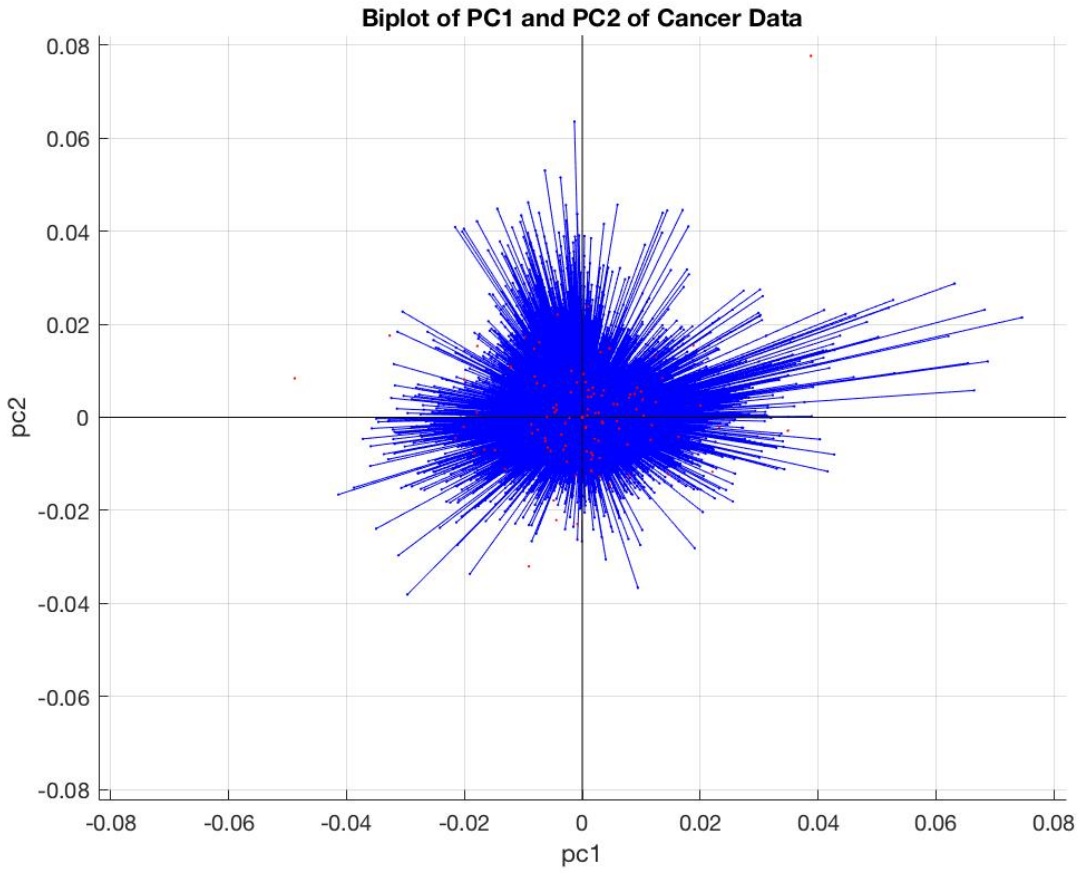

Figure 8: Biplot of PC1 and PC2 of Cancer Data

This helps show some trends in each gene between the two PCs. For example, in terms of the first principal component, the difference between the lines pointing to the right and the lines pointing to the left accounts for the most variance in the data. Each of these lines represents a measurement (in this case, the expression level of a specific gene), so this biplot helps show relationships among the various genes, which can be interesting.

```
cancerMatrix = geoseriesread('GSE5325_series_matrix.txt');
cancerData = double(cancerMatrix.Data);
% clean data
cancerData(isnan(cancerData)) = 0;
[cancerSignals, cancerPc, cancerVars] = pca(cancerData);

% calculate total vars
totalVar = sum(cancerVars);
[vm, vn] = size(cancerVars);
varProportions = arrayfun(@(x) x / totalVar, cancerVars);
plot(1:vm, varProportions);
title('Proportion of Variation Explained by PCs');
```

```
xlabel('Principal Component');
ylabel('Proportion of Variance (%)');

figure
plot(1:vm, cumsum(varProportions));
title('Cumulative Proportion of Variation Explained by PCs');
xlabel('Principal Component');
ylabel('Proportion of Variance (%)');

figure
plot(cancerSignals(:,1), cancerSignals(:,2), '*')
title('PC2 vs. PC1 of Cancer Data')
xlabel('pc1')
ylabel('pc2')

figure
biplot(cancerPc(:, 1:2), 'scores', cancerSignals(:, 1:2))
title('Biplot of PC1 and PC2 of Cancer Data')
xlabel('pc1')
ylabel('pc2')
```

7 Analysis of Dataset 2

This dataset can be found here[3].

