Question # 1

Below is th R code that applied K-means algorithm on given image for K = 3,5, and 10. Figure. 1 , Figure. 2 , and Figure. 3 shows the results of k-means for foregoing values of K.

```
fileName <- "DM2016.txt"
conn <- file(fileName,open="r")
linn <-readLines(conn)

rgb = matrix(NA,1018*1554,5)
index=1

for (i in c(2:1019))
{
  y=1
  row = strsplit(linn[i]," ")
  row = unlist(row)

  for(pcol in seq(2,4663,3))
  {
    rgb[index,1] = i-1
    rgb[index,2] = y
    rgb[index,3] =  as.numeric(row[pcol])/255
    rgb[index,4] = as.numeric(row[pcol+1])/255
    rgb[index,5] =  as.numeric(row[pcol+2])/255
    y = y + 1
    index = index + 1
  }

}
close(conn)

rgb_data = data.frame(x = as.vector(rgb[,1]), y = as.vector(rgb[,2]), r.value=as.
    vector(rgb[,3]), g.value=as.vector(rgb[,4]),b.value=as.vector(rgb[,5]))
#head(rgb_data)

kColors <- 10
kMeans <- kmeans(rgb_data[, c(3,4,5)],
                 centers = kColors)

clusterColour <- rgb(kMeans$centers[kMeans$cluster, ])

plot(y ~ x , data=rgb_data, main="k-mean cluster analysis",
     col = clusterColour, asp = 1, pch = ".",
     axes=FALSE, ylab="",
     xlab="k-means cluster analysis of 10 colours")
```

Conclusion : K-means clusters the colours in an image that's why it can't recover the shuffled image.

Figure 1: K-mean clustering with k=3

Figure 2: K-mean clustering with k=5

Figure 3: K-mean clustering with k=10

Question # 2

Following code applied dimensionality reduction technique "PCA" on given image using function "prcomp" in R.

```
pca = prcomp(img[,1:ncol(img)], scale. = FALSE, retx = TRUE, center = FALSE)
```

Below is the R code that shows the commulative summary of variance captured by principal compnonents in Figure. 4. We can see, the first 3 principal components captures more than 90% of the varaince.

```
plot((cumsum(pca$sdev^2)/sum(pca$sdev^2))*100,type="b",pch=4,col="blue",ylab="
    cumulative variance captured",xlab="principal components",main="Comulative
    Variance (%) Captured \n by Principal Components")
```

