# Whiskey Taste Indicators (PCA & k-means clustering)

## MINHCHAU

```
library(dplyr)
library(tidyr)
library(car)
library(psych) # for pairs.panels()
library(factoextra) # for fviz_cluster()
library(ggplot2)
```

- Dataset required: `whiskies.csv`

This will be an exploratory question using k-means clustering to examine a dataset of Whiskey Taste Indicators. The dataset can be obtained from https://outreach.mathstat.strath.ac.uk/outreach/nessie/nessie_whisky.html (https://outreach.mathstat.strath.ac.uk/outreach/nessie/nessie_whisky.html).

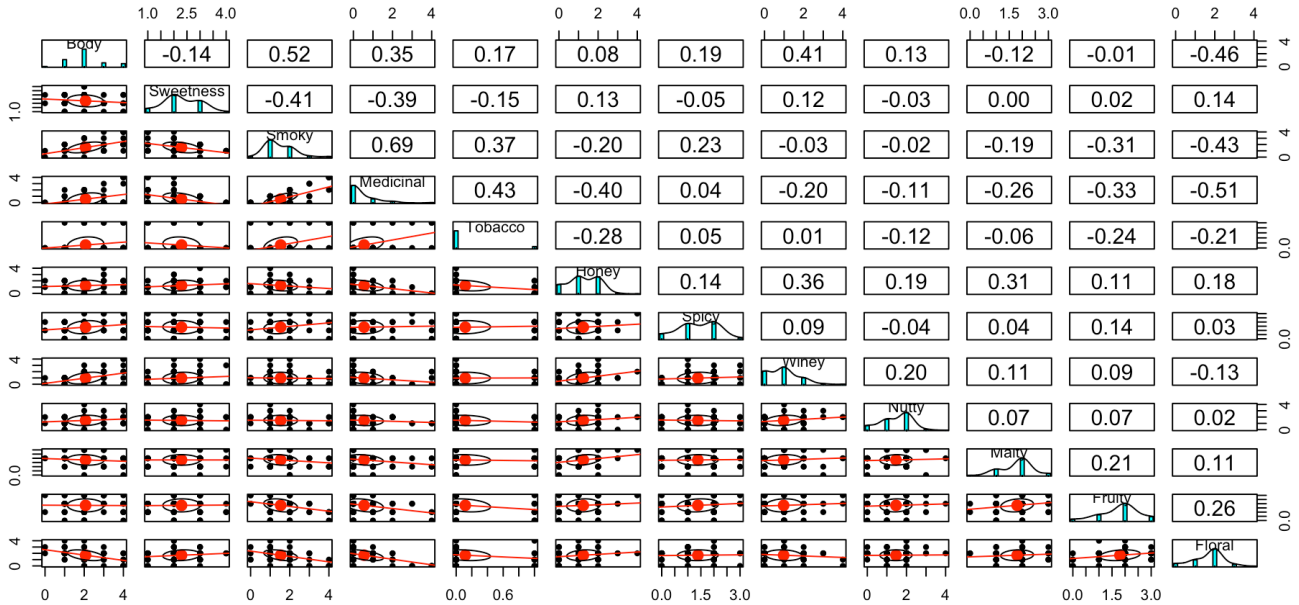
It consists of 86 (Single-Malt) Whiskies that are rated from 0-4 on 12 different taste categories: `Body` , `Sweetness` , `Smoky` , `Medicinal` , `Tobacco` , `Honey` , `Spicy` , `Winey` , `Nutty` , `Malty` , `Fruity` , `Floral` .

