STAT 46700/CS 59000 Topics in Data Science Spring 2025

Final-Part II

Name: Vaishak Balachandra - 0037831852

Due: May 7, 2025 (5:00 PM CST)

Please provide the complete solutions of all problems.

Please work individually and provide detail solution with R code embedded with your answers.
I affirm that I didn't give or receive any unauthorized help on this exam and that all work is my own [....................***VAISHAK BALACHANDRA***........................]

Q.N. 1) A slightly modified data set consisting of 753 married women in the United States and information about whether they participate in the labor market (either they have a job or are actively looking for one) and background information on them and their families are attached. The variables included in the data are

inlf : In Labor Force which is a dummy variable equal to 1 if the woman is in the labor force and 0 if not.
kidslt6 : Number of children under age of 6.
age : age of the woman
educ : Number of years of education of the woman
hushrs : Number of hours per year that the husband works.
huseduc : Number of years of education of the husband.
amotheduc : Number of years of education of the woman's mother.
fatheduc : Number of years of education of the woman's father.

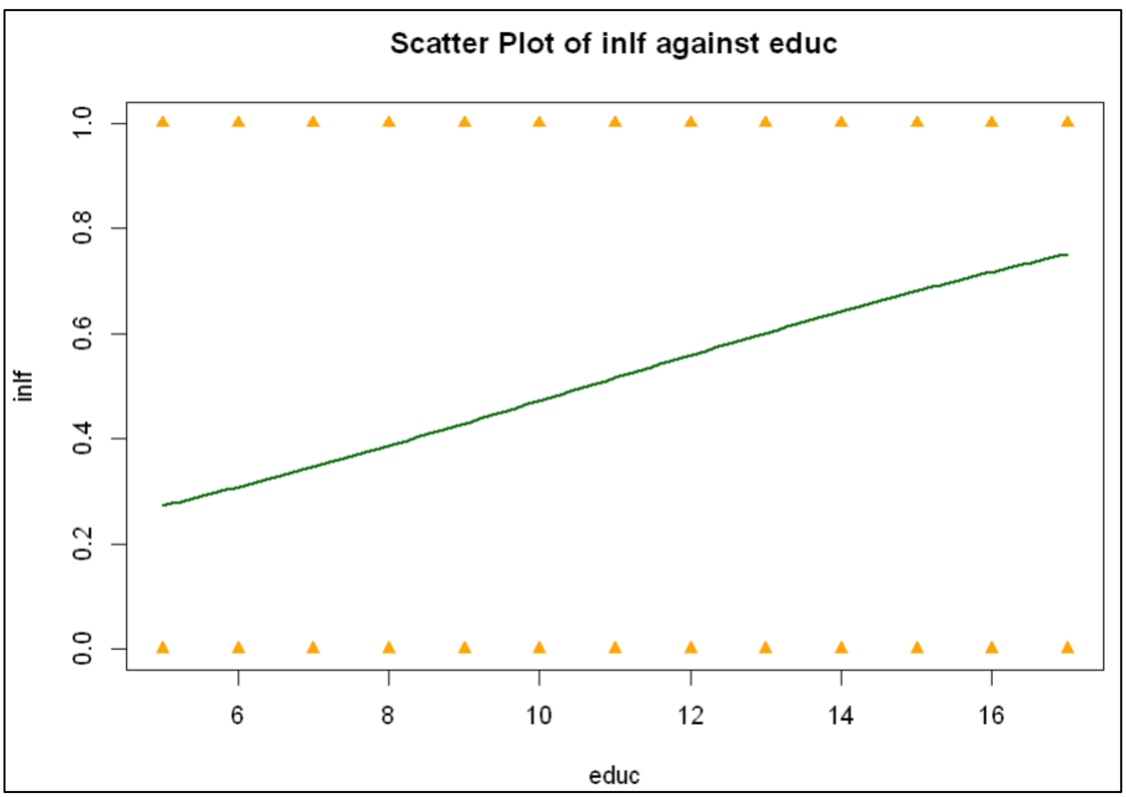
(a) Fit a simple logistic regression model to describe the relationship between the *inlf* and the years of education. Please be sure to state the model equation

```
> # Q1
>
> install.packages("readxl")
> library(readxl)
> Q1 <- read_excel("Labor_data.xlsx")
> head(Q1)
# A tibble: 6 × 8
   inlf kidslt6   age  educ hushrs huseduc motheduc fatheduc
  <dbl>   <dbl> <dbl> <dbl>  <dbl>   <dbl>    <dbl>    <dbl>
1     1       1    32    12   2708      12       12        7
2     1       0    30    12   2310       9        7        7
3     1       1    35    12   3072      12       12        7
4     1       0    34    12   1920      10        7        7
5     1       1    31    14   2000      12       12       14
6     1       0    54    12   1040      11       14        7
> dim(Q1)
[1] 753   8
> names(Q1)
[1] "inlf"     "kidslt6"  "age"      "educ"     "hushrs"   "huseduc"  "motheduc" "fatheduc"
> attach(Q1)
>
```

```
> # (a)   Fit a simple logistic regression model to describe the relationship between the inlf and t
he years of education. Please be sure to state the model equation
> model = glm(inlf~educ, data = Q1, family = binomial)
> summary(model)

Call:
glm(formula = inlf ~ educ, family = binomial, data = Q1)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.85199    0.42828  -4.324 1.53e-05 ***
educ         0.17398    0.03465   5.022 5.12e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1029.7  on 752  degrees of freedom
Residual deviance: 1002.7  on 751  degrees of freedom
AIC: 1006.7

Number of Fisher Scoring iterations: 4

> cat("Fitted Simple Logistic Regression:
+ inlf = [1 + exp(1.85199 - 0.17398*educ)]^-1")
Fitted Simple Logistic Regression:
inlf = [1 + exp(1.85199 - 0.17398*educ)]^-1
```

