Stat 437 HW4A

Audrey Kimball (11659795)

```
#load libraries
library(knitr)
library(ElemStatLearn)
library(MASS)
library(klaR)
library(reshape2)
library(ggplot2)
opts_chunk$set(fig.align="center",tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

General rule

Please show your work and submit your computer codes in order to get points. Providing correct answers without supporting details does not receive full credits. This HW covers

- Discriminant analysis

For an assignment or project, you DO NOT have to submit your answers or reports using typesetting software. However, your answers must be well organized and well legible for grading. Please upload your answers in a document to the course space. Specifically, if you are not able to knit a .Rmd/.rmd file into an output file such as a .pdf, .doc, .docx or .html file that contains your codes, outputs from your codes, your interpretations on the outputs, and your answers in text (possibly with math expressions), please organize your codes, their outputs and your answers in a document in the format given below:

```
Problem or task or question ...
Codes ...
Outputs ...
Your interpretations ...
```

It is absolutely not OK to just submit your codes only. This will result in a considerable loss of points on your assignments or projects.

Conceptual exercises: III (Discriminant analysis)

3. Exercise 2 of Section 4.7 of the Text, which starts with "It was stated in the text that classifying an observation to the class for which (4.12) is largest is equivalent to classifying an observation to

the class for which (4.13) is largest. Prove that this is the case." (Helpful information on how to prove this is contained in the lecture video on LDA and "LectureNotes5b_notes.pdf".)

In order to prove that observation x using the posterior probability $p_k(x)$ is the equivalent to using the discriminant function $\delta_k(x)$ this will be done by maximizing $p_k(x)$ over k is equivalent to maximizing $\delta_k(x)$ over k.

The posterior probability $p_k(x)$ is:

$$p_k(x) = \frac{\pi_k \cdot \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{-2\sigma^2}(x-\mu_k)^2)}{\sum_{l=1}^{K} \pi_l \cdot \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{-2\sigma^2}(x-\mu_l)^2)}$$

Since the denominator is independent of k, maximizing $p_k(x)$ is equivalent to maximizing numerator:

$$\pi_k \cdot \exp(-\frac{1}{2\sigma^2}(x-\mu_k)^2)$$

take log of the numerator

$$\log(\pi_k \cdot \exp(-\frac{1}{2\sigma^2}(x-\mu_k)^2)) = \log(\pi_k) - \frac{1}{2\sigma^2}(x-\mu_k)^2$$

Expand the square

$$\log(\pi_k) - \frac{1}{2\sigma^2}(x^2 - 2x\mu_k + \mu_k^2)$$

Group terms:

$$= -\frac{x^2}{2\sigma^2} + \frac{x\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

The term $-\frac{x^2}{2\sigma^2}$ is constant with respect to k so it doesn't affect which k maximizes the expression. So we can for for classification purposes:

$$\delta_k(x) = \frac{x\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

This is the discriminant function:

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

this is very clearly the exact same equation in the end

4. Exercise 3 of Section 4.7 of the Text, which starts with "This problem relates to the QDA model, in which the observations within each class are drawn from a normal distribution with a class specific mean vector and a class specific covariance matrix. We consider the simple case where p = 1; i.e. there is only one feature." (Helpful information on how to prove this is contained in the lecture video on QDA and "LectureNotes5b_notes.pdf".)

