MSiA 421
Professor Edward Malthouse

Homework 2

Due: January 23, Start of class

Submit one homework per group. Put all names on the homework.

1. This data uses data from `tradeshow.csv`. You have the following variables:

   - **Buy**. Evaluate and compare specific equipment for purchase, place orders, find new suppliers and solutions.
   - **Social**. Spend time with others, network with colleagues, extend professional network.
   - **Education**. Keep up-to-date on industry trends, attend continuing education sessions, attend keynotes.

   (a) Estimate K-means on the three variables and find the 3-cluster solution. Give the cluster sizes, means and RMSE values. Describe each of the three clusters *Answer: Cluster 1: attends for social, but not to buy; moderate on education. Cluster 2: not there for social, moderate on education and buying. Cluster 3: High on all three, social, education and buying.*

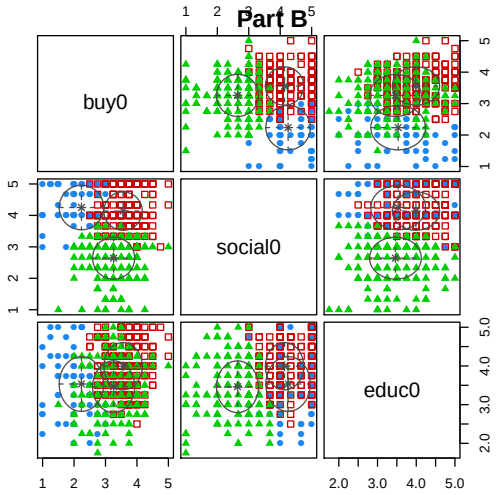
```
> fit = kmeans(dat, 3, iter.max=100, nstart=100)
> summary(fit)
    n  Pct buy0 social0 educ0   RMSE
1 106 0.24 2.25    4.10  3.48 0.6514
2 169 0.38 3.31    2.47  3.47 0.6064
3 170 0.38 3.70    4.17  4.03 0.5567
  445 1.00 3.21    3.51  3.69 0.6006
```

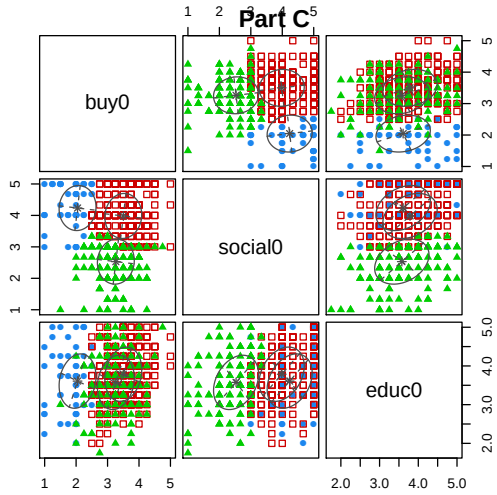
   (b) Estimate a Gaussian mixture using the three variables in R with the options `G=3` for three clusters and `modelNames="VII"` for unequal variance, round clusters (`spherical` in Python). Submit a classification plot. Compare the solution to K-means.

```
> fit = Mclust(dat, G=3, modelNames="VII") # part b
> round(data.frame(pct=fit$parameters$pro, t(fit$parameters$mean),
  sigma=sqrt(fit$parameters$variance$sigmasq)),2)

   pct buy0 social0 educ0 sigma
1 0.17 2.24    4.25  3.53  0.71
2 0.40 3.56    4.14  3.99  0.61
3 0.43 3.26    2.64  3.46  0.67

> fit2 = Mclust(dat, G=3)    # part c
> round(data.frame(pct=fit2$parameters$pro, t(fit2$parameters$mean)),2)
      pct    buy0 social0   educ0
  1   0.15    2.05    4.23    3.61
  2   0.50    3.50    3.98    3.79
  3   0.35    3.26    2.53    3.57
```