(b) Display the fitted logistic regression model using the probability curve.

```
> # (b)   Display the fitted logistic regression model using the probability curve.
> plot(educ, inlf, main = "Scatter Plot of inlf against educ", pch = 17, col = "orange")
> curve(predict(model, newdata = data.frame(educ = x), type = "response"), add = TRUE, col = "darkg
reen", lwd = 2)
```



(c) Split the dataset with 70% training data and 30% test data. Please be sure to use set.seed and use your PUID number for reproducibility of the results.

```
> # (c)  Split the data-set with 70% training data and 30% test data. Please be sure to use set.see
d  and use your PUID number for reproducibility of the results.
> set.seed(037831852)
> train_indices <- sample(1:nrow(Q1), size = 0.7*nrow(Q1))
> train_data <- Q1[train_indices, ]
> test_data <- Q1[-train_indices, ]
> dim(train_data)
[1] 527   8
> dim(test_data)
[1] 226   8
```

(d) Fit a multiple logistic regression model using *inlf* as the outcome variable and all other variables as explanatory variables. Identify all significant variables.

```
> # (d)  Fit a multiple logistic regression model using inlf as the outcome variable and all other
variables as explanatory variables. Identify all significant variables.
> model1 = glm(inlf~., data = train_data, family = binomial)
> summary(model1)

Call:
glm(formula = inlf ~ ., family = binomial, data = train_data)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.6669926  0.9546566   1.746   0.0808 .
kidslt6     -1.4718459  0.2311941  -6.366 1.94e-10 ***
age         -0.0660215  0.0139882  -4.720 2.36e-06 ***
educ         0.3285862  0.0631352   5.204 1.95e-07 ***
hushrs      -0.0002851  0.0001618  -1.762   0.0781 .
huseduc     -0.1024689  0.0412102  -2.486   0.0129 *
motheduc    -0.0225550  0.0372873  -0.605   0.5452
fatheduc    -0.0105843  0.0344658  -0.307   0.7588
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 719.87  on 526  degrees of freedom
Residual deviance: 637.00  on 519  degrees of freedom
AIC: 653

Number of Fisher Scoring iterations: 4

> cat("Significant Variables (alpha = 0.05):
+ 1. kidslt6 -- highly significant,
+ 2. age -- highly significant,
+ 3. educ -- highly significant,
+ 4. huseduc -- marginally significant")
Significant Variables (alpha = 0.05):
1. kidslt6 -- highly significant,
2. age -- highly significant,
3. educ -- highly significant,
4. huseduc -- marginally significant
```

