PHP2650: Statistical Learning and Big Data

Assignment 2 - Dimension Reduction

Antonella Basso

March 4, 2022

Problem 1: Simulation of Eigenvalues

One rule of thumb some people use when thinking about the number of principal components to keep is to consider all those with eigenvalues above 1. Consider the setting in which we have a matrix X with n observations of p independent variables each following a standard normal distribution $N(0, 1)$. Simulate the matrix X 1,000 times and find the distribution of eigenvalues for the corresponding principal components. Comment on your results.

Next, design a simulation experiment to show how this distribution changes as the variables become more correlated. You should consider how you will generate the matrix X and describe what you expect the eigenvalues to look like based on your design. For example, you may generate each observation using a multivariate normal distribution and change the covariance matrix for each experiment.

Solution

Part I: Assuming an $n \times p$ matrix, X, such that each variable (column) $p_i \sim N(0, 1)$, we can obtain its eigenvalues by applying the `eigen` R command to the corresponding $p \times p$ covariance matrix. Given that the first principal component (eigenvector given by the first and largest eigenvalue) captures the most information or variance within the data compared to the others, and that only principal components whose eigenvalues are greater than 1 ought to be considered, we can get a sense for how this "largest" eigenvalue is distributed by simulating the matrix X 1,000 times and constructing a histogram of "first" eigenvalues observed. Letting $n = 1000$ and $p = 10$, we see (in Figure 1 below) that this "first" eigenvalue is (relatively) normally distributed between 1.2 and 2, with a mean of ≈ 1.6. Although the simulated data demonstrated a few instances of larger more extreme values (possible outliers), we see in Figure 1 that out of all simulated "first" eigenvalues of X, none were at or fell below 1. This tells us that when our variables are *iid* on $N(0, 1)$, we will (almost suely) always obtain at least 1 principal component (the first) that explains the data better than a single original variable. Moreover, Figure 2 shows the distribution of the optimal number of principal components for this kind of data. That is, the total number of eigenvalues greater than 1 observed in each realization of X. Here, we see that the majority of the time, either 4 or 5 eigenvalues, out of 10 computed, were greater than 1. Specifically, around half of the simulated X matrices had 5 eigenvalues greater than 1 and around 400 of them had 4 greater than 1. It is also worth noting that no realizations of X had more than 7 or less than 3 eigenvalues greater than 1. This indicates that for this type of data, we wouldn't likely obtain more than 7 principal components that are better at explaining the variation within it than the variables already present, and similarly, that at least 3 of them will capture even more information.

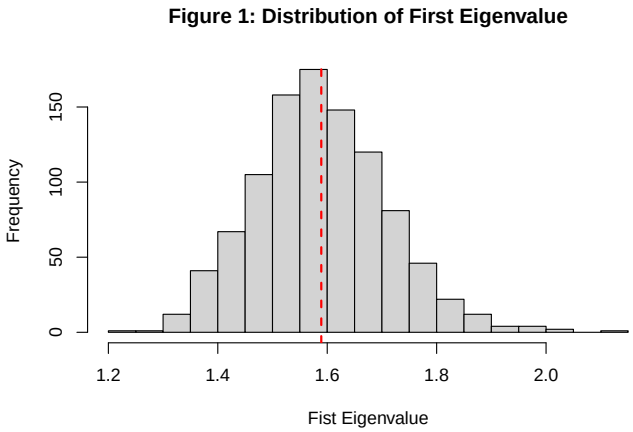


Figure 1: Distribution of First Eigenvalue

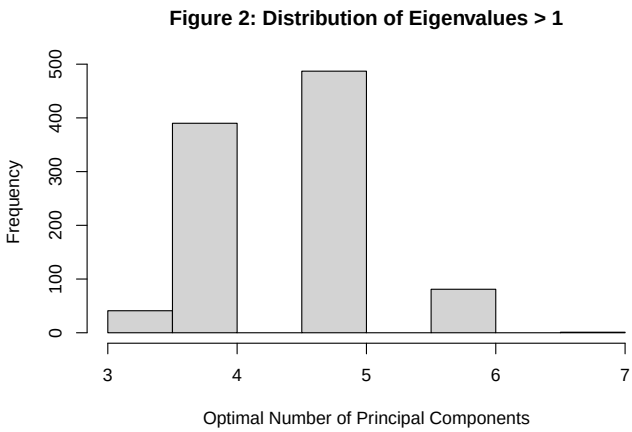


Figure 2: Distribution of Eigenvalues > 1

Part II: To observe how the number of "significant" principal components and the amount of variation explained by them changes as the variables in our data become more correlated, we may consider an experiment in which we simulate both uncorrelated and correlated data; obtain their principal components; and plot the corresponding (cumulative) proportion of the variance they explain. Here, we use a multivariate normal distribution to simulate the data, as it allows us to specify a covariance matrix (Sigma) which can be constructed to reflect different levels of correlation. Specifically, letting X be a 100×10 matrix, we use the 10×10 identity matrix to specify the first covariance pattern (Sigma) we want our data to emulate. With 1's along the diagonal and 0's elsewhere, this covariance pattern reflects the lack of covariance (and hence, lack of correlation) that we would want to condition the multivariate normal distribution on to obtain the most "uncorrelated" data possible. Conversely, to simulate more correlated data, we would simply generate an $n \times 10$ matrix from an arbitrary distribution (here we use 10 Poisson random variables) and use its covariance matrix to parametarize the multivariate normal distribution. This way, we specify a covariance pattern that is guaranteed to yield, at the very least, slightly more correlated data than the previous pattern. Generating 1,000 realizations of X from a multivariate normal distribution with each of these covariance patterns, we obtain the corresponding cumulative and noncumulative proportions of variances explained to produce Figures 3-6 below. Particularly, Figures 3 and 4 display the cumulative variances explained by the principal components for the uncorrelated and correlated data, respectively. By looking at these curves, it is clear that more correlation between variables results in fewer principal components that explain much greater proportions of the variation within the data. This difference is shown both in Figures 3-4, as the points deviate from the red line with added correlation, and in Figures 5-6, as the proportion of variance explained by each principal component decreases at a much faster rate. Specifically, the curvature displayed in the graphs produced with the correlated data, which contrasts the almost linear patterns reflected in the other graphs, is enough to prove that having more correlation between variables in our data allows us to reduce its dimension more drastically. That is, it allows us to capture the same information in fewer dimensions compared to highly uncorrelated data, for which we would need a much larger space to represent. Thus proving that PCA is only useful when our data is correlated enough to be captured by a number of principal components that yields a truly valuable dimension reduction.