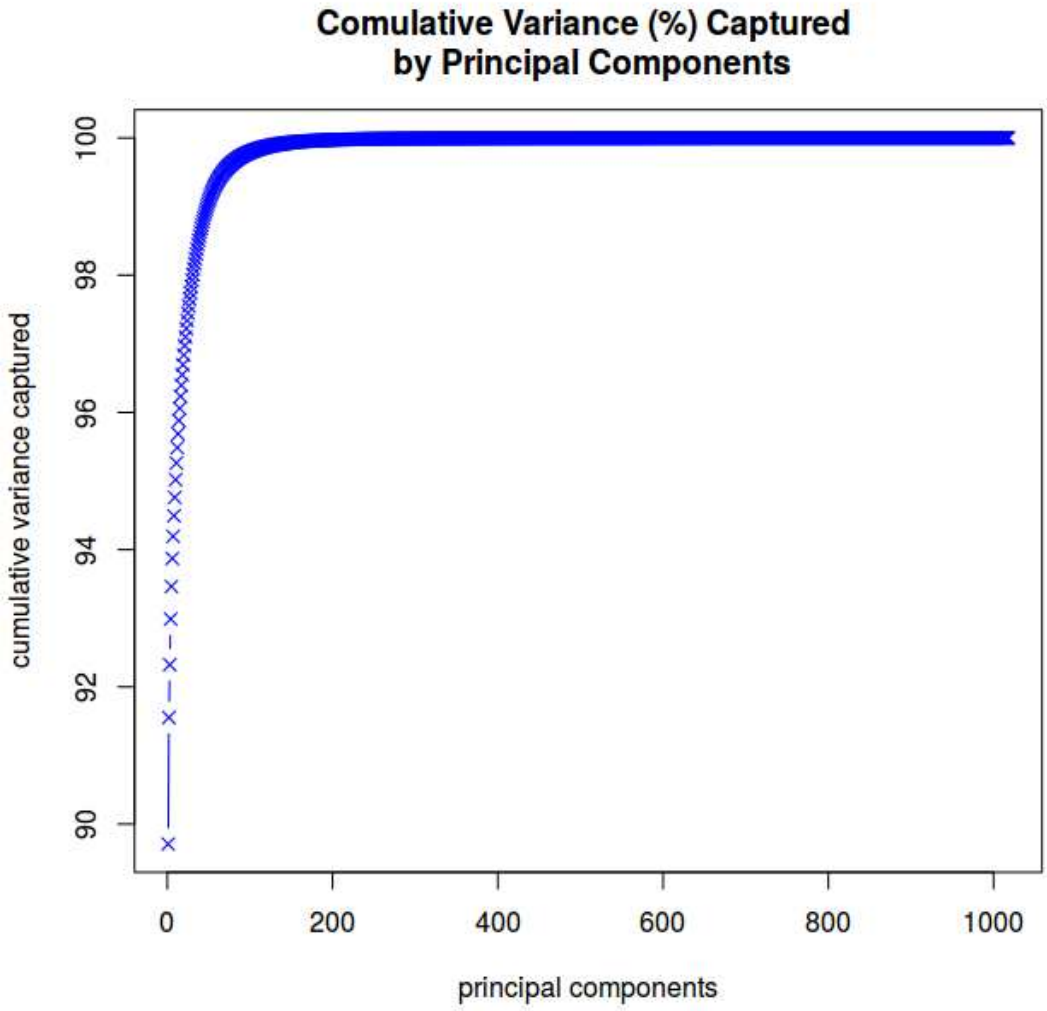

Figure 4: Comulative Variance (%) captured by principal components

Figure. 5 illustrates the 3D plot of first 3 principal components.

```
library(scatterplot3d)
with(mtcars, {
  scatterplot3d(pca$x[,1],
                pca$x[,2],
```

```
                pca$x[,3],
                color = "blue",
                xlab = "pca 1",
                ylab = "pca 2",
                zlab = "pca 3",
                main="3-D Scatterplot of PCA 1, PCA 2, PCA 3")
})
```