Here's what the dataset looks like:

```
wh = read.csv('whiskies.csv', header=T)

# Selecting out the independent variables "X".
whX <- wh %>% select(c("Body", "Sweetness", "Smoky", "Medicinal", "Tobacco", "Honey",
"Spicy", "Winey", "Nutty", "Malty", "Fruity", "Floral"))

# using pairs.panel() to look at the data
pairs.panels(whX, lm=T)
```

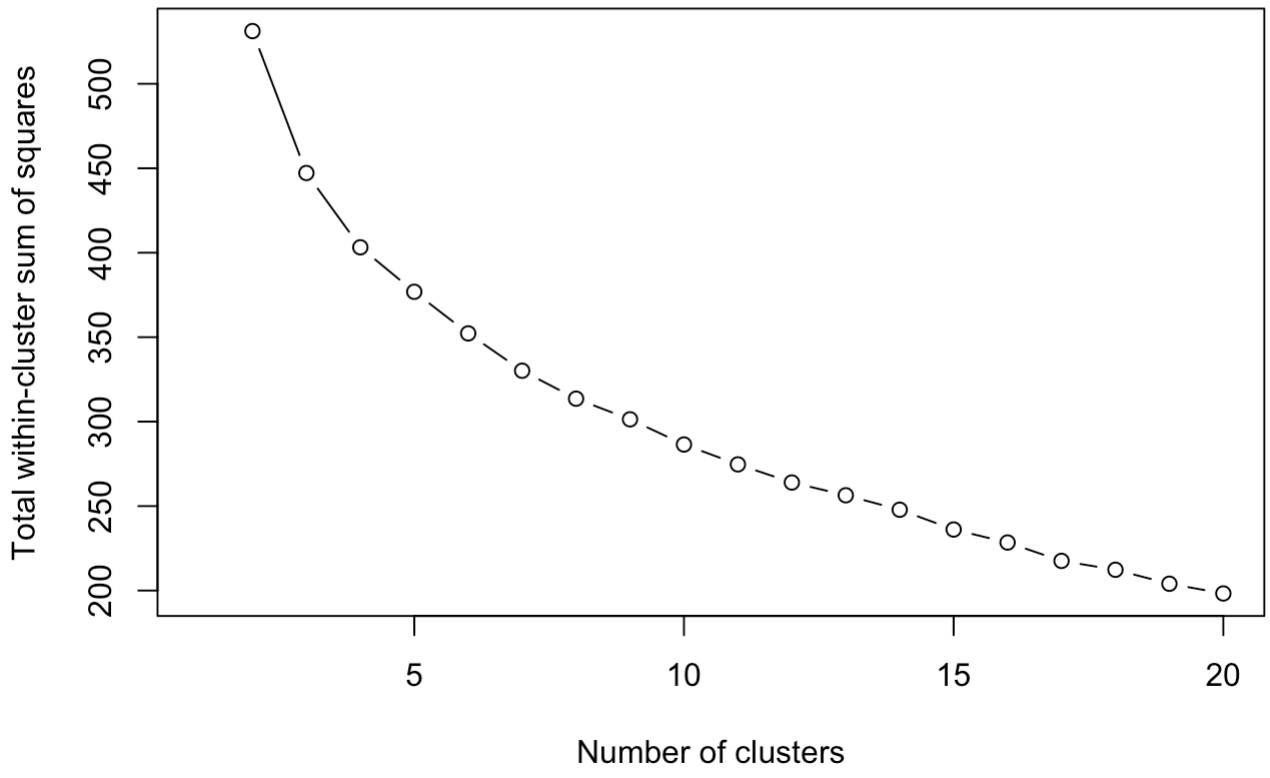


The main purpose of this question is to try clustering a real dataset, and try to interpret the clusters via looking at the cluster centers (in the dimensions of the independent variables), and generating “profiles” for each cluster.

(2a) The different whiskies based on their taste profile. First, let’s use the Elbow method to pick the best number of clusters.2Calculate the Within-Cluster Sum of Squares from k=2 to k=20 clusters using `whX` , and plot the Within-Cluster Suma of Squares against number of clusters.

