

HOMEWORK 12

KENIGBOLO MEYA STEPHEN

1. Apply any clustering techniques (hierarchical, SOM, K-Means) that you wish and try to recover what is pictured on the image.



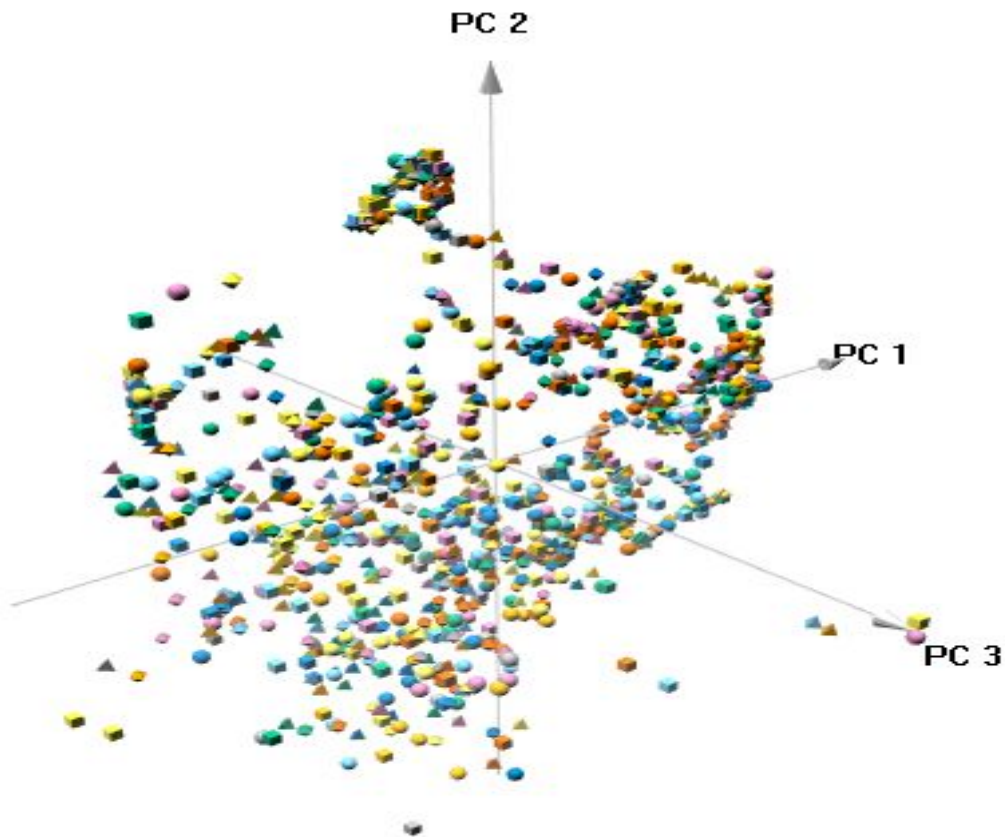
So I applied several clustering techniques ranging from summing the rows and sorting it in descending order, applying K-Means with centers of 20, 40, 60 and even 100, applying hierarchical clustering and several other techniques I haven't documented for the sake of space. However the above images are a collage of the images from the applied techniques. Full resolution images can be found in the folder named "Question 1". Below you will find the best reproduction of the original image I got.



Reproduction of image using k means with number of clusters as 100

The R scripts written for this task can be found in the Rmd file attached.

2. Use the same data matrix from task 1 and run a PCA analysis on it. Plot first three principal components as 2-dimensional plots PC1-PC2, PC1-PC3, PC2-PC3 of these data or as a 3D plot. Check out PCA [example](#)



3D Visualization snapshot of the Principal Component Analysis

For this task I made use of the R Library called `pca3d` which is basically a library of functions that signify the presentation of PCA models in 3D interactive representation using 'rgl'. I performed the principal components analysis by running the `prcomp` function on my preprocessed data. The `prcomp` function returns a `pca` object. In order to generate a 3D model I invoke the `pca3D` function for rendering the PC1, PC2 and PC3 in 3D format. However to double check my 3D plot I plotted 2-Dimensional plots for PC1 vs PC2, PC2 vs PC3 and PC1 vs PC3. The images of the plot are attached to the Question 2 folder.

3. Grab US census data (e.g. medium size) in here - <http://biit.cs.ut.ee/~vilo/edu/Data/census2000/>
 Make Pivot table summary about people's earnings based on various variables. E.g. the gender and education level. Make sure to apply heatmaps on top of pivot table.