(e) Create the confusion matrix to assess the classification accuracy (assume that probabilities exceeding 0.5 as predicted to be in the labor force based on your model)

```
> # (e)   Create the confusion matrix to assess the classification accuracy (assume that probabiliti
es exceeding  0.5 as predicted to be in the labor force based on your model)
> pred <- predict(model1, newdata = test_data, type = "response")
> predictions <- ifelse(pred > 0.5, 1, 0)
> conf_matrix <- table(Predicted = predictions, Actual = test_data$inlf)
> cat("Confusion Matrix: \n")
Confusion Matrix:
> print(conf_matrix)
         Actual
Predicted  0  1
        0 50 29
        1 49 98
> accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
> cat("Classification Accuracy: ", accuracy*100 , "%")
Classification Accuracy:  65.48673 %
> # or
> install.packages("caret")
> library(caret)
> confusionMatrix(data = factor(predictions), reference = factor(test_data$inlf), positive = "1")
> cat("Classification Accuracy: 65.49%")
Classification Accuracy: 65.49%
```

Q.N. 2) The dataset below has 4 features (Color, Size, Act, Age) that each balloon can have two values and a binary label (Inflated?). Use this data to answer the following questions.

Color	**Size**	**Act**	**Age**	**Inflated?**
Red	Large	Stretch	Adult	F
Red	Large	Stretch	Child	T
Red	Large	Dip	Child	F
Blue	Large	Dip	Adult	T
Blue	Large	Stretch	Child	F
Blue	Large	Dip	Child	F
Red	Small	Dip	Child	F
Blue	Small	Dip	Adult	T
Red	Small	Stretch	Child	F
Red	Small	Dip	Adult	T

a) Calculate the entropy of the inflated status.

```
> # Q2
>
> Color = c("Red", "Red", "Red", "Blue", "Blue", "Blue", "Red", "Blue", "Red", "Red")
> Size = c("Large", "Large", "Large","Large", "Large", "Large", "Small", "Small", "Small", "Small")
> Act = c("Stretch", "Stretch", "Dip", "Dip", "Stretch", "Dip", "Dip", "Dip", "Stretch", "Dip")
> Age = c("Adult", "Child", "Child", "Adult", "Child", "Child", "Child", "Adult", "Child", "Adult")
```