*Note:* If the variables are on very different scales, we should standardize the variables (to have mean 0 and sd 1). Since the variables are already on a similar scale (0-4), it’s fine NOT scale the variables. Just run kmeans on `whX` .

```
# type your code here
set.seed(1)
wss <- rep(NA, 20)
for(k in c(2:20)) {
  wss[k] = kmeans(whX, k, nstart=10)$tot.withinss
}
plot(wss, type="b", xlab="Number of clusters", ylab="Total within-cluster sum of squa
res")
```




How to know the optimal number of cluster k? => Look for the elbow point Examine the plot to find a point where the Total within-cluster sum of square begins to decrease at a slower rate (the “elbow”). This point suggests a good balance between model complexity and performance

(2b)

From the plot, there is no clear “elbow”. The Within-Cluster Sum of Squares seem to keep decreasing, and there doesn’t seem to be a clear stopping point. This may happen in real datasets. We will use our own judgment to decide on the number of clusters.

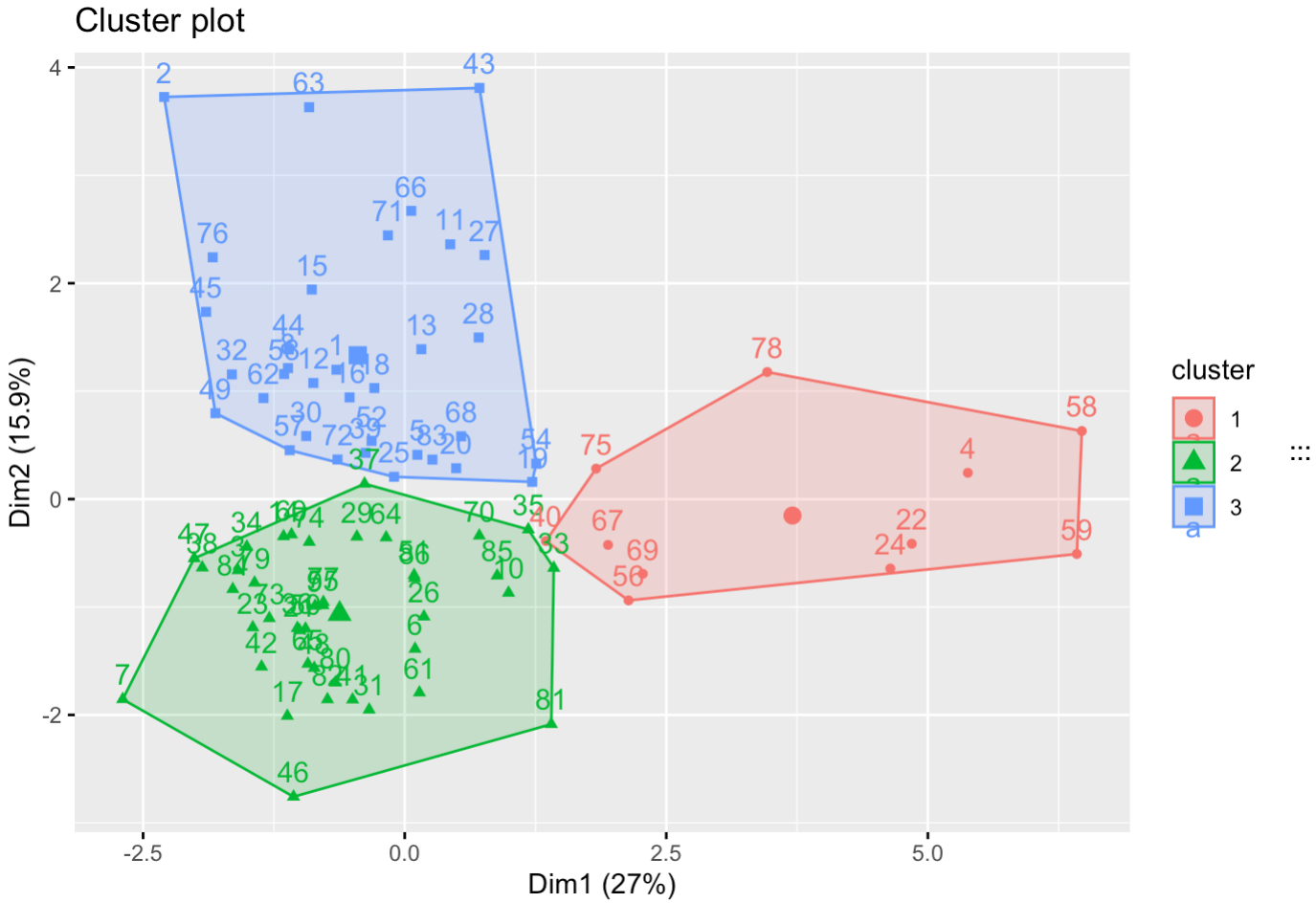
Ok, let's say our local business partner applies his expert intuition, and suggests that that `k=3` is a good starting point.

