Exploring heterogeneity in clinical trials with latent class analysis

Zhongheng Zhang[1], Abdallah Abarda[2], Ateka A. Contractor[3], Juan Wang[4], C. Mitchell Dayton[5]

[1]Department of Emergency Medicine, Sir Run-Run Shaw Hospital, Zhejiang University School of Medicine, Hangzhou 310016, China; [2]National School of Applied Science, Ibn Tofail University, Kenitra, Morocco; [3]Department of Psychology, University of North Texas, Denton, TX, USA; [4]Unit of Statistical Genetics, Kyoto University Graduate School of Medicine, Kyoto, Japan; [5]Department of Measurement, Statistics, and Evaluation, College of Education, University of Maryland, College Park, MD, USA

Correspondence to: Zhongheng Zhang. No. 3, East Qingchun Road, Hangzhou 310016, China. Email: zh_zhang1984@zju.edu.cn.

Abstract: Case-mix is common in clinical trials and treatment effect can vary across different subgroups. Conventionally, a subgroup analysis is performed by dividing the overall study population by one or two grouping variables. It is usually impossible to explore complex high-order intersections among confounding variables. Latent class analysis (LCA) provides a framework to identify latent classes by observed manifest variables. Distal clinical outcomes and treatment effect can be different across these classes. This paper provides a step-by-step tutorial on how to perform LCA with R. A simulated dataset is generated to illustrate the process. In the example, the classify-analyze approach is employed to explore the differential treatment effects on distal outcomes across latent classes.

Keywords: Latent class analysis (LCA); information criteria; heterogeneity; subgroup; classify-analyze

Submitted Nov 25, 2017. Accepted for publication Jan 03, 2018.
doi: 10.21037/atm.2018.01.24
View this article at: http://dx.doi.org/10.21037/atm.2018.01.24

Introduction

RCTs, as the gold standard trials, applicable into the assessment of effectiveness of treatments in clinic routines, aim to reduce the bias affected by confounding factors by means of double blind and randomization assignments design, so that the accuracy of assessment could be improved and the effectiveness of drugs could be interpreted more objectively. However, the quality of a clinical trial is often compromised by case mix or heterogeneity of the study population. Typically, the patient inclusionary and exclusionary criteria include clinically observed indicators such as the diagnosis, stage of disease, age (categorized into age groups), gender, medical history, and comorbidities. Although every effort has been made to purify the study population that the biological efficacy can be fully exploited (1), the heterogeneity induced by unmeasurable factors such as genomics and socioeconomic status (e.g., occupation, income, and education) cannot be fully addressed. As a result, even the most carefully selected patient population can exhibit remarkable heterogeneity in clinical trials. Thus, meaningful subgroups of patients based on endorsed response patterns need to be identified to maximize the beneficial effects of a given intervention. For example, the study population in a clinical trial may comprise of two sub-phenotypes wherein the studied effective intervention may benefit one sub-phenotype however not benefit or harm the other sub-phenotype. Additionally, the subgroups of patients may differ in the course, prognosis, and even comorbidity patterns of the disease/disorder of interest for a clinical trial; such information may also create differential effects for the assessed intervention.

Acute respiratory distress syndrome (ARDS) is a common disease in the intensive care unit and it can be diagnosed in the presence of hypoxia induced by bilateral lung inflammation. Although patients with ARDS have long been considered as a study population in clinical trials, this group of patients actually encompasses a heterogeneous population. Investigators have successfully identified sub-phenotypes of ARDS by using inflammatory biomarkers, and the results showed that these sub-phenotypes responded differently to fluid therapy (2,3).

Typically, subgroup analysis performed in clinical trials divides study population by one or two factors. For example, patients can be categorized into subgroups with or without diabetes. The complex relationship (e.g., interactions, higher-order intersections and etc.) between manifest (observed) variables cannot be adequately explored with conventional subgroup analysis. In situations when there are, for example, 6 manifest variables and each has 2 response levels, there will be a total of 2^6=64 possible response patterns. However, some of the response patterns may only be present in a handful of patients, or even not exist in the real-world setting (4).

A popular method to address patient heterogeneity is finite mixture models, which is a statistically sophisticated framework for identifying meaningful subgroups of patients that are not directly observable. Latent class analysis (LCA) and latent profile analysis (LPA) are special forms of the finite mixture models; which allows us to identify a finite number of latent subgroups and to explore how treatment effect varies across these subgroups (5). The former assesses categorical symptom indicators while the latter assesses continuous indicators for latent group classification (6). Such person-centered approaches transcend limitations imposed by diagnostic categories and classify patients into latent homogenous classes based on similar response patterns (6). Latent subgroups of patients are compared with reference to shape (qualitative differences) symptom levels (quantitative differences) (7). LCA/LPA is more robust and reliable compared to analytically similar cluster analyses because they account for measurement error and use objective criteria to determine the optimal class solution. Identification of such meaningful latent subgroups allows investigators to explore differential treatment effects across these subgroups (8). This article aims to provide a step-by-step tutorial on how to perform LCA in R. In clinical trials, we prefer LCA because continuous variables can be easily converted to categorical variables, but the reverse is not true.

Brief description of LCA

The LCA is based on probabilistic models to create classes or subgroups in a heterogeneous population. This model assumes the existence of unobservable classes which we can measure or observe the consequences or effects.

To describe the latent class model, we adopt the following notations: Let c be the latent class $c = 1,\ldots,C$ and v the manifest variable, $v=1,\ldots,V$. We denote s the response patterns or the outcome vector, $s = 1,\ldots,S$, where S represents a list of responses. Let $s(v)$ be the response levels of the variable v , $s(v)=1,\ldots,I_s$.