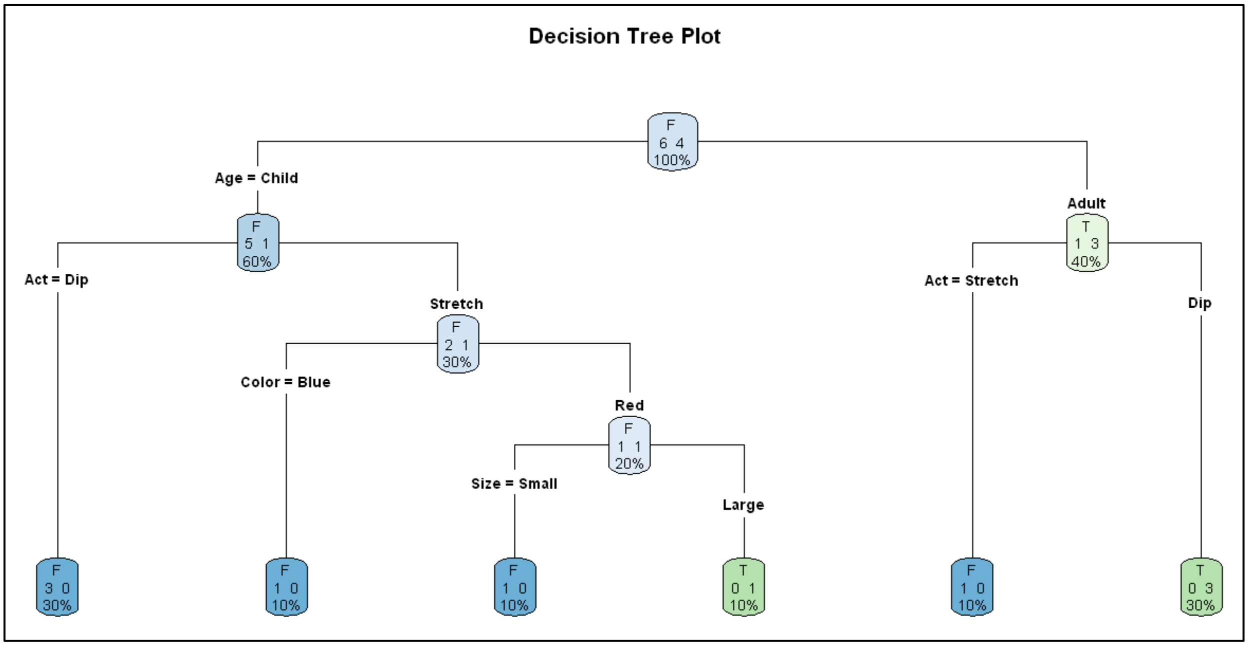
```
> Inflated = c("F", "T", "F", "T", "F", "F", "F", "T", "F", "T")
> Q2 = data.frame(Color, Size, Act, Age, Inflated)
> head(Q2)
  Color  Size     Act   Age Inflated
1   Red Large Stretch Adult        F
2   Red Large Stretch Child        T
3   Red Large     Dip Child        F
4  Blue Large     Dip Adult        T
5  Blue Large Stretch Child        F
6  Blue Large     Dip Child        F
> dim(Q2)
[1] 10  5
> names(Q2)
[1] "Color"    "Size"     "Act"      "Age"      "Inflated"
>
> # (a)  Calculate the entropy of the inflated status.
> # install.packages("DescTools")
> library(DescTools)
> Q2$Inflated <- factor(Q2$Inflated, levels = c("F", "T"))
> entropy <- Entropy(table(Q2$Inflated), base = 2)
> cat("Entropy of Inflated Variable:", entropy)
Entropy of Inflated Variable: 0.9709506
> cat("INFERENCE: Indicates a high level of uncertainty!!")
INFERENCE: Indicates a high level of uncertainty!!
```

b) Identify the root node of the above data by calculating the information gain.

```
> # (b)  Identify the root node of the above data by calculating the information gain.
> # install.packages("FSelector")
> library(FSelector)
> info_gain <- information.gain(Inflated~., data = Q2)
> print(info_gain)
      attr_importance
Color      0.01384429
Size       0.01384429
Act        0.03218930
Age        0.17774088
> cat("Seeing the importance of all the attribute
+ 'AGE' can be the root node!!")
Seeing the importance of all the attribute
'AGE' can be the root node!!
```

c) Construct a decision tree for the subject data using R.

```
> # (c)  Construct a decision tree for the subject data using R.
> # install.packages("rpart")
> # install.packages("rpart.plot")
> library(rpart)
> library(rpart.plot)
> tree_model <- rpart(Inflated ~ ., data = Q2, method = "class", parms = list(split = "information"
), control = rpart.control(cp = 0, minsplit = 2, minbucket = 1))
> rpart.plot(tree_model, type = 4, extra = 101, main = "Decision Tree Plot", fallen.leaves = TRUE,
cex = 0.55)
```



Q.N. 3) Datasets containing average scores on math, reading, and science together with standard errors for all OECD countries are provided in the `csranks` package in R. These are from the 2018 and 2022 editions of Program for International Student Assessment (PISA) study by the Organization for Economic Cooperation and Development (OECD). The average scores are over all 15-year-old students in the study.

