Unsupervised Learning with the Palmer Penguins

MATH38161

Yann McLatchie (Student ID: 10161640)

```
library(dplyr); library(graphics); library(GGally); library(palmerpenguins); library(cluster); library(gridExtra)
library(factoextra); library(ape); library(mclust); library(cowplot)
set.seed(123) # initialisation for PAM
load("penguins.rda")
palette.penguins = c("#008080", "#FFA500", "#800080")
palette(palette.penguins)
```

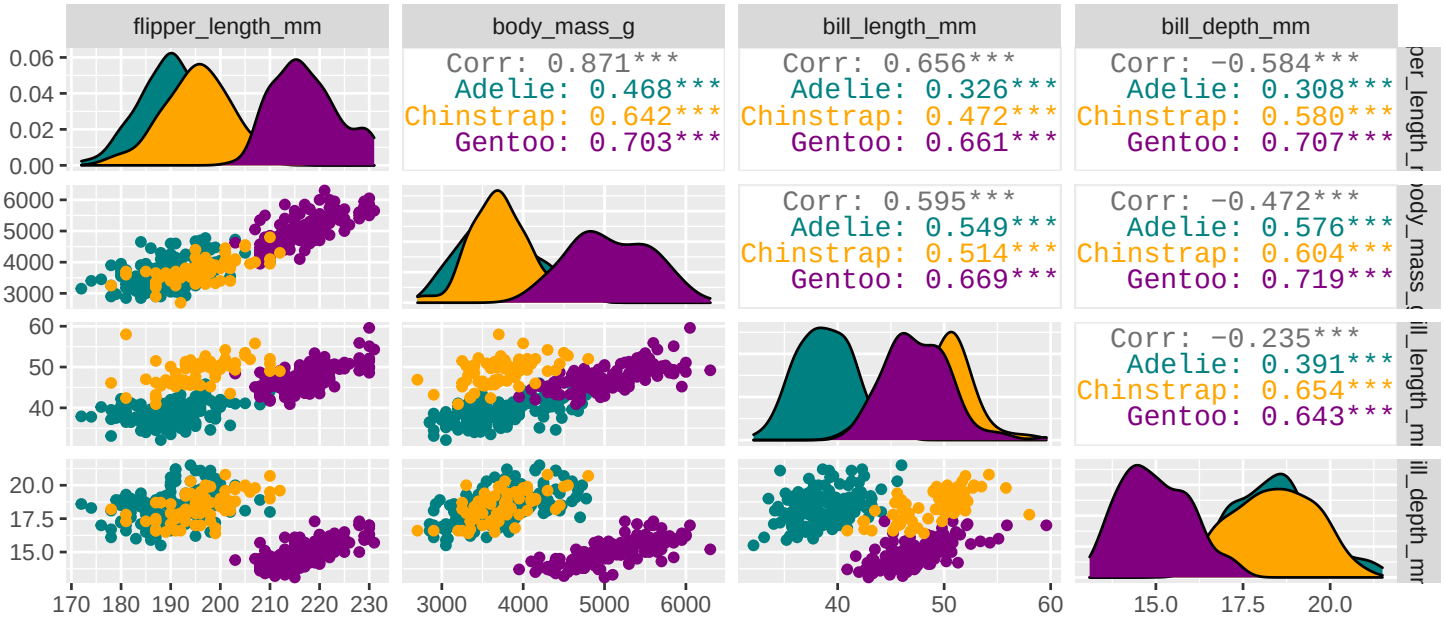
1 Dataset

The dataset used is courtesy of Horst, Hill, and Gorman (2020), where we are provided four body measurements (bill length, bill depth, flipper length, and body mass) for 333 penguins, who belong to one of three different species (Adélie, Chinstrap, and Gentoo). The aim of our analysis on these data is to cluster these penguins using their body measurements.

1.1 Summary Statistics

To begin with, let us plot the different species' body measurement distributions to get a better idea of the data. The code for the following plot is adapted from Horst, Hill, and Gorman (2020).

```
penguins %>% select(species, body_mass_g, ends_with("_mm")) %>% ggpairs(
    aes(color=species), columns=c("flipper_length_mm", "body_mass_g","bill_length_mm", "bill_depth_mm")
  ) + scale_colour_manual(values=palette.penguins) + scale_fill_manual(values=palette.penguins)
```



Immediately, we can see from the distribution of penguins and the correlation figures that `bill length` is the most orthogonal to the other features, and may be more informative in our models to come. The scatter plots of `bill length` against other features produce the most clear clusters. We find from the distributions that Gentoo penguins are larger than both Adélie and Chinstrap, who have similar body mass and flipper length measurements. We will see later that we have some larger female Adélie penguins that could be confused with smaller male Chinstrap penguins. Chinstrap and Adélie can be differentiated by their bill measurements, with Chinstraps having longer bills than Adélie, and both of them having deeper bills than Gentoo. Statistics for the distribution across islands were also produced but were less informative. The primary finding was that

overall, penguins on Biscoe had larger flippers, larger body mass, and flatter bills since they are mostly Gentoo. Given this, we will consider our clustering results exclusively with reference to the penguin species.