We denote P_s the probability of the outcome vector s which can be written as (9)

$$p_s = \sum_{c=1}^{c} p_{s,c}$$

where $p_{s,c}$ denotes the unobserved probabilities of falling simultaneously in the categories defined by vector s and the latent class c. The probability of outcome vector c conditional on latent class c is denoted by $p_{s/c}$. Assuming conditional independence, $p_{s,c}$ can be written as

$$p_{s,c} = \sum_{c=1}^{c} p_c \prod_{v=1}^{v} p_{v,s(v)/c}$$

The parameters p_c (the size of class) and $p_{v,s(v)/c}$ [probability of category $s(v)$ for variable v conditional on class c] are estimated by using the EM algorithm (10). This algorithm is made up of two important steps: expectation step denoted by E and the likelihood maximization step denoted by M. The first step consists on calculating the expectation of the log-likelihood assuming that we have the information about classes. The second step consists of the maximization of log-likelihood function.

The posterior probabilities that permit to affect individuals to the latent classes is given by:

$$p_{s/c} = \frac{p_c p_{s/c}}{\sum_{t-1}^{t=c} p_t p_{s/t}}$$

Dataset simulation

In this tutorial, an artificial dataset is generated by using the poLCA.simdata() function shipped with the poLCA package (11). We first install and load the package.

```
> install.packages("poLCA")
> library(poLCA)
```

Then, we proceed to generate a simulated dataset.

```
> set.seed(8)
> probs <- list(matrix(c(0.6,0.2,0.2, 0.6,0.3,0.1,
```



```
0.3,0.1,0.6 ),ncol=3,byrow=TRUE), # Y1
matrix(c(0.2,0.8, 0.7,0.3, 0.3,0.7 ),
ncol=2,byrow=TRUE), # Y2
matrix(c(0.3,0.6,0.1, 0.1,0.3,0.6, 0.3,0.6,0.1 ),
ncol=3,byrow=TRUE), # Y3
matrix(c(0.1,0.1,0.5,0.3, 0.5,0.3,0.1,0.1, 0.3,0.1,0.1,0.5),
ncol=4,byrow=TRUE), # Y4
matrix(c(0.1,0.2,0.7, 0.1,0.8,0.1, 0.8,0.1,0.1 ),
ncol=3,byrow=TRUE)) # Y5
> simdat <- poLCA.simdata(N=1000,probs,P
=c(0.2,0.3,0.5))
```

The *probs* object is a list of matrices with dimensions equal to the number of classes (the number of rows) by the number of responses (the number of column). Each matrix corresponds to one manifest variable (from Y1 to Y5), and each row contains the class-conditional outcome probabilities. Note that each row sums to one. For example, the manifest variable Y2 contains two responses and Y4 contains 4 responses. The ability to incorporate polytomous manifest variables is a feature of the poLCA package. Also note that all matrices have the same number of rows because it represents the number of classes. The poLCA.simdata() function generates an artificial dataset. The number of observations is 1000, and the prior class probabilities are 0.2, 0.3 and 0.5 for the three classes.

In many situations, it is of interest to investigate whether the treatment effect varies across latent classes. Thus, we also simulate a variable *trt* representing the treatment group and an *outcome* variable representing a binary clinical outcome.

```
#run together with the above simulation for reproduc-
ibility
> trt<-as.factor(sample(c("trt","ctrl"),replace=T,
size=1000))
> z <- 1 - as.numeric(trt)-simdat$trueclass+
 0.5*as.numeric(trt)*simdat$trueclass
> pr <- 1/(1+exp(-z))
> outcome <- rbinom(1000,1,pr)
> dat<-data.frame(simdat$dat,trt=trt,
outcome=outcome)
```

In the second line, a two-level factor variable *trt* was created with equal size in both levels. The object z is a linear predictor of a logistic regression model. The coefficient of each variable is arbitrarily assigned. The factor variable is converted to a numeric variable by the as.numeric() function. The linear predictor is then converted to probability by logit link function. Then, the outcome variable is generated by assuming a binomial distribution.

To choose the best number of classes

The correct selection of the number of latent classes represents a critical problem because it can significantly affect substantive interpretations (12). Indeed, an incorrect selection of latent classes can lead to an incorrect interpretation of the studied phenomenon. The definition of the number of classes from a population is commonly achieved by using a likelihood ratio test (LMR). This is often used to compare two models (nested models deriving from each other by adding or deleting terms) under the assumption that these two models correctly fit the data. When many models need to be compared, the risk of rejecting the null hypothesis when it is true increases substantially. A number of methods, including information criteria (13), parametric resampling, etc., have been proposed to choose the number of classes. Information criteria including Akaike Information Criterion (AIC) (14), Bayesian Information Criterion (BIC) (15), consistent Akaike Information Criterion (cAIC), adjusted Bayesian Information Criterion (aBIC) (16) are among the most practical methods and require much less computational effort than other methods such as parametric resampling. The AIC is generally valid for the small sample models, though it is not useful for determining the number of classes in general. Other proposed method to judge model fit include Lo-Mendell-Rubin adjusted likelihood ratio test (LMR) (17), Likelihood ratio/deviance statistic, Bootstrap likelihood ratio test (BLRT) (18), and entropy. Typically, several latent class models with various numbers of classes are fit, and their statistics of model fit are compared to choose the best one. Sometimes, the subject-matter knowledge should also be considered when considering the number of classes. Thus, a combination of fit indices (rather than sole reliance on one index), coupled with a consideration of theory and interpretability is recommended to determine the optimal latent class model (8,19). According to the recommended fit indices, the optimal class solution would have the lowest BIC values, lowest aBIC values, a significant LMR value, a

significant BLRT *p* value, relatively higher entropy values, and conceptual and interpretive meaning (8,20,21). When comparing a *K*-class model with a *K-1* class model, a significant LMR test indicates that the model with K classes is optimal (8).

