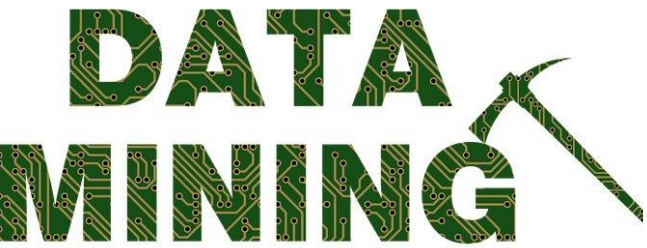
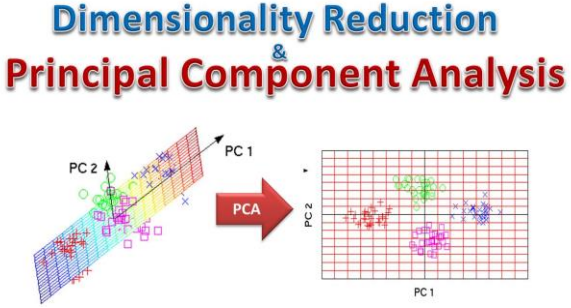
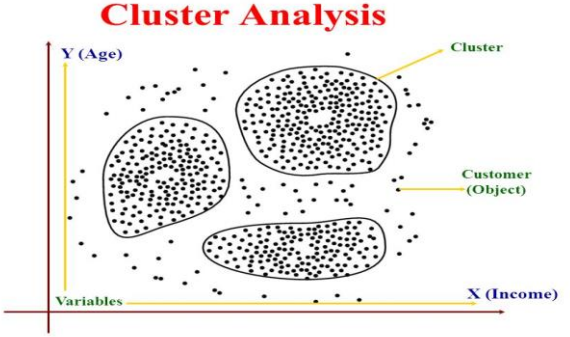






TBA2102 2020/2021 Semester 2
Tutorial 9: Data Mining

STRUCTURE OF TUTORIALS

Duration:

45 mins

Content:

- Data mining concepts
- Tutorial 9 (Questions 1& 2)

Data Mining Concepts

DATA DIMENSIONALITY REDUCTION

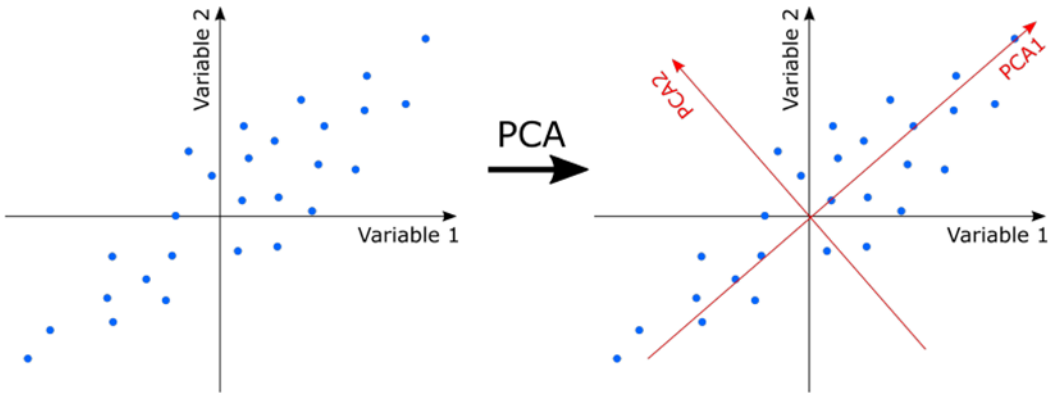
General idea

- Reduce the number of variables of a data set, while preserving as much information as possible.