Figure 3: Mean Cumulative Variance Explained by PC's
Uncorrelated Variables

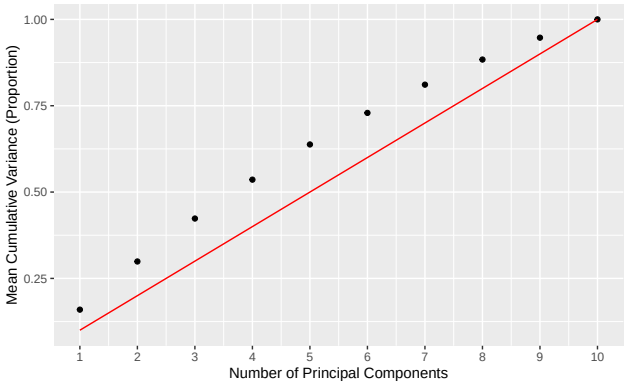


Figure 4: Mean Cumulative Variance Explained by PC's
Correlated Variables

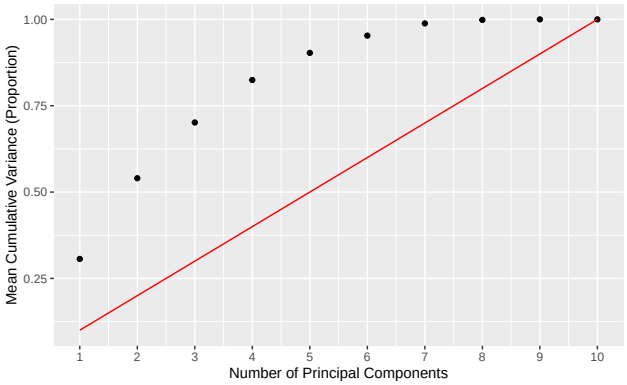


Figure 5: Mean Variance Explained by Each PC
Uncorrelated Variables

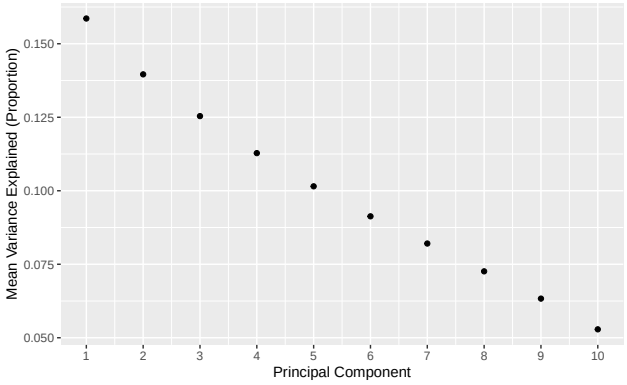
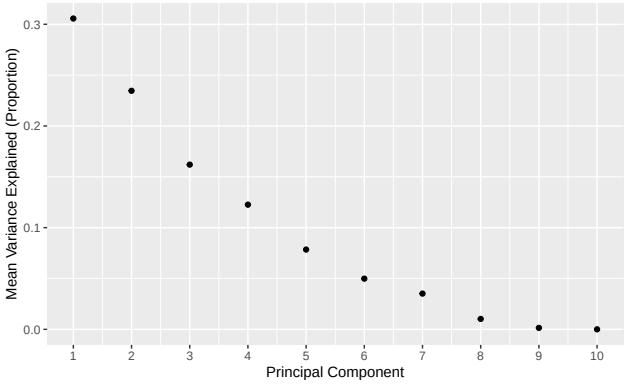


Figure 6: Mean Variance Explained by Each PC
Correlated Variables



Problem 2: ISL Problem 10.6

A researcher collects expression measurements for 1,000 genes in 100 tissue samples. The data can be written as a 1,000 by 100 matrix, which we call X, in which each row represents a gene and each column a tissue sample. Each tissue sample was processed on a different day, and the columns of X are ordered so that the samples that were processed earliest are on the left, and the samples that were processed later are on the right. The tissue samples belong to two groups: control (C) and treatment (T). The C and T samples were processed in a random order across the days. The researcher wishes to determine whether each gene's expression measurements differ between the treatment and control groups.

As a pre-analysis (before comparing T versus C), the researcher performs a principal component analysis of the data, and finds that the first principal component (a vector of length 100) has a strong linear trend from left to right, and explains 10% of the variation. The researcher now remembers that each patient sample was run on one of two machines, A and B, and machine A was used more often in the earlier times while B was used more often later. The researcher has a record of which sample was run on which machine.

a) Explain what it means that the first principal component explains 10% of the variation.
b) The researcher decides to replace the $(i,j)^{\text{th}}$ element of X with $x_{ij} - z_{i1}\phi_{j1}$ where z_{i1} is the i^{th} score, and ϕ_{j1} is the j^{th} loading, for the first principal component. They will then perform a two-sample t-test on each gene in this new data set in order to determine whether its expression differs between the two conditions. Critique this idea, and suggest a better approach.
c) Design and run a small simulation experiment to demonstrate the superiority of your idea.