To prove that under QDA model where each class has its own variance σ_k^2 the Bayes classifier is not linear and in fact the decision boundary is quadratic.

In general the Bayes classifer assigns a point x to class k that maximizes the posterior probability:

$$P(Y = k \mid X = x) \propto \pi_k \cdot f_k(x)$$

- π_k is the prior probability for class k
- $f_k(x)$ is the class conditional density equation

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)$$

So the classifier chooses the class k that maximize:

$$\pi_k \cdot \frac{1}{\sqrt{2\pi}\sigma_k} \exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)$$

Take the log to simplify

$$\log(\pi_k \cdot \frac{1}{\sqrt{2\pi}\sigma_k} \exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)) = \log(\pi_k) - \log(\sqrt{2\pi}\sigma_k) - \frac{1}{2\sigma_k^2}(x-\mu_k)^2$$

Group it:

$$\delta_k(x) = -\frac{1}{2\sigma_k^2}(x^2 - 2x\mu_k + \mu_k^2) + \log(\pi_k) - \log(\sqrt{2\pi}\sigma - k)$$

The discriminant function $\delta_k(x)$ contains and x^2 term and the coefficient of the x^2 depends on the class-specific variance σ_k^2. Therefore, when we compare two classes, the difference in discriminants $\delta_k(x) - \delta_l(x)$ will include a quadratic term in x. That is:

$$\delta_k(x) - \delta_l(x) = Ax^2 + Bx + C$$

so the decision boundary, defined by $\delta_k(x) = \delta_l(x)$, is a quadratic equation in x. When the class variances σ_k^2 are not equal, the discriminant function includes a quadratic term, and Bayes classifier is quadratic, not linear.

5. Exercise 5 of Section 4.7 of the Text, which starts with "We now examine the differences between LDA and QDA." (Hint: for this question, you may also use information from Figures 4.9, 4.10 and 4.11 in the Text.)

a) If the Bayes decision boundary is linear, then on the training set, QDA will likely perform better because it is more flexible and can fit the training data more closely—even if the true decision boundary is linear. However, on the test set, we expect LDA to perform better. Since the true boundary is linear, LDA models it appropriately without unnecessary flexibility, resulting in lower variance and better generalization. In contrast, QDA might overfit the training data due to its added complexity.

b) If the Bayes decision boundary is non-linear, then on the training set, QDA is expected to perform better because of its flexibility, allowing it to capture non-linear patterns more effectively. On the test set, QDA will generally perform better if the sample size is sufficiently large, as it can approximate complex, non-linear boundaries. However, if the dataset is small, QDA's flexibility may lead to overfitting, and LDA might perform better due to its simplicity and lower variance.

c) As the sample size n increases, the test prediction accuracy of QDA relative to LDA improves. This is because QDA estimates more parameters (specifically, a separate covariance matrix for each class) and therefore requires more data to estimate these parameters reliably. With more data, QDA overfits less and becomes better at approximating complex decision boundaries. So, QDA's relative performance improves as n increases.