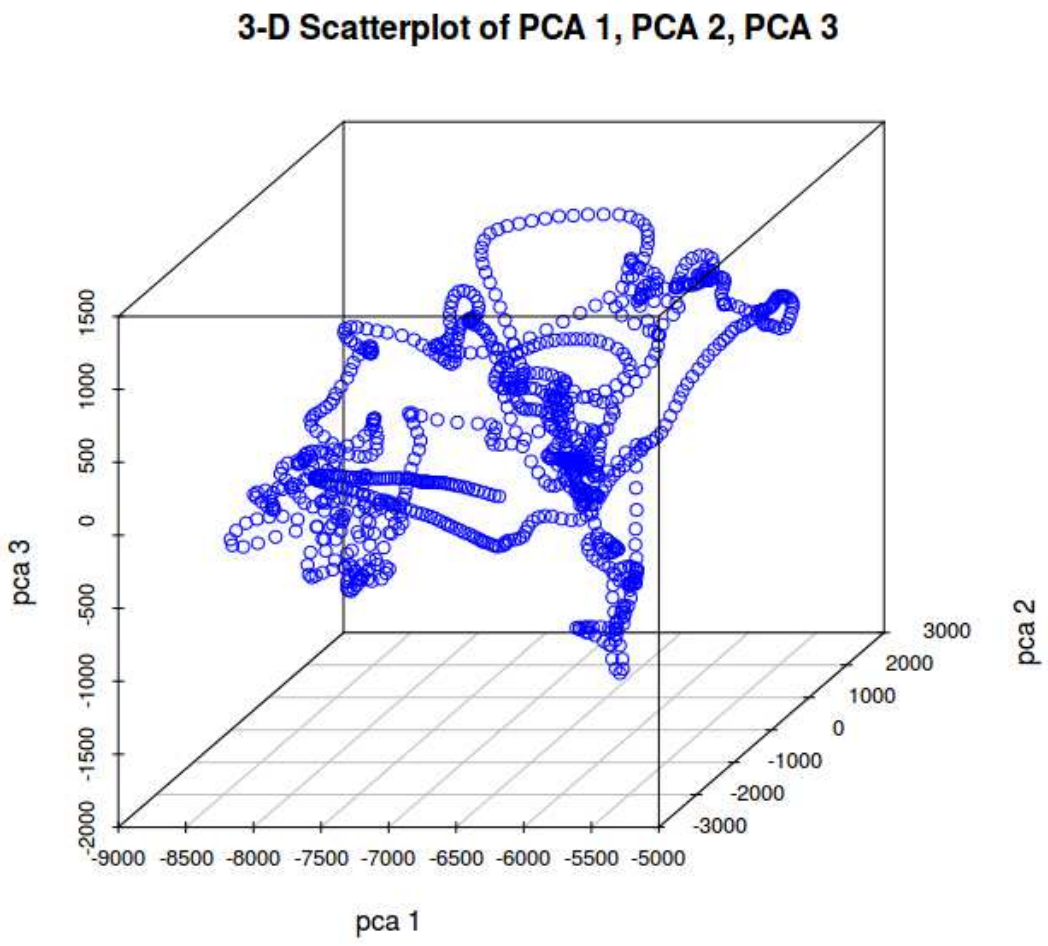

Figure 5: 3D scatter plot of PCA-1, PCA-2, and PCA-3

Question # 3

Following is the R code to compute pivot table summeary using "dplyr" package in R.

```
library(dplyr)
library(RColorBrewer)
library(gplots)

data = read.csv("extract_medium.csv",sep = ";")

data$Marriage = factor(data$Marriage,level= c(1,2,3,4,5), labels=c("Married","Widowed","Divorced","Seperated","Never married (includes under 15 years)"))
education_labels = c("Not in universe (Under 2 years)","No schooling completed","Nursery school to 4th grade","5th grade or 6th grade","7th grade or 8th grade","9th grade","10th grade","11th grade","12th grade,no diploma")
education_labels = c(education_labels,c("High school graduate","Some college, but less than 1 year","One or more years of college, no degree","Associate degree","Bachelor's degree"))
education_labels = c(education_labels,c("Master's degree","Professional degree","Doctorate degree"))

data$Education = factor(data$Education,levels=c(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),labels = education_labels)
data$Sex= factor(data$Sex,level=c(1,2), labels = c("Male","Female"))

group= group_by(data, Sex, Education)
sum_earnings = summarise(group,
                         avg_earnings = mean(Earnings))
summ_t = dcast(sum_earnings,  Education ~ Sex , value.var = "avg_earnings")
row.names(summ_t) <- summ_t$Education
summ_t <- summ_t[,2:3]

dt_matrix <- data.matrix(summ_t)

 heatmap(dt_matrix, Rowv=NA, Colv=NA,  col = brewer.pal(8, "Blues"), scale="column", margins=c(5,10),cexRow = .75,cexCol = .90)
```

Below is the pivot table summary of people's earning based on gender and education level.

Education	Male	Female
Not in universe(Under 2 years)	0.0000	0.0000
No schooling completed	3504.7847	947.7903
Nursery school to 4th grade	690.8592	387.3430
5th grade or 6th grade	5083.6720	2771.0552
7th grade or 8th grade	4073.9616	2256.1374
9th grade	9596.4985	3326.6576
10th grade	11185.8485	4281.3388
11th grade	11167.6389	5003.1646
12th grade,no diploma	19404.3564	9200.8858
High school graduate	24012.2758	11515.8326
Some college, but less than 1 year	28201.2106	16305.0455
One or more years of college no degree	28488.3473	16949.3175
Associate degree	35081.0018	21991.4803
Bachelor's degree	53294.2643	33193.7135
Master's degree	70755.1736	44412.2318
Professional degree	94245.2049	46821.9613
Doctorate degree	61467.6768	41476.0656