Solution

a) If the **first** principal component of our $1,000 \times 100$ matrix, that is, the first eigenvector (of the covariance matrix of our data) explains 10% of the variation within the data, this means that:

- The most information or variance between genes that can be captured by a single principal component (one dimension) is 10%
- There exists some (linear) relationship or correlation between (at least two) tissue samples (columns) in the data (which can be reduced to a single dimension to preserve 10% of gene information via orthogonal projection)
- 90% of the information or variance between genes not captured by the first principal component is captured by the remaining principal components

b) Before performing a two-sample t-test it might be beneficial to incorporate into the dataset the known information regarding the machines used to run patient samples. Particularly since the first principal component identifies a strong linear trend among sequential samples, and we're told that machine A was used more frequently on those obtained earlier, while machine B was used more frequently on those obtained later, the inclusion of this information could significantly improve the outcome of the test as well as the level of variation explained by the first principal component.

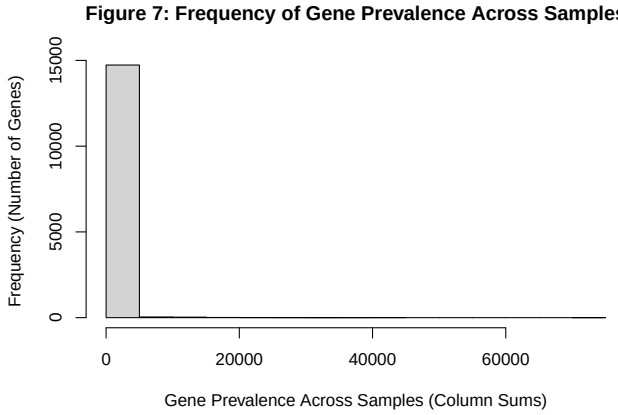
c) Assuming that this information is included as an additional row in the data, we set the $1,001^{\text{th}}$ row of a $1,000 \times 100$ simulated matrix (from a standard normal distribution with the first 200 entries reflecting a linear trend from left to right) to equal a vector whose first 50 and last 50 entries have the values 10 and 0, representing the use of machines A and B, respectively. Computing the proportion of variance explained by the first principal component for both this matrix and one which excludes the last "machine" row, we see that the former results in 10.3% of variance explained, while the latter (excluding the information about machine use) only 7.2%. Thus, in using the most amount of information available, this approach increases the level of variation in the data captured by (at least) the first principal component and may hence, also improve the outcome of a t-test.

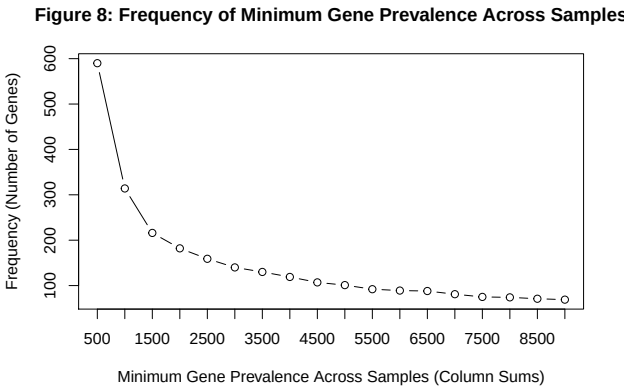
Problem 3: Application to Sequencing Data

The data set `rna_seq_data.csv` in the Data folder on Canvas has recorded transcription counts of 32,738 genes from 2,000 samples of blood mononuclear cells. There is also information about the cell type. You can read more about the data here: https://www.nature.com/articles/ncomms14049. First, load in and explore the data. Then, consider why researchers might consider dimension reduction on such data and complete a dimension reduction analysis. You should justify any modifications you make to the data and interpret your results. As part of your dimension reduction, you should consider informative ways to visualize the data. Write your overall analysis as a report.