H	I	J	K	L	M	N	O
	EDUCATION	SEX					
		1	2				
	0	0	0				
	1	3504.785	947.7903			Scale	Color
	2	690.8592	387.343			0 - 5000	
	3	5083.672	2771.055			5001 - 10000	
	4	4073.962	2256.137			10001 - 25000	
	5	9596.499	3326.658			>25000	
	6	11185.85	4281.339				
	7	11167.64	5003.165				
	8	19404.36	9200.886				
	9	24012.28	11515.83				
	10	28201.21	16305.05				
	11	28488.35	16949.32				
	12	35081	21991.48				
	13	53294.26	33193.71				
	14	70755.17	44412.23				
	15	94245.2	46821.96				
	16	61467.68	41476.07				

Pivot Table Summary Heatmap in Excel for Earnings according to Education and Sex

I performed this task using Excel and R. I began by extracting the data and reading it into R. In R I created a function which iterates through the combination of values for Education and Sex and then prints out the mean of those values. Basically to get the earnings of people with Education 0 and Sex 1, I iterate through all the values where Education is equal to 0 and Sex equal to one and then subset the table after which I find the mean of the earnings column for the subsetted table.

It would have been easier to apply the heatmap function in R if I simply had created a separate dataframe and spewed the outputs of if iteration into the dataframe however I preferred to do heat maps in Excel as this helps me fine tune it to my taste.

Rscript and Excel file for this task can be found in the Question 3 folder.

4. On the same data - try to visualize other relationships in data - based on ancestry, industry, marital status and education, for example.

	H	I	J	K	L	M	N	O	P	Q
	RACE									
EDUCATION	1	2	3	4	5	6	7	8	9	
0	0	0	0	NaN	0	0	0	0	0	0
1	1764.239	0	81.81818	NaN	0	2188.811	NaN	3662.691	1268.056	
2	264.387	0	954.5455	NaN	2875	1350.909	0	971.2766	330.7087	
3	2526.801	630.303	909.0909	9200	2666.667	2441.358	NaN	7413.196	2206.333	
4	2121.628	1115.385	290.9091	NaN	11100	3049.294	0	5798.189	3083.077	
5	6172.234	233.3333	7362.5	NaN	4333.333	5001.176	NaN	7722.902	6285.714	
6	7381.979	3166.667	1750	NaN	21300	3324.189	NaN	10799.38	7635.946	
7	8798.571	3196.296	9971.429	NaN	1375	3690	20000	7662.832	7529.524	
8	15046.41	8457.442	11508.33	22000	9416	13684.26	NaN	13781.36	13618.52	
9	18405.8	14493.41	15045.71	23000	10097.5	13501.61	26814.29	15980.7	16442.66	
10	22665.55	18029.62	12425	NaN	9750	17769	NaN	18626.42	19474.07	
11	23237.95	20886.6	21457.14	NaN	27317.14	18837.91	11000	22997.86	22444.71	
12	30044.49	20095.24	17050	NaN	50540	22401.84	14000	23230.34	36413.04	
13	46518.5	36228.45	58240	NaN	27833.33	32605.83	40542.86	26848.74	37908.54	
14	60266.88	41323.08	28000	NaN	NaN	47429.63	60000	31826.67	43396	
15	78746.18	50487.5	50000	NaN	0	68384.81	NaN	36172.22	169533.3	
16	56387.81	NaN	NaN	NaN	1500	35463.64	NaN	40600	103250	
			SCALE	COLOR						
			NAN							
			0 - 5000							
			5001 - 15000							
			15001 - 30000							
			>30000							

Pivot Table Summary Heatmap in Excel for Earnings according to Race and Education

Like in the previous task, I performed this task in a similar manner by iterating through the various values and printing their different outputs to the console and then proceeding to read it into Excel to produce a heat map. In this task however I discovered something really interesting and it was the fact that for a certain race which in the data is represented as 4, there was basically no earnings for them in almost all educational levels which brings me to the thought that this is probably an underrepresented race in the US. This could have been a result of unavailability of data or simply show mass neglect of a specific race or even low population density for the said Race.

For the bonus Task I am hoping to explore more on this in order to get some more insights. Code can be found in the zip file while heatmap is located in the folder Question 4.

