—----------Question 4.1—----------

Describe a situation or problem from your job, everyday life, current events, etc., for which a clustering model would be appropriate. List some (up to 5) predictors that you might use.

Answer:
My sister-in-law has a small dog sitting business where she takes care of variety of dogs. Each time she has a client, she measures and weigh their dog(s). By organizing her clients into size categories (Small, Medium, Large, and X-Large), she can track how many dogs of each size she takes care of on a weekly, monthly, quarterly, or yearly basis. A clustering model will be a great tool to groups dogs based on their measurements.

1. Their heigh:
    a. Small size dog would be around 13”-17” tall.
    b. Medium size dog would be around 18”-25” tall.
    c. Large size dog would be around 26”-31” tall.
    d. X-Large dog would be around more than 32” tall.
2. Their weigh:
    a. Small size dog would be around 22lbs.
    b. cMedium size dog would be around 23lbs-57lbs.
    c. Large size dog would be around 58lbs- 99lbs.
    d. X-Large size dog would be around >99lbs

For each dog she takes care of, they will have their own measurement. If any of the measurements fall between 2 categories (for example, 15” tall but 25lbs), it will be assigned to the larger group- medium in this case.

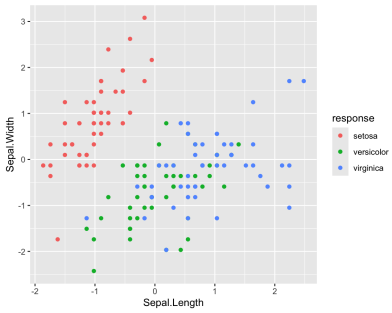
So, the cluster chart will have an x-axis for the weight, and the y-axis for the heigh, then each cluster will represent each dog she took care of.

—----------QUESTION 4.2—----------

The *iris* data set `iris.txt` contains 150 data points, each with four predictor variables and one categorical response. The predictors are the width and length of the sepal and petal of flowers and the response is the type of flower. The data is available from the R library datasets and can be accessed with iris once the library is loaded. It is also available at the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Iris). *The response values are only given to see how well a specific method performed and should not be used to build the model.*
Use the R function `kmeans` to cluster the points as well as possible. Report the best combination of predictors, your suggested value of k, and how well your best clustering predicts flower type.

Answer:

```
#install package tidyverse to use ggplot2 to visualize the data:
install.packages("tidyverse")
library(tidyverse)
library(ggplot2)

#set the seed for reproducibility:
set.seed(1)
View(iris)

#Split the response column from the data:
iris_data <- iris[1:4]
View(iris_data)
response <- iris$Species

# Explain why all predictors were used- based on office hour on Jan 20th:

# I used all four features: Sepal.Length, Sepal.Width, Petal.Length, and Petal.Width because they each
# provide meaningful information for distinguishing between the species. Petal.Length and Petal.Width
# are the most distinguishing, while Sepal.Length and Sepal.Width also help in separating Setosa.
# Using all four ensures we capture the full variance in the data.

#Scale the data after being split
scale(iris_data)
iris_data_scaled <- scale(iris_data)

#take a look at the data, this code is from office hour on Jan 20:
ggplot(iris_data_scaled, aes(Sepal.Length, Sepal.Width, color=response)) + geom_point()
```



#I'm about to use the kmeans function. However, I need to run centers in a loop to see what's the best centers number:
#Use Elbow method to pick the optimal k by using tot.withinss from the kmeans:
#The idea is from office hour on Jan 20th, and those codes are referenced from ChatGPT:

wss <- vector() #create a vector called wss to store the total-within cluster sum of squares for each k.