Solution

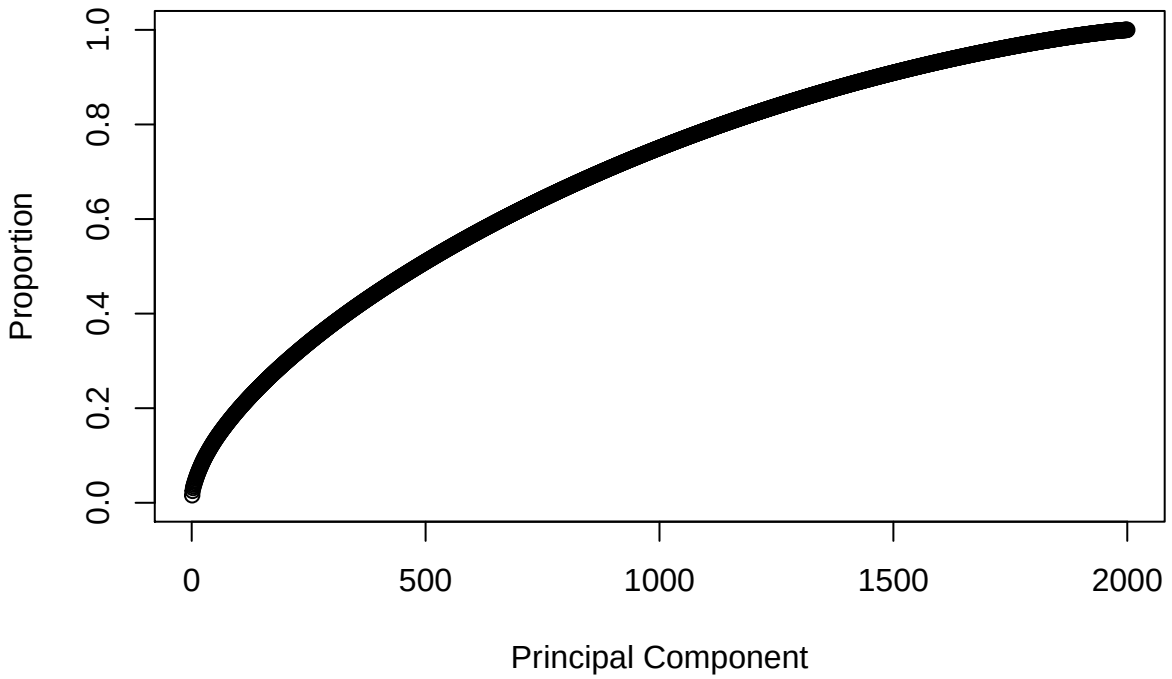
1) **Data Exploration**: In exploring the $2,000 \times 32,739$ data, we notice that the last column gives the cell type (of which there are 10) of the sample being taken (row). Additionally, we see that $\frac{64357392}{65478000} \approx 98.29\%$ of all observations are zero values, and $\frac{17907}{32739} \approx 54.70\%$ of columns have all zero values. This not only implies that most cell samples lack a significant portion of these genes, but that more than half of them fail to show up in any sample at all. For this reason (in addition to the PCA algorithm failing to run on the whole dataset), it serves us best to direct our attention to the subset of genes that were actually observed. To get a sense of gene prevalence across samples for such genes (that appear in at least one sample), we construct a histogram (Figure 7 below) of column (gene) sums and a line plot (Figure 8 below) displaying the number of genes whose counts were least $500, 1000, 1500, ..., 9000$ between all samples. From these plots, we gather that most of the genes observed had very low prevalence, while only a small number of them (relative to the number of genes observed) had significantly large counts across samples (which seem to decay almost exponentially with respect to gene frequency).





2) **PCA**: The first method of dimension reduction implemented on the data was simple PCA, the results of which were not ideal. Figure 9 below illustrates the resulting cumulative proportion of variance explained by the principal components, which exhibit a rate of increase that researchers may not find compelling enough to adopt this method of dimension reduction. That is, implementing this method would force us to keep around 1,250 (of 2,000 possible) principal components to retain only 80% of the information within the data, which is not vastly different from what could be explained by 1,600 of the original variables (assuming they each explain the same amount of information). The results obtained from running simple PCA also tell us that there isn't significant linear correlation among genes (variables) in the data, meaning that their relationships (if present) can't be captured by this algorithm.

Figure 9: Cumulative Variance Explained by PC's



3) **Kernel PCA**: Given the observations made by implementing simple PCA, which tell us that this set of data is not linearly separable, we proceed the dimension reduction analysis by running a kernel PCA algorithm instead, which (by projection onto a higher dimensional space) makes the data linearly separable to yield more optimal dimensionality reduction. Figures 10 and 11 below show the cumulative proportion of variance explained by the resulting principal components, which display a much more

desirable outcome. Specifically, the plots tell us that, under this version of PCA, roughly 50% of the variance within the data can be explained with only the first principal component. And subsequently, that nearly 70% and 80% of information is captured by the first two and first ten principal components, respectively. Moreover, we see that while the first principal component contributes 50% to the total variance explained and the second almost 20%, the remaining (1,997) principal components, each rapidly contribute significantly less to the total variance explained, meaning that we would need a notable quantity of them (and vastly greater dimensionality) to account for even an additional 5% or 10% of information. Thus, it is perhaps naturally best to observe only the first two more closely. Figures 12 and 13 illustrate their independent distributions across all samples in the data, from which we may gather that the majority falls within a small range, while only a few samples (relative to the rest) are widely spread out around more extreme values of principal components 1 and 2. Given the frequency of gene prevalence (column sums) touched on previously, this is not a surprising feature for these principal components to have (as they are essentially combinations of the variables/genes/columns in the data). This pattern the data displays with regards to the first two principal components (and genes more broadly, since these capture roughly 70% of the data's variation) can also be observed in Figure 14 below, which allows us to visualize the general spread of samples as it pertains to their corresponding cell-type categories. Here, we not only see that most samples are generally clustered around a particular small range of principal components 1 and 2, but for two cell types in particular ("CD34+" and "Dendritic"), a number of samples are widely spread along greater values of principal component 1 (to the right on the graph), and while those of cell-type "CD34+" steadily grow in values of principal component 2, those of cell-type "Dendritic" decline in a similar fashion (forming the kind of mirror-image we observe in the plot).