1.2 PCA Analysis

The code for this PCA analysis is adapted from Strimmer (2020) and the description of PCA methodology is adapted from Strimmer (2020), and James et al. (2013).

We use PCA to identify a set of representative variables that effectively explain the majority of the variance in the data using as low a dimension as possible. This is an unsupervised method since we do not deal with data labels. We will use PCA as a visualisation tool, producing a lower dimensional representation of our data from which we are able to visualise the distribution of our data in fewer plots.

Assume we have a random vector $\vec{X}$ with $\text{Var}(\vec{X}) = \mathbf{\Sigma} = \mathbf{U\Lambda U}^T$, then PCA is just a transformation of the random vector, $\vec{t}_{\text{PCA}} = \mathbf{U}^T\vec{x}$, such that its resulting components are orthogonal, and equivalently has diagonal covariance matrix. We obtain $\mathbf{U}$ by first estimating the covariance matrix of $\vec{x}$, $\hat{\mathbf{\Sigma}}$, calculate its eigendecomposition, and take $\mathbf{U}$ as the matrix its eigenvectors.

The first principal component of a set of features $X_1, X_2, ..., X_d$ is given as

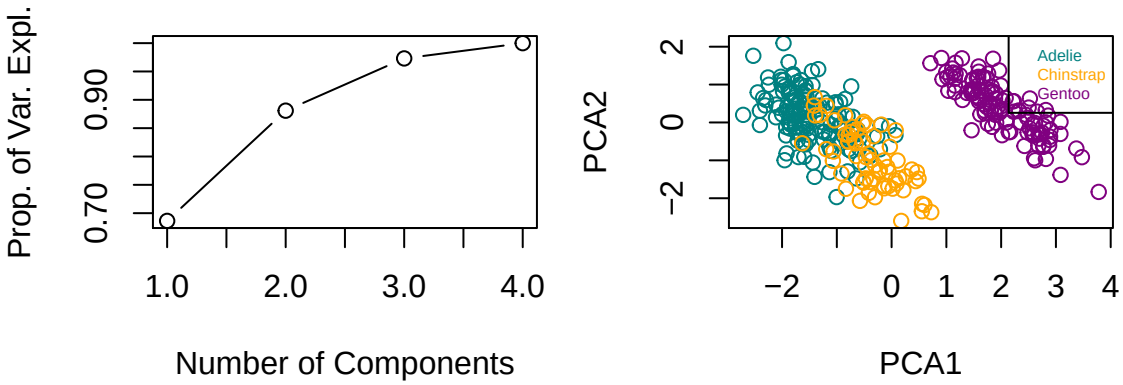
$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{d1}X_d$$

where $\overrightarrow{\phi}_1 = (\phi_{11}, \phi_{21}, ..., \phi_{d1})^T$ is called the *loading vector* of this first principal component.

We can interpret these PCA values geometrically using their loading vectors as follows. Each loading vector defines a direction in the parameter space along which the data varies the most, with each loading vector being orthogonal to each other. When we project our data onto these directions, we achieve the PCA values. By taking the two first principal components, we are thus examining the data across the two directions of most variance.

This is implimented in R as follows.

```
X.scaled = scale(X.penguins, center = TRUE, scale = TRUE)
# PCA calculations
S = cov(X.scaled)
U = eigen(S)$vector
lambda = eigen(S)$values
penguin.PCA = X.scaled %*% U
par(mfrow = c(1, 2), mar = c(4, 4, 1, 1))
plot(cumsum(lambda)/sum(lambda), xlab = "Number of Components", ylab = "Prop. of Var. Expl.", type = "b")
plot(penguin.PCA[,1], penguin.PCA[,2], col=L.species, xlab="PCA1", ylab="PCA2")
legend("topright", legend=levels(L.species), text.col=palette.penguins, cex=0.5)
```



1.3 Split By Sex

From the first two PCs of all the data without splitting, we can easily identify Gentoo, while Adélie and Chinstrap are more confused. The reason for this difficulty in identification between Adélie and Chinstrap is due to the fact that the larger female Adélie penguins resemble the smaller male Chinstrap, as we can see in the following table.

```
df = penguins %>% group_by(species, sex) %>% summarize(
  avg_flipper=mean(flipper_length_mm), avg_mass=mean(body_mass_g), var_flipper=var(flipper_length_mm), var_mass=var(body_mass_g)
  )
df[((df$species=='Adelie' & df$sex=='male')|(df$species=='Chinstrap' & df$sex=='female')) & complete.cases(df),]
```

species	sex	avg_flipper	avg_mass	var_flipper	var_mass
Adelie	male	192.4110	4043.493	43.55099	120278.25