The following loop generates a series of latent class models with one to five classes.

```
> f<-with(dat, cbind(Y1,Y2,Y3,Y4,Y5)~1)
> k=5
> for(i in 1:k){
 assign(paste("lc",i,sep=""),
 poLCA(f, dat, nclass=i, maxiter=3000,
 tol=1e-5, na.rm=FALSE,
 nrep=10, verbose=TRUE, calc.se=TRUE))
}
```

The first line creates a formula in the form of response~predictors. The manifest variables Y1 to Y5 are response variables that characterize the latent class. Note that only non-zero integer variables are allowed, negative or decimal values will return an error message. For continuous variables, users need to convert them into categorical variables. In the formula, no predictor of latent class is added, thus a numeral 1 is added to the right of the "~" symbol. The second line assigned a numeral 5 to k, indicating a maximum of 5 classes will be allowed in the following latent class models. In the for loop, the assign() function is employed to assign a value to a name. The values are a series of objects of the class *poLCA*, and the object names are different in each loop cycle. The poLCA() is the main function that it estimates latent class models for polytomous outcome variables. The first argument is a formula defined previously. The data argument is a data frame containing variables in the formula. After the loop, five latent class models are created in the environment with the names lc1, lc2, lc3, lc4 and lc5.

```
> tab.modfit<-data.frame(matrix(rep(999,7),nrow=1))
> names(tab.modfit)<-c("log-likelihood",
"resid. df","BIC",
 "aBIC","cAIC","likelihood-ratio","Entropy")
```

The above codes prepare to create a table containing statistics reflecting model fit. The first line creates a data frame with arbitrary values, but there should be 7 columns. Then the column names are assigned to represent the 7 most commonly used model fit statistics.

The entropy-based measures can be a poor tool for model selection, as stated by Collins LM that "*Latent class assignment error can increase simply as a function of the number of latent classes, so indices like entropy often decrease as the number of latent classes increases. In other words, class assignment can look better purely by chance in a two-latent-class model than in a comparable model with three or more latent classes.*" (22). However, since entropy is widely used in research practice, we illustrate how to compute entropy here. The poLCA.entropy() shipped with the poLCA package is able to calculate entropy when the number of response is the same for all manifest variables. However, when the numbers of response are not equal for all variables, the poLCA.entropy() function will report an error. Thus, we modified the function to incorporate circumstances when the number of response is unequal. In poLCA package, the entropy is defined as (11):

$$H = -\sum_c p_c \times \log(p_c),$$

where p_c is the share of the probability in the cth cell of the cross-classification table. The function can be rewritten as:

```
> entropy.poLCA<-function (lc)
{
 K.j <- sapply(lc$probs, ncol)
 if(length(unique(K.j))==1){
        fullcell <- expand.grid(data.frame(sapply(K.j,
seq, from = 1)))
 } else{
 fullcell <- expand.grid(sapply(K.j, seq, from = 1))
 }
 P.c <- poLCA.predcell(lc, fullcell)
 return(-sum(P.c * log(P.c), na.rm = TRUE))
}
```

More commonly, the mathematical equation for entropy is given by (23) (https://www.statmodel.com/download/relatinglca.pdf):

$$EN(p) = -\sum_{i=1}^{N}\sum_{c=1}^{C} p_{ic} \log\, p_{ic}$$

Where p_{ic} denotes the estimated posterior probability for individual i in class c. C is the number of classes and

N is the number of observations. The entropy equation is bounded from [0, ∞), with higher values indicated a larger amount of uncertainty in classification. Thus, the function for entropy can be written as follows:

```
> entropy<-function(lc){
        return(-sum(lc$posterior*log(lc$posterior),
 na.rm=T))
}
```

The "lc$posterior" extracts the posterior probabilities for all observations belonging to a class.

```
> lc3$posterior[1,]
[1] 0.09989867 0.08565670 0.81444463
```

The above code examines the posterior distributions of the first observation in the *lc3* model. It appears that this observation is most likely to belong to class 3 (e.g., with a posterior probability of 0.81). Other observations can be examined in the same way.

Relative entropy is a rescaled version of entropy by the following equation (23):

$$E = 1 - \frac{EN(p)}{N \times \log(J)}$$

where J is the number of classes. The R function for computation of relative entropy is as follows:

```
> relative.entropy<-function(lc){
        en<--sum(lc$posterior*
log(lc$posterior),na.rm=T)
e<-1-en/(nrow(lc$posterior)*log(ncol(lc$posterior)))
        return(e)
}
```

Then we proceed to calculate the model fit statistics for all fitted latent class models.

```
> for(i in 2:k){
tab.modfit<-rbind(tab.modfit,
 c(get(paste("lc",i,sep=""))$llik,
 get(paste("lc",i,sep=""))$resid.df,
 get(paste("lc",i,sep=""))$bic,
 (-2*get(paste("lc",i,sep=""))$llik) +
 ((log((get(paste("lc",i,sep=""))$N + 2)/24)) *
 get(paste("lc",i,sep=""))$npar),
 (-2*get(paste("lc",i,sep=""))$llik) +
 get(paste("lc",i,sep=""))$npar *
 (1 + log(get(paste("lc",i,sep=""))$N)),
get(paste("lc",i,sep=""))$Gsq,
relative.entropy(get(paste("lc",i,sep="")))
 ))
}
> tab.modfit<-round(tab.modfit[-1,],2)
> tab.modfit$Nclass<-2:k
```