Figure 10: Cumulative Variance Explained by PC's

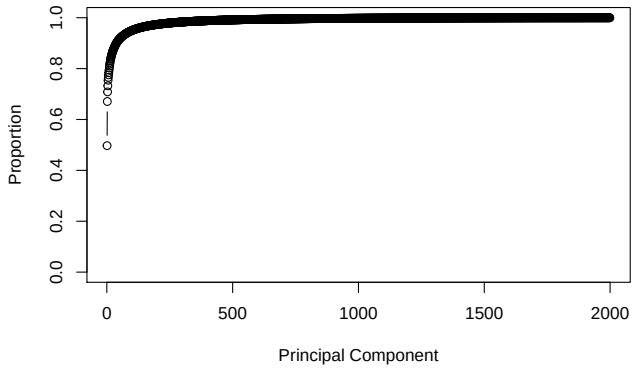


Figure 11: Cumulative Variance Explained by PC's

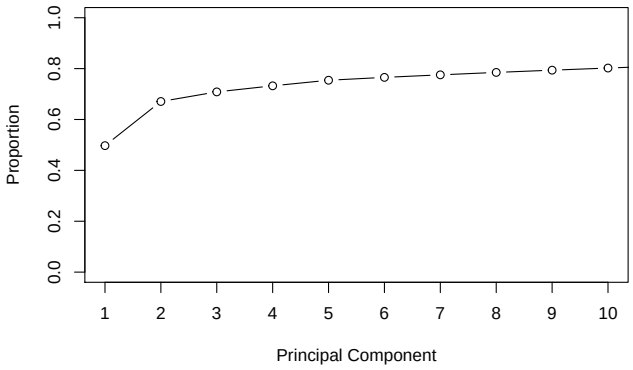


Figure 12: Distribution of Principal Component 1

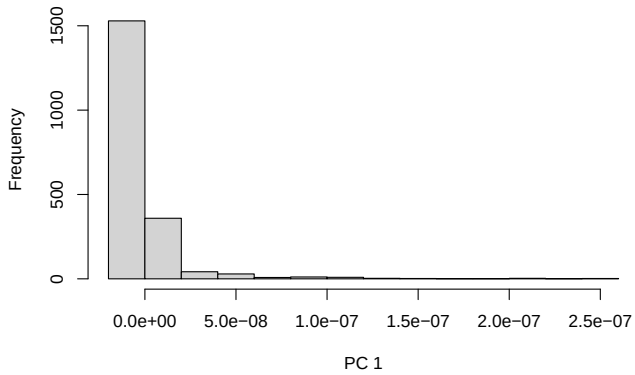


Figure 13: Distribution of Principal Component 2

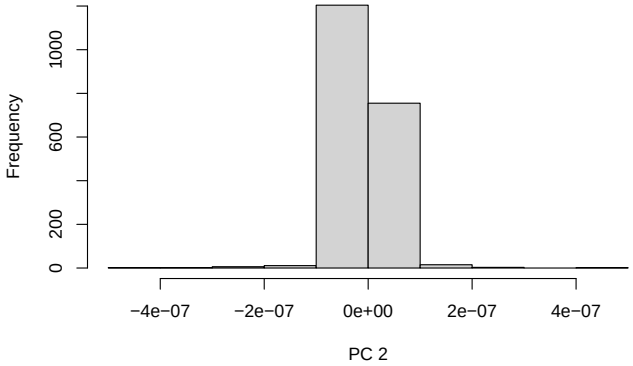
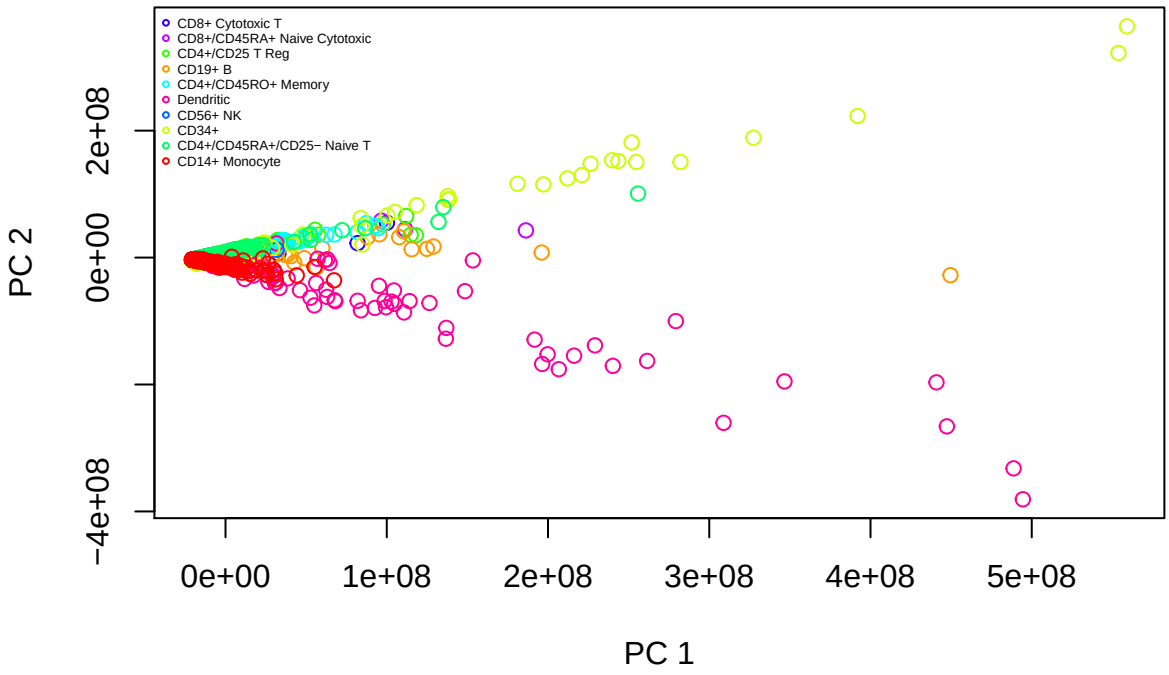
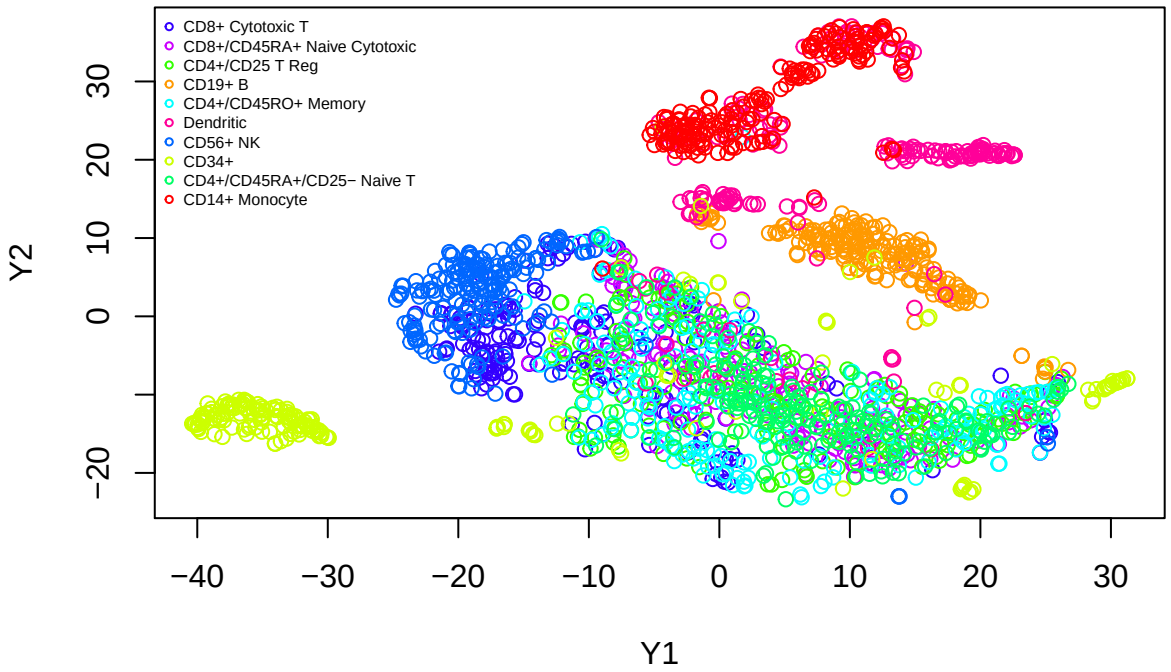