There are 54,675 dimensions with 54 samples. More specifically, gene expression levels were measured in blood samples of male subjects who ingested alcohol. These blood samples were taken at five different time points, when their blood alcohol content was 0%, 0.04%, 0.08%, 0.04%, and 0.02%. Because the dataset was too large, 20,000 dimensions were randomly selected for analysis.

The analysis of this dataset is very similar to the analysis of the previous one.

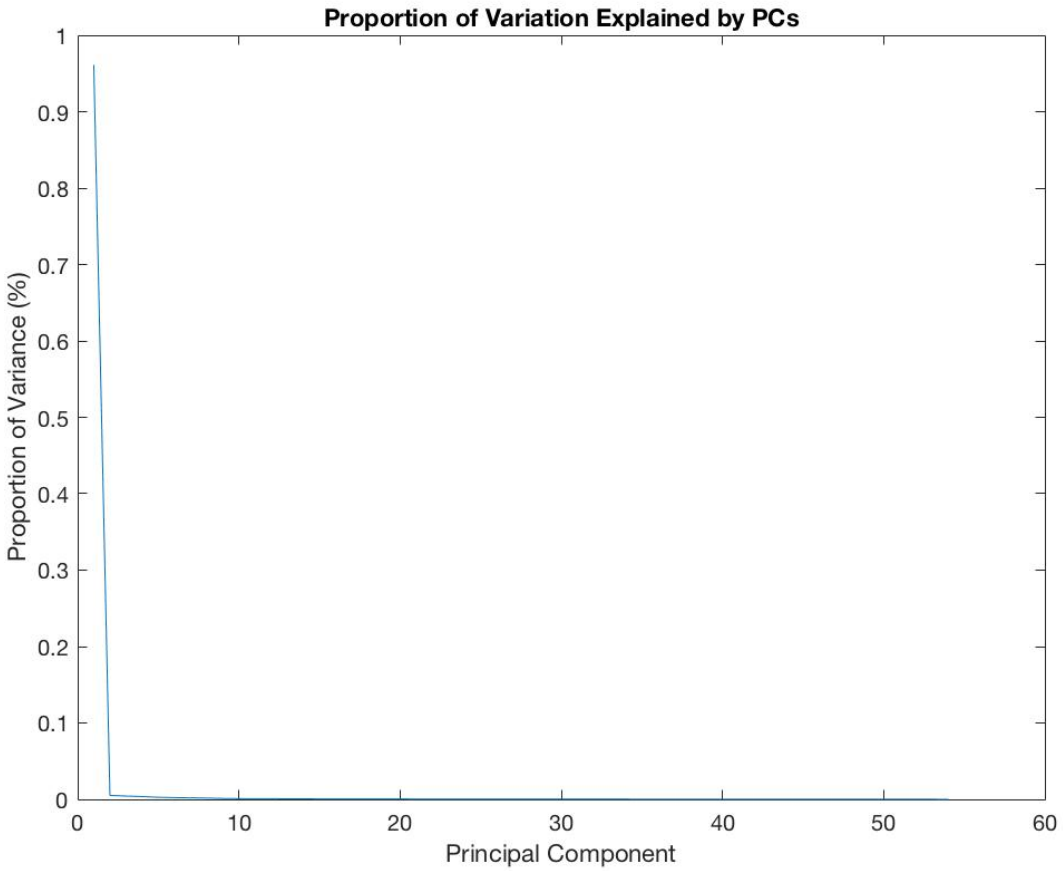

Figure 9: Proportion of Variation Explained by Principal Components of etoh Data

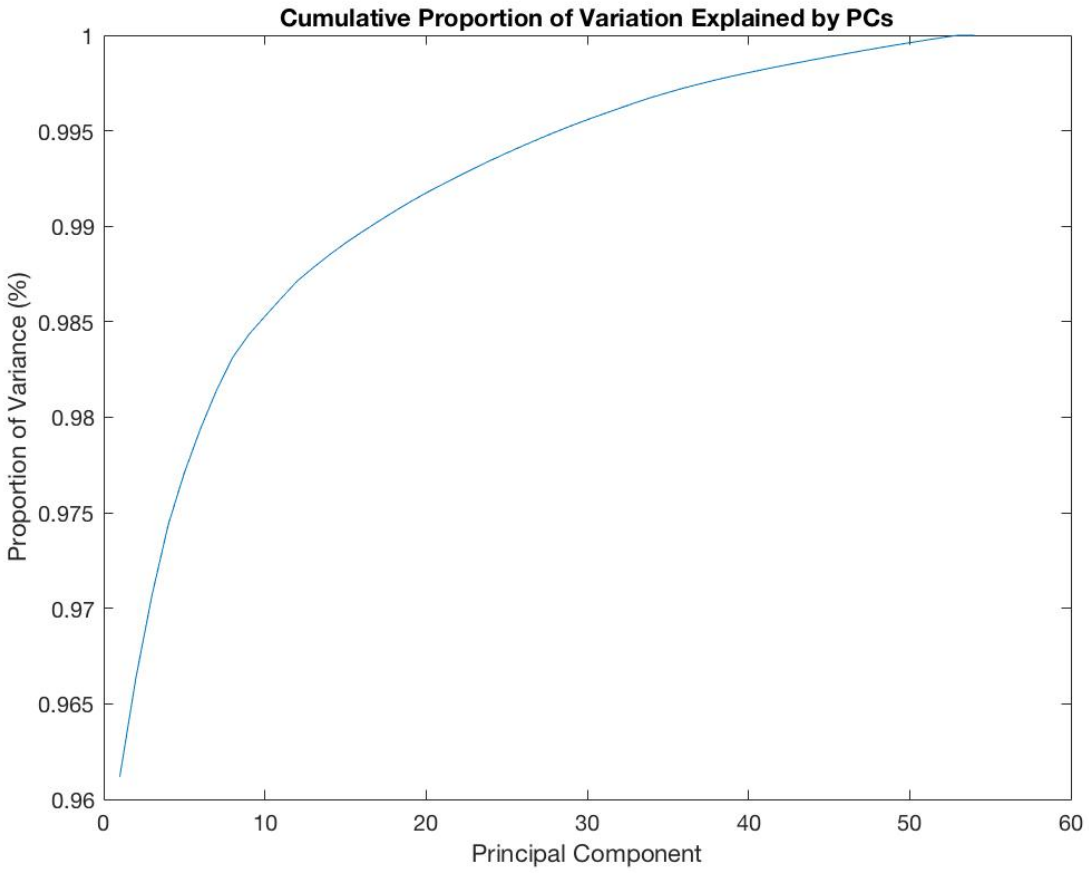

Figure 10: Cumulative Proportion of Variation Explained by Principal Components of etoh Data

Similar to the previous distributions, this distribution of the proportion of variation explained by each principal component seems reasonable since the first few principal components typically disproportionately represent the most variation. However, it is interesting to note that the first few principal components really account for the vast majority of the variance. In fact, the first PC, accounts for about 96.12% of the variance of the data. A graph of the second PC vs the first PC is shown below.