Motivation

- Reduce overfitting
- Costly to use all predictors
- Multicollinearity

THE GENERAL IDEA OF PRINCIPAL COMPONENT ANALYSIS

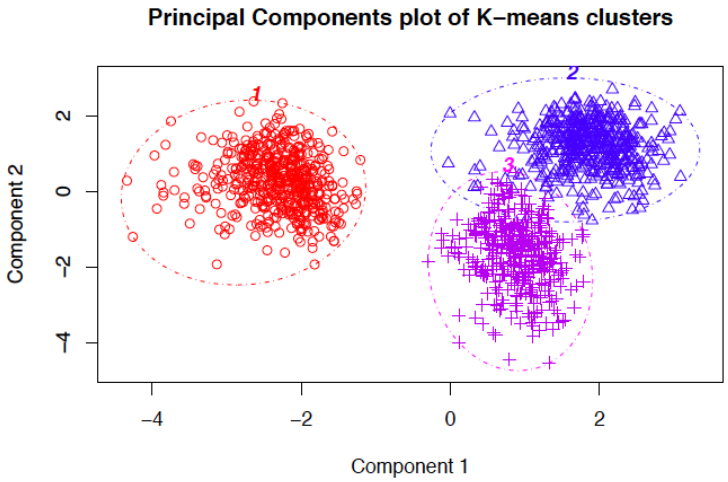


- *K* predictors = *K* principal components (remember to exclude the outcome variable).
- Each PC is a linear combination of ALL independent variables X's.
- The 1st PC accounts for the largest possible variance in the data set.
- PCs are orthogonal
- The PCs may or may not have any clear interpretation.
- Standard PCA cannot handle categorical variables.
- Standardise the variables

CLUSTERING

When we cluster observations of a dataset:

- We seek to partition them into distinct groups
- So that the observations within each group are quite similar to one another,
- While observations in different groups are quite different from each other.



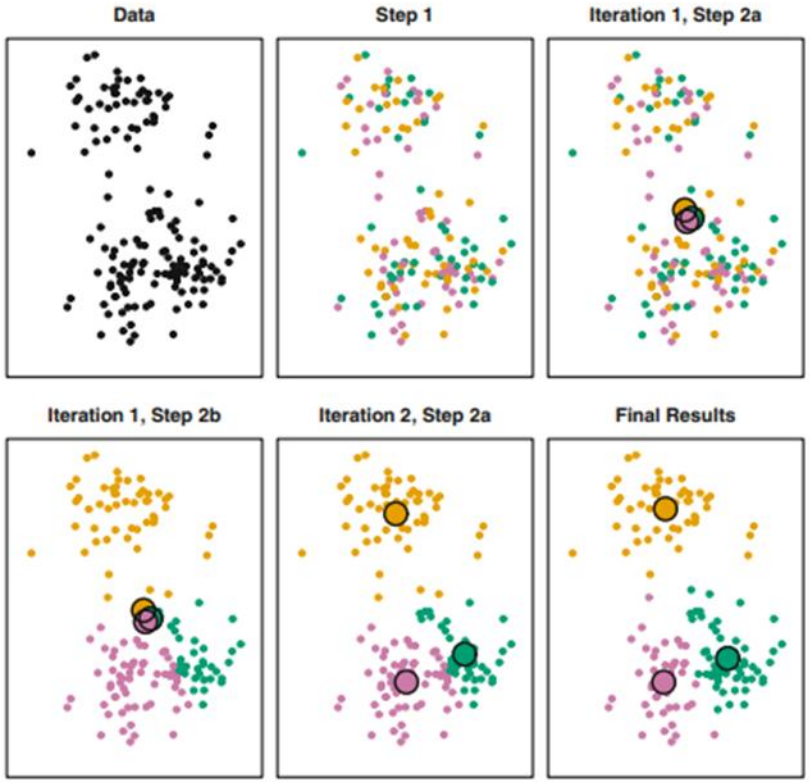
Both clustering & PCA seek to simplify the data via a small number of summaries, but their mechanisms are different.

- **PCA looks to find a low-dimensional representation of the observations that explain a good fraction of the variance.**

- **Clustering looks to find homogenous subgroups among the observations.**

K-MEANS CLUSTERING

- A simple & elegant approach for partitioning a data set into K distinct clusters.
- k-means partitions n observations into k clusters in which each observation is assigned to the nearest centroid (mean) & within cluster distance is minimized.
- Example of unsupervised learning.