Figure 14: Principal Components 1 vs. 2



4) **t-SNE**: The last dimension reduction technique used on the data was t-SNE, the results of which yield strategic means for visualizing clusters and naturally-occurring spread patterns in the data. Specifically, Figure 15 below illustrates the (probabilistic) mapping of our high-dimensional data onto the two-dimensional space given by "Y1" and "Y2". Having preserved the color scheme from Figure 14, this graph allows us to get a sense for the between- and within-sample differences in the context of cell type. In particular, we notice that cell-types "Dendritic", "CD19+ B", "CD34+", and "CD14+ Monocyte" form pretty distinct individual clusters, while several others (in addition to some samples from the aforementioned cell-types) seem to come together to form a larger and less distinguishable cloud that appears to blend into the "CD56+ NK" cell-type. Moreover, not only do the "CD34+" and "Dendritic" cell-types form clusters that are relatively isolated from the rest, but unlike the other isolated cell-types, they have a considerable number of samples in the larger mixed group and lie on opposite sides of it, similar to what was observed in Figure 14.

Figure 15: t−SNE



5) **Conclusion**: As was observed in the data exploration, most genes have considerably low counts, appearing very infrequently across samples, while a small fraction of them are extremely prevalent. And, since the first principal component identified by the kernel PCA algorithm explains nearly 50% of the variation in the data, it is quite possible for principal component 1 to be some large combination of very rare genes, while others combinations of more prevalent genes (of which there are very few). Moreover, given the results from the t-SNE implementation above, it is also possible that some cell-types have more distinct patterns of gene prevalence, while many others are very similar in this respect. It could also be the case that this cluster of samples (from mixed cell-types) are those for which there is very low gene prevalence, and (potentially) reflective of principal component 1 as well as Figures 7 and 8 from the data exploration above. These however, are mere speculations, and given the vast differences in the algorithms used, more extensive analyses would need to take place in order to make more definitive claims. Nevertheless, variables (genes/columns) are such that a large portion of them can (and hence, ought to) be reduced to a lower dimension as a way to more effectively gauge the relative similarities and differences between samples.