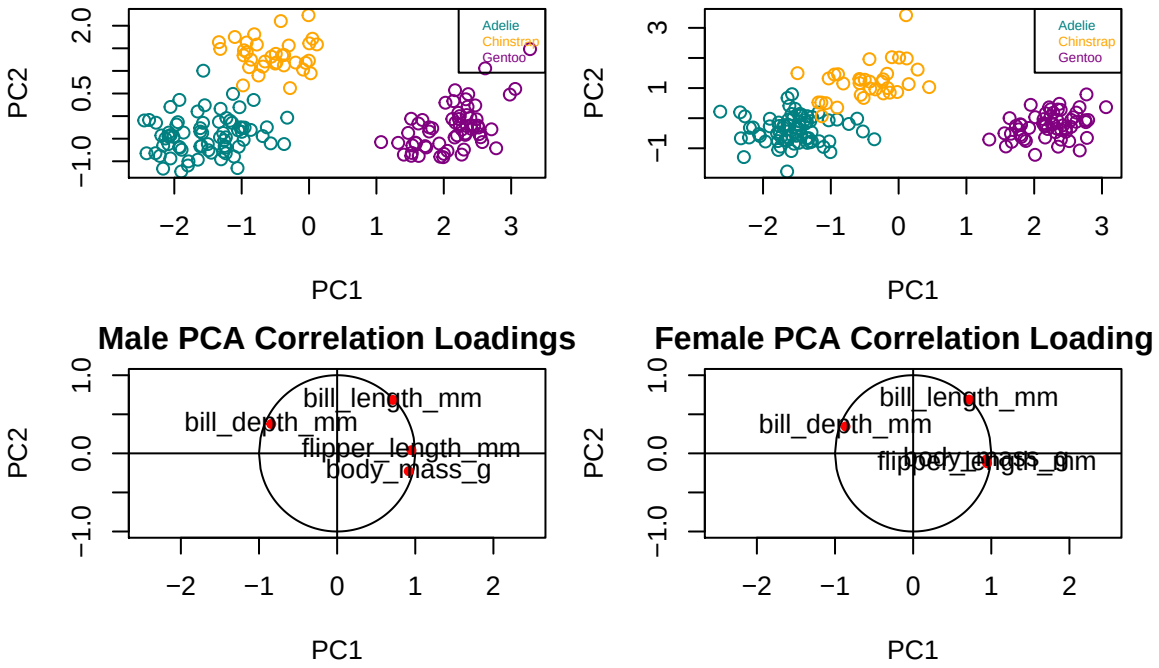
species	sex	avg_flipper	avg_mass	var_flipper	var_mass
Chinstrap	female	191.7353	3527.206	33.10963	81415.44

By separating our data by sex and performing PCA analysis on both sexes individually, we will hopefully achieve a much clearer image.

```
# split the data by sex
M.species = L.species[L.sex == 'male']; F.species = L.species[L.sex == 'female']
M.penguins = scale(X.penguins[L.sex == 'male', ], center = TRUE, scale = TRUE)
F.penguins = scale(X.penguins[L.sex == 'female', ], center = TRUE, scale = TRUE)
```

We scale and center the data so that the four features can be compared, since they are all given in different units and so when we calculate the Euclidean distance between them for our models, the values will not be informative and may even have a negative effect on our model. By scaling and centering them, we ensure that all features are comparable.

```
# compute PCA
M.penguins.PCA = prcomp(M.penguins); F.penguins.PCA = prcomp(F.penguins)
M.PCA = M.penguins.PCA$x; F.PCA = F.penguins.PCA$x
# plot first two PC components
par(mfrow = c(2, 2), mar = c(4, 4, 2, 1))
plot(M.PCA[,1], M.PCA[,2], col=M.species, xlab="PC1", ylab="PC2")
legend("topright", legend=levels(M.species), text.col=palette.penguins, cex=0.5)
plot(F.PCA[,1], F.PCA[,2], col=F.species, xlab="PC1", ylab="PC2")
legend("topright", legend=levels(F.species), text.col=palette.penguins, cex=0.5)
# correlation loadings plot
M.Psi = cor(M.PCA, M.penguins); F.Psi = cor(F.PCA, F.penguins)
plot(1, xlim = c(-1,1), ylim=c(-1,1), type='n', asp = 1, xlab="PC1", ylab="PC2", main="Male PCA Correlation Loadings")
curve(( sqrt(1 - x^2) ), add=TRUE, from=-1 , to =1); curve((-sqrt(1 - x^2) ), add=TRUE, from=-1 , to =1); abline(h=0); abline(v=0)
points( M.Psi[1,], M.Psi[2,], pch=20, col="red" ); text( M.Psi[1,], M.Psi[2,], labels=names(M.Psi[1,]))
plot(1, xlim = c(-1,1), ylim=c(-1,1), type='n', asp = 1, xlab="PC1", ylab="PC2", main="Female PCA Correlation Loadings")
curve(( sqrt(1 - x^2) ), add=TRUE, from=-1 , to =1); curve((-sqrt(1 - x^2) ), add=TRUE, from=-1 , to =1); abline(h=0); abline(v=0)
points( F.Psi[1,], F.Psi[2,], pch=20, col="red" ); text( F.Psi[1,], F.Psi[2,], labels=names(F.Psi[1,]))
```



Much clearer! It was calculated that the first two PCs also contribute about 90% of the total variation for both male and female penguins, so we have effectively reduced the dimensionality of our data while preserving information.