i. Do the cluster means tell the same story, or are there differences? *Answer: The solutions are similar and tell the same basic story. Kmean1=mclus1: social and education means larger higher with mclust. Kmean2=mclus3: all three means are slightly larger with mclust. Kmean3=mclus2: all three means are slightly larger with kmeans.*

ii. Comment on the K-means vs. GMM cluster sizes. *Answer: Mclust1 is smaller than kmeans1. The other two mclust clusters are larger than their corresponding kmeans segments. This is consistent with kmeans being biased toward equal-sized clusters.*

iii. Comment on the within cluster standard deviations (vs. RMSE for K-means). *Answer: There is more variation for mclust. Recall that there is a downward bias in the estimates of variances for kmeans because of the hard cluster assignments.*

iv. How many variance parameters are estimated in total? *Answer: 2 priors, 9 means and 3 variances for a total of 14 parameters. This is confirmed by looking at* `fit$df`

(c) Estimate Gaussian mixtures using three variables only with the `G=3` option (use `tied` in Python). *Answer:*

i. Do the cluster means tell the same story, or are there differences? *Answer: The basic story is the same, but mclust1 and mclust2 are smaller and mclust2 is larger.*

ii. Generate a classification plot. *Answer: See above*

iii. Which variance model did Mclust pick (it should be EEE)? Describe in words the shape of the class-conditional distributions. *Answer: The contours are ellipses that are not aligned with the coordinate axes, i.e., there is a within-class correlation. The EEE model requires that all clusters have the same covariance matrix.*

iv. How many variance parameters are estimated in total? *Answer: 2 priors, 9 means, and 6 covariances for a total of 17df.*

(d) Which of the three solutions do you prefer? *Answer: The biases of K-means are evident, and for this reason I would use one of the Mclust solutions. The BIC criterion suggests using EEE.*

2. This problem uses a data set from the Nuoqi retailer in China. You have five factors measuring attitudes toward fashion: Cross, fashion enthusiast, functional, impressive, self-expression. See the Powerpoint for the actual questions that were asked of consumers the alpha values.

```
> nuoqi = read.csv("teach/421/hw/nuoqi.csv")
> set.seed(12345)
> fit = kmeans(nuoqi[,1:5], 5, nstart=100)  # part a
> summary(fit)
    n  Pct impress selfexpress functional cross fashethus   RMSE
1 158 0.16    3.53        4.13       3.98  3.61      3.41 0.4488
2 146 0.15    3.20        3.23       3.32  3.24      3.48 0.5144
3 147 0.15    4.16        4.46       4.17  3.05      4.25 0.4431
4 303 0.30    4.07        4.33       4.18  4.12      4.21 0.3699
5 240 0.24    4.68        4.82       4.65  4.54      4.68 0.3033
  994 1.00    4.02        4.27       4.13  3.85      4.09 0.4056

> #part b
> nuoqi$impressI = nuoqi$impress - nuoqi$xbar
> nuoqi$selfexpressI = nuoqi$selfexpress - nuoqi$xbar
> nuoqi$functionalI = nuoqi$functional - nuoqi$xbar
> nuoqi$crossI = nuoqi$cross - nuoqi$xbar
> nuoqi$fashethusI = nuoqi$fashethus - nuoqi$xbar

> # part c
> set.seed(12345)
> fit = kmeans(nuoqi[,10:14], 5, nstart=100)
> summary(fit)
    n  Pct impressI selfexpressI functionalI crossI fashethusI   RMSE
1 171 0.17    -0.01         0.55        0.23   0.03      -0.39 0.3702
2 180 0.18     0.05        -0.28        0.01  -0.04       0.22 0.3960
3 309 0.31     0.43         0.50        0.36   0.36       0.41 0.3251
4 161 0.16    -0.33         0.57        0.39   0.07       0.43 0.3511
5 173 0.17     0.21         0.52        0.16  -0.90       0.21 0.4118
  994 1.00     0.13         0.38        0.24  -0.04       0.20 0.3672

> # part d
> set.seed(12345)
> ans = data.frame(k=2:6, sse=rep(NA,5), Rsqr=rep(NA,5), F=rep(NA,5))
> for(k in 2:6){
    fit = summary(kmeans(nuoqi[,1:5], k, nstart=100))
    ans$sse[k-1] = fit$sse
    ans$Rsqr[k-1] = fit$Rsqr
    ans$F[k-1] = fit$F
}
> ans
  k       sse      Rsqr        F
1 2 1217.9321 0.3816140 612.1761
2 3  997.3443 0.4936140 483.0026
3 4  890.6949 0.5477636 399.7068
4 5  813.4389 0.5869891 351.4025
5 6  754.9399 0.6166911 317.9111
```

```
> # now do it for ipsatized data
set.seed(12345)
ans = data.frame(k=2:6, sse=rep(NA,5), Rsqr=rep(NA,5), F=rep(NA,5))
for(k in 2:6){
fit = summary(kmeans(nuoqi[,10:14], k, nstart=100))
ans$sse[k-1] = fit$sse
ans$Rsqr[k-1] = fit$Rsqr
ans$F[k-1] = fit$F
}
ans
> ans
  k      sse      Rsqr        F
1 2 899.1399 0.1912410 234.5707
2 3 796.8534 0.2832458 195.8109
3 4 714.3881 0.3574217 183.5561
4 5 666.8154 0.4002125 164.9793
5 6 624.0464 0.4386823 154.4288
```

