Before all coding questions were attempted, after the respective packages were downloaded, the following commands were run in R:

```
library(ggplot2)
library(gridExtra)
library(dynRB)
```

1.a) What is the best number of clusters to choose and why?

Assuming all 4 silhouette plots are for the same set of data, 2 is the best number of clusters to choose. When comparing all 4 plots, graph 1 has the highest average silhouette width. Silhouette width describes the degree of separation between clusters. It can range from -1 to 1, 1 meaning the data is far away from neighbouring clusters, and -1 meaning that the data is likely assigned to the wrong cluster. The further away data is from neighbouring clusters, the better, because it means that the data is distinctly clustered/separated. Not only do the two clusters in graph 1 have the highest silhouette width (0.77 and 0.63), the average is also higher ($\mathbf{0.68} > 0.42 > 0.35 > 0.32$).

1.b) What is the second best number of clusters to choose and why?

The second best number of clusters to choose would be 4 clusters, as shown in graph 3. It has the second highest average silhouette width of 0.42, and also has no negative silhouette widths, unlike graph 2 (6 clusters) and graph 4 (8 clusters). This means that when the data is grouped into 4 clusters, none of the data seems to be in the wrong cluster, however, when it is grouped into 6 or 8 clusters, it seems like there is at least one data point that is in the wrong cluster.

2.a) Describe k-means clustering and k-medoids clustering.

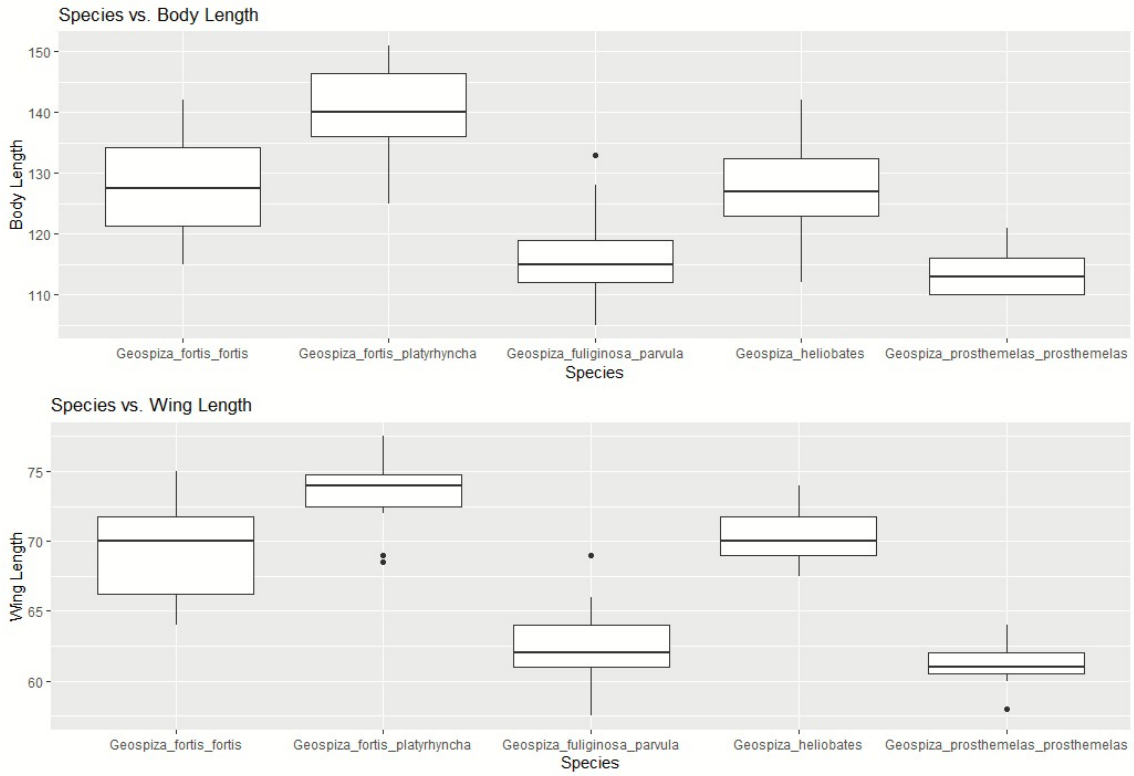
K-means clustering and k-medoids clustering are both partitioning methods where the idea is to group the data into k number of clusters, maximizing the distance between points and their neighbouring clusters, while also minimizing the distance between points within clusters as well as their cluster centres. We want to maximize the distance between points and their neighbouring clusters because ideally, we want each cluster to be distinct without having overlapping points with other clusters. This is so that the data is either in cluster a or cluster b, and it is clear which cluster it should belong to. We want to minimize the distance between points within clusters as well as their cluster centres so that the points are more similar to those that are within their own cluster, and not similar to points in other clusters, or none of the clusters. Using the elbow method or the silhouette method, we can select the best value of k. The main difference between k-means clustering and k-medoids clustering is the fact that the the cluster centres in k-means clustering are means (averages, not actual data points), whereas the cluster centres in k-medoids clustering are medoids (actual data points that are closest to the centre of each cluster). It should also be noted that both partitioning methods work best when dealing with spherical clusters, otherwise, errors are more prone to occurring when using the algorithms.