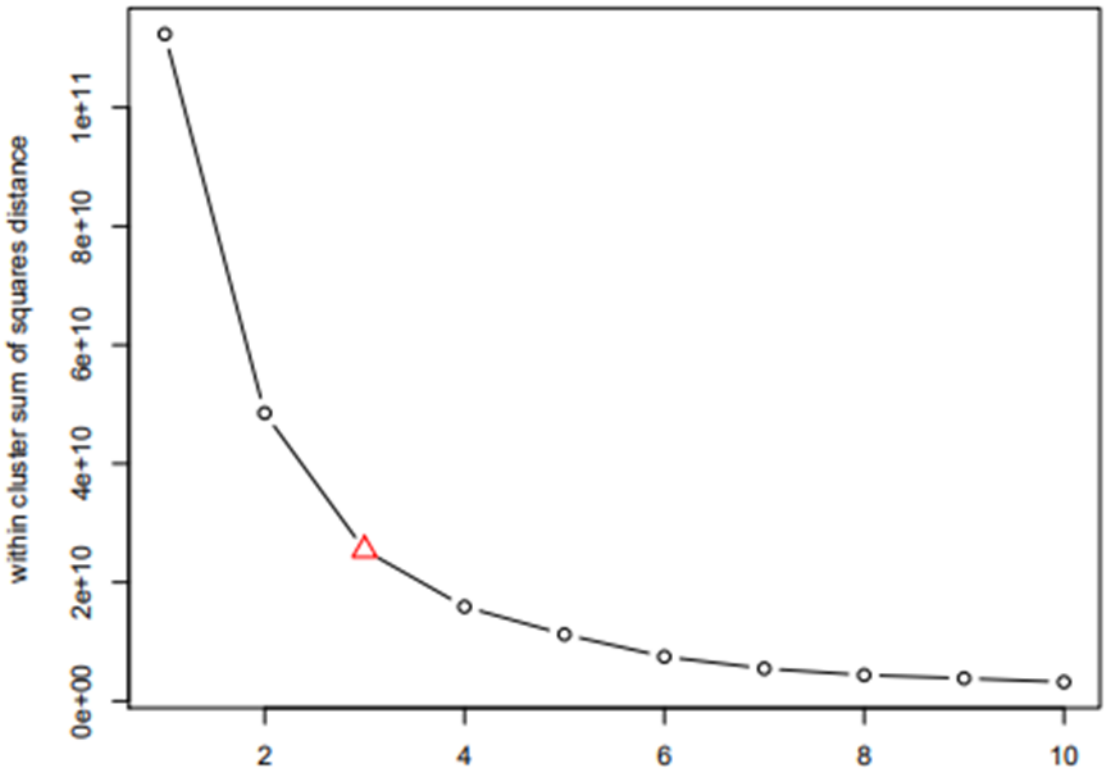
Initialization Step: Place the centroids of k clusters on k randomly chosen datapoints. (here k=3).

Assignment Step: Distance from each datapoint to all centroids are computed such that datapoints are “assigned” to the cluster with the closest centroid.

Update Step: Update the centroid position to be the mean of all points assigned to that cluster.

Iteration: Until convergence.

HOW DO WE DETERME THE VALUE OF K?



What is the value of k? what do we do when k is not clear?

