BIOST 2155: Assignment 4

YOUR NAME HERE (YOUR PITT EMAIL HERE)

Released: 2026-10-30. **Due by**: 2026-11-13, 11:59 PM

Instructions

- This file serves as a template for your assignment. **Remember to set eval = TRUE below when compiling the report.**

```
pacman::p_load(formatR)
knitr::opts_chunk$set(echo = TRUE, eval = FALSE,
                      fig.height = 4, fig.width = 6, fig.align = "center",
                      message=FALSE, tidy.opts = list(width.cutoff = 70), tidy = TRUE,
                      class.output  = "shadebox",
                      class.message = "shadebox",
                      class.warning = "shadebox",
                      class.error   = "shadebox")
```

- Use comments to annotate and explain your code whenever possible.
- This assignment covers **Clustering II** (Gaussian mixture models) and **Dimensionality Reduction** (principal component analysis). It uses two data sets: `Palmer Penguins` (`penguins.csv`, continued from Assignment 3) for Part A, and `Wisconsin Breast Cancer` (`assign3_data.csv`, continued from Assignment 3) for Part B.

Theory (15 points)

T1. Hierarchical clustering (7 + 7 + 1 = 15 points)

ISLR Problem 12.6.2, page 546. Suppose that we have four observations, for which we compute a dissimilarity matrix, given by

$$\begin{pmatrix} & 0.3 & 0.4 & 0.7 \\ 0.3 & & 0.5 & 0.8 \\ 0.4 & 0.5 & & 0.45 \\ 0.7 & 0.8 & 0.45 & \end{pmatrix}$$

For instance, the dissimilarity between the first and second observations is 0.3, and the dissimilarity between the second and fourth observations is 0.8.

(a) (7 points) On the basis of this dissimilarity matrix, sketch the dendrogram that results from hierarchically clustering these four observations using **complete** linkage. Be sure to indicate on the plot the height at which each fusion occurs, as well as the observations corresponding to each leaf in the dendrogram.

(b) (7 points) Repeat (a), this time using **single** linkage.

(c) (1 point) Suppose that we cut the dendrogram obtained in (a) such that two clusters result. Which observations are in each cluster?

Show your working — you may upload handwritten answers/sketches for this problem.

Application, Part A: Clustering II — Gaussian Mixture Models (44 points)

Dataset: `Palmer Penguins`, saved as `penguins.csv` (same data as Assignment 3, Part B).

In Assignment 3 you used k-means and hierarchical clustering — both "hard" assignment methods — to recover penguin species from morphological measurements. Here we revisit the same problem with Gaussian Mixture Models (GMMs), a "soft" probabilistic clustering method in which each observation has a probability of belonging to each cluster, and each cluster is modeled as a multivariate Gaussian component. **Please set a random seed (`set.seed(42)`) before any step involving randomization.**

```
rm(list = ls())
pacman::p_load(tidyverse, mclust)
set.seed(42)
penguins <- read_csv("penguins.csv") |>
    drop_na()
features <- penguins |>
    dplyr::select(bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g)
features_scaled <- scale(features)
```

Using `mclust`. In Lab 8 we wrote EM by hand. Here we use the `mclust` package, which fits the same model in one line.

- `model <- Mclust(features_scaled, G = 1:6)` fits every combination of number of components and covariance structure, and keeps the one with the best BIC. `summary(model)` reports the choice.
- The chosen fit is in `model$G` (number of components) and `model$modelName` (covariance structure). `model$z` holds the soft posterior probabilities, one row per penguin, and `model$classification` the hard cluster labels.
- In `mclust`, **higher BIC is better** (its BIC is defined with the opposite sign to some textbooks).
- Covariance structures have three-letter names for the **volume, shape and orientation** of the clusters, each either `E` (equal across clusters) or `V` (varies across clusters). An `I` means spherical (shape) or axis-aligned (orientation). For example, `EII` is spherical clusters of equal size, close to the k-means assumption, and `VVV` lets every cluster have its own full covariance, the "full" model from Lab 8.