Code

```
############################  QUESTION 1 PART 2 ############################

library(MASS)
library(tidyverse)

set.seed(47)

####### Functions for Variance and Cumulative Variance Explained #######

#### A: Getting Variance Explained by Each Principal Component
get_var <- function(p, Sigma){
  #p MVN variables, each with mean 0, and given covariance matrix
  x <- mvrnorm(100, mu=rep(0,p), Sigma)
  pc_x <- prcomp(x) #principal components (eigenvectors)
  pc_x_var <- pc_x$sdev^2 #principal component variances (eigenvalues)
  var <- pc_x_var/sum(pc_x_var) #variance explained by each principal component
  return(var)
}

##### B: Getting Cumulative Variance Explained by Principal Components
get_cum_var <- function(p, Sigma){
  #p MVN variables, each with mean 0, and given covariance matrix
  x <- mvrnorm(100, mu=rep(0,p), Sigma)
  pc_x <- prcomp(x) #principal components (eigenvectors)
  pc_x_var <- pc_x$sdev^2 #principal component variances (eigenvalues)
  cum_var <- cumsum(pc_x_var/sum(pc_x_var)) #cumulative variance explained
  return(cum_var)
}


####### Covariance Matrices for MVN Uncorrelated and Correlated Data #######

#### Covariance Matrix 1: Identity
#each variable has a variance of 1
#variables have no covariance (thus, they are not correlated)
Sigma <- matrix(0, nrow=10, ncol=10)
diag(Sigma) <- 1

#### Covariance Matrix 2: Poisson Data Covariance
#simulating Poisson data to get covariance matrix for MVN (more correlated than identity)
pois_x <- matrix(rpois(1000, 4), nrow=10, ncol=10, byrow=FALSE) #matrix X (nxp)
Sigma2 <- cov(pois_x)


####### Simulation Plots #######

#### Simulation A.1
#each row gives the cumulative variance explained at each principal component (10)
#each column represents a different iteration (1000)
sim_A1 <- replicate(1000, get_cum_var(10, Sigma))

# Mean Cumulative Variance Explained
```

```r
#getting the mean for each principal component (row=1)
mean_cum_var <- apply(sim_A1, 1, mean)

# Figure 3: Mean Cumulative Variance Explained Across Principal Components
ggplot() +
  geom_point(aes(x=1:10, y=mean_cum_var)) +
  scale_x_continuous(breaks=seq(0, 10, by=1)) +
  geom_line(aes(x=1:10, y=seq(0.1,1, 0.1)), color="red") +
  labs(x="Number of Principal Components",
       y="Mean Cumulative Variance (Proportion)",
       title="Figure 3: Mean Cumulative Variance Explained by PC's",
       subtitle="Uncorrelated Variables")

#### Simulation A.2
#each row gives the cumulative variance explained at each principal component (5)
#each column represents a different iteration (1000)
sim_A2 <- replicate(1000, get_cum_var(10, Sigma2))

# Mean Cumulative Variance Explained
#getting the mean for each principal component (row=1)
mean_cum_var2 <- apply(sim_A2, 1, mean)

# Figure 4: Mean Cumulative Variance Explained Across Principal Components
ggplot() +
  geom_point(aes(x=1:10, y=mean_cum_var2)) +
  scale_x_continuous(breaks=seq(0, 10, by=1)) +
  geom_line(aes(x=1:10, y=seq(0.1,1, 0.1)), color="red") +
  labs(x="Number of Principal Components",
       y="Mean Cumulative Variance (Proportion)",
       title="Figure 4: Mean Cumulative Variance Explained by PC's",
       subtitle="Correlated Variables")

##### Simulation B.1
#each row gives the variance explained by each principal component (10)
#each column represents a different iteration (1000)
sim_B1 <- replicate(1000, get_var(10, Sigma))

# Mean Variance Explained
#getting the mean for each principal component (row=1)
mean_var <- apply(sim_B1, 1, mean)

# Figure 5: Mean Variance Explained by Each Principal Component
ggplot() +
  geom_point(aes(x=1:10, y=mean_var)) +
  scale_x_continuous(breaks=seq(0, 10, by=1)) +
  labs(x="Principal Component",
       y="Mean Variance Explained (Proportion)",
       title="Figure 5: Mean Variance Explained by Each PC",
       subtitle="Uncorrelated Variables")

#### Simulation B.2
#each row gives the variance explained by each principal component (10)
#each column represents a different iteration (1000)
```

```r
sim_B2 <- replicate(1000, get_var(10, Sigma2))

# Mean Variance Explained
#getting the mean for each principal component (row=1)
mean_var2 <- apply(sim_B2, 1, mean)

# Figure 6: Mean Variance Explained by Each Principal Component
ggplot() +
  geom_point(aes(x=1:10, y=mean_var2)) +
  scale_x_continuous(breaks=seq(0, 10, by=1)) +
  labs(x="Principal Component",
       y="Mean Variance Explained (Proportion)",
       title="Figure 6: Mean Variance Explained by Each PC",
       subtitle="Correlated Variables")

rm(list=c("get_var", "get_cum_var", "Sigma", "Sigma2", "pois_x", "sim_A1", "mean_cum_var", "sim_A2", "m
```

```r
############################  QUESTION 2 PART C ############################

set.seed(47)

# Matrix 1: not including machine information
gene_X1 <- matrix(rnorm(1000*100, 0, 1), nrow=1000, ncol=100, byrow=FALSE)
for (i in 1:200){
  gene_X1[i, ] <- (seq(-1, 0.98, 0.02)) # linear trend from left to right
}
# PCA
gene_X1_pca <- prcomp(gene_X1)
# Proportion of Variance Explained by PC's
gene_X1_pca$sdev^2/sum(gene_X1_pca$sdev^2)

# Matrix 2: including machine information
gene_X2 <- rbind(gene_X1, c(rep(10, 50), rep(0, 50))) #1001th row
# PCA
gene_X2_pca <- prcomp(gene_X2)
# Proportion of Variance Explained by PC's
gene_X2_pca$sdev^2/sum(gene_X2_pca$sdev^2)

rm(list=c("gene_X1", "gene_X1_pca", "gene_X2", "gene_X2_pca", "i")) #freeing up RAM