Heat of the above table is shown in Figure. 6

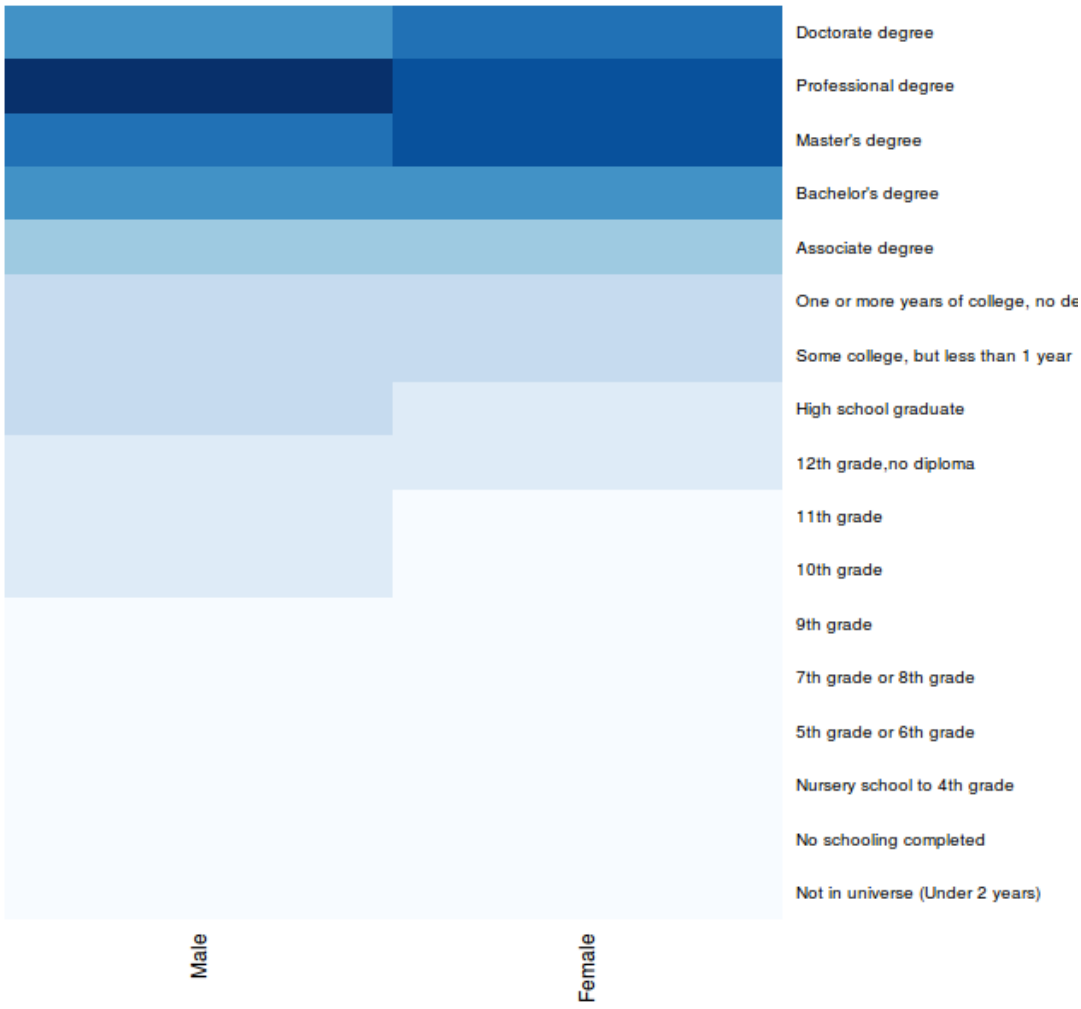

Figure 6: peoples' Earnings based on gender and education level

The message is clear and upsetting :-(, professional and master's degree holders are earning better than people with doctoral degree :-)