The poLCA() function automatically calculates all statistics assessing model fit, and we just need to extract them from the returned poLCA objects and put them in a data frame. The last two lines round the returned values to 2 decimal places and remove the first row. Then, a new variable *Nclass* is added to denote the number of classes. The results can be viewed as follows:

```
> tab.modfit
```

	log-likelihood	resid.df	BIC	aBIC	cAIC	likelihood-ratio	Entropy	Nclass
2	-4952.31	194	10049.69	9982.99	10070.69	329.13	0.62	2
3	-4898.03	183	10017.10	9915.47	10049.10	220.57	0.64	3
4	-4889.70	172	10076.43	9939.86	10119.43	203.91	0.62	4
5	-4882.70	161	10138.41	9966.91	10192.41	189.90	0.68	5

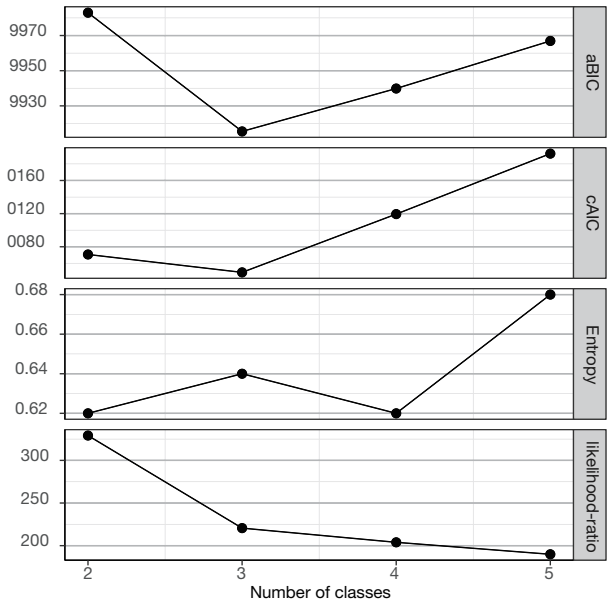

Figure 1 Elbow-Plot showing the parsimony and goodness-of-fit for models with varying number of classes.

The model fit can also be visualized to facilitate interpretation. First of all, we need to reformat the data frame *tab.modfit*.

```
> install.packages("forcats")
> library("forcats")
> tab.modfit$Nclass <-as.factor(tab.modfit$Nclass)
> results2<-tidyr::gather(tab.modfit,label,value,4:7)
> results2
```

	log-likelihood	resid.df	BIC	Nclass	label	value
1	-4952.31	194	10049.69	2	aBIC	9982.99
2	-4898.03	183	10017.10	3	aBIC	9915.47
3	-4889.70	172	10076.43	4	aBIC	9939.86
4	-4882.70	161	10138.41	5	aBIC	9966.91
5	-4952.31	194	10049.69	2	cAIC	10070.69
6	-4898.03	183	10017.10	3	cAIC	10049.10
7	-4889.70	172	10076.43	4	cAIC	10119.43
8	-4882.70	161	10138.41	5	cAIC	10192.41
9	-4952.31	194	10049.69	2	likelihood-ratio	329.13
10	-4898.03	183	10017.10	3	likelihood-ratio	220.57
11	-4889.70	172	10076.43	4	likelihood-ratio	203.91
12	-4882.70	161	10138.41	5	likelihood-ratio	189.90
13	-4952.31	194	10049.69	2	Entropy	0.62
14	-4898.03	183	10017.10	3	Entropy	0.64
15	-4889.70	172	10076.43	4	Entropy	0.62
16	-4882.70	161	10138.41	5	Entropy	0.68

The results2 data frame is a long format that the labels for model fit statistics are formatted in long style. The value column is the values for respective statistics. Then, this data frame can be passed to the ggplot() function (24).

```
> fit.plot<-ggplot(results2) +
 geom_point(aes(x=Nclass,y=value),size=3) +
 geom_line(aes(Nclass, value, group = 1)) +
 theme_bw()+
 labs(x = "Number of classes", y="", title = "") +
 facet_grid(label ~. ,scales = "free") +
 theme_bw(base_size = 16, base_family = "") +
 theme(panel.grid.major.x = element_blank() ,
 panel.grid.major.y = element_line(colour="grey",
size=0.5),
 legend.title = element_text(size = 16, face = 'bold'),
 axis.text = element_text(size = 16),
 axis.title = element_text(size = 16),
 legend.text= element_text(size=16),
 axis.line = element_line(colour = "black"))
> fit.plot
```

The result is a plot showing the changing values of model fit statistics by varying number of classes (*Figure 1*). It appears that latent class model with 3 classes has the smallest values in aBIC and cAIC. However, the relative entropy is not optimal with 3 classes (as mentioned above, entropy can be a poor tool for model selection). Although the 5-class model has a greater entropy value than the 3-class model, three of the 5 classes have very low population shares. Collectively, the 3-class model is the optimal model, taking into account of the combination of statistical fit indices, parsimony and interpretative value.

```
> lc5$P
[1] 0.06606815 0.09553452 0.26579377 0.49230406
0.08029950
> lc3$P
```

```
[1] 0.3888389 0.1610742 0.4500869
```