- $k = 3$ or $k = 4$
- At times: theory, experience and/or intuition

Tutorial 9

DATASET REQUIRED

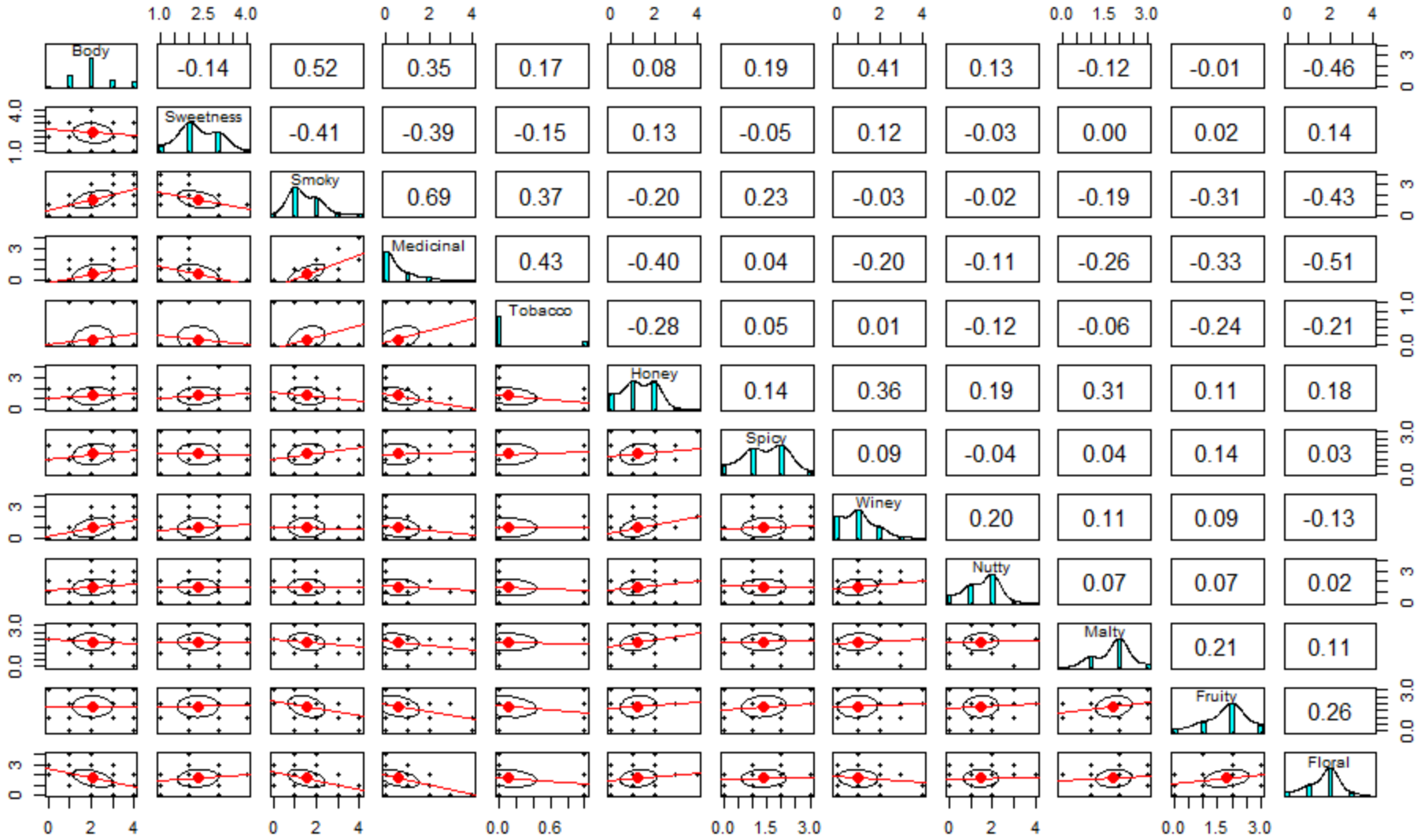
Tutorial8_whiskies.csv

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.

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`.

CORRELATIONS

```
pairs.panels(d1x, lm=T)
```



QUESTION 1B

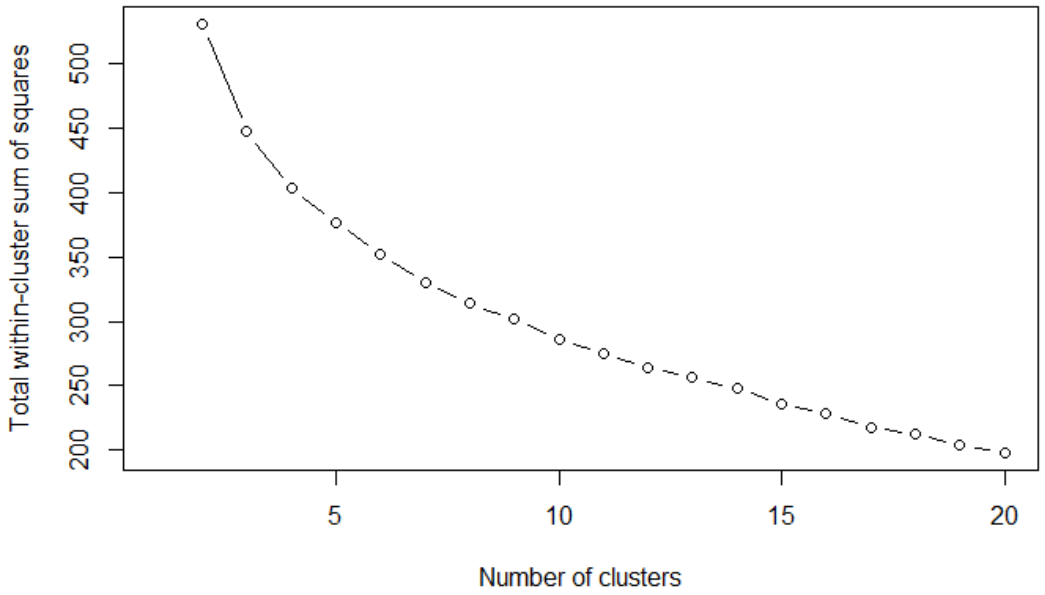
Next, use Kmeans clustering to group the different whiskies based on their taste profile. Recall that we can use the Elbow method to pick the number of clusters to use. Using the code in the lecture, calculate the Within-Cluster Sum of Squares from k=2 to k=20 clusters using d1X, and plot the Within-Cluster Sum of Squares against number of clusters.