Set the random seed to ensure consistent results across all (students+TAs+instructors) by running set.seed(1) before fitting the k-means model, so that all of us can get the same results.

Then use the `fviz_cluster()` function from the `factoextra` package to plot the results of this clustering.

What do you notice from the graph? Discuss this with your TA and fellow students. (Recall that the graph dimensions will be along the top two principal components.)

```
set.seed(1)
km_obj <- kmeans(whX, 3)
fviz_cluster(km_obj, whX)
```




{style="color: red"}

There seems to be three clearly separated clusters: One (In the graph above, Cluster 1 in red) that's much higher on Principal Component 1 (PC1) than the rest (i.e., on the right of the graph). The other two (Clusters 2, and 3 in green n blue) are on the left side of the graph, but they are separated by Principal Component 2 (PC2), such that Cluster 3 is higher on PC2 and Cluster 2 is lower on Principal Component 2 (PC2). If there are more than two dimensions (variables) fviz_cluster will perform principal component analysis (PCA) and plot according to the first two principal components (as it explains majority of the variance) Dim1 - Principal Component 1 (27%) Dim2 - Principal Component 2 (15.9%) :::

(2c)

Ok, let's extract the cluster centers using $center (where `<kmeans_object_name>` is the name of the kmeans model you fitted earlier

Try to interpret the clusters based on their taste profiles. I'll provide you with one observation,

- For example, I notice that Cluster 1 has the highest Body compared to Clusters 2 and 3.

Try to generate at least four more observations about the clusters. and summarize your observations into a Taste Profile for each Cluster.

- For example, Cluster 1 high in Smoky, Tobacco-y, Spicyness.

If a client prefers whiskies that are rich in Smoky, Medicinal, and Tobacco flavors, which cluster would you recommend? (e.g., if this were a real client, you could go back and look at the Distilleries in `wh` and generate a list of those in the same cluster.)

```
km_obj$centers
```

```
##       Body Sweetness    Smoky Medicinal    Tobacco     Honey    Spicy     Winey
## 1 2.909091  1.545455 2.909091 2.7272727 0.45454545 0.4545455 1.454545 0.5454545
## 2 1.487805  2.463415 1.121951 0.2682927 0.07317073 0.9268293 1.146341 0.5121951
## 3 2.500000  2.323529 1.588235 0.1764706 0.05882353 1.8823529 1.647059 1.6764706
##      Nutty    Malty   Fruity    Floral
## 1 1.545455 1.454545 1.181818 0.5454545
## 2 1.146341 1.658537 1.878049 2.0000000
## 3 1.823529 2.088235 1.911765 1.7058824
```

Cluster 1 will be fuller bodied, less sweet, more smoky, more medicinal, more tobacco, less honey, less fruity and less floral than the rest. This is probably what Principal Component 1 (PC1) is picking up on (what we saw in Q1b). Cluster 2 and 3 are relatively more similar to each other, but compared to Cluster 3, Cluster 2 has: less body, less honey, less “winey” and nutty tastes.