You will work with `pisa2022` data if your PUID has last digit higher than 4 otherwise you will use `pisa2018` data.

a) Access your dataset and print the jurisdiction (country, from which data were collected).

```
> # Q3
>
> # (a)   Access your dataset and print the jurisdiction (country, from which data were collected).
> # install.packages("csranks")
> library(csranks)
> data("pisa2018")
> head(pisa2018)
  jurisdiction science_score science_se reading_score reading_se math_score  math_se
1    Australia      502.9646   1.795398      502.6317   1.634343   491.3600 1.939833
2      Austria      489.7804   2.777395      484.3926   2.697472   498.9423 2.970999
3      Belgium      498.7731   2.229240      492.8644   2.321973   508.0703 2.262662
4       Canada      517.9977   2.153651      520.0855   1.799716   512.0169 2.357476
5        Chile      443.5826   2.415280      452.2726   2.643766   417.4066 2.415888
6     Colombia      413.3230   3.052402      412.2951   3.251344   390.9323 2.989559
> dim(pisa2018)
[1] 38  7
> names(pisa2018)
[1] "jurisdiction"  "science_score" "science_se"    "reading_score" "reading_se"    "math_score"
[7] "math_se"
> # printing the jurisdiction
> print(pisa2018$jurisdiction)
 [1] "Australia"       "Austria"         "Belgium"       "Canada"        "Chile"         "Colombia"
 [7] "Costa Rica"      "Czech Republic"  "Denmark"       "Estonia"       "Finland"       "France"
[13] "Germany"         "Greece"          "Hungary"       "Iceland"       "Ireland"       "Israel"
[19] "Italy"           "Japan"           "Korea"         "Latvia"        "Lithuania"     "Luxembourg"
[25] "Mexico"          "Netherlands"     "New Zealand"   "Norway"        "Poland"        "Portugal"
[31] "Slovak Republic" "Slovenia"        "Spain"         "Sweden"        "Switzerland"   "Türkiye"
```

```
[37] "United Kingdom"   "United States"
```

b) Clean the dataset by removing the missing values.

```
> # (b)  Clean the dataset by removing the missing values.
> sum(is.na(pisa2018))
[1] 2
> cat("Has 2 NA values!!")
Has 2 NA values!!
> pisa_clean = na.omit(pisa2018)
> dim(pisa_clean)
[1] 37  7
```

c) Choose 25 countries at random using sample function. Please make sure you used your PUID to set the seed for reproducibility of your work.