From the correlation loadings plot we see that the first principal component for both the male and female data is highly correlated with flipper length, body mass, and bill depth, and slightly correlated to bill length while the second component is only correlated with bill length. Knowing what these components represent, their distribution conforms with our initial observations from the pairs plot, with Gentoo being physically larger and having flatter bills, and then Adélie having shorter bills than Chinstrap.

2 Methods

The descriptions of the methods involved in this coursework are adapted from Strimmer (2020), James et al. (2013), and Murphy (2013).

2.1 Algorithmic Clustering

For this section, we will make use of the Partition Around Medoids (PAM) clustering algorithm, which is similar to K-Means only using mediods. This method of clustering requires us to state the number of clusters in our data, K, in advance.

We consider a 'good' clustering to have minimal *within-group* variance, where elements in the same cluster are as similar to each other as possible. We define similarity as the Euclidean distance between elements, hence the need for scaling, although when using PAM we have more flexibility when defining this than with K-means.

For n observations, we have K^n possibly clusterings, which are prohibitively expensive to calculate when n gets large. Instead, we implement an algorithm that seeks to find a *local optimum*, in much less time. The algorithm is as follows:

1. Randomly assign the n penguins $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ to clusters $C(\boldsymbol{x}_i) \in \{1, \ldots, K\}$.
2. Estimate the cluster mediods, $\hat{\boldsymbol{\mu}}_k$.
3. Update group assignment; Penguin $\boldsymbol{x}_i$ is assigned to the cluster k with the nearest $\hat{\boldsymbol{\mu}}_k$ to it.
4. Repeat steps (2) and (3) until the algorithm converges and no more penguins are reassigned.

Note that the sum of within-group variances of all our clusters ever only ever decrease, up to a point when it will plateau. This means that once our algorithm finds a local minimum, it will not escape it. As such, our algorithm is sensitive to the initial assignations. We must thus run the algorithm multiple times with different initial configurations and select the clustering with the lowest total within-group variance.

We have chosen to use PAM over K-Means because its use of mediods over means is more robust to noise and outliers in our data, and is thus more likely to converge. However, we make this choice at an increased computation cost with PAM having a complexity of $O(K * (n - K)^2)$, more expensive than K-Means.

2.2 Hierarchical Clustering

There are two types of hierarchical, or tree-based, clustering: divisive and agglomerative. In this analysis we will focus on the latter, where we build the tree from the leaves-upward. The primary advantage of hierarchical clustering over algorithmic methods, is that we need not specify the number of clusters at the beginning. The results are given in a tree-like representation, called a *dendrogram*, which aids interpretability, and we gain the clustering results by cutting the dendrogram at a given height. Cutting a dendrogram at different heights return clusterings with different numbers of clusters, and thus from one execution of the algorithm, we achieve multiple clusterings.

The algorithm is as follows:

1. Compute the distance matrix between all pairs of penguins.
2. Recursively identify the pair of penguins, or pair of clusters, with the smallest distance separation and merge them into a cluster. Create a node in the tree at each merge event and update the distance matrix. The algorithm terminates once all penguins belong to the same cluster.

As a distance measure for pairs of clusters, A and B, we will use Ward's minimum variance approach. We merge the two clusters A and B that, when merged, will have the smallest within-group variance, or equivalently the smallest total within-group sum of squares. Thus scaling was once more necessary. We have chosen to use Ward's clustering in this analysis, as the dendrogram produced is the most stable, and the clusters are the most balanced. That is to say that subtrees in cuts predominantly represent the same class.

We can interpret the dendrogram as follows. Each leaf is an observation of the data with zero within-group variance. As leaves are merged, the within-group variance increases, and the higher up the tree the clustering occurs, the more the within-group variance increases. We thus measure the similarity (or dissimilarity) of elements by the height at which they are clustered.

Hierarchical clustering works best when the data have a nested or hierarchical structure (country, region, city, etc.). We thus recognise that this method may not be suited to our dataset.

2.3 Probabilistic Clustering

In probabilistic clustering, we estimate a mixture model to best describe the data. The set up is as follows. We have K discrete latent states, or classes, and each of these classes $k \in \{1, 2, ..., K\}$ has its own distribution F_k with density $f_k(\vec{x}) = f(\vec{x} \mid k)$

and parameters $\vec{\vartheta}$. We call the *mixing weight* of each class, the probability $\mathbb{P}(k) = \pi_k$, from which we can build the joint density for each latent state $f(\vec{x}, k) = f(\vec{x} \mid k)\mathbb{P}(k) = f_k(\vec{x})\pi_k$. We achieve the mixture model by summing these joint densities with their associated mixing weights.