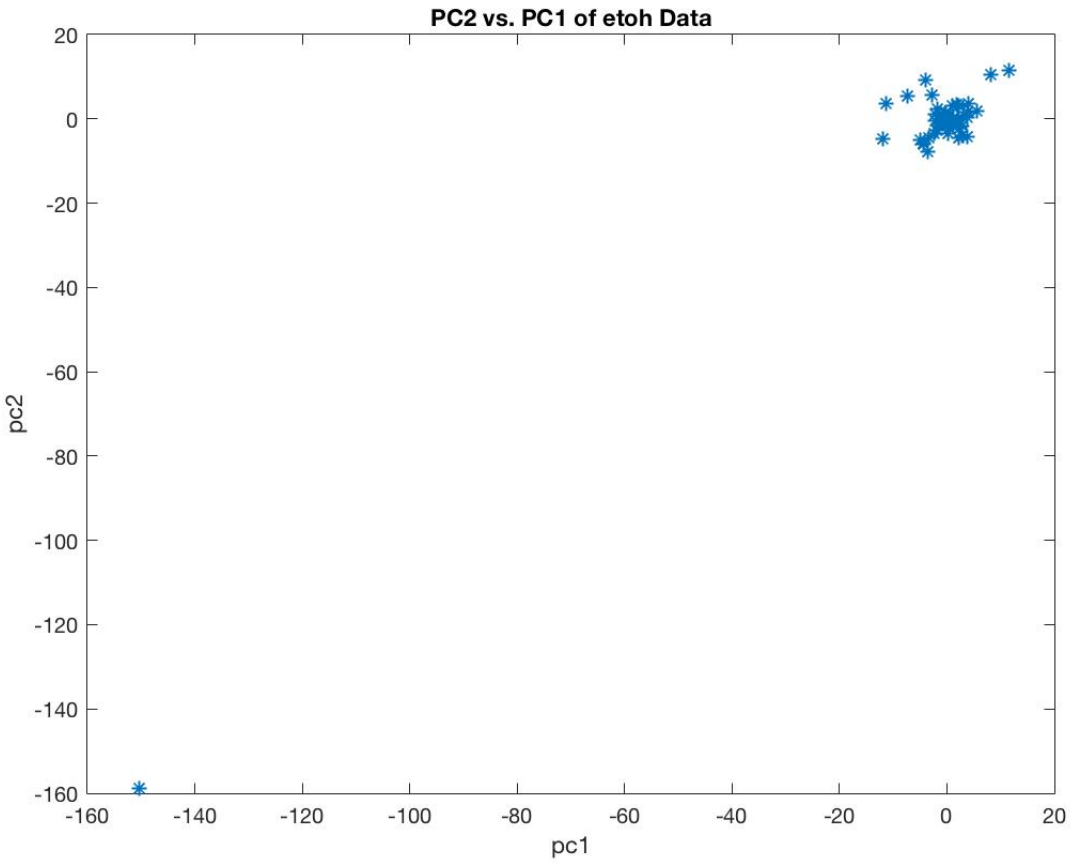

Figure 11: PC2 vs PC1 of etoh Data

This shows that the two principal components are independent of each other. Furthermore, the biplot of the two PCs is shown below.