5. Read the [Jim Gray - Data Cube abstraction](https://en.wikipedia.org/wiki/OLAP_cube#Operations). Describe the key operators from this article using examples based on above census data (tasks 3-4). (Alternative list of operations - https://en.wikipedia.org/wiki/OLAP_cube#Operations)

Extracting - This refers to moving the aggregated data from the database into a file or table and in the bonus task this was done with MYSQL.

Dice - As stated in wikipedia, this allow us to pick specific values of multiple dimensions and this is basically the technique we used in observing the relationship between variables.

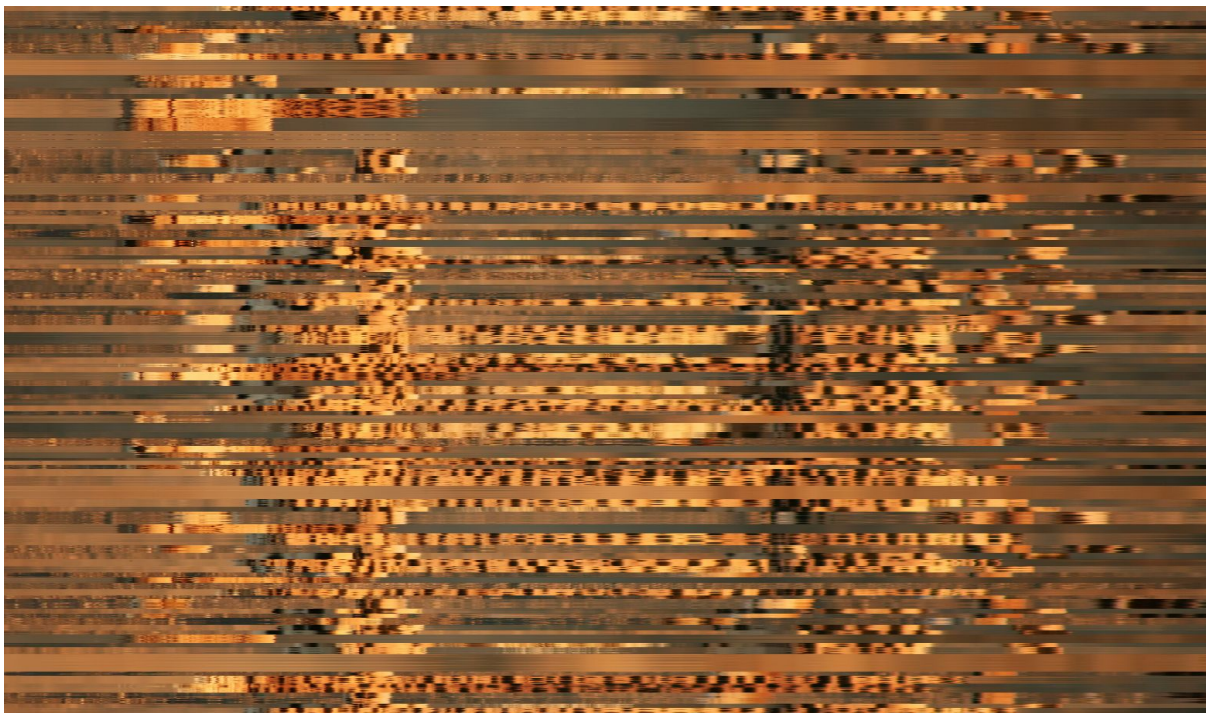
Drill Down/Up - This allows the user to navigate between different data levels and this technique was basically used in selecting the different levels for every variable that the relationship was checked.

Roll-up - A roll-up refers to summarizing the data along a dimension and this is what the task was all about. The summarization formula was the average of the relationship column (Earnings) for each level of the dependent variables.

Visualizing - This refers to visualizing the data and it was accomplished with the aid of heatmaps.

6. (Bonus 2p) Attempt running a TSP or other techniques to recover as well as possible the original image of tasks 1.-2.

To perform this task I tried several algorithms. I first began with trying out TSP but it was really slow and time consuming as well as giving me errors after spending several hours fine tuning and waiting. I used several swapping algorithm functions but they were all really not giving me any visibly nice image so I switched over to Fuzzy c-means clustering algorithm and the results improved albeit not as good as I would have loved it to be.