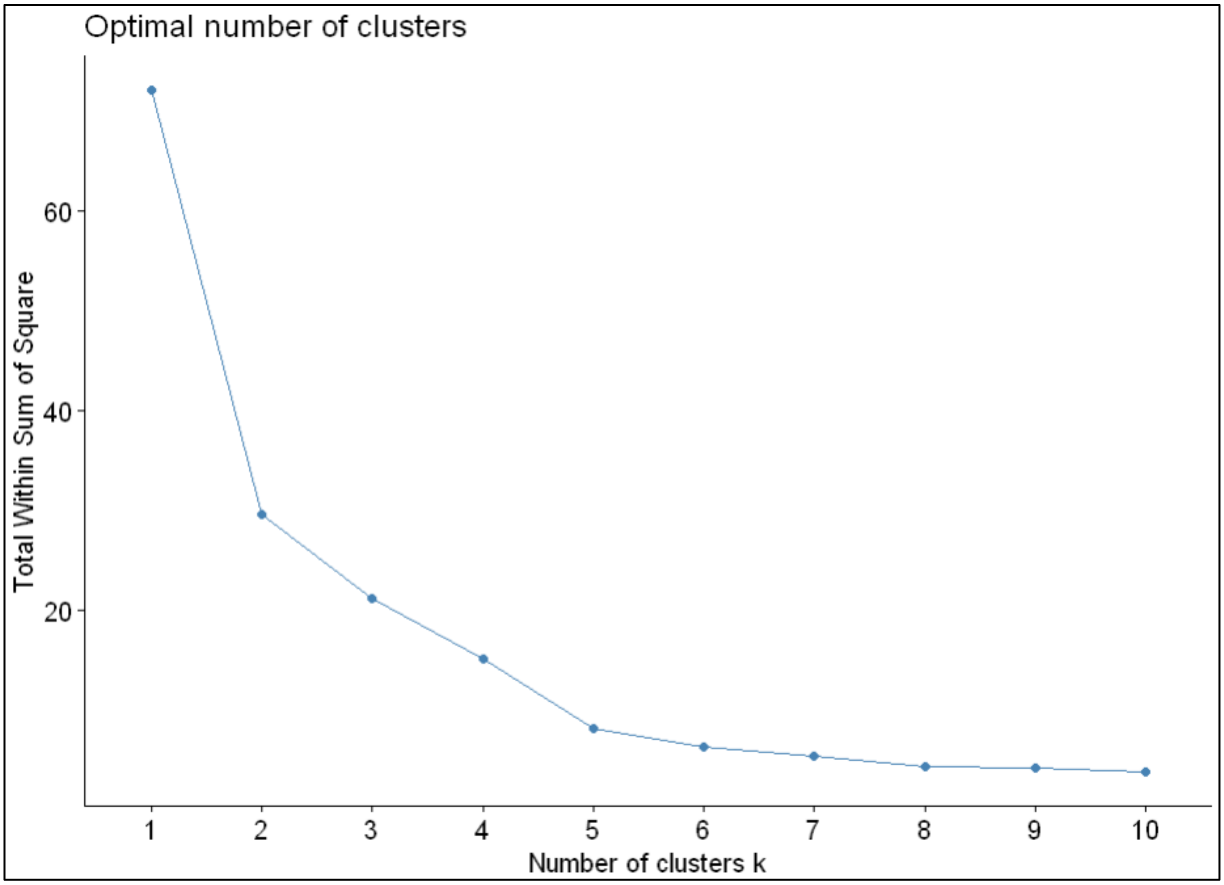
```
> # (c)  Choose 25 countries at random using sample function. Please make sure you used your PUID t
o set the seed for reproducibility of your work.
> set.seed(037831852)
> random_countries = sample(pisa_clean$jurisdiction, 25)
> index = match(random_countries, pisa_clean$jurisdiction)
> pisa_subset = pisa_clean[index,]
> head(pisa_subset)
      jurisdiction science_score science_se reading_score reading_se math_score  math_se
34          Sweden      499.4447   3.069711      505.7852   3.024772   502.3877 2.654251
24      Luxembourg      476.7694   1.220843      469.9854   1.125891   483.4215 1.097632
36         Türkiye      468.2996   2.013049      465.6317   2.171214   453.5078 2.260407
30        Portugal      491.6773   2.773163      491.8008   2.428931   492.4874 2.684570
38   United States      502.3800   3.317920      505.3528   3.568673   478.2447 3.235444
4           Canada      517.9977   2.153651      520.0855   1.799716   512.0169 2.357476
```

d) Extract the numerical variables (test scores) and standardize them.

```
 > # (d)Extract the numerical variables (test scores) and standardize them.
> score_data <- scale(pisa_subset[, c("math_score", "reading_score", "science_score")])
> head(score_data)
    math_score reading_score science_score
34  0.30739179    0.64272417    0.24805116
24 -0.71233676   -1.18732838   -0.86753491
36 -2.32066285   -1.40988649   -1.28423431
30 -0.22490278   -0.07214656   -0.13409004
38 -0.99066837    0.62061775    0.39246507
4   0.82511295    1.37374161    1.16082630
```

e) Perform the cluster analysis using the K-means clustering to identify the members in each clusters.

```
> # (e)  Perform the cluster analysis using the K-means clustering to identify the members in each
clusters.
> set.seed(037831852)
> # install.packages("factoextra")
> library(factoextra)
> fviz_nbclust(score_data, kmeans, method="wss")
```