The lca_select() function

The lca_select() function makes it easier to choose the right model based on the information criteria (created by A.A.). In this function, we added more information criteria such as Hurvich and Tsai Criterion (HT) (25), Modified AIC (mAIC) (26), Hannan and Quinn Criteria (HQ) (27), Corrected Akaike Information Criterion (AICc) (28). The function is defined as follows:

```
> lca_select <- function(f,dat,nb_var,k,nbr_repet)
# f is the selected variables
# dat is the data
# nb_var is the number of selected variables
# k is the number of latent class generated
# nbr_repet is the number of repetition to
# reach the convergence of EM algorithm
{
N=length(t(dat[,1]))
tab.modfit<-data.frame(matrix(rep(999,12),nrow=1))
names(tab.modfit)<-c("Df","Gsq","Llik","AIC",
"mAIC","AICc","HT",
 "cAIC","AICc","BIC","aBIC","HQ")
for(i in 2:k){
  assign(paste("lc",i,sep=""),
  poLCA(f, dat, nclass=i, maxiter=3000,
  tol=1e-5, na.rm=FALSE,
  nrep=nbr_repet, verbose=TRUE, calc.se=TRUE))
tab.modfit<-rbind(tab.modfit, c(
get(paste("lc",i,sep=""))$resid.df, #df
get(paste("lc",i,sep=""))$Gsq, #gsq
get(paste("lc",i,sep=""))$llik, #llik
-2*get(paste("lc",i,sep=""))$llik+
 2*get(paste("lc",i,sep=""))$npar, #AIC
-2*get(paste("lc",i,sep=""))$llik+
 3*get(paste("lc",i,sep=""))$npar, #AIC3
-2*get(paste("lc",i,sep=""))$llik+
 2*get(paste("lc",i,sep=""))$npar+
 (2*get(paste("lc",i,sep=""))$npar*get(paste("lc",
 i,sep=""))$npar+1)/(N-get(
 paste("lc",i,sep=""))$npar-1), #AICC
-2*get(paste("lc",i,sep=""))$llik+
2*get(paste("lc",i,sep=""))$npar+
(2*(get(paste("lc",i,sep=""))$npar+1)*(get(paste("lc",
 i,sep=""))$npar+2))/(N-get(
 paste("lc",i,sep=""))$npar-2), #HT
-2*get(paste("lc",i,sep=""))$llik+get(
paste("lc",i,sep=""))$npar*(log(N)+1), #CAIC
-2*get(paste("lc",i,sep=""))$llik+
2*get(paste("lc",i,sep=""))$npar+
(2*get(paste("lc",i,sep=""))$npar*get(paste("lc",
 i,sep=""))$npar+1)/(N-get(paste("lc",i,sep=""))$
npar-1)+
N*log(N/(N-get(paste("lc",i,sep=""))$npar-1)), #CAIU
-2*get(paste("lc",i,sep=""))$llik+
get(paste("lc",i,sep=""))$npar*log(N), #BIC
-2*get(paste("lc",i,sep=""))$llik+
get(paste("lc",i,sep=""))$npar*log((N+2)/24), #ABIC
-2*get(paste("lc",i,sep=""))$llik+
2*get(paste("lc",i,sep=""))$npar*log(log(N)) #HQ
))
}
tab.modfit<-round(tab.modfit[-1,],2)
tab.modfit$Nclass<-2:k
print(tab.modfit)
plot(tab.modfit$AIC,type="l",lty=2,lwd=1,
xaxt="n",
ylim=c(min(tab.modfit$AIC,tab.modfit$aBIC)-
100,round(max(tab.modfit$BIC,tab.modfit$aBIC))+100),
col="black",
xlab="Number of classes",ylab="Information criteria",
main="Comparison of information criteria to choose the
number of classes")
axis(1,at=1:length(tab.modfit$Nclass),
labels=tab.modfit$Nclass)
lines(tab.modfit$AIC,col="black",type="l",lty=2,lwd=1)
lines(tab.modfit$BIC,col="red",type="l",lty=2,lwd=1)
lines(tab.modfit$aBIC,col="green",type="l",lty=2,
lwd=1)
lines(tab.modfit$cAIC,col="orange",type="l",lty=2,
lwd=1)
lines(tab.modfit$HQ,col="blue",type="l",lty=2,lwd=1)
#lines(dd$caiu,col="purple",type="l",lty=7,lwd=2)
#lines(dd$bica,col="grey",type="l",lty=8,lwd=2)
```

Table 1 Information criteria for the choice of the number of classes

Df	Gsq	Llik	AIC	mAIC	AICc	HT	cAIC	AICc	BIC	aBIC	HQ	N class
194	329.13	−4,952.31	9,946.63	9,967.63	9,947.53	9,947.66	1,0070.69	9,969.77	10,049.69	9,982.99	9,985.80	2
183	220.57	−4,898.03	9,860.06	9,892.06	9,862.18	9,862.38	10,049.10	9,895.73	10,017.10	9,915.47	9,919.75	3
172	203.91	−4,889.70	9,865.40	9,908.40	9,869.27	9,869.54	10,119.43	9,914.26	10,076.43	9,939.86	9,945.61	4
161	189.90	−4,882.70	9,873.39	9,927.39	9,879.57	9,879.92	10,192.41	9,936.14	10,138.41	9,966.91	9,974.12	5
150	173.22	−4,874.36	9,878.71	9,943.71	9,887.76	9,888.19	10,262.71	9,956.04	10,197.71	9,991.27	9,999.95	6
139	159.98	−4,867.73	9,887.47	9,963.47	9,899.98	9,900.50	10,336.46	9,980.11	10,260.46	10,019.08	10,029.23	7
128	143.28	−4,859.38	9,892.77	9,979.77	9,909.37	9,909.96	10,406.74	10,001.48	10,319.74	10,043.43	10,055.05	8

Df, degree of freedom; Gsq, Likelihood ratio/deviance statistic; Llik, maximum log-likelihood; AIC, Akaike Information Criterion; BIC, Bayesian Information Criterion; cAIC, consistent Akaike Information Criterion; HT, Hurvich and Tsai Criterion; AIC3, Modified AIC; aBIC, adjusted Bayesian Information Criterion; HQ, Hannan and Quinn Criteria; AICc, Corrected Akaike Information Criterion; Nclass, number of classes.