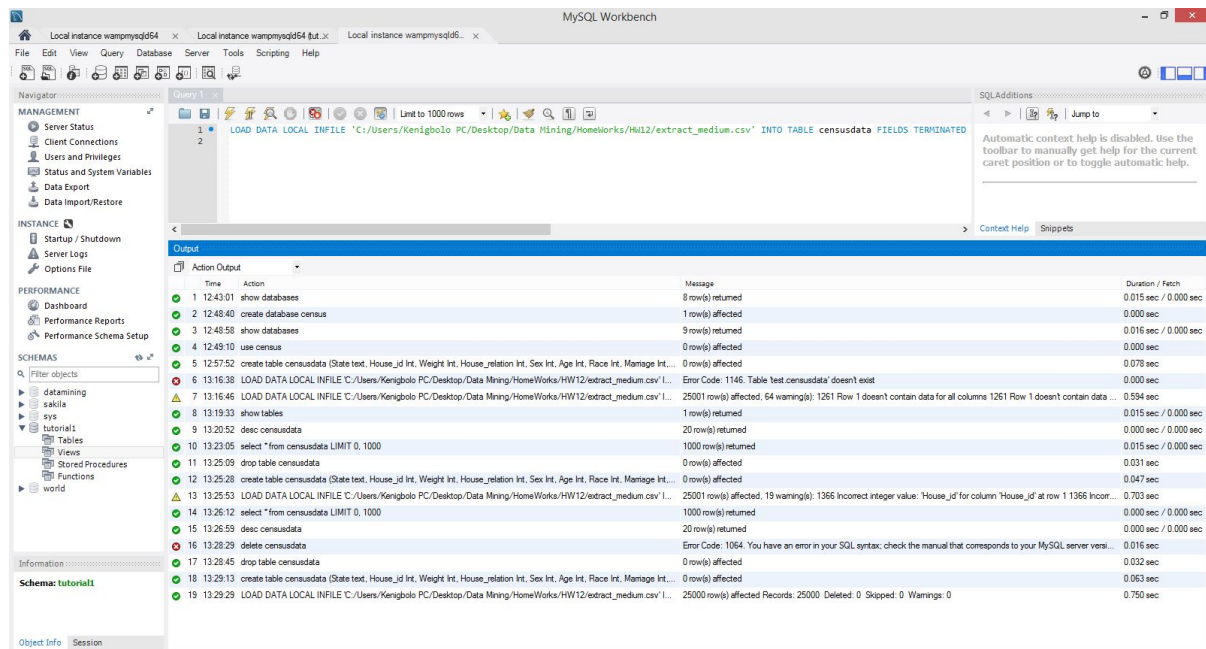


The fuzzy c-means algorithm basically works by assigning membership to each data point which corresponds to the center of each cluster based on the distance between the center of the cluster and the data point. The more the data is near to the cluster center the more is its membership towards the particular cluster center. "Summation of membership of each data point should be equal to one" according to the documentation on it. After each iteration, membership and cluster centers are updated.

7. (Bonus 2p) Load the same census data sets (you can attempt larger ones, too) into a DB and run SQL queries to achieve summarization as in pivot tables.

STEP ONE

For this task I made use of the MySQL Workbench tool. First I proceeded to perform the task by creating a database in MySQL workbench called **census** and then I created a table called **censusdata** which contained columns as obtainable in the abstracted data. I proceeded to preprocess the csv file for import into MySQL by deleting the first row which corresponds to the column names I already created in the table. Finally I imported the data into the created **censusdata** table in the **census** database (the command can be found in the appendix section) I created then exported this file as an sql database (i.e. for the purpose of verification. if needed the file can be found inside the dumps folder of Question 7). Below is a screenshot of the commands run for this first stage. All commands can be found in the Appendix of this document