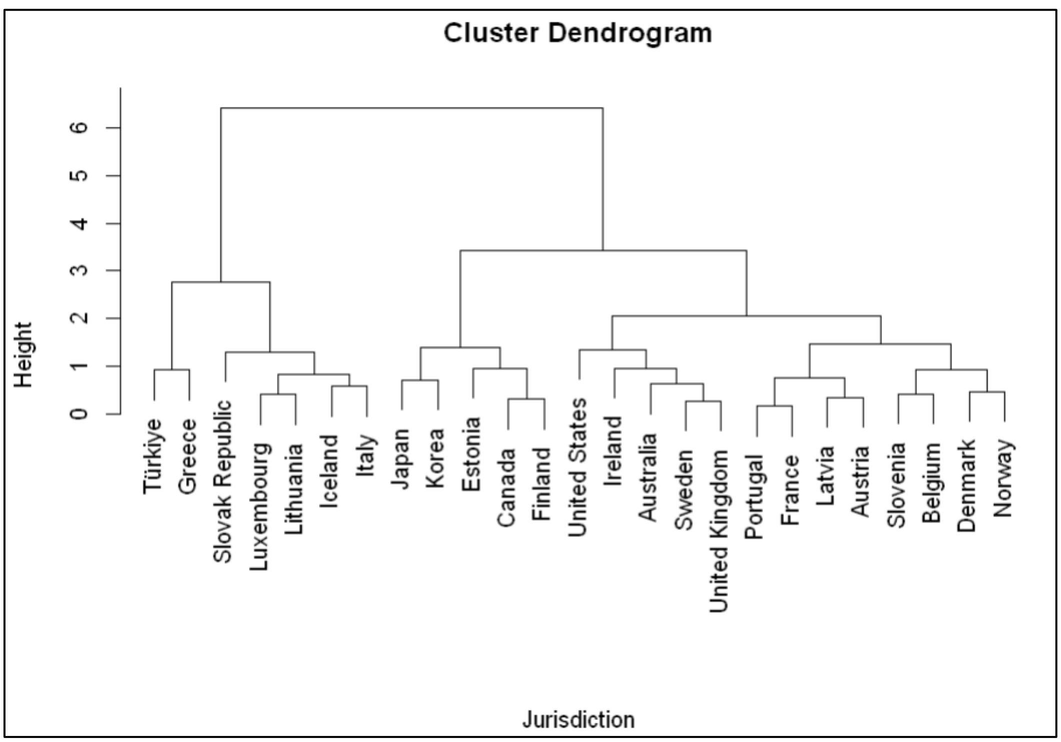



```
> cat("Optimal Cluster Size from the Elbow plot: 3")
Optimal Cluster Size from the Elbow plot: 3
> kmeans_model <- kmeans(score_data, centers = 3, nstart = 25)
> kmeans_model
K-means clustering with 3 clusters of sizes 5, 13, 7

Cluster means:
  math_score reading_score science_score
1  1.2074334     1.1760338      1.437760
2  0.1115561     0.2371453      0.109178
3 -1.0696280    -1.2804368     -1.229731

Clustering vector:
34 24 36 30 38  4  9 22 16 17 12  1 32 14 11 31  2 23  3 20 10 21 37 28 19
 2  3  3  2  2  1  2  2  3  2  2  2  2  3  1  3  2  3  2  1  1  1  2  2  3

Within cluster sum of squares by cluster:
[1] 1.836651 6.924486 7.579493
 (between_SS / total_SS =  77.3 %)

Available components:

[1] "cluster"     "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
> kmeans_model$size
[1]  5 13  7
> cat("K-means clustering with 3 clusters of sizes '5 13 7'
+ i.e 5 - math_score
+     13 - reading_score
+     7 - science_Score")
K-means clustering with 3 clusters of sizes '5 13 7'
i.e 5 - math_score
```