(a) Use K-means to find the five-cluster solution using the first five variables in the data frame. Give the usual sizes, means, and RMSE values. Comment on the solution. *Answer: Three clusters are basically low (cluster 2), medium (cluster 4) and high (cluster 5) on all variables. Cluster 3 is very similar to 4 except that it is low on cross. Cluster 1 is moderate on functional and self-expression and low on the others.*

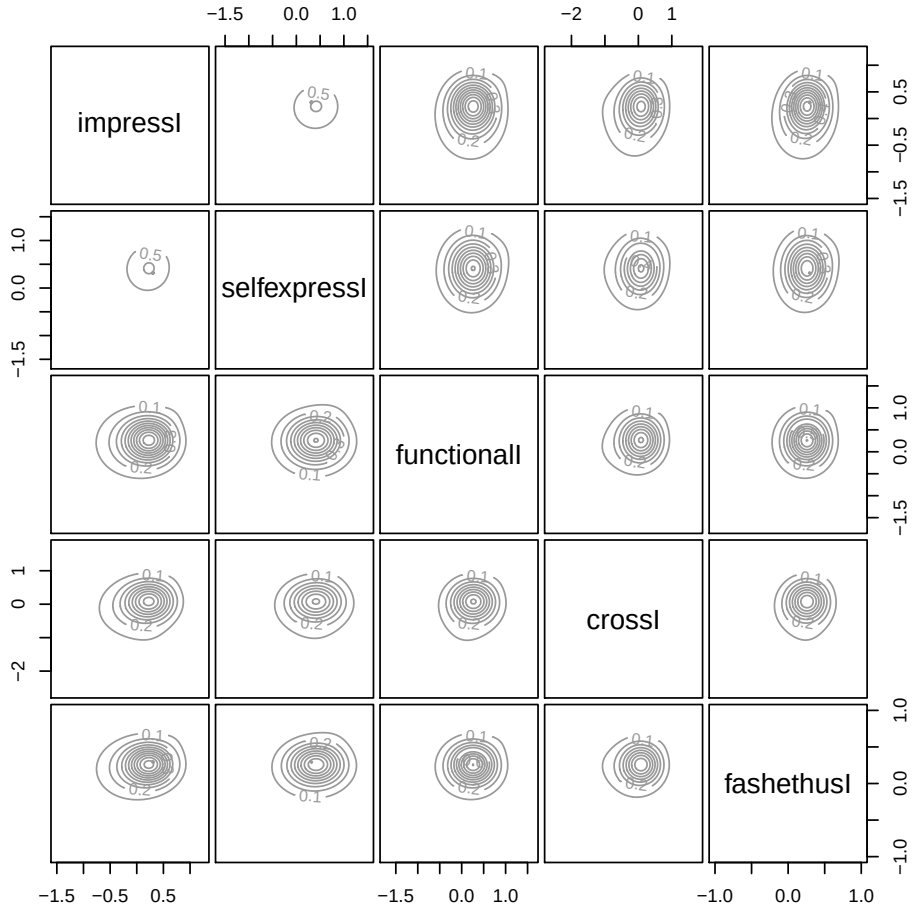
(b) Suppose that there are individual differences in the way that different respondents use the sales, where some are systematically more positive and others are more negative. The variable `xbar` is the average response *for the given respondent* to all 5-point scales on the survey. Compute five new variables equal to the original variable minus xbar, e.g., `nuoqi$impressI = nuoqi$impress-nuoqi$xbar`. This is called *ipsatization*, and it will be important to us with recommender systems. *Answer: See above for code.*