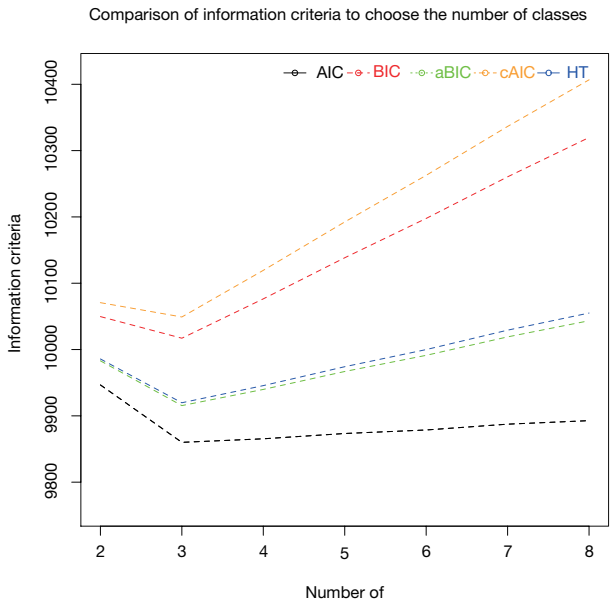

Figure 2 Comparison of information criteria to choose the number of classes.

```
#lines(dd$hq,col="pink",type="l",lty=9,lwd=2)
legend("topright",legend=c("AIC","BIC","aBIC","cAIC",
"HT"),
pch=21,col=c("black","red","green","orange","blue"),
ncol=5,bty="n",cex=0.8,lty=1:9,
text.col=c("black","red","green","orange","blue"),
inset=0.01)
}
```

Then we proceed to compute and display model fit statistics with the following code.

```
> lca_select(with(dat, cbind(Y1,Y2,Y3,Y4,Y5)~1),
 dat, k=8, nbr_repet=10)
```

In the example, the number of classes to be chosen is 8 and the information criteria as well as other statistics is shown in *Table 1*. Furthermore, the information criteria to choose the number of classes are displayed in *Figure 2*. It appears that the 3-class model has the lowest values on all information criteria, and it is reasonable to choose the 3-class model.

Model visualization

The generic function plot() can be applied directly to the poLCA object to visualize the latent class model.

```
> plot(lc3)
```

The result is shown in *Figure 3*. The overall population is divided into 3 classes by the five manifest variables. The graphic presents probabilities of categories $s(v)$ (response value) for variable v conditional on class c. For example, the class 2 is characterized by a large probability of response value 2 in Y2 and 3 in Y5. These characteristics can be interpreted with subject-matter knowledge.

The latent class model can also be visualized in 2D plot.

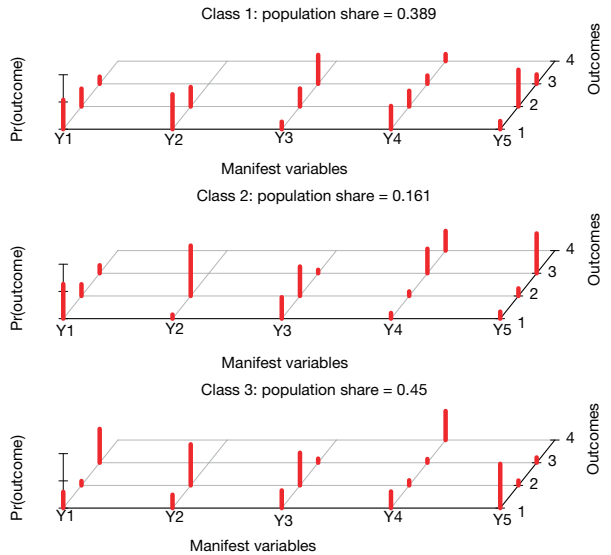

Figure 3 Posterior probability of manifest variable responses across classes.