```

```r
#########################  QUESTION 3: Data Exploration  #########################

# loading data
rna_seq <- read.csv("/Users/antonellabasso/Desktop/PHP2650/DATA/rna_seq_data.csv")
#head(rna_seq)

# dimensions
#sum(rna_seq == 0) #zero values
#sum(rna_seq != 0) #non-zero values
#dim(rna_seq)

# cell types
#unique(rna_seq[,32739])
```

```r
# matrix with cell-type column removed
rna_seq2 <- rna_seq[,-32739]

# matrix with cell-type column & zero columns removed
rna_seq3 <- rna_seq2[, colSums(rna_seq2 != 0) > 0]
```

```r
# Figure 7: Gene Prevalence Across Samples (Column Sums)
hist(colSums(rna_seq3),
     main="Figure 7: Frequency of Gene Prevalence Across Samples",
     xlab="Gene Prevalence Across Samples (Column Sums)",
     ylab="Frequency (Number of Genes)")

# Figure 8: Frequency of Minimum Gene Prevalence Across Samples
gene_counts_ge <- c(seq(500, 9000, 500))
freq_gc_ge <- c()
for (i in gene_counts_ge){
  gc <- sum(colSums(rna_seq3) >= i)
  freq_gc_ge <- c(freq_gc_ge, gc)
}
plot(x=gene_counts_ge,
     y=freq_gc_ge,
     main="Figure 8: Frequency of Minimum Gene Prevalence Across Samples",
     xlab="Minimum Gene Prevalence Across Samples (Column Sums)",
     ylab="Frequency (Number of Genes)",
     xaxt="n",
     type="b")
axis(1, at=c(seq(500, 9000, 500)))

rm(list=c("gene_counts_ge", "freq_gc_ge", "rna_seq2", "gc", "i")) #freeing up RAM
```

```r
################################  QUESTION 3: PCA  ################################

# PCA
rna_pca <- prcomp(rna_seq3, center=TRUE, scale=TRUE)
#summary(rna_pca)

# Figure 9: Cumulative Variance Explained by PC's
var_pca <- rna_pca$sdev^2
cum_var_pca <- cumsum(var_pca)/sum(var_pca)
plot(cum_var_pca,
     main="Figure 9: Cumulative Variance Explained by PC's",
     xlab="Principal Component",
     ylab="Proportion",
     ylim=c(0,1),
     type="b")

rm(list=c("rna_pca", "var_pca", "cum_var_pca", "rna_seq3")) #freeing up RAM
```

```r
# #installing packages
#
# #kernel PCA
# install.packages("kernlab")
library(kernlab)
#
```

```r
# #t-SNE
# install.packages("Rtsne")
library(Rtsne)
```

```r
#transforming data

#subsetting data for visualization
rna_ct <- data.frame(x=rna_seq[, colSums(rna_seq != 0) > 0],
                     y=as.factor(rna_seq[,32739]))
#levels(rna_ct$y) #cell-type categories

#creating a color for each cell-type category
colors <- rainbow(length(unique(rna_ct$y)))
```

```r
############################  QUESTION 3: kernel PCA  ############################

# kernel PCA
ker_pca_rna <- kpca(~., data=rna_ct,
                    kernel="polydot",
                    kpar=list(scale=0.5, degree=3))
#dim(pcv(ker_pca_rna)) #principal components
#dim(rotated(ker_pca_rna)) #projected data

# Plotting cumulative variance (kernel PCA):

# for all PC's - Figure 10: Cumulative Variance Explained by PC's
var_kpca <- eig(ker_pca_rna)
cum_var_kpca <- cumsum(var_kpca)/sum(var_kpca)
plot(cum_var_kpca,
     main="Figure 10: Cumulative Variance Explained by PC's",
     xlab="Principal Component",
     ylab="Proportion",
     xlim=c(1,2000),
     ylim=c(0,1),
     type="b")

# # for the first 100 PC's
# plot(cum_var_kpca,
#      main="Cumulative Variance Explained by PC's",
#      xlab="Principal Component",
#      ylab="Proportion",
#      xlim=c(1,100),
#      ylim=c(0,1),
#      type="b")
#
# # for the first 20 PC's
# plot(cum_var_kpca,
#      main="Cumulative Variance Explained by PC's",
#      xlab="Principal Component",
#      ylab="Proportion",
#      xlim=c(1,20),
#      ylim=c(0,1),
#      xaxt="n",
#      type="b")
```

```r
# axis(1, at=c(2,4,6,8,10,12,14,16,18,20))

# for the first 10 PC's - Figure 11: Cumulative Variance Explained by PC's
plot(cum_var_kpca,
     main="Figure 11: Cumulative Variance Explained by PC's",
     xlab="Principal Component",
     ylab="Proportion",
     xlim=c(1,10),
     ylim=c(0,1),
     xaxt="n",
     type="b")
axis(1, at=c(1:10))
```

```r
# Figures 12-13: Distribution of Principal Components 1-2

hist(pcv(ker_pca_rna)[,1],
     main="Figure 12: Distribution of Principal Component 1",
     xlab="PC 1")
hist(pcv(ker_pca_rna)[,2],
     main="Figure 13: Distribution of Principal Component 2",
     xlab="PC 2")
# hist(pcv(ker_pca_rna)[,3],
#      main="Distribution of Principal Component 3",
#      xlab="PC 3")
# hist(pcv(ker_pca_rna)[,4],
#      main="Distribution of Principal Component 4",
#      xlab="PC 4")
```

```r
# Figure 14: Principal Components 1 vs. 2
plot(rotated(ker_pca_rna),
     col=colors[rna_ct$y],
     main="Figure 14: Principal Components 1 vs. 2",
     xlab="PC 1",
     ylab="PC 2")
legend("topleft",
       legend=unique(rna_ct$y),
       col=unique(colors[rna_ct$y]),
       pch=1, cex=0.4, bty="n")

rm(list=c("ker_pca_rna", "var_kpca", "cum_var_kpca")) #freeing up RAM
```

```r
################################  QUESTION 3: t_SNE  ################################
set.seed(3919)

# t-SNE for 2D visualization
rna_tsne <- Rtsne(rna_seq[, colSums(rna_seq != 0) > 0], dims=2)
#names(rna_tsne) #algorithm names/components

# Figure 15: t-SNE
plot(rna_tsne$Y, t="n",
     main="Figure 15: t-SNE",
     xlab="Y1",
     ylab="Y2")
points(rna_tsne$Y, col=colors[rna_ct$y])
```

```
legend("topleft",
       legend=unique(rna_ct$y),
       col=unique(colors[rna_ct$y]),
       pch=1, cex=0.5, bty="n")

rm(list=c("rna_tsne", "rna_ct", "colors", "rna_seq")) #freeing up RAM
```