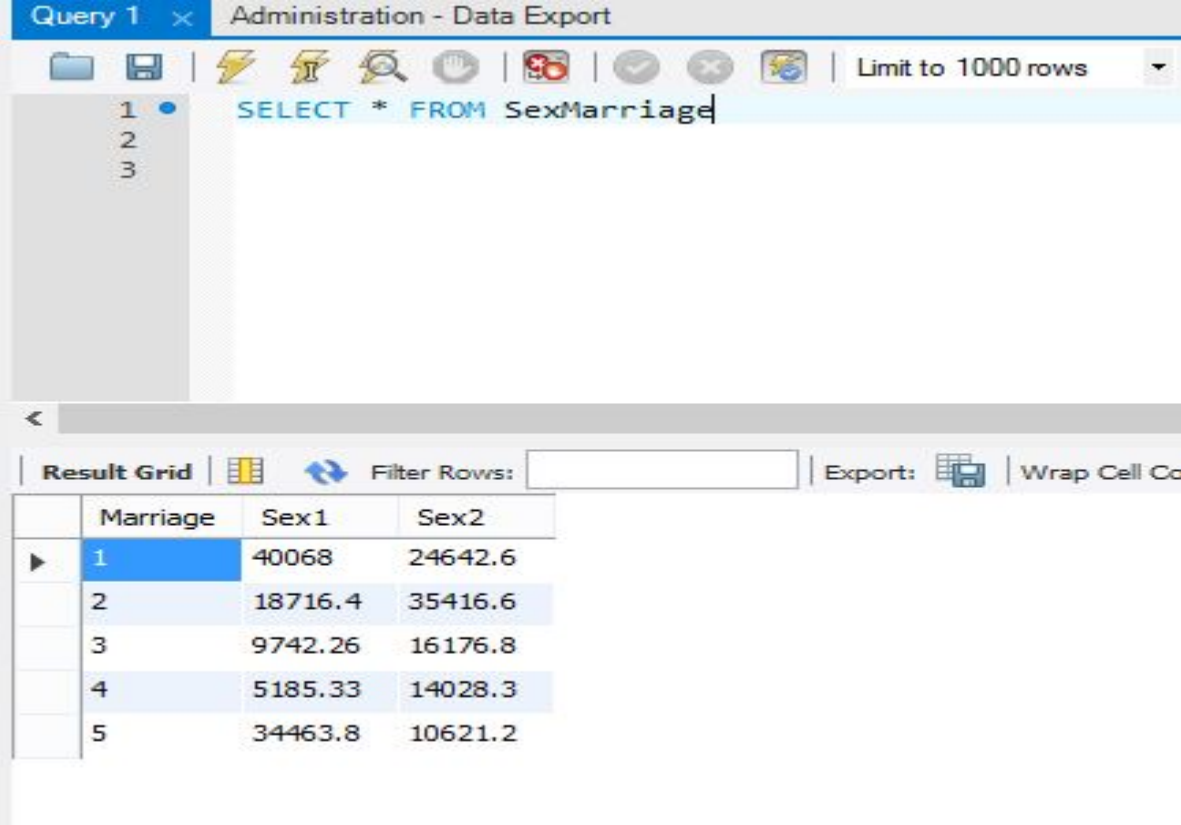


STEP TWO

For Pivot table summarization of earnings I decided to use the following variables for two different summarization pivot tables.

1. Sex and Marriage - To find out if the marital status of an individual affects their earnings. This is interesting as although the marital status and sex labels are not defined I believe it might give an overview about if marriage affects the earnings of women.
2. Race and Language - To find out if people of a certain race earn more based on the language that they speak. Once again although the labels aren't given, from the results of my summarization in Question 4 I believe this will also be an interesting avenue to explore.

Summarization was done by selecting all rows that meet a certain condition (the different values for the variables) and then finding the mean of the resulting table's earnings column. This value was stored manually and then inserted into the various pivot tables. Screenshots of the tables can be found below (as well as in high resolution in question 7 folder) as well as all queries attached in the appendix of this document.



The screenshot shows a database query interface. At the top, there's a tab labeled 'Query 1' and a window title 'Administration - Data Export'. Below the title bar is a toolbar with various icons. The main area displays a SQL query: `SELECT * FROM SexMarriage`. To the left of the query is a list of row numbers 1, 2, and 3. Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows' input field and an 'Export' button. The grid itself contains five rows of data with columns labeled 'Marriage', 'Sex1', and 'Sex2'. The first row is highlighted in blue.