d) False. Even if the Bayes decision boundary is linear, we will not necessarily achieve a superior test error rate using QDA. While QDA is flexible enough to model a linear boundary as a special case, it estimates more parameters than LDA—each class has its own variance estimate—which introduces higher variance into the model. This increased variance can lead to overfitting, especially when the true boundary is linear and the sample size is small or moderate. LDA is more efficient in this case and typically results in lower test error due to its lower variance.

6. Let Y be the random variable for the class label of a random vector $X \in \mathbb{R}^p$ (where p is the number of features), such that $Y \in \mathcal{G} = \{1, \ldots, K\}$ and $\Pr(Y = k) = \pi_k$ for Class k with $k \in \mathcal{G}$, where $K \geq 2$ is the number of classes. Consider the Gaussian mixture model such that the conditional density of X when it comes from Class k is $f_k(x) \sim \text{Gaussian}(\mu_k, \Sigma_k)$. Given the training set $\mathcal{T}$ of n observations $(x_1, y_1), \ldots, (x_n, y_n)$ on (X, Y), where y_i is the class label of observation x_i, do the following:

6.1) Provide the MLEs of π_k, μ_k and Σ_k for each $k \in \mathcal{G}$ respectively for the case where all Σ_k's are equal and for the case where not all Σ_k's are equal. When $p > n$, is the MLE of Σ_k still accurate? If not, recommend a different estimator for estimating Σ_k and provide details on this estimator.

Case 1: Shared Covariance Matrix ($\Sigma_k = \Sigma$ for all k) LDA

Given n_k observations in class k out of total n the MLEs are:

- Class Prior: $\hat{\pi}_k = \frac{n_k}{n}$
- Class Mean:
$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$
- Pooled Covariance Matrix:
$$\hat{\Sigma} = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

Case 2: Class-specific Covariance Matrices(Σ_k distinct) QDA

- Class Prior and Mean: same as LDA above.
- Class-Specific Covariance Matrix:
$$\hat{\Sigma}_k = \frac{1}{n_k - 1} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

  When $p > n$: In high-dimensional settings (more features than samples), the sample covariance matrices $\hat{\Sigma}_k$ are not accurate. They become singular (i.e., non-invertible), which means they cannot be used directly in LDA or QDA since these methods rely on matrix inversion.

One option is shrinkage:
$$\tilde{\Sigma}_k = (1 - \lambda)\hat{\Sigma}_k + \lambda I$$

- $\lambda \in [0, 1]$ is a tuning parameter typically chosen through cross-validation.
- I is the identity matrix
- This estimator shrinks the sample covariance toward a well-conditioned target (here, the identity), reducing variance and improving stability.

6.2) Assume $p = 2$ and $K = 2$ and $k = 1$. For the density $f_k(x) \sim \text{Gaussian}(\mu_k, \Sigma_k)$, what shape do its contours take, and how does Σ_k control the shape of these contours? How do you check if the conditional density of X given that it comes from Class k is Gaussian?

The contours of a multivariate Gaussian density are ellipses centered at μ_k, because the density function is:

$$f_k(x) \propto \exp(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k))$$

- The shape and orientation of the ellipses are controlled by:
  - Eigenvalues of $\Sigma_k \rightarrow$ length of axes
  - Eigenvectors of $\Sigma_k \rightarrow$ direction of axes

To verify the $X \mid Y = k$ is Gaussian you can:

- Visualize the data: look for elliptical contours in scatter plot
- Multivariate Normality tests
- Examine Residuals

6.3) Is it true that discriminant analysis will perform badly if the Gaussian assumption is violated? (Hint: for this question, you may also use the information provided by Figures 4.10 and 4.11 of the Text.) Let $X = (X_1, \ldots, X_p)^P$, i.e., X_1 up to X_p are the feature variables. Can discriminant analysis be applied to observations of X when some of $X_j, j = 1 \ldots, p$ is a discrete variable (such as a categorical variable)? Explain your answer.

Discriminant analysis does not necessarily perform poorly when the Gaussian assumption is violated. LDA, in particular, is quite robust to mild departures from normality, as shown in Figures 4.10 and 4.11 of the text. However, severe violations—like multimodal or highly skewed distributions—can reduce its effectiveness, in which case other classifiers may be better suited.

While discriminant analysis assumes that all features are continuous and follow a multivariate Gaussian distribution within each class, it can still be applied with categorical variables if they are properly encoded (e.g., one-hot encoding). However, this violates the model's assumptions, and other methods or models designed for mixed data types may be more appropriate.

6.4) What is a ROC curve, and what is AUC? How is AUC used to gauge the performance of a classifier? If you apply the same classifier, say, LDA or QDA under the same Gaussian mixture model, to two data sets that are independently generated from the same data generating process, i.e., that are independently generated from (X, Y) for classification problems, and obtain two ROC curves, would the two ROC curves be quite different? Explain your answer. When there are 3 or more classes, are the codes provided in the lecture notes able to obtain ROC curves and their AUC's for LDA and QDA?

A ROC curve (Receiver Operating Characteristic curve) is a graphical representation that illustrates the performance of a binary classifier by plotting the true positive rate against the false positive rate at various threshold settings. The AUC, or Area Under the Curve, is a numerical summary of this plot and indicates the classifier's overall ability to distinguish between the two classes.

When applying the same classifier, such as LDA or QDA, to two independently generated datasets from the same underlying Gaussian mixture model, the resulting ROC curves are expected to be generally similar but not identical. Differences may arise due to sampling variability and finite sample effects, especially with smaller datasets. However, with sufficiently large samples, these differences typically diminish, and the ROC curves tend to converge due to the consistency of the classifier.

For problems involving three or more classes, ROC curves and AUC—originally designed for binary classification—can still be applied using a one-vs-rest approach. In this setup, each class is treated as the positive class while all others are grouped as the negative class, allowing for the computation of a separate ROC curve and AUC for each class. Whether this is supported by the lecture code depends on its implementation. If the code includes functionality for one-vs-rest classification or uses a library that supports multiclass evaluation, then ROC curves and AUCs can be obtained for LDA and QDA in multi-class settings.

