Question 4.1:

Consider the high dimensional DNA microarray data set called `Human_Tumor_Microarray` available on Ultra, which consists of 6830 rows representing individual genes and 64 columns representing samples from patients with human tumor. The main objective is to cluster the samples, each of which is a vector of length 6830 corresponding to expression values for the genes. Each sample has a label such as breast (for breast cancer), melanoma, and so on (see the file `Human_Tumor_Microarray_labels` on Ultra). These labels are not used in the clustering, but we will examine posthoc which labels fall into which clusters.

(a) First load the data set `Human_Tumor_Microarray` into R using the command `read.csv("Human_Tumor_Microarray.csv", header=TRUE)`, and then create a matrix of actual data for clustering by removing the first column which is the labels of samples. Transpose the data matrix to have the samples in rows and the genes in columns (i.e., $n \times p$) as in the lecture notes for clustering purposes. Also, centre the data so that all columns have zero mean.

(b) Apply the K-means clustering method to the `Human_Tumor_Microarray` data, using the R function `kmeans` with 3 clusters. Interpret the result of K-means clustering. Also, plot the clustering results using the R function `plot`.

(c) Choose an optimal number of clusters for K-means clustering using a scree plot for `tot.withinss` being the total within sum of squares of distances from the cluster means, obtained with the number of clusters from one to ten.

(d) Apply the hierarchical clustering method with the average linkage to the `Human_Tumor_Microarray` data, using the R function `hclust`. Interpret the result, also by getting the dendrogram of the hierarchical clustering outcome using the R function `plot`.

(e) Apply the sparse PCA to the `Human_Tumor_Microarray` data (with all columns being centred as above), using the R function `spca` available in the R package `sparsepca` which must be installed. For this, use only the 20 leading eigenvectors to avoid long computations. Print the loadings for these sparse PCs which all should be sparse. Compare the sparse PCA result with the standard PCA.

Question 4.2:

This practical question concerns a high dimensional finance data set, called `SP500data`, obtained from the US stock return data during the first year of the COVID-19 pandemic. The data set is available on Ultra. The US market data for different time periods can be obtained at https://www.finance.yahoo.com. The data set we use here holds the daily closing prices of stocks from the S&P 500 index during the first year of COVID-19 pandemic between 1st January 2020 and 30th June 2020, which results in $n = 125$ time points and $p = 496$ stocks (note $p > n$). During this time period, the S&P 500 index showed much volatility and dropped by about 20% in early March 2020 entering into a bear market. While the drop was the steepest one-day fall since 1987, S&P 500 index began to recover at the start of April 2020.

(a) First load the data set `SP500data` into R using the command `load(file="SP500data.rda")`, and then scale the data so that all columns have mean 0 and variance 1.

(b) Apply the K-means clustering method to the `SP500data`, using the R function `kmeans` with 3 clusters. Interpret the result of K-means clustering, also by plotting the clustering results using the R function `plot`.

(c) Obtain an optimal number of clusters for K-means clustering using a scree plot for `tot.withinss` being the total within sum of squares of distances from the cluster means, obtained with the number of clusters from one to ten.

(d) Apply the hierarchical clustering method with the average linkage to the `SP500data`, using the R function `hclust`. Interpret the result, also by getting the dendrogram of hierarchical clustering outcome using the R function `plot`.

(e) Now apply both K-means clustering and hierarchical clustering methods to the `SP500data` based on the dimensionality reduction by PCA, from Algorithm 3 in the lecture notes. Compare the results with Part (b) and Part (d).

(f) Use the Rand index to compare the performance of the K-means clustering in Part (b) and the K-means clustering based on PCA in Part (e). For this, use the R function `rand.index`. Do the same with the adjusted Rand index, using the R function `adj.rand.index`. Note that, as the truth is unknown with real data, the actual purpose here is to see if there is any difference by the Rand index when using K-means clustering based on PCA over usual K-means clustering.