Recall, if the variables are on very different scales, we should standardize the variables (to have mean 0 and sd 1). But in this case, all the variables are on the same scale (0-4) so it is fine not to scale the variables.
Let's try clustering the different whiskies based on their taste profile. First, let's use the Elbow method to pick the best number of clusters.

QUESTION 1B

```
set.seed(1)
wss <- rep(NA, 20)
for(k in c(2:20)) {
  wss[k] = kmeans(d1X, k, nstart=10)$tot.withinss
}
plot(wss, type="b", xlab="Number of clusters", ylab="Total within-cluster sum of
squares")
```

- Why do we set seed?
- What is nstart?

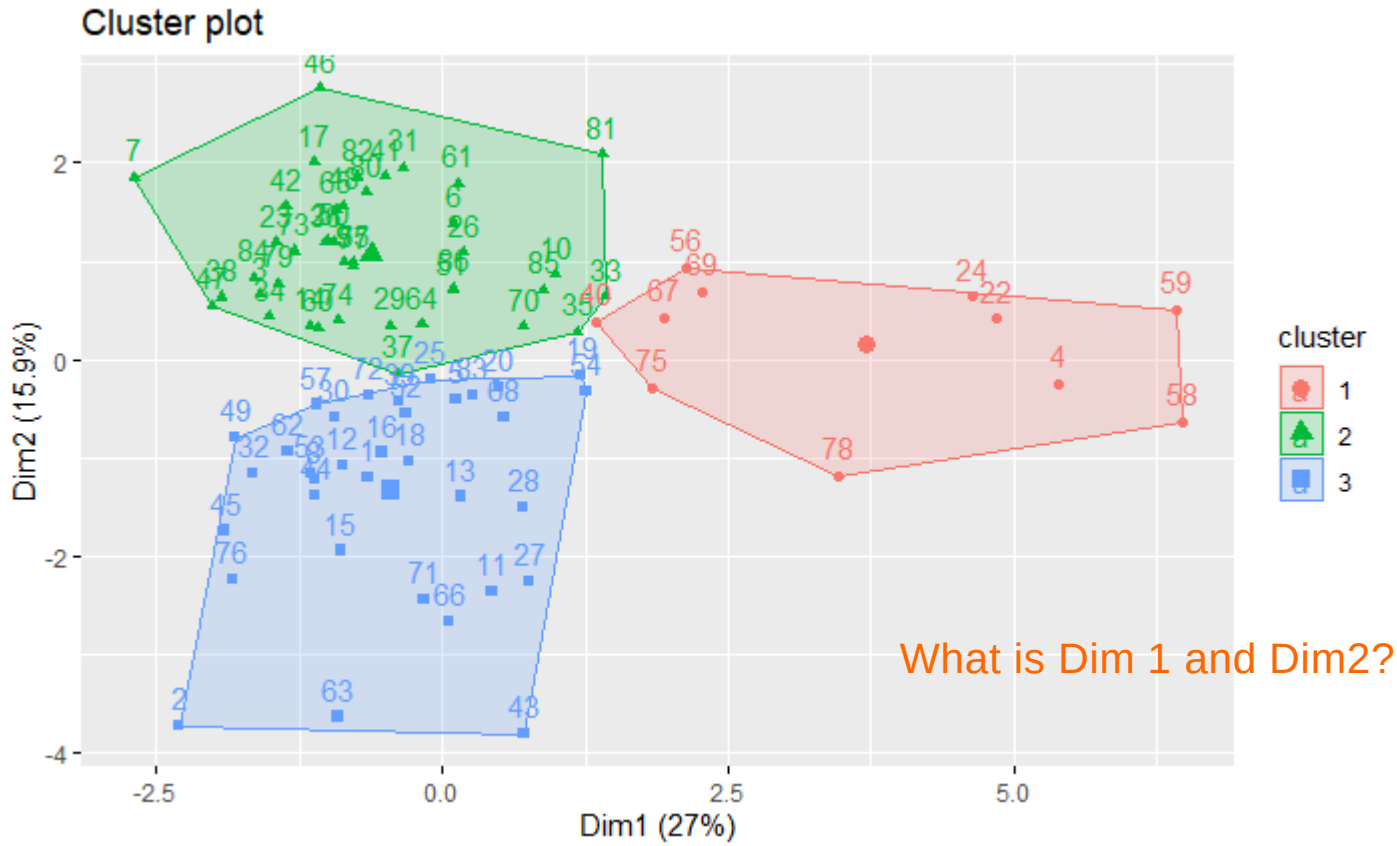


- Is there a clear elbow?
- What can we do?

QUESTION 1C: RUNNING K-MEANS CLUSTER ANALYSIS

```
set.seed(1)
km_obj <- kmeans(d1x, 3)
fviz_cluster(km_obj, d1x)
```

Our local business partner applies his expert intuition, and tells us to try fitting kmeans with **3** clusters



What is Dim 1 and Dim2?

QUESTION 1D

Use `<kmeans_object_name> $center` (where `<kmeans_object_name>` is the name of the kmeans model you fit above) to extract the centers of the 3 clusters.