$$f_{\text{mixture}}(\vec{x} \mid \vec{\vartheta}) = \sum_{k=1}^{K} \pi_k f_k(\vec{x} \mid \vec{\vartheta}).$$

We call a mixture model with multivariate normally distributed components, a *Gaussian Mixture Model* (GMM). In order to use GMMs for clustering, we first fit the mixture model to our data, and compute the posterior probability that the point $\vec{x}$ belongs to the cluster k, $z_k = f(k \mid \vec{x}, \vec{\vartheta})$. Calculating this probability for each point $\vec{x}$ we perform a *soft clustering*, which results a vector $\vec{z} = (z_1, \ldots, z_K)^T$ containing the probability that it belongs to each class. Once we have these soft clusterings, it may be valuable to make *hard clusterings*. To do this, we assign each point $\vec{x}_i$ to a cluster according to the rule $C(\vec{x}_i) = \arg\max_k \log(z_{ik})$. If we assume that the π_k are all equal, then this simplifies to $C(\vec{x}_i) = \arg\min_k (\vec{x}_i - \hat{\vec{\mu}}_k)^T(\vec{x}_i - \hat{\vec{\mu}}_k)$, or K-means clustering.

We will use the `MClust` R package, which estimates the parameters of the mixture model with the help of the EM algorithm to avoid issues with likelihood methods such as singularities and incomplete data, which follows the procedure:

1. Guess the parameters of the model $\vec{\vartheta}^{(1)}$ and proceed to step 2. This guess can be informed by some a priori information, such as running K-means first, or derived completely randomly.
2. **(Expectation Step)** Use Bayes' Theorem to compute the allocation probabilities using the current parameter estimates (at iteration p)
3. **(Maximisation Step)** Maximise the expected complete data log-likelihood using the $z_{ik}^{(p+1)}$ calculated in the E step, and update the parameter estimates.
4. Repeat steps 2 and 3 until $\vec{\vartheta}^{(p+1)}$ converges towards an estimate of the mixture model parameters.

We choose this value K using the penalised log likelihood framework of Bayesian Information Criterion (BIC)

$$\text{BIC} = 2 \log L - K \log(n)$$

which we look to maximise. This metric will look for the model with the highest log liklihood while penalising the number of parameters it uses, favouring less complex models.

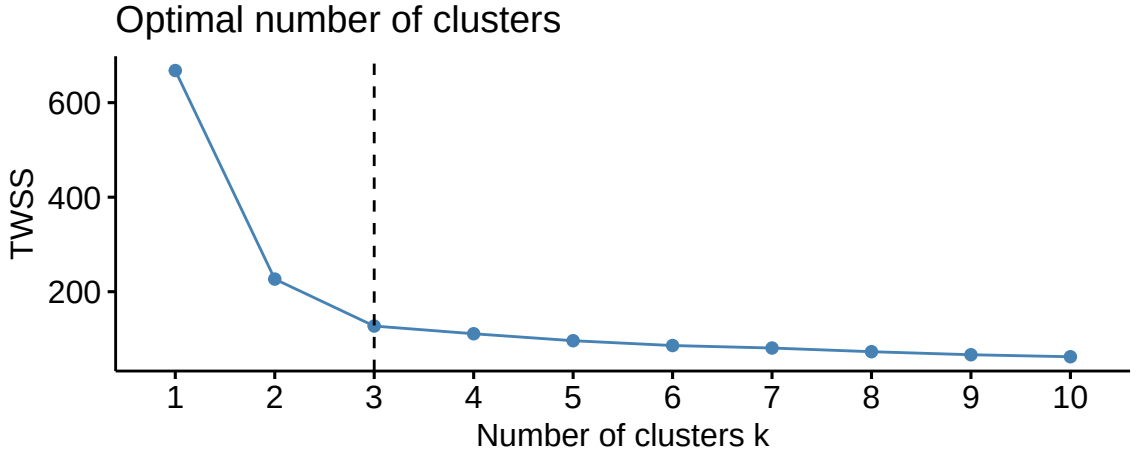
3 Results and Discussion

There is a cardinality difference in our three groups, with nearly twice as many Gentoo and Adélie as Chinstrap. This could cause a reduction in performance when predicting Chinstrap compared to the other two species. The code for the following analyses is adapted from Strimmer (2020).