A1. Model Selection via BIC (8 + 4 = 12 points)

1. Fit Gaussian mixture models on the scaled features using `mclust::Mclust()`, letting the function search over `G = 1:6` components and all available covariance structures. Plot the BIC values across the candidate models (`plot(model, what = "BIC")`) and report the number of components and covariance structure selected by BIC.

```
# your code here
```

2. Does the BIC-selected number of components match the known number of penguin species (3)? Briefly comment.

```
# your code here
```

A2. Fitting and Interpreting the GMM (6 + 8 = 14 points)

1. Using the BIC-optimal model from A1, extract the hard cluster assignments (`model$classification`) and the soft posterior probabilities of component membership (`model$z`) for the first 10 observations. Print these posterior probabilities and identify the penguin(s), if any, for which the model is most "uncertain" (i.e. no single component has posterior probability close to 1).

```
# your code here
```

2. Create a cross-tabulation (`table()`) comparing the true `species` labels to the GMM hard cluster assignments. How well does the GMM recover the species groups?

```
# your code here
```

A3. Comparing GMM to k-means (10 + 8 = 18 points)

1. Re-run k-means with `centers = 3` (as in Assignment 3) on the same scaled features and the same seed. Compare the k-means and GMM cross-tabulations against the true species labels — which method achieves a higher agreement with species? Report a simple agreement measure (e.g. the proportion of observations whose cluster label, after matching clusters to the majority species, equals the true species).

```
# your code here
```

2. *Interpretation: GMMs allow elliptical, differently-shaped and differently-oriented clusters (depending on covariance structure), while k-means assumes roughly spherical, equal-variance clusters. Using the covariance structure selected by BIC in A1, does this flexibility seem to help on the penguin data? More generally, describe one real scenario in which the "soft" probabilistic assignments from a GMM would be more useful to a downstream analyst than the hard assignments from k-means.*

Application, Part B: Dimensionality Reduction — PCA (41 points)

Dataset: `Wisconsin Breast Cancer`, saved as `assign3_data.csv` (same data as Assignment 3, Part A).

We return to the breast cancer dataset from Assignment 3. Here, instead of classifying tumors directly from the 10 raw features, we will use Principal Component Analysis (PCA) to reduce dimensionality first, and examine what that buys (or costs) us. **Please set a random seed (`set.seed(42)`) before any step involving data splitting or randomization.**

```
pacman::p_load(tidyverse, caret, ggsci)
set.seed(42)
data <- read_csv("assign3_data.csv") |>
    dplyr::select(-1) |>
    drop_na()
data$diagnosis <- factor(data$diagnosis, levels = c("B", "M"))

train_index <- createDataPartition(data$diagnosis, p = 0.75, list = FALSE)
train <- data[train_index, ]
test <- data[-train_index, ]
```

B1. Preprocessing and PCA (3 + 8 = 11 points)

1. Explain, in 2-3 sentences, why standardizing features is a critical preprocessing step before applying PCA. Then, standardize the 10 features in the training set, and apply the *same* centering/scaling (learned from the training set) to the test set.

```
# your code here
```

2. Perform PCA on the standardized training features. Plot the proportion of variance explained (PVE) by each component (scree plot) and the cumulative PVE. How many components are required to explain at least 80% of the variance?

```
# your code here
```

B2. Visualizing the Components (6 + 6 = 12 points)

1. Create a scatterplot of PC1 vs. PC2 scores for the training data, colored by `diagnosis`. Do the first two components alone appear to separate malignant from benign tumors well?

```
# your code here
```

2. Create a biplot of PC1 vs. PC2 showing the loading vectors. Which original features load most heavily on PC1? Which load most heavily on PC2? *Hint: base R's `biplot()` takes a `prcomp()` object directly. The loadings themselves are in its `$rotation` matrix, as in Lab 9.*

```
# your code here
```

B3. Principal Component Regression for Classification (12 + 6 = 18 points)

1. Using the number of components chosen in B1.2, project both the training and test sets onto that many principal components. Fit a logistic regression model predicting `diagnosis` using these PC scores as predictors, and report the confusion matrix and accuracy on the test set.

```
# your code here
```

2. *Interpretation: Compare this PC-based logistic regression to what you would expect from fitting logistic regression on all 10 raw features directly (you do not need to fit that model — reason about it). What do we gain by reducing to a handful of components before modeling, and what, if anything, do we lose? When would you recommend PCA as a preprocessing step before classification, and when would you avoid it?*