2019-01-29- LCA data modeling Seth-Josh

```
knitr::opts_chunk$set(echo = TRUE, cache = FALSE, message = FALSE, warning = FALSE, echo = TRUE)
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60),  # For code
                      width = 60)  # For output

options(cli.width = 60)  # For tidyverse loading messages
set.seed(20180925)
knitr::clean_cache()
```

```
## NULL
```

1. Loading, setting up

```
library(tidyverse)
library(poLCA)
library(readxl)
```

Getting data from Google Sheets

```
library(googlesheets)
library(readr)

g <- gs_title("Observations_segment_Units_1-7_2013-14-with-duplicates-identified")
d <- gs_read(g, col_types =
                 readr::cols(
                     `ClassObservation::Observer` = col_character(),
                     `ClassObservation::ObsNickname` = col_double(),
                     `Teacher::TeacherID` = col_double(),
                     `Teacher::First Name` = col_character(),
                     `Teacher::Last Name` = col_character(),
                     `Teacher::Condition` = col_character(),
                     `ClassObservation::Unit` = col_double(),
                     `ClassObservation::Date` = col_datetime(format = ""),
                     Notes = col_character(),
                     ObsNN = col_double(),
                     SegNum = col_double(),
                     `Segment::StartStamp` = col_datetime(format = ""),
                     `Segment::EndStamp` = col_datetime(format = ""),
                     fWhole = col_double(),
                     fGroups = col_double(),
                     fSeat = col_double(),
                     sInvented = col_double(),
                     sConceptual = col_double(),
                     sProcedural = col_double(),
                     sEngagement = col_character(),
                     tInitSelect = col_double(),
                     tCompare = col_double(),
                     tDiscussQ = col_double(),
                     tPressExplain = col_double(),
                     tConnectOthers = col_double(),
                     tConnectBigIdeas = col_double(),
                     tConventional = col_double(),
                     tProcedural = col_double(),
                     iPrecision = col_double(),
                     iCenter = col_double(),
                     iDIsplay = col_double(),
                     iOther = col_double(),
                     iOrder = col_double(),
                     iScale = col_double(),
                     iGrouping = col_double(),
                     iShape = col_double(),
                     iShow = col_double(),
                     iHide = col_double(),
                     iMode = col_double(),
                     iMedian = col_double(),
                     iMean = col_double(),
                     iRange = col_double(),
                     iCenterClump = col_double(),
                     iDeviation = col_double(),
                     iReplicability = col_double(),
                     iGeneralizability = col_double(),
                     iLinkVisDist = col_double(),
                     iLinkImagDist = col_double(),
                     ITheoreticalProb = col_double(),
                     IEmpiricalProb = col_double(),
                     IOdds = col_logical(),
                     ISampleSize = col_double(),
                     ISamplingDistrib = col_double(),
                     ICenterStats = col_double(),
                     IVariabilityStats = col_double(),
                     `Segment::iIntelligibility` = col_double(),
                     `Segment::iModelFit` = col_double(),
                     `Segment::iDistribution` = col_double(),
                     `Segment::iRandomComponents` = col_double(),
                     `Segment::iNonRandomComponents` = col_double(),
                     `Segment::iMedianDistr` = col_double(),
                     `Segment::iIQRDistr` = col_logical(),
                     `Segment::iNewMedian` = col_double(),
                     `Segment::iNewIQR` = col_logical(),
                     `Segment::iRegions` = col_double(),
                     `Segment::iQuantRegions` = col_double(),
                     number_of_segments = col_double(),
                     `Duplicate Condition` = col_character()
                 ))

d <- dplyr::rename(d, condition = `Teacher::Condition`)

d <- d %>%
    mutate(condition = ifelse(str_detect(condition, "2"), 0,
                              ifelse(str_detect(condition, "1"), 1, NA)))

library(readxl)
u <- read_xlsx("Observations_summary_Units_1-7_2012-13-mod.xlsx")

g1 <- gs_title("Observations_segment_Units_1-7_2012-13-with-duplicates-identified")
d1 <- gs_read(g1)
d1 <- rename(d1, Teacher_ID = handl)
#d1 <- unite(d1, Teacher, `Teacher::First Name`, `Teacher::Last Name`, sep = " ")
d1 <- d1 %>% left_join(u, by = "Teacher_ID")
d1 <- rename(d1, condition = Group)
```

```
add_one <- function(x) {
    x + 1
}

ds <- d %>%
    dplyr::select(sInvented, sProcedural, sConceptual, tInitSelect, tCompare, tDiscussQ, tConnectBigIdea
    map_df(replace_na, 0) %>%
    modify_at(c(1:9), add_one) %>%
    mutate(groups = case_when(
        fGroups == 1 ~ "small_groups",
        fSeat == 1 ~ "seat",
        fWhole == 1 ~ "whole"
    )) %>%
    dplyr::select(-fGroups, -fSeat, -fWhole)

ds1 <- d1 %>%
    dplyr::select(sInvented, sProcedural, sConceptual, tInitSelect, tCompare, tDiscussQ, tConnectBigIdea
    map_df(replace_na, 0) %>%
    modify_at(c(1:9), add_one) %>%
    mutate(groups = case_when(
        fGroups == 1 ~ "small_groups",
        fSeat == 1 ~ "seat",
        fWhole == 1 ~ "whole"
    )) %>%
    dplyr::select(-fGroups, -fSeat, -fWhole)

dd <- bind_rows(ds, ds1)

dds <- filter(dd, `Duplicate Condition` != "D" & `Duplicate Condition` != "d")
```

3. Choosing the number of classes/profiles