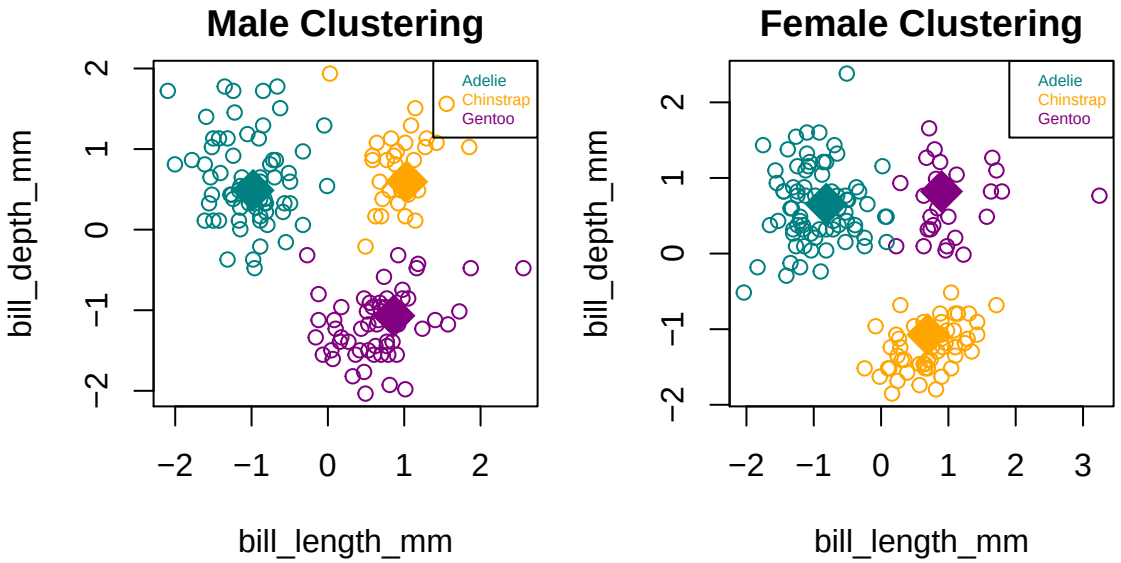
3.1 PAM

When performing a clustering analysis, we usually do not know the true number of classes in the data. In order to determine this, we run our algorithm using different values of K taking note of the total within-group sum of squares (TWSS), the metric which PAM aims to optimise, and select the lowest value of K such that the explained variation, $B = \frac{SSW}{n}$, is not significantly worse that the next largest value of K. We perform this for both male and female models, and find that they are in fact the same so only the graph for the male cluster is shown.

```
fviz_nbclust(M.penguins, pam, method="wss") + geom_vline(xintercept=3, linetype=2) + ylab("TWSS")
```



```
M.clust.pam = pam(M.penguins, k = 3)
F.clust.pam = pam(F.penguins, k = 3)
par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))
plot(M.penguins,col=M.clust.pam$cluster, main="Male Clustering")
points(M.clust.pam$medoids,col=1:3,pch=18,cex=3)
legend("topright", legend=levels(M.species), text.col=palette.penguins, cex=0.5)
plot(F.penguins,col=F.clust.pam$cluster, main="Female Clustering")
points(F.clust.pam$medoids,col=1:3,pch=18,cex=3)
legend("topright", legend=levels(F.species), text.col=palette.penguins, cex=0.5)
```



We find that once the data is split by sex, using Partinioning Around Mediods produces easily identifiable clusters where bills measurements are most informative, as was expected from PCA. The mediods for each cluster are clearly separated across these dimensions and with minimal overlap between clusters.

```
M.t = tableGrob(table(M.clust.pam$clustering, M.species))
F.t = tableGrob(table(F.clust.pam$clustering, F.species))
grid.arrange(M.t, F.t, ncol=2, left="Male", right="Female")
```

Male

	Adelie	Chinstrap	Gentoo
1	72	0	0
2	1	34	0
3	0	0	61

	Adelie	Chinstrap	Gentoo
1	73	4	0
2	0	0	58
3	0	30	0

Female

As a result there is only one male Adélie penguin that is misclustered as a Chinstrap, and only 4 female Chinstrap penguins that our misclustered as Adélie. We knew already that this would be an issue with our data, although the algorithmic method was able to cluster them effectively.

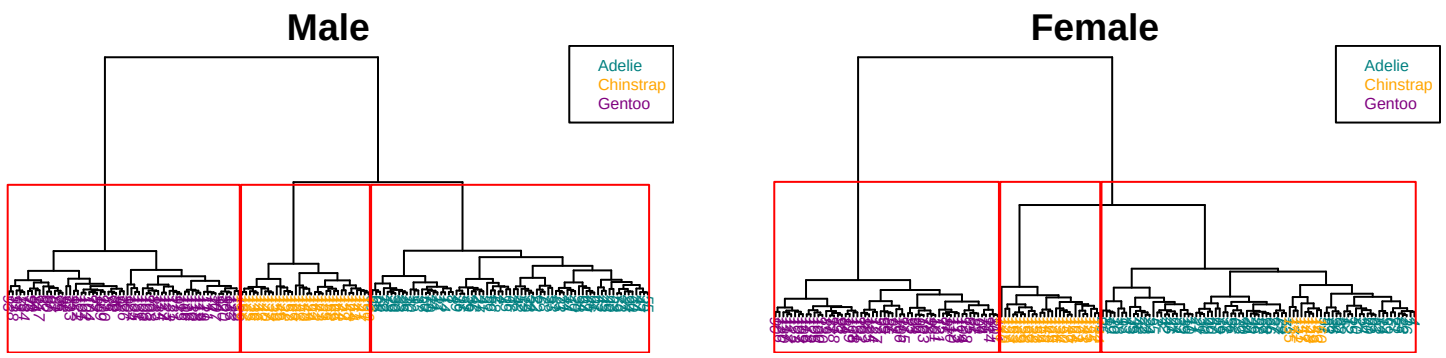
The PAM clustering algorithm was run several times with different initial configurations, and on each run the results were similar and good. This indicates that the use of mediods is indeed robust to noise and outliers and thus less sensitive to initial configurations as K-Means.