	Marriage	Sex1	Sex2
1		40068	24642.6
2		18716.4	35416.6
3		9742.26	16176.8
4		5185.33	14028.3
5		34463.8	10621.2

Summarization of Earnings for Sex and Marriage Relationship

Result Grid										
Filter Rows: <input type="text"/>										
Export: Wrap Cell Content:										
Language	Race1	Race2	Race3	Race4	Race5	Race6	Race7	Race8	Race9	
0	25156.6	20673.2	15539.2	34977.8	6252.63	10892	NULL	16164.4	16350	
1	12756	19872.2	10562.5	12014	NULL	10948.7	8444.44	6368.96	3992.86	
2	13854.5	7491.51	9200	16825.7	11236.4	11237.9	9706.11	22250	NULL	
3	22500	13666	9276.43	12318.2	18059	250	NULL	0	7648.02	
4	12836	20559.7	23202.8	17813.3	NULL	5150	12616.1	NULL	7181.48	

RaceLanguage 17 x										
Output										
Action Output										
	Time	Action	Message							
54	16:32:30	CREATE TABLE RaceLanguage (Language Int, Race1 Float, Race2 Float, Race3 Float, Race4 Float, Race5 Float, R...	0 row(s) affected							
55	16:32:44	INSERT INTO RaceLanguage VALUES (0, 25156.58, 20673.16, 15539.24, 34977.78, 6252.632, 10892, Null, 16164....	1 row(s) affected							
56	16:32:59	INSERT INTO RaceLanguage VALUES (1, 12756.02, 19872.22, 10562.5, 12013.95, Null, 10948.73, 8444.444, 6368....	1 row(s) affected							
57	16:33:08	INSERT INTO RaceLanguage VALUES (2, 13854.46, 7491.513, 9200, 16825.71, 11236.36, 11237.91, 9706.108, 22....	1 row(s) affected							
58	16:33:18	INSERT INTO RaceLanguage VALUES (3, 22500, 13666.01, 9276.429, 12318.22, 18059.02, 250, Null, 0, 7648.015)	1 row(s) affected							
59	16:33:31	INSERT INTO RaceLanguage VALUES (4, 12836, 20559.68, 23202.76, 17813.33, Null, 5150, 12616.14, Null, 7181.4...	1 row(s) affected							
60	16:33:48	SELECT * FROM RaceLanguage LIMIT 0, 1000	5 row(s) returned							

Summarization of Earnings for Race and Language Relationship

APPENDIX

<!--CREATE AN SQL DATABASE-->

CREATE DATABASE census

<!--SWITCH TO USE DATABASE CREATED-->

USE census

<!--CREATE A TABLE INSIDE OF THE SELECTED DATABASE-->

CREATE TABLE censusdata (State text, House_id Int, Weight Int, House_relation Int, Sex Int, Age Int, Race Int, Marriage Int, Education Int, Ancestry Int, Language Int, Employment_status Int, Traveltime Int, Industry Int, Occupation Int, Hours Int, Weeks Int, Salary Int, Income Int, Earnings Int)

<!--IMPORT CSV FILE INTO THE CREATED DATABASE-->

LOAD DATA LOCAL INFILE 'C:/Users/Kenigbolo PC/Desktop/Data Mining/HomeWorks/HW12/extract_medium.csv' INTO TABLE censusdata FIELDS TERMINATED BY ','

<!--OVERVIEW OF THE UPDATED TABLE-->

DESC censusdata

<!--VIEW THE UPDATED DATABASE TABLE-->

SELECT * FROM censusdata LIMIT 0, 25000

<!-- GET THE AVERAGE FOR ALL MARRIAGE VALUES EARNINGS WHEN SEX VALUE IS 1 -->

**SELECT AVG(Earnings)
FROM censusdata
WHERE Sex = 1 AND Marriage =1 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Sex = 1 AND Marriage =2 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Sex = 1 AND Marriage =3 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Sex = 1 AND Marriage =4 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Sex = 1 AND Marriage =5 LIMIT 0, 25000**

<!-- GET THE AVERAGE FOR ALL MARRIAGE VALUES EARNINGS WHEN SEX VALUE IS 2 -->

**SELECT AVG(Earnings)
FROM censusdata
WHERE Sex = 2 AND Marriage =1 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Sex = 2 AND Marriage =2 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Sex = 2 AND Marriage =3 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Sex = 2 AND Marriage =4 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Sex = 2 AND Marriage =5 LIMIT 0, 25000**

<!-- CREATE SUMMARIZATION TABLE AND INPUT SUMMARIZATION VALUES FOR SEX VS MARRIAGE EARNINGS-->

CREATE TABLE SexMarriage (Marriage Int, Sex1 Float, Sex2 Float)

INSERT INTO SexMarriage VALUES (1,40067.98,24642.57)

INSERT INTO SexMarriage VALUES (2,18716.37,35416.55)

INSERT INTO SexMarriage VALUES (3,9742.265,16176.84)

INSERT INTO SexMarriage VALUES (4,5185.334,14028.28)