#Run kmeans for k from 1 to 10:
```
for (k in 1:10) {
  optimal_kmeans <- kmeans(iris_data_scaled, centers=k, nstart=25) #nstart is from office hour on Jan 20th
  wss[k] <- optimal_kmeans$tot.withinss
}
```

wss #this means the more clusters, the tighter the group
```
> wss
 [1] 596.00000 220.87929 138.88836 113.33162  90.20190  79.46523  70.53640
 [8]  62.02699  54.74103  46.99885
```

optimal_kmeans

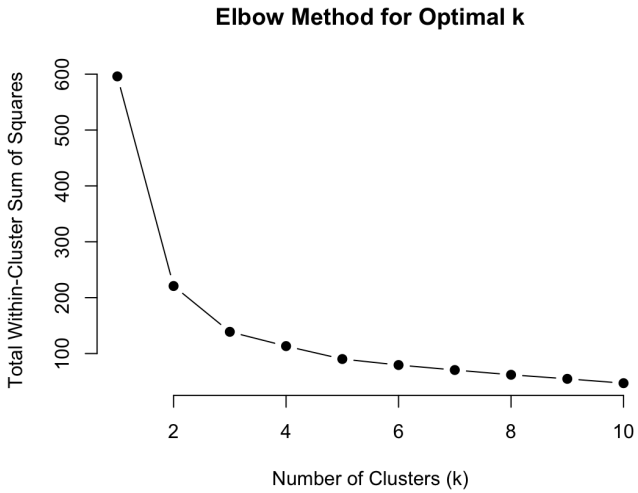
optimal_kmeans	list [9] (S3: kmeans)	List of length 9
cluster	integer [150]	3 1 1 1 3 5 ...
centers	double [10 x 4]	−1.3949 −0.1597 −0.9667 0.6622 −0...
totss	double [1]	596
withinss	double [10]	5.16 4.03 3.40 8.67 3.95 3.09 ...
tot.withinss	double [1]	46.99885
betweenss	double [1]	549.0011
size	integer [10]	17 18 21 24 12 9 ...
iter	integer [1]	3
ifault	integer [1]	0

#Plot to see the elbow:

#This following code is referenced from https://www.r-bloggers.com/2017/02/finding-optimal-number-of-clusters/

```
plot(1:10, wss, type="b", pch=19, frame=FALSE,
    xlab="Number of Clusters (k)", ylab="Total Within-Cluster Sum of Squares",
    main="Elbow Method for Optimal k")
```

#Based on the elbow, we can see that k=3.



#Run the kmeans with k=3:

final_kmeans <- kmeans(iris_data_scaled, centers=3, nstart = 25)

```
> final_kmeans$centers
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1  -1.01119138  0.85041372   -1.3006301  -1.2507035
2  -0.05005221 -0.88042696    0.3465767   0.2805873
3   1.13217737  0.08812645    0.9928284   1.0141287
```

#Compare clustering result to the original dataset:
#This code is referenced from office hour on Jan 20tth.

table(final_kmeans$cluster, iris$Species)

```
> table(final_kmeans$cluster, iris$Species)

  setosa versicolor virginica
1     50         0         0
2      0        39        14
3      0        11        36
```

#the original iris dataset:
table(iris$Species)

```
> table(iris$Species)

    setosa versicolor  virginica
        50         50         50
```

#from what I learned on https://stats.stackexchange.com/questions/144616/comparing-k-means-results-to-original-data-how-to-interpret-the-resulting-plots#145337,

#K-means is meant to cluster, rather than compare clusters to known groups.

#So, I am going to copy this method from the link above, to compare the kmeans model to its original dataset.

library(MASS)
#The MASS package contains many useful functions for statistical analysis, including the Linear Discriminant Analysis (LDA) function, which is used in the next steps.

iris.lda <- lda(Species~., iris)
#Fits a Linear Discriminant Analysis (LDA) model using the Species as the target and all other columns (Sepal.Length, Sepal.Width, Petal.Length, Petal.Width) as predictors. The result is saved in iris.lda

#LDA is used to classify iris flowers based on their features.

iris.predict <- predict(iris.lda, iris[1:4]) #predict the species (Species) based on the first four columns (features) of the iris data

#Creates a confusion matrix to compare predicted species (iris.predict$class) with actual species (iris$Species):

```
table(
  iris.predict$class,      #The predicted species for each flower based on the LDA model.
  iris$Species)            #The actual species of each flower (the true values)

sum(              #Adds up the correct predictions to calculate the accuracy of the model
  diag(           #Extracts the correct predictions (diagonal values)
    prop.table(   #Converts the confusion matrix to proportions.
      table(iris.predict$class, iris$Species))))
```

```
> sum(diag(prop.table(table(iris.predict$class, iris$Species))))
[1] 0.98
```

So this kmeans model provide the accuracy of 98%.