Try to interpret the clusters.

```
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 tobaccoy, less honey, less fruity and less floral than the rest. This is probably what PC1 is picking up on.

QUESTION 2

Dataset required: T9_breast-cancer.csv

In this question, we will be doing a simple Principal Component Analysis, building a simple logistic regression classifier, then assessing the output of that classifier.

The dataset for this question is available at: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29

THE DATASET

Here are the variables in the dataset:

SampleID	Sample code number: The ID number of the sample.
Thickness	Clump Thickness: 1 - 10
SizeUniformity	Uniformity of Cell Size: 1 - 10
ShapeUniformity	Uniformity of Cell Shape: 1 - 10
MarginalAdhesion	Marginal Adhesion: 1 - 10
EpithelialCellSize	Single Epithelial Cell Size: 1 - 10
BareNuclei: Bare Nuclei	1 - 10
BlandChromatin	Bland Chromatin: 1 - 10
NormalNucleoli	Normal Nucleoli: 1 - 10
Mitoses: Mitosis	1 - 10
Class	2 for benign, 4 for malignant

DATA PREPARATION

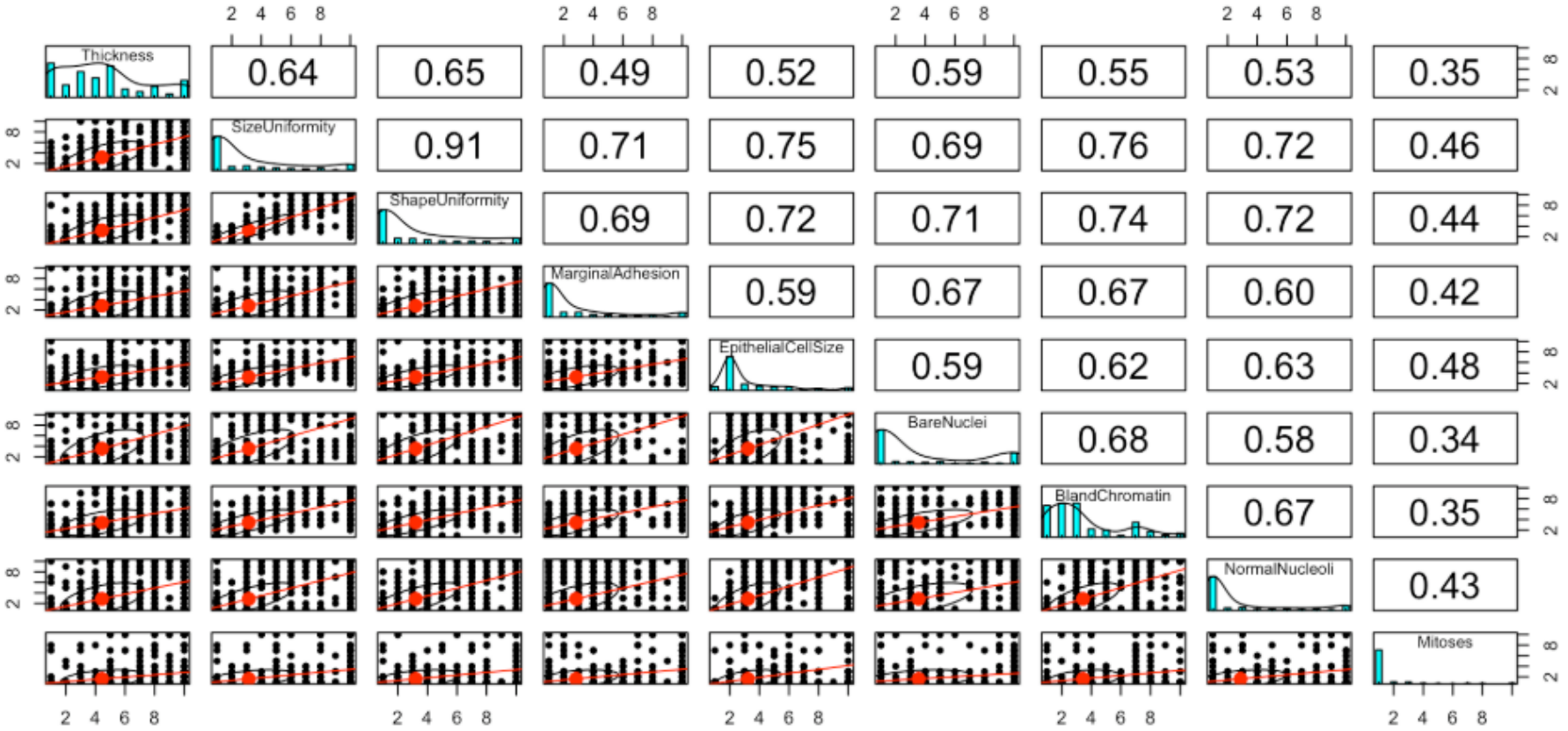
```
d2 = read.csv("T9_breast-cancer.csv", header=T)