Question # 4

Figure. 7 explored the earnings based on marital status and education level. Prosperity formula, earn a profession degree and stay seperated. Another advice is never go for a doctoral degree :-)

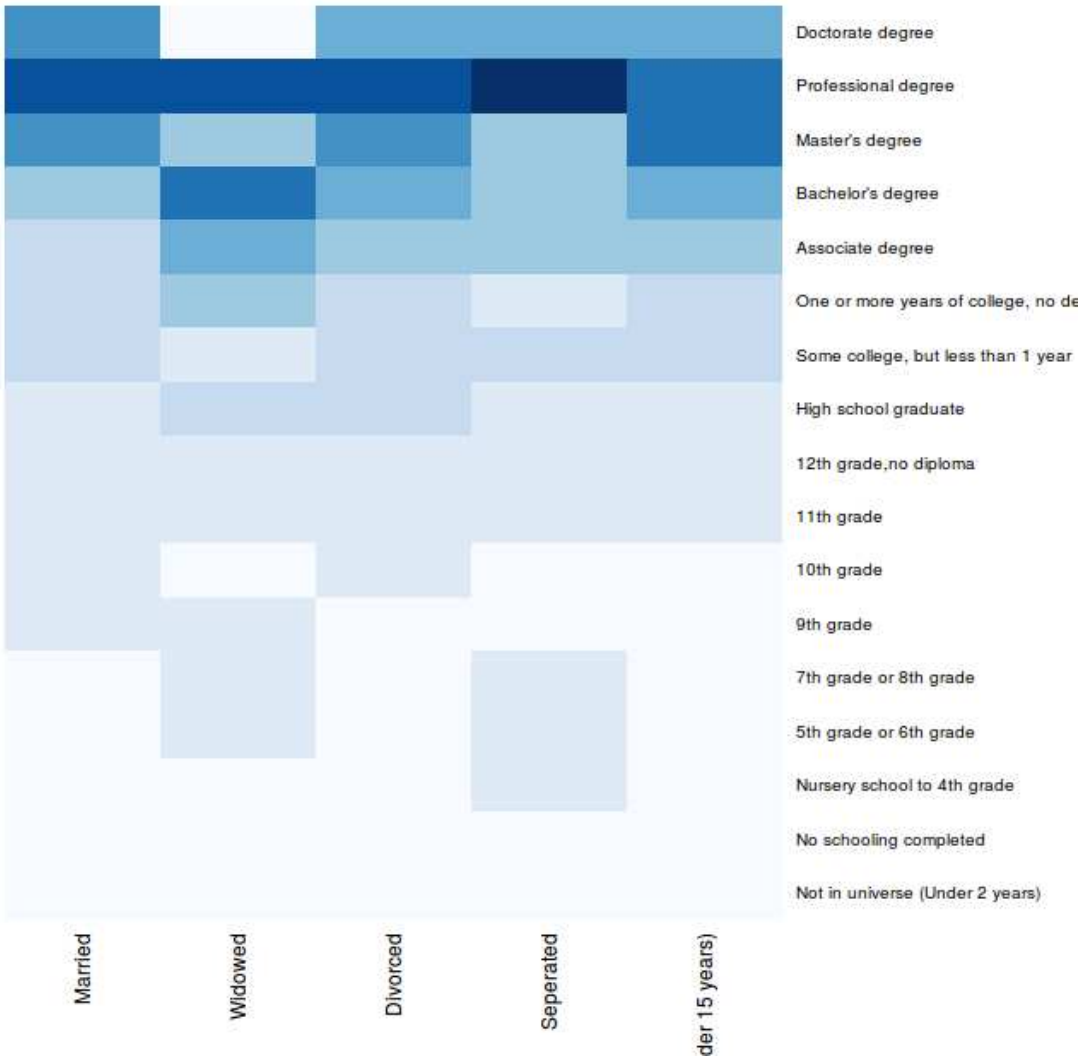

Figure 7: peoples' Earnings based on gender and education level

Figure. 8 looked into the people's earning based on gender and marital status. The male with married status are earning more than single or divorced.