2.b) Would you expect k-means clustering to work reasonably well on the 3 clusters depicted below? Explain your answer.

Yes, I would expect k-means clustering to work reasonably well on the clusters depicted below. For starters, the data can easily be divided into 3 distinct clusters (meaning that $k = 3$ easily), where the points in each cluster are close to each other but are not very close to neighbouring clusters. We do not see many overlapping points between each cluster (only a few between the yellow and orange clusters, but overall not many), which means that each point can be distinctly placed into a cluster. Along with this, the clusters are somewhat spherical (as opposed to scattered or in the form of lines for example), which works well with both k-means clustering as well as k-medoids clustering.

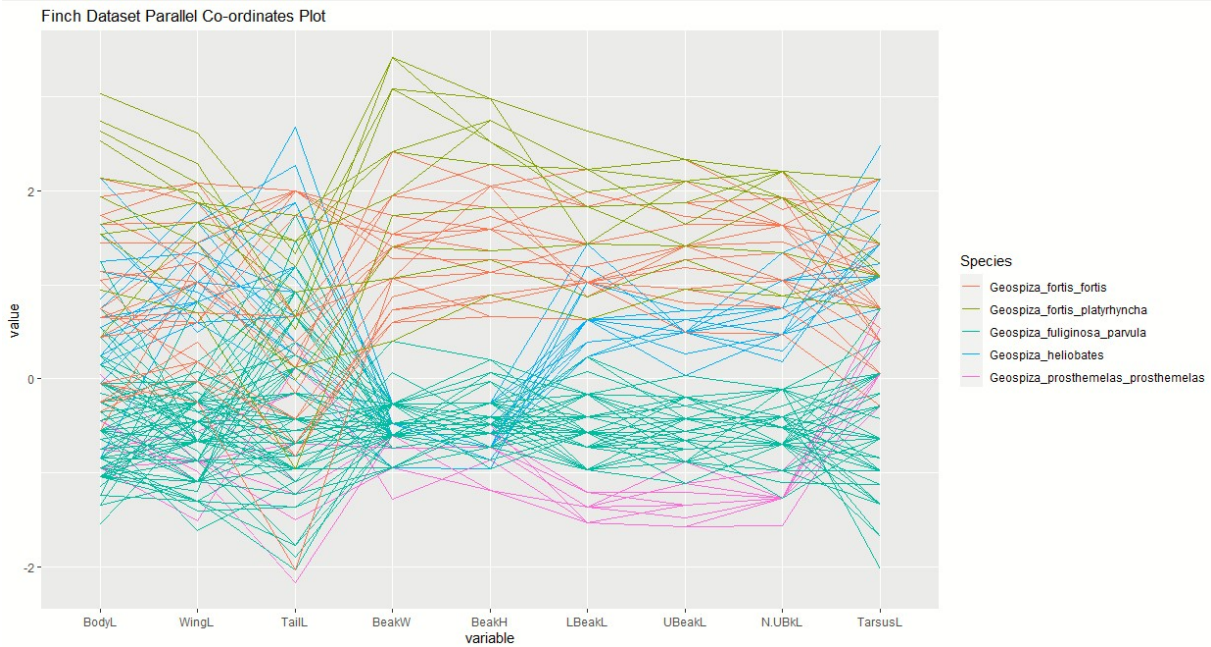
3.a) Create box and whisker plots for body length (BodyL) against Species, and for wing length (WingL) against Species. Display the 2 graphs in one image, placing the plot for body length on top, and the plot for wing length underneath, (i.e. image with 1 column, 2 rows).

```
# to check name of variables
?finch

# load the data
data(finch)

a <- ggplot(finch, aes(Species, BodyL)) + geom_boxplot() +
labs(title = "Species vs. Body Length", y = "Body Length", x = "Species")

b <- ggplot(finch, aes(Species, WingL)) + geom_boxplot() +
labs(title = "Species vs. Wing Length", y = "Wing Length", x = "Species")

grid.arrange(a, b)

# I could've also used grid.arrange(a, b, ncol = 1, nrow = 2) if I wanted
to specify the columns and rows I want
```



3.b) Create a parallel co-ordinates plot using all 9 predictor variables. The plot should be colour coded by Species.

```
ggparcoord(finch, columns = 2:10, groupColumn = "Species",
title ="Finch Dataset Parallel Co-ordinates Plot")
```



4.) Name a good predictor variable for separating out the responses.

A good predictor variable for separating out response A and response B would be variable 5. Based on the parallel co-ordinates plot, we can see that when we sort by variable 5, response A (red) and response (blue) are completely separate, and do not have any overlapping points. When we compare this to other variables such as variable 2, which has massive overlap of response A and response B, it is clear to see that variable 5 is better. Variable 4 wouldn't be bad either, but there is still slight overlap between response A and response B.