6.5) Describe the key similarities and differences, respectively, between LDA and logistic regression. Provide a situation where discriminant analysis can still be sensibly applied but logistic regression is not well-defined.

LDA and logistic regression are both used for classification and produce linear decision boundaries in the binary case. The key difference is that LDA is a generative model that assumes Gaussian distributions within each class and a shared covariance matrix, while logistic regression is a discriminative model that directly models class probabilities without assuming a distribution for the predictors. Logistic regression handles categorical variables more easily. However, when there is perfect separation in the data, logistic regression may fail to converge, whereas LDA can still be applied since it estimates class distributions directly. This makes LDA useful in small-sample settings or when logistic regression is not well-defined.

Applied exercises: III (Discriminant analysis)

8. The following is on software commands:

(8.1) What is the main cause of the message "Warning in `lda.default`(x, grouping, ...): variables are collinear"? What is the main cause of the message "Error in `qda.default`(x, grouping, ...) : some group is too small for 'qda' "?

The main cause of the warning message in lda.default(x, grouping, ...): "variables are collinear" is that the predictor variables are linearly dependent or perfectly correlated—meaning one variable can be expressed as a linear combination of others. This results in a singular (non-invertible) covariance matrix, which linear discriminant analysis (LDA) requires to be invertible.

The main cause of the error message in qda.default(x, grouping, ...): "some group is too small for 'qda' " is that one or more of the groups have too few observations to estimate their own class-specific covariance matrices, which are necessary for quadratic discriminant analysis (QDA). Without sufficient observations in each group, the covariance matrices cannot be computed reliably.

(8.2) Provide details on the `list` that `predict{MASS}` returns.

When using the predict() function from MASS package on a fitted LDA or QDA model the returned object is a list typically containing

- class: A factor vector of predicted Class labels for the observations
- posterior: A matrix of posterior probabilities for each class. Each row corresponds to an observation, and each column to a class.
- X (only for LDA): The values of the linear discriminants

(8.3) The arguments `gamma` and `lambda` of `rda{klaR}` are usually determined by cross-validation. Can they be set manually?

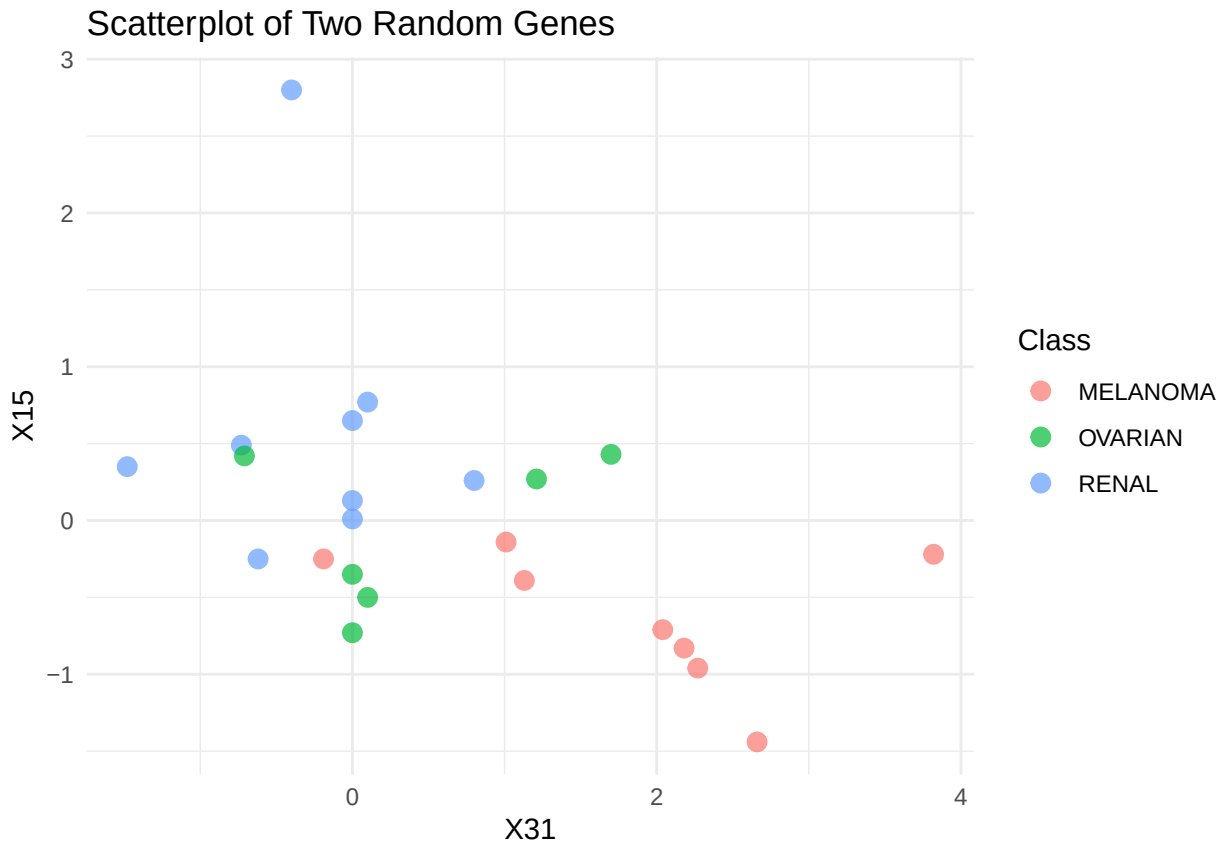
Gamma and Lambda can be set manually when using the rda() function from the klaR package. Gamma controls the weight between LDA and QDA.Lambda shrinks the class-specific covariance matrices toward a diagonal matrix.

9. We will use the human cancer microarray data that were discussed in the lectures and are provided by the R library `ElemStatLearn` (available at https://cran.r-project.org/src/contrib/Archive/ElemStatLearn/). Pick 3 cancer types "MELANOMA", "OVARIAN" and "RENAL", and randomly select the same set of 60 genes for each cancer type. Please use `set.seed(123)` for the whole of this exercise. Your analysis will be based on observations for these genes and cancer types.