3.2 Hierarchical

Much like for PAM, we calculate the total within-group sum of squares for different cut heights and choose from these values the optimal K. Similarly to before, we find this to be $K = 3$ which matches our prior knowledge.

```
M.dmat = dist(M.penguins, method="euclidean")
F.dmat = dist(F.penguins, method="euclidean")
M.clust.tree = hclust(M.dmat, method ="ward.D2")
F.clust.tree = hclust(F.dmat, method ="ward.D2")
par(mfrow = c(1, 2), mar = c(1, 1, 1, 1))
plot(as.phylo(M.clust.tree), cex=0.5, direction="d", tip.color=palette.penguins[as.integer(M.species)], main="Male")
rect.hclust(M.clust.tree, k=3, border="red")
legend("topright", legend=levels(M.species), text.col=palette.penguins, cex=0.5, bg="white")
plot(as.phylo(F.clust.tree), cex=0.5, direction="d", tip.color=palette.penguins[as.integer(F.species)], main="Female")
```

```
rect.hclust(F.clust.tree, k=3, border="red")
legend("topright", legend=levels(F.species), text.col=palette.penguins, cex=0.5, bg="white")
```



We find that using the Ward's method of minimum variance, we do indeed achieve a balanced tree, where there is little variance in the subtrees beneath the cuts. This means that our clusters will be more stable. There is however the exception of a small portion of female Chinstraps that are classified as Adélie. In fact, they are merged at a low height, which suggests they strongly resemble Adélie penguins. This shows that the hierarchical method was not as robust to the outliers present in the female dataset as PAM. We did not however expect tree-based methods to be well-suited to this dataset, as our data are not hierarchical in their interpretation.

```
M.t = tableGrob(table(cutree(M.clust.tree, 3), M.species))
F.t = tableGrob(table(cutree(F.clust.tree, 3), F.species))
grid.arrange(M.t, F.t, ncol=2, left="Male", right="Female")
```

Male

	Adelie	Chinstrap	Gentoo
1	73	0	0
2	0	0	61
3	0	34	0

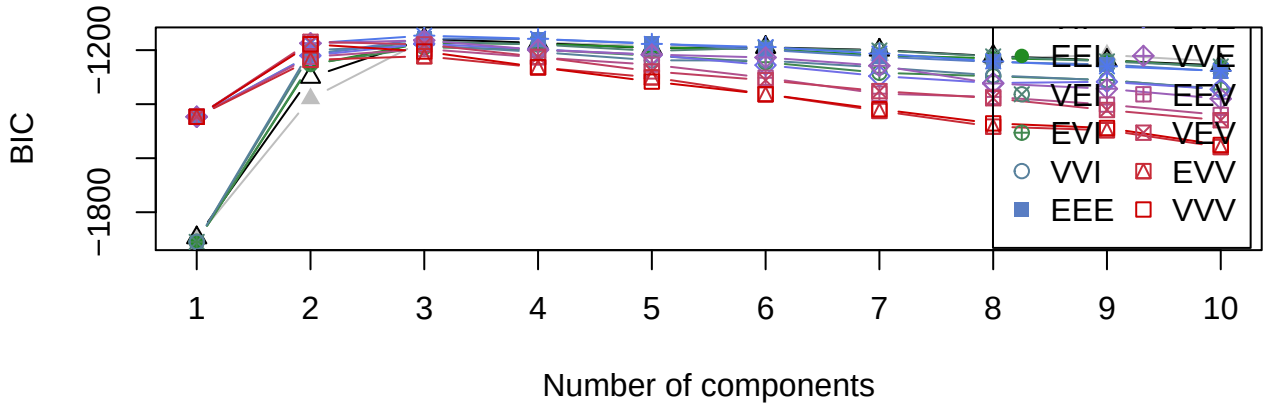
	Adelie	Chinstrap	Gentoo
1	73	8	0
2	0	0	58
3	0	26	0