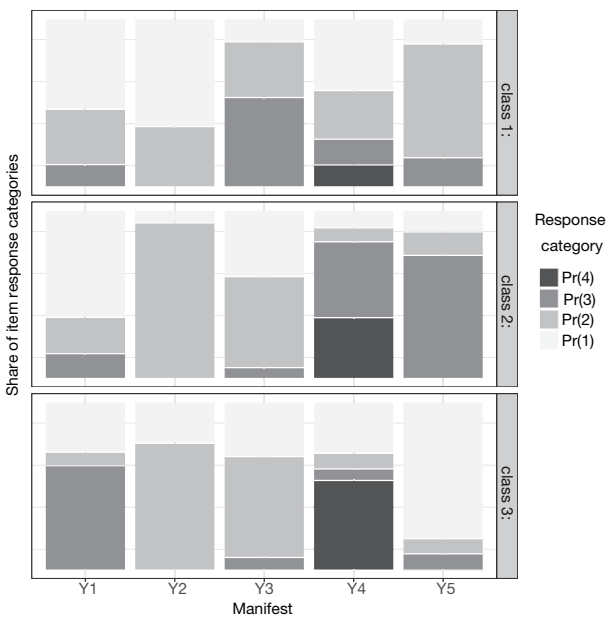

Figure 4 2-dimensional plot showing the posterior probability of manifest variable responses across classes.

```
> lcmodel <- reshape2::melt(lc3$probs, level=2)
> zp1 <- ggplot(lcmodel,aes(x = L2, y = value, fill =
Var2))+
geom_bar(stat = "identity", position = "stack")+
facet_grid(Var1 ~ .)+
scale_fill_brewer(type="seq", palette="Greys") +
theme_bw()+
labs(x = "Manifest variables",
 y="Share of item response categories",
 fill ="Response
 category")+
theme(axis.text.y=element_blank(),
 axis.ticks.y=element_blank(),
 panel.grid.major.y=element_blank())+
guides(fill = guide_legend(reverse=TRUE))
> print(zp1)
```

The *Figure 4* gives the same information as that of the *Figure 3*, but the responses are displayed in 2-D format.

Differential treatment effects across latent classes

The 3-class model is considered as the best fitted model and the 3 classes can be considered as subgroups of the overall study population. The next task is to investigate whether the treatment effects vary across latent classes. As mentioned earlier, we have generated the variable *trt* and *outcome*, corresponding to the treatment group and the binary outcome. Lanza ST and colleagues described two methods to examine differential treatment effects (4): (I) a classify-analyze approach involving logistic regression model, and (II) a model-based approach. The classify-analyze approach involves two steps, "classify" step and the "analyze" step. The first step is to obtain the posterior probability of each patient and assign them to the latent class with the maximum probability, which is based on the maximum-probability assignment rule (29,30). Next, the "analyze" step involves building a logistic regression model with the outcome variable as the dependent variable. The treatment and latent class membership are included in an interaction term (31). This is the conventional regression/analysis of variance approach to test differential treatment effects across subgroups. The model-based approach involves multiple group LCA which is not currently implemented in the poLCA package.

```
> mod<-glm(outcome~trt*as.factor(lc3$predclass),
 family="binomial")
```

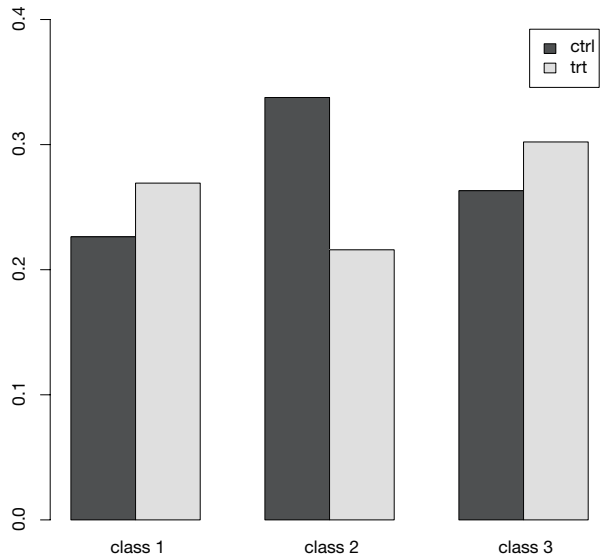

Figure 5 Proportion of treatment and control group patients in each class reporting outcome 1.

```
> round(summary(mod)$coefficients,3)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.229	0.173	-7.090	0.000
trttrt	0.231	0.241	0.958	0.338
as.factor(lc3$predclass)2	0.556	0.297	1.871	0.061
as.factor(lc3$predclass)3	0.200	0.230	0.870	0.384
trttrt:as.factor(lc3$predclass)2	-0.847	0.428	-1.978	0.048
trttrt:as.factor(lc3$predclass)3	-0.038	0.317	-0.121	0.904

The logistic regression model is fitted with generalized linear model (32). The *trt* and latent class group membership variables are included as an interaction term. The "lc3$predclass" extracts the latent class membership for each individual patient, and it is converted to a factor variable because the class membership values 1, 2 and 3 have no numeric relationship. In the model, class 1 is regarded as the reference class, against which other classes are compared. The result shows that the treatment effect is not significantly different for the class 1 *vs.* class 2 comparison (β=2.031, SE=0.241, P=0.338), but the effect is significantly different for the class 1 *vs.* class 2 comparison (β=0.847, SE=0.428, P=0.048). The treatment seems to be more effective in reducing adverse outcome for patients in class 2. For subject-matter audience, it is convenient to transform the regression coefficients to odds ratios. Patients in the class 2 group assigned to treatment group are $e^{0.231-0.847}$=0.54 times as likely to report outcome 1 compared to the control group. In other words, the treatment results in a 50% risk reduction for patients in class 2.