```
#set seed
set.seed(123)

#get data dimensions
samples <- dim(nci)[2]
genes <- dim(nci)[1]

#randomly select 60 genes
rand_select <- sample(1:genes, size = 60, replace = FALSE)

#select cancer types
cancer_types <- colnames(nci) %in% c("MELANOMA", "OVARIAN", "RENAL")
check_select <- which(cancer_types)

#create subset
nci_sub <- nci[rand_select, check_select]
colnames(nci_sub) <- colnames(nci)[check_select]

#transpose and adding class labels
cancer_data <- data.frame(t(nci_sub))
cancer_data$Class <- colnames(nci_sub)
rownames(cancer_data) <- NULL
```

9.1) Pick 2 features and visualize the observations using the 2 features. Do you think it is hard to classify the observations based on the amount of overlap among the 3 neighborhoods of observations in each of the 3 classes? Here "a neighborhood of observations in a class" is a "open disk that contains the observations in the class".

```
#reset seed because for some reason without it the genes keep changing even with
#seed set above
set.seed(123)

#randomly pick 2 genes
gene_cols <- sample(1:60, size = 2, replace = FALSE)
gene1 <- colnames(cancer_data)[gene_cols[1]]
gene2 <- colnames(cancer_data)[gene_cols[2]]

#plot
ggplot(cancer_data, aes(x = .data[[gene1]], y = .data[[gene2]], color = Class)) +
  geom_point(size = 3, alpha = 0.7) +
  labs(title = "Scatterplot of Two Random Genes", x = gene1, y = gene2) +
  theme_minimal()
```



Given the overlap in the scatter plot, it would be difficult to perfectly classify the observations using only two genes. The open disks around each class cluster are not fully separate, suggesting that these two genes are not sufficient for accurate classification.

9.2) Apply LDA and report the classwise error rate for each cancer type.

```
#apply lda
lda_fit <- lda(Class ~ ., data = cancer_data )
```

```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

```
#predict using fitted model
lda_pred <- predict(lda_fit, cancer_data)

#True and predicted labels
true_labels_92 <- cancer_data$Class
pred_labels_92 <- lda_pred$class

#confusion matrix
conf_matrix_92 <- table(Predicted = pred_labels_92, Actual = true_labels_92)
print(conf_matrix_92)
```

```
##           Actual
## Predicted  MELANOMA OVARIAN RENAL
##   MELANOMA        6       0     0
##   OVARIAN         0       5     0
##   RENAL           2       1     9
```

```
#classwise error rates
class_totals_92 <- colSums(conf_matrix_92)
class_errors_92 <- class_totals_92 - diag(conf_matrix_92)
class_error_rate_92 <- round(class_errors_92 / class_totals_92, 4)

#output error rates
class_error_rate_92
```

```
## MELANOMA  OVARIAN    RENAL
##   0.2500   0.1667   0.0000
```

Melanoma: 0.2500 Ovarian: 0.1667 Renal: 0.0000

9.3) Use the library `klaR`, and apply regularized discriminant analysis (RDA) by setting the arguments `gamma` and `lambda` of `rda{klaR}` manually so that the resulting classwise error rate for each cancer type is zero.

```
#apply rda

rda_fit <- rda(Class ~ ., data = cancer_data, gamma = 0.05, lambda = 0.01)

#predict using rda model
```

```r
rda_pred <- predict(rda_fit, cancer_data)

#true predicted labels
true_labels_93 <- cancer_data$Class
pred_labels_93 <- rda_pred$class

#confusion matrix
conf_matrix_93 <- table(Predicted = pred_labels_93, Acutual = true_labels_93)
print(conf_matrix_93)
```

```
##           Acutual
## Predicted  MELANOMA OVARIAN RENAL
##   MELANOMA        8       0     0
##   OVARIAN         0       6     0
##   RENAL           0       0     9
```

```r
#classwise error rates
class_totals_93 <- colSums(conf_matrix_93)
class_errors_93 <- class_totals_93 - diag(conf_matrix_93)
class_error_rate_93 <- round(class_errors_93 / class_totals_93, 4)

class_error_rate_93
```