# Create a new variable "Malignant" that is TRUE when class is 4 and FALSE when class is 2 (benigh), just so it's
 clear what Class means
d2$Malignant <- ifelse(d2$Class=="4", 1, 0)

# removing the 16 rows with incomplete data, just to avoid some programming issues later with PCA and missing dat
a.
d2 <- d2[complete.cases(d2),]

# Selecting out the independent variables "X".
d2X <- d2 %>% select(c("Thickness", "SizeUniformity", "ShapeUniformity", "MarginalAdhesion", "EpithelialCellSize"
, "BareNuclei", "BlandChromatin", "NormalNucleoli", "Mitoses"))
```

- Dependent variable: class
- SampleID is not a useful independent variable.
- So everything else, from Thickness to Mitoses, would be possible Ivs.

QUESTION 2A

- Start by using the pairs.panels() function from psych package to see what the

```
psych::pairs.panels(d2X, lm=TRUE)
```



QUESTION 2B

- Summarize the data using Principal Component Analysis.

```
d2pca <- prcomp(d2X, center = TRUE, scale = TRUE)
summary(d2pca)
```

- Discuss the arguments center and scale.

- What is the cumulative proportion of variance explained by the first three PCs?

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.4289 0.88088 0.73434 0.67796 0.61667 0.54943 0.54259
## Proportion of Variance 0.6555 0.08622 0.05992 0.05107 0.04225 0.03354 0.03271
## Cumulative Proportion  0.6555 0.74172 0.80163 0.85270 0.89496 0.92850 0.96121
##                           PC8     PC9
## Standard deviation     0.51062 0.29729
## Proportion of Variance 0.02897 0.00982
## Cumulative Proportion  0.99018 1.00000
```

QUESTION 2C

Check the loadings on the first 3 PCs. What do you notice?