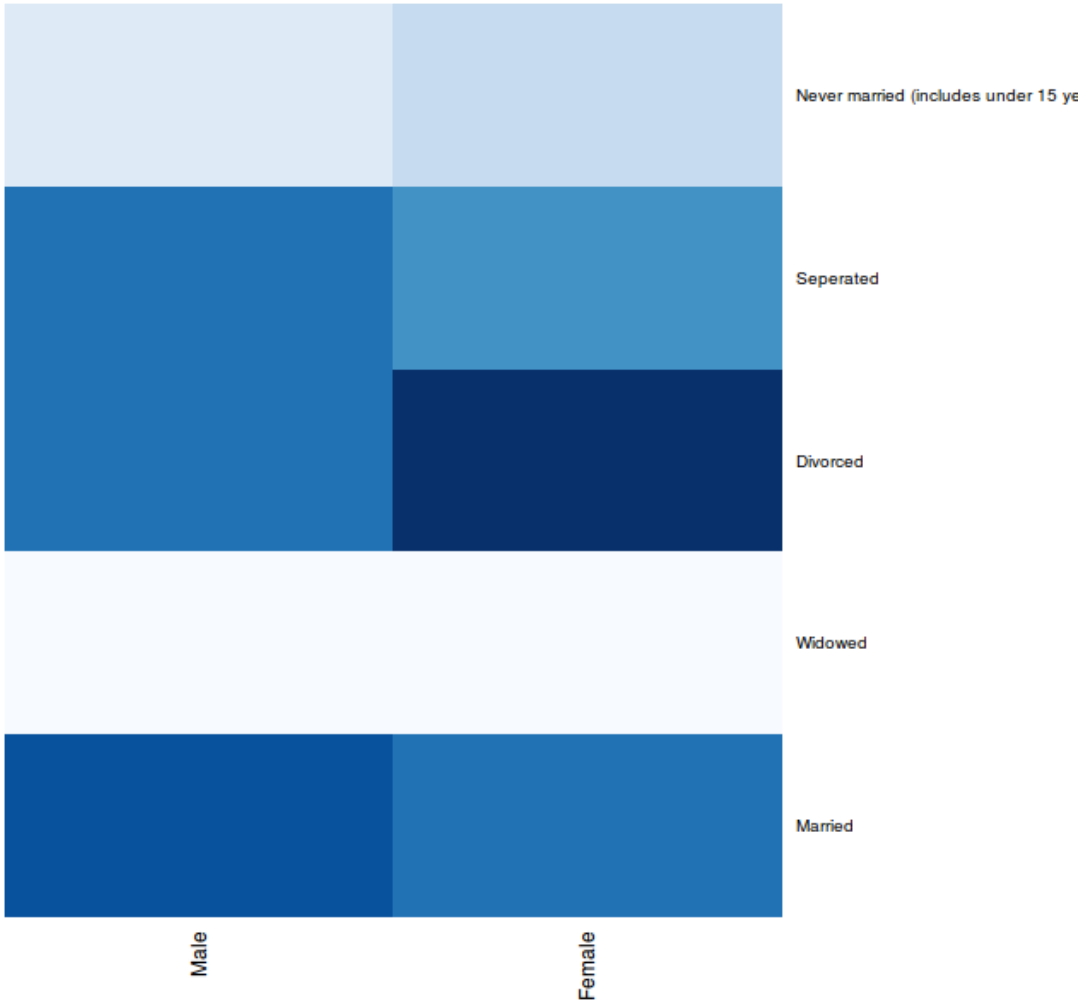

Figure 8: peoples' Earnings based on gender and education level

Question # 5

For this task , i want to present **Marital Status, Education, and Hours** as a data cube.

Slicing data can be sliced by differnt martical status values.

Dicing Using dice operation, i can pick a subcube of certain group of values (Hours, week, months etc.)

Drill up and Drill down We can drill up from hours to days and then to months. Similarly, we can drill down from months to days and then to hours.