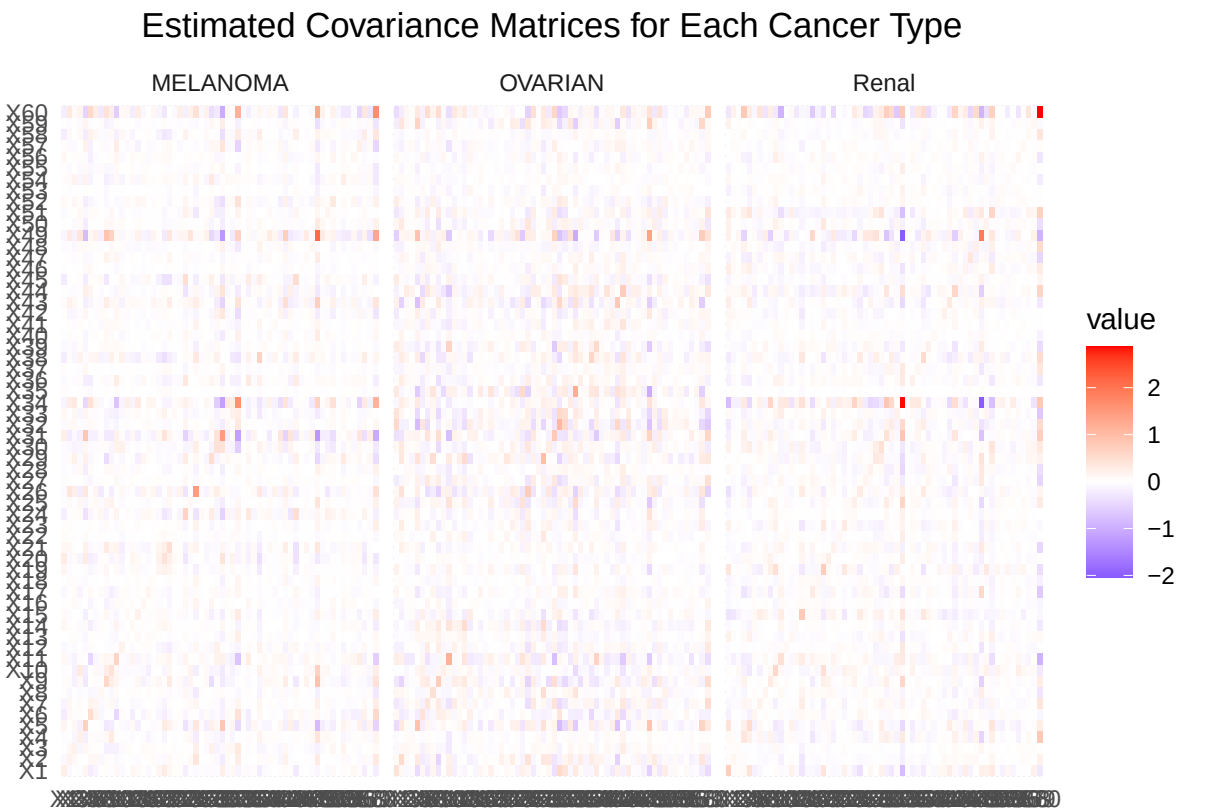
```
## MELANOMA  OVARIAN    RENAL
##        0        0        0
```

Melanoma: 0 Ovarian: 0 Renal: 0

9.4) Obtain the estimated covariance matrices from the RDA and visualize them using the same strategy in Example 3 in "LectureNotes5c_notes.pdf". What can you say about the degree of dependence among these genes for each of the three cancer types? (Hint and caution: the class labels "MELANOMA", "OVARIAN" and "RENAL" will be ordered alphabetically by R. So, you need to keep track on which estimated covariance matrix is for which class. Otherwise, you will get wrong visualization.)

```r
#extract covariance matrices from rda
sigma_melan <- rda_fit$covariances[, , 1] #melanoma
sigma_ovar <- rda_fit$covariances[, , 2] #ovarian
sigma_ren <- rda_fit$covariances[, , 3] #renal

#melt matrices
melt_melan <- melt(sigma_melan)
melt_ovar <- melt(sigma_ovar)
melt_ren <- melt(sigma_ren)

#add cancer type label
```

```r
melt_melan$Cancer <- "MELANOMA"
melt_ovar$Cancer <- "OVARIAN"
melt_ren$Cancer <- "Renal"

#combine into one data frame
sigma_all <- rbind(melt_melan, melt_ovar, melt_ren)
sigma_all$Cancer <- factor(sigma_all$Cancer)

#plot the covariance matrices
ggplot(data = sigma_all, aes(x = Var1, y = Var2, fill = value)) + geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint =  0) +
  facet_grid(~ Cancer) + xlab("") + ylab ("") +
  ggtitle("Estimated Covariance Matrices for Each Cancer Type") +
  theme_minimal() + theme(plot.title = element_text(hjust = 0.5))
```



The covariance heatmaps show that melanoma has the strongest gene-gene dependence, with clear positive and negative covariance patterns. Ovarian displays weaker, more scattered covariances, suggesting the genes act more independently. Renal shows moderate dependence with some structured patterns. This indicates that gene relationships can differ across cancer types, which may impact classification.