- How can we extract the loading?

```
d2pca$rotation[,1:3]
```

```
##                         PC1         PC2          PC3
## Thickness          -0.3020626 -0.14080053  0.866372452
## SizeUniformity     -0.3807930 -0.04664031 -0.019937801
## ShapeUniformity    -0.3775825 -0.08242247  0.033510871
## MarginalAdhesion   -0.3327236 -0.05209438 -0.412647341
## EpithelialCellSize -0.3362340  0.16440439 -0.087742529
## BareNuclei         -0.3350675 -0.26126062  0.000691478
## BlandChromatin     -0.3457474 -0.22807676 -0.213071845
## NormalNucleoli     -0.3355914  0.03396582 -0.134248356
## Mitoses            -0.2302064  0.90555729  0.080492170
```

- Could you make a guess of the sign of the coefficient if you are to use PC1 to predict Malignancy?

QUESTION 2D

- Extract the first three PCs back into d2.
- Construct and run a logistic regression, predicting Malignant from the first three principal components. Which coefficients are significant?
- Using a model with all three PCs, use predict(<glm_object>, type='response') to ask the model to predict the probability of Malignant.
- Let’s make the assumption that if the probability is >=0.50, that the model says “Yes, it is Malignant”, and if it’s <0.50, the model says “No, it is not Malignant”. Store the binary predictions as a variable prediction in d2.
- How many “Yes” and “No” predictions did the model make?

QUESTION 2D

```
#extract PCs into d2
d2$pc1 <- d2pca$x[,"PC1"]
d2$pc2 <- d2pca$x[,"PC2"]
d2$pc3 <- d2pca$x[,"PC3"]

d2regpc <- glm(Malignant ~ pc1 + pc2 + pc3, d2, family='binomial')
summary(d2regpc)
```

```
##
## Call:
## glm(formula = Malignant ~ pc1 + pc2 + pc3, family = "binomial",
##     data = d2)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.08261  -0.12833  -0.06843   0.03255   2.78989
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.1474     0.2844  -4.034 5.47e-05 ***
## pc1          -2.3108     0.2276 -10.154  < 2e-16 ***
## pc2          -0.3795     0.3765  -1.008    0.313
## pc3           0.7350     0.3020   2.433    0.015 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 884.35  on 682  degrees of freedom
## Residual deviance: 113.14  on 679  degrees of freedom
## AIC: 121.14
##
## Number of Fisher Scoring iterations: 8
```

Which PCs are significant?

QUESTION 2D

```
d2$prediction = round(predict(d2regpc, type='response'))
table(d2$prediction)
```

```
##
##   0   1
## 443 240
```

The model makes 240 “Yes” predictions and 443 “No” predictions.

QUESTION 2E

Construct a confusion matrix. You can either use the confusionMatrix () function in the caret package, or use table(x1, x2) with both your model's "Yes/No" predictions and the actual Malignant values.

- How many True Positives are there?
- How many True Negatives are there?
- How many False Positives are there?
- How many False Negatives are there?
- What is the model's overall classification accuracy, recall, precision, specificity and F1 scores?

What would you say about the performance of this model?

```
table(d2$Malignant, d2$prediction)
```

```
##
##       0   1
##   0 433  11
##   1  10 229
```

```
cm = confusionMatrix(as.factor(d2$prediction), as.factor(d2$Malignant), positive = "1")
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 433  10
##          1  11 229
##
##                Accuracy : 0.9693
##                  95% CI : (0.9534, 0.9809)
##     No Information Rate : 0.6501
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9325
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9582
##             Specificity : 0.9752
##          Pos Pred Value : 0.9542
##          Neg Pred Value : 0.9774
##              Prevalence : 0.3499
##          Detection Rate : 0.3353
##    Detection Prevalence : 0.3514
##       Balanced Accuracy : 0.9667
##
##        'Positive' Class : 1
##
```

- True Positive = 229
- True Negative = 433
- False Positive = 11
- False Negative = 10

- Classification Accuracy = 229+433 / = 96.9%
- Precision = 229/(229+11) = 95.4%
- Recall = 229/(229+10) = 95.8%
- F1 = 95.6% (allow some rounding error for F1)

THANK YOU.
SEE YOU NEXT WEEK.