(c) Use K-means to find the five-cluster solution using the ipsatized versions of the first five variables in the data frame. Give the usual sizes, means, and RMSE values. Comment on the solution. Is there improvement? *Answer: The clusters are not great. Cluster 1 is high on self-expression and low on fashionista. Cluster 2 is moderate on all dimensions, Cluster 3 is high on all dimensions. Cluster 4 is low on impression, and high on self-expression, functional and fashionista. Cluster 5 is low on cross, high on self-expression, and moderate on the others.*

(d) Run K-means solutions for the K =2–6 solutions and examine the fit statistics (SSE, R-Squared, Pseudo F). Try both the raw and ipsatized data. Which do you suggest? *Answer: See above. There are no clear spikes in F or elbows for SSE.*

(e) Try Gaussian mixture models and look at the plots. What is the underlying problem when trying to cluster this data set? *Answer: See the code below. The density plots show that the data is really unimodal.*

```
> fit2= Mclust(nuoqi[,10:14], G=5)
> plot(fit2, what="density")
```



3. Write a function to generate data for this problem with parameter μ. There are $K = 2$ equal-sized clusters with one cluster sampled from $\mathcal{N}(-\mu, \sigma^2)$ and the other from $\mathcal{N}(\mu, \sigma^2)$, where $\sigma^2 = 1$. Assume $n_1 = n_2 = 3000$ observations from each (`Mclus` will start to have problems for larger n, but Python should be able to handle somewhat larger sample sizes, and K-means can easily handle much larger n). Estimate GMM and K-means models for $\mu = 0.5, 1$ and 2. Report the estimated means and variances. Discuss the results, especially the biases discussed in class and how they are affected by the separation of the means.

Answer: The biases (both $\bar{x}$ and s) are stronger when the clusters are not well separated ($\mu = \pm 0.5$). As the separation increases (e.g., $\mu = \pm 2$) the biases diminish. For $\mu = \pm 0.5$ the GMM estimates are substantially off, but closer than the K-means estimates.

	K-means				GMM		
μ	$\bar{x}_1$	$\bar{x}_2$	s_1	s_2	$\bar{x}_1$	$\bar{x}_2$	s_{pool}
± 0.5	-0.92	0.86	0.6667	0.6628	-0.60	0.554	0.8959
± 1	-1.15	1.18	0.8036	0.7807	-1.01	0.998	0.9665
± 2	-2.01	2.02	0.9645	0.9475	-1.99	2.00	0.9784

```
library(mclust)
makedat = function(mu, sigma=1, n=1000, seed=12345){
  set.seed(12345)
  data.frame(x=c(
    rnorm(n, -mu, sigma),
    rnorm(n, mu, sigma)
  ))
}


doone = function(mu, n=3000){
  dat = makedat(mu=mu, n=n)
  plot(density(dat$x, bw=.3))
  summary(kmeans(dat$x, 2, 20, 20))
  fit = Mclust(dat$x, G=2)
  cat("pro=(", fit$parameters$pro,
    "); means=(",fit$parameters$mean,
    "); SD=", fit$parameters$variance$sigmasq)
}

doone(2)
doone(1)
doone(.5)
```