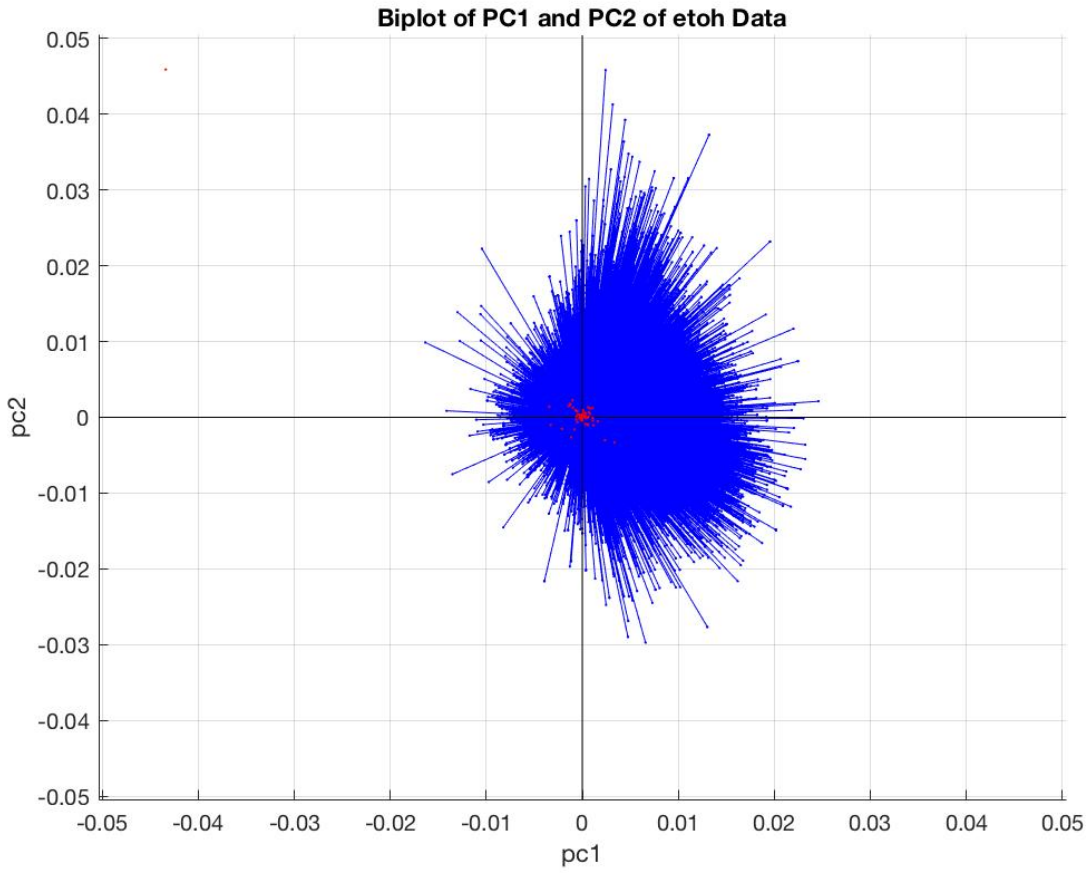

Figure 12: Biplot of PC1 and PC2 of etoh Data

Similar to the previous dataset, this helps show some trends in each gene between the two PCs. For example, in terms of the first principal component, the difference between the lines pointing to the right and the lines pointing to the left accounts for the most variance in the data. Each of these lines represents a measurement (in this case, also the expression level of a specific gene), so this biplot helps show relationships among the various genes, which can be interesting. This biplot has far more lines to the right, so it could mean that the expression levels of most of the measured genes are additive.

```
etohMatrix = geoseriesread('GSE20489_series_matrix.txt');
etohData = double(etohMatrix.Data);
n = 20000; % arbitrarily chosen
sampleEtohData = datasample(etohData, n);
% clean data
[etohSignals, etohPC, etohVars] = pca(sampleEtohData);

% calculate total vars
totalVar = sum(etohVars);
[vm, vn] = size(etohVars);
```

```
varProportions = arrayfun(@(x) x / totalVar, etohVars);
plot(1:vm, varProportions);
title('Proportion of Variation Explained by PCs');
xlabel('Principal Component');
ylabel('Proportion of Variance (%)');

figure
plot(1:vm, cumsum(varProportions));
title('Cumulative Proportion of Variation Explained by PCs');
xlabel('Principal Component');
ylabel('Proportion of Variance (%)');

figure
plot(etohSignals(:,1), etohSignals(:,2), '*')
title('PC2 vs. PC1 of etoh Data')
xlabel('pc1')
ylabel('pc2')

figure
biplot(etohPC(:, 1:2), 'scores', etohSignals(:, 1:2))
title('Biplot of PC1 and PC2 of etoh Data')
xlabel('pc1')
ylabel('pc2')
```

8 Conclusion

It is clear that PCA is important, and understanding it would certainly be useful when analyzing and visualizing high-dimensional data. However, it has some limitations, which have been addressed in various modified forms. For example, it is assumed that the data can be represented with linear models, which may not always be the case. Thus, when non-linear transformations are performed prior to PCA, the process is often called kernel PCA. Furthermore, PCA assumes that the data can be represented with normal distributions, but this may also not be the case, so by removing this assumption, nonlinear optimizations can be made with Independent Component Analysis (ICA). It should be noted, though, that with enough samples, the Central Limit Theorem makes it reasonable to assume normal distributions. Still, these modified forms could be interesting to examine further.

9 References

1. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE5325
2. Ringnr, M. (2008). What is principal component analysis? Nature Biotechnology, 26(3), 303-304. doi:10.1038/nbt0308-303

3. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE5325
4. Shlens, J. (2003). A TUTORIAL ON PRINCIPAL COMPONENT ANALYSIS Derivation, Discussion and Singular Value Decomposition.
5. Jauregui, J. (2012). Principal Component Analysis with Linear Algebra.
6. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: with applications in R. New York: Springer.