INSERT INTO SexMarriage VALUES (5,34463.77,10621.21)

<!-- DISPLAY SEX AND MARRIAGE EARNINGS SUMMARIZATION PIVOT TABLE -->

SELECT * FROM SexMarriage

<!-- GET THE AVERAGE FOR ALL MARRIAGE VALUES EARNINGS WHEN LANGUAGE IS 0 -->

**SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 0 AND Race = 1 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 0 AND Race = 2 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 0 AND Race = 3 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 0 AND Race = 4 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 0 AND Race = 5 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 0 AND Race = 6 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 0 AND Race = 7 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 0 AND Race = 8 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 0 AND Race = 9 LIMIT 0, 25000**

<!-- GET THE AVERAGE FOR ALL MARRIAGE VALUES EARNINGS WHEN LANGUAGE IS 1 -->

**SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 1 AND Race = 1 LIMIT 0, 25000**

**SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 1 AND Race = 2 LIMIT 0, 25000**

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 1 AND Race = 3 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 1 AND Race = 4 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 1 AND Race = 5 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 1 AND Race = 6 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 1 AND Race = 7 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 1 AND Race = 8 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 1 AND Race = 9 LIMIT 0, 25000
```

<!-- GET THE AVERAGE FOR ALL MARRIAGE VALUES EARNINGS WHEN LANGUAGE IS 2 -->

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 2 AND Race = 1 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 2 AND Race = 2 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 2 AND Race = 3 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 2 AND Race = 4 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 2 AND Race = 5 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 2 AND Race = 6 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 2 AND Race = 7 LIMIT 0, 25000
```



```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 2 AND Race = 8 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 2 AND Race = 9 LIMIT 0, 25000
```

<!-- GET THE AVERAGE FOR ALL MARRIAGE VALUES EARNINGS WHEN LANGUAGE IS 3 -->

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 3 AND Race = 1 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 3 AND Race = 2 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 3 AND Race = 3 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 3 AND Race = 4 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 3 AND Race = 5 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 3 AND Race = 6 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 3 AND Race = 7 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 3 AND Race = 8 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 3 AND Race = 9 LIMIT 0, 25000
```

<!-- GET THE AVERAGE FOR ALL MARRIAGE VALUES EARNINGS WHEN LANGUAGE IS 4 -->

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 4 AND Race = 1 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 4 AND Race = 2 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
```

```
FROM censusdata
WHERE Language = 4 AND Race = 3 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 4 AND Race = 4 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 4 AND Race = 5 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 4 AND Race = 6 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 4 AND Race = 7 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 4 AND Race = 8 LIMIT 0, 25000
```

```
SELECT AVG(Earnings)
FROM censusdata
WHERE Language = 4 AND Race = 9 LIMIT 0, 25000
```

```
<!-- CREATE SUMMARIZATION TABLE AND INPUT SUMMARIZATION VALUES FOR RACE VS
LANGUAGE EARNINGS-->
```

```
CREATE TABLE RaceLanguage (Language Int, Race1 Float, Race2 Float, Race3 Float, Race4
Float, Race5 Float, Race6 Float, Race7 Float, Race8 Float, Race9 Float)
```

```
INSERT INTO RaceLanguage VALUES (1, 25156.58, 20673.16, 15539.24, 34977.78, 6252.632,
10892, Null, 16164.44, 16350)
```

```
INSERT INTO RaceLanguage VALUES (2, 12756.02, 19872.22, 10562.5, 12013.95, Null, 10948.73,
8444.444, 6368.955, 3992.865)
```

```
INSERT INTO RaceLanguage VALUES (3, 13854.46, 7491.513, 9200, 16825.71, 11236.36,
11237.91, 9706.108, 22250, Null)
```

```
INSERT INTO RaceLanguage VALUES (4, 22500, 13666.01, 9276.429, 12318.22, 18059.02, 250,
Null, 0, 7648.015)
```

```
INSERT INTO RaceLanguage VALUES (5, 12836, 20559.68, 23202.76, 17813.33, Null, 5150,
12616.14, Null, 7181.481)
```

```
<!-- DISPLAY RACE AND LANGUAGE EARNINGS SUMMARIZATION PIVOT TABLE -->
```

```
SELECT * FROM RaceLanguage
```

Reference for FCM

<https://sites.google.com/site/dataclusteringalgorithms/fuzzy-c-means-clustering-algorithm>