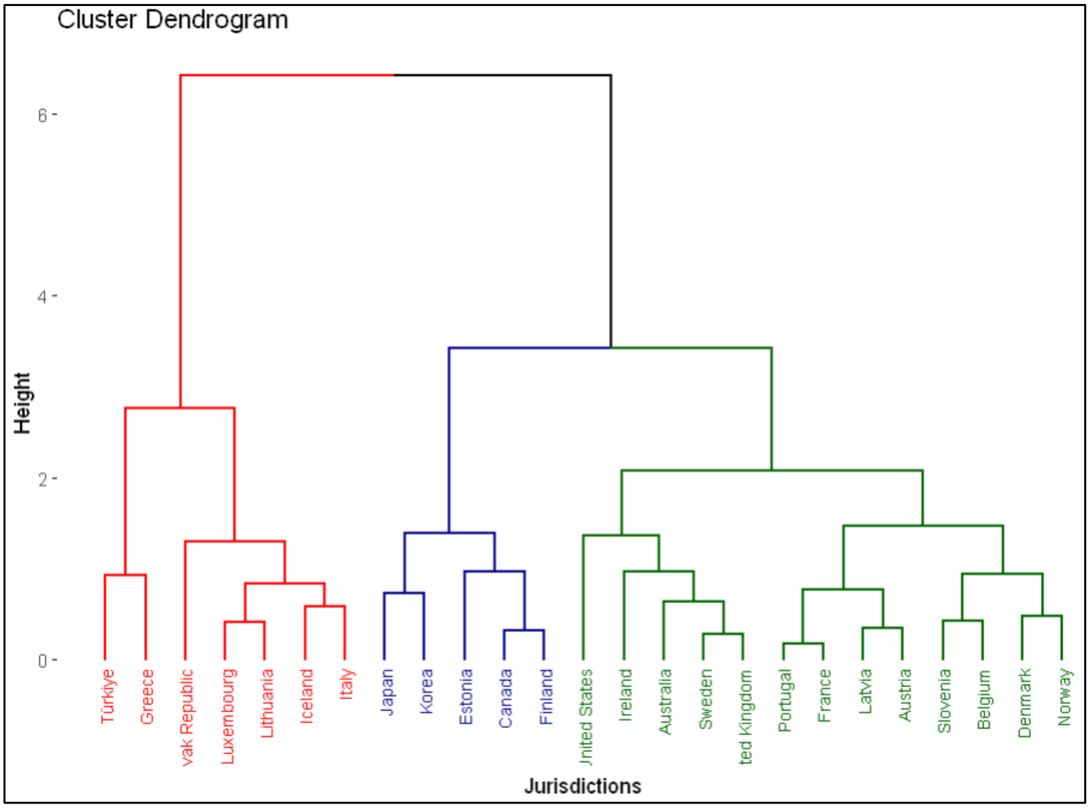
```
     13 - reading_score
     7 - science_Score
> # Contigency Table/Confusion Matrix: having 3 rows representing each cluster, and 1 - present, an
d 0 - absent
> class = pisa_subset$jurisdiction
> table(kmeans_model$cluster, class)
   class
    Australia Austria Belgium Canada Denmark Estonia Finland France Greece Iceland Ireland Italy Japan Korea
  1         0       0       0      1       0       1       1      0      0       0       0     0     1     1
  2         1       1       1      0       1       0       0      1      0       0       1     0     0     0
  3         0       0       0      0       0       0       0      0      1       1       0     1     0     0
   class
    Latvia Lithuania Luxembourg Norway Portugal Slovak Republic Slovenia Sweden Türkiye United Kingdom
  1      0         0          0      0        0               0        0      0       0              0
  2      1         0          0      1        1               0        1      1       0              1
  3      0         1          1      0        0               1        0      0       1              0
   class
    United States
  1             0
  2             1
  3             0
> pisa_subset$cluster <- kmeans_model$cluster
```

f) Display the Cluster Dendrogram.

```
> # (f) Hierarchical clustering and dendrogram
> distance_matrix <- dist(score_data)
> hc <- hclust(distance_matrix)
> plot(hc, main = "Cluster Dendrogram", xlab = "Jurisdiction", sub = "", labels = pisa_subset$juris
diction)
```




```
> # or
> hc$labels <- as.character(pisa_subset$jurisdiction)
> fviz_dend(hc, k = 3, cex = 0.6, k_colors = c("red", "darkblue", "darkgreen"), main = "Cluster Den
drogram", xlab = "Jurisdictions")
```



Cluster Dendrogram
Height
6
4
2
0
Türkiye
Greece
vak Republic
Luxembourg
Lithuania
Iceland
Italy
Japan
Korea
Estonia
Canada
Finland
Jnited States
Ireland
Australia
Sweden
ted Kingdom
Portugal
France
Latvia
Austria
Slovenia
Belgium
Denmark
Norway
Jurisdictions