Female

3.3 GMM

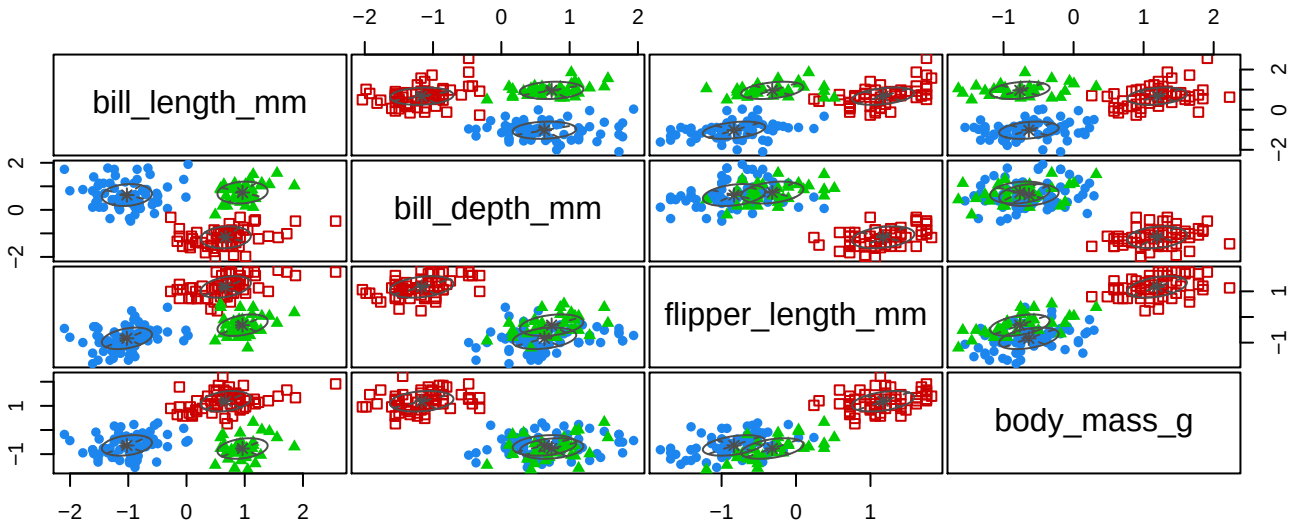
In order to find the optimal value of K, we build multiple GMMs with varying values of K, calculating their associated BIC, and take K from the model with the highest BIC. In this case, we find that indeed the largest BIC is the case when $K = 3$ as expected.

```
gmm.opt <- Mclust(F.penguins, G=1:10, verbose=FALSE)
plot(gmm.opt, what="BIC")
```



From the plot below, we see a clear separation of the densities of the components in the mixture model, and thus a clear separation of classes, allowing for a clear clustering result. Indeed, we see very good separation of the three species across the first two PCs. We only show the male clustering model knowing that the female model is nearly identical.

```
M.clust.gmm = Mclust(M.penguins, G=3, verbose=FALSE)
F.clust.gmm = Mclust(F.penguins, G=3, verbose=FALSE)
plot(M.clust.gmm, what="classification")
```



The uncertainty charts of the clustering was also analysed, where I found that there were in fact very few uncertain data points, and that the clusters are very stable. Those data points that were most uncertain were once more Adélie penguins that resembled Chinstrap and vice versa. By using probabilistic methods, we have more flexibility in converting from soft to hard clustering in the situation where there are uncertain results, and can tweak the threshold accordingly.

```
M.t = tableGrob(table(M.clust.gmm$classification, M.species))
F.t = tableGrob(table(F.clust.gmm$classification, F.species))
grid.arrange(M.t, F.t, ncol=2, left="Male", right="Female")
```

Male

	Adelie	Chinstrap	Gentoo
1	73	0	0
2	0	0	61
3	0	34	0

	Adelie	Chinstrap	Gentoo
1	73	3	0
2	0	31	0
3	0	0	58

Female

We achieve a perfect clustering result for the male penguins, and for the female penguins there are only 3 Adélie that are misclustered as Chinstrap. An excellent result! We will tend to favour probabilistic methods when latency is not a consideration, as they are much more computationally expensive than their algorithmic and hierarchical counterparts although very powerful and interpretable.

3.4 Conclusion

PAM clustering is computable with a lower complexity than hierarchical. GMM is a probabilistic clustering method where the EM algorithm is computationally expensive as it involves large calculations at each iteration, and singularities in the data may prevent log-likelihood calculations. Thus if latency is a factor in the clustering, then one should favour PAM. Hierarchical methods are less intuitive for this dataset as the features aren't nested or hierarchical so will not be our primary choice. GMM was the most performant with this dataset, and provides more information about the uncertainty in its conclusions which could be useful in the situation where certain data is consistently misclassified, and we can modify the threshold for the translation from soft to hard clusterings. We thus favour GMM and probabilistic methods more generally if latency is not an issue.

References

Horst, Allison Marie, Alison Presmanes Hill, and Kristen B Gorman. 2020. *Palmerpenguins: Palmer Archipelago (Antarctica) Penguin Data*. https://doi.org/10.5281/zenodo.3960218.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning: With Applications in R*. Springer. https://faculty.marshall.usc.edu/gareth-james/ISL/.

Murphy, Kevin P. 2013. *Machine Learning : A Probabilistic Perspective*. Cambridge, Mass. [u.a.]: MIT Press. https://www.cs.ubc.ca/~murphyk/MLbook/.

Strimmer, Korbinian. 2020. *Lecture Notes on Multivariate Statistics and Machine Learning*. http://www.strimmerlab.org/publications/lecture-notes/MATH38161.