Pivoting is about rotating the cube in space to see the data from different perspective.

Question #6

Below is the code that tried to recover the image using method **"repetitive_nn"** of TSP.

```
d_matrix = dist(img)
tsp = TSP(d_matrix)
x = solve_TSP(tsp,method="repetitive_nn")
img_order = x
```

Below is the code written in R to show the recovered image. The recovered image is shown in Figure. 9

```
order_img = matrix(NA,1018,4662)
for(i in c(1:1018))
{
  order_img[i,]= img[img_order[i],]
}

img1 = array(NA,c(1018,1554,3))
for (i in c(1:1018))
{
  row = order_img[i,]
  y=1
  for(pcol in seq(1,4662,3))
  {
    img1[i,y,1] = as.numeric(row[pcol])/255
    img1[i,y,2] = as.numeric(row[pcol+1])/255
    img1[i,y,3] = as.numeric(row[pcol+2])/255
    y = y + 1
  }
}

require('jpeg')
res = dim(img1)[1:2]
plot(1,1,xlim=c(1,res[1]),ylim=c(1,res[2]),asp=1,type='n',xaxs='i',yaxs='i',xaxt='n'
    ,yaxt='n',xlab='',ylab='',bty='n')
rasterImage(img1,1,1,res[1],res[2])
```

Figure 9: Recovered image using method “repetitive_nn” of TSP

Figure. shows the recovered image using **“nearest_insertion”** method of TSP.

Figure 10: Recovered image using method“nearest_insertion” of TSP

Question #7

To import data in MySQL , we need similar kind of Table structure to save data. Below SQL command creates the schema to store census data.

```
1
2 CREATE TABLE tbl_Census (
3 State varchar(50) DEFAULT NULL,
4 House_id varchar(50) DEFAULT NULL,
5 Weight varchar(50),
6 House_relation varchar(50),
7 Sex varchar(20),
8 Aage varchar(50),
9 Race varchar(50),
10 Marriage varchar(50) DEFAULT NULL,
11 Education varchar(50) DEFAULT NULL,
12 Ancestry varchar(50) DEFAULT NULL,
13 Language varchar(50) DEFAULT NULL,
14 Employment_Status varchar(50) DEFAULT NULL,
15 Traveltime varchar(50) DEFAULT NULL,
16 Industry varchar(50) DEFAULT NULL,
17 Occupation varchar(50) DEFAULT NULL,
18 Hours varchar(50) DEFAULT NULL,
19 Weeks varchar(50) DEFAULT NULL,
```

```
Salary varchar(50) DEFAULT NULL,
Income varchar(50) DEFAULT NULL,
Earnings int(20)DEFAULT NULL
);
```

Data is imported from large version of the csv file into table using following command.

```
LOAD DATA INFILE '/var/lib/mysql_files/extract_large.csv'
INTO TABLE tbl_Census
FIELDS TERMINATED BY ';'
```

In the end, I executed "group by" statement of SQL to generate the pivot table.

```
SELECT Sex, Education, avg(Earnings)
FROM tbl_Census
group by Sex, Education
```

Sex	Education	Avg(Earnings)
1	00	0.0000
1	01	3436.9163
1	02	1018.5143
1	03	5719.2300
1	04	4585.9244
1	05	7892.1493
1	06	8702.8604
1	07	10341.6579
1	08	16704.6764
1	09	22399.7355
1	10	26831.3657
1	11	29900.6087
1	12	35249.8174
1	13	52814.0881
1	14	63902.4224
1	15	97160.4380
1	16	69790.5412
2	00	0.0000
2	01	1358.9230
2	02	386.2795
2	03	1996.0910
2	04	1656.8459
2	05	3048.7202
2	06	3878.9171
2	07	4711.3237
2	08	7794.4182
2	09	10865.3802
2	10	15102.8411
2	11	16310.1780
2	12	20708.8457
2	13	27804.7783
2	14	35058.9334
2	15	44529.5096
2	16	44325.0832

Figure 11: pivot table using SQL group by statement