```
> dat$predclass<-lc3$predclass
> prop<-rbind(ctrl=prop.table(table(dat[dat$trt=="ctrl",
]$predclass,
dat[dat$trt=="ctrl",]$outcome),1)[4:6],
trt=prop.table(table(dat[dat$trt=="trt",]$predclass,
dat[dat$trt=="trt",]$outcome),1)[4:6])
> colnames(prop)<-c('class 1',"class 2","class 3")
> barplot(prop,beside =T,
 legend.text=c('ctrl',"trt"),
 ylim = c(0,0.4))
```

The first line attaches the class membership variable *predclass* to the original data frame. Then a data frame containing proportions is generated, which can be passed to the barplot() function. The result shows that the treatment is able to reduce the risk of outcome 1 by nearly 50% (*Figure 5*), whereas such a beneficial treatment effect is not observed in the other two classes.

Acknowledgements

Funding: The study was supported by Zhejiang provincial natural science foundation of China (LGF18H150005).

Footnote

Conflicts of Interest: The authors have no conflicts of interest to declare.

References

1. Nallamothu BK, Hayward RA, Bates ER. Beyond the randomized clinical trial: the role of effectiveness studies in evaluating cardiovascular therapies. Circulation 2008;118:1294-303.
2. Calfee CS, Delucchi K, Parsons PE, et al. Subphenotypes in acute respiratory distress syndrome: latent class analysis of data from two randomised controlled trials. Lancet Respir Med 2014;2:611-20.
3. Famous KR, Delucchi K, Ware LB, et al. Acute respiratory distress syndrome subphenotypes respond differently to

randomized fluid management strategy. Am J Respir Crit Care Med 2017;195:331-8.
4. Lanza ST, Rhoades BL. Latent class analysis: an alternative perspective on subgroup analysis in prevention and treatment. Prev Sci 2013;14:157-68.
5. Rindskopf D, Rindskopf W. The value of latent class analysis in medical diagnosis. Stat Med 1986;5:21-7.
6. McCutcheon A. Latent Class Analysis. Newbury Park, CA: SAGE Publications, 1987.
7. Nugent NR, Koenen KC, Bradley B. Heterogeneity of posttraumatic stress symptoms in a highly traumatized low income, urban, African American sample. J Psychiatr Res 2012;46:1576-83.
8. Nylund KL, Asparouhov T, Muthén BO. Deciding on the number of classes in latent class analysis and growth mixture modeling: a monte carlo simulation study. Struct Equ Modeling 2007;14:535-69.
9. Mooijaart A, van der Heijden PG. The EM algorithm for latent class analysis with equality constraints. Psychometrika 1992;57:261-9.
10. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 1977;39:1-38.
11. Linzer DA, Lewis JB. poLCA: An RPackage for Polytomous Variable Latent Class Analysis. Journal of Statistical Software 2011;42:1-29.
12. Yang CC. Evaluating latent class analysis models in qualitative phenotype identification. Computational Statistics & Data Analysis 2006;50:1090-104.
13. Lin TH, Dayton CM. Model Selection Information Criteria for Non-Nested Latent Class Models. Journal of Educational and Behavioral Statistics 1997;22:249-64.
14. Akaike H. Factor analysis and AIC. In: Regression modeling strategies. New York, NY: Springer New York; 1998:371-86.
15. Schwarz G. Estimating the Dimension of a Model. The Annals of Statistics 1978;6:461-4.
16. Sclove SL. Application of model-selection criteria to some problems in multivariate analysis. Psychometrika 1987;52:333-43.
17. Lo Y. Testing the number of components in a normal mixture. Biometrika 2001;88:767-78.
18. McLachlan G, Peel D. Finite Mixture Models. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2005.
19. Tein JY, Coxe S, Cham H. Statistical power to detect the correct number of classes in latent profile analysis. Struct Equ Modeling 2013;20:640-57.
20. Nylund K, Bellmore A, Nishina A, et al. Subtypes, severity, and structural stability of peer victimization: what does latent class analysis say? Child Dev 2007;78:1706-22.
21. DiStefano C, Kamphaus RW. Investigating Subtypes of Child Development. Educational and Psychological Measurement 2016;66:778-94.
22. Collins LM, Lanza ST. Latent class and latent transition analysis: with applications in the social, behavioral, and health sciences. In: Lanza ST. editor. Latent class and latent transition analysis: with applications in the social, behavioral, and health sciences (wiley series in probability and statistics). Hoboken, NJ: John Wiley & Sons, Inc., 2010:295.
23. Jedidi K, Ramaswamy V, Desarbo WS. A maximum likelihood method for latent class regression involving a censored dependent variable. Psychometrika 1993;58:375-94.
24. Ito K, Murphy D. Application of ggplot2 to Pharmacometric Graphics. CPT Pharmacometrics Syst Pharmacol 2013;2:e79-16.
25. Hurvich CM, Tsai CL. Regression and time series model selection in small samples. Biometrika 1989;76:297-307.
26. Andrews RL, Currim IS. A comparison of segment retention criteria for finite mixture logit models. Journal of Marketing Research 2003;40:235-43.
27. Hannan EJ, Quinn BG. The Determination of the Order of an Autoregression. Journal of the Royal Statistical Society 1979;41:190-5.
28. Sugiura N. Further analysts of the data by akaike' s information criterion and the finite corrections. Communications in Statistics - Theory and Methods 2007;7:13-26.
29. Goodman LA. On the Assignment of Individuals to Latent Classes. Sociological Methodology 2016;37:1-22.
30. Bakk Z, Tekle FB, Vermunt JK. Estimating the Association between Latent Class Membership and External Variables Using Bias-adjusted Three-step Approaches. Sociological Methodology 2013;43:272-311.
31. Bray BC, Lanza ST, Tan X. Eliminating Bias in Classify-Analyze Approaches for Latent Class Analysis. Struct Equ Modeling 2015;22:1-11.
32. Zhang Z. Model building strategy for logistic regression: purposeful selection. Ann Transl Med 2016;4:111-1.

Cite this article as: Zhang Z, Abarda A, Contractor AA, Wang J, Dayton CM. Exploring heterogeneity in clinical trials with latent class analysis. Ann Transl Med 2018;6(7):119. doi: 10.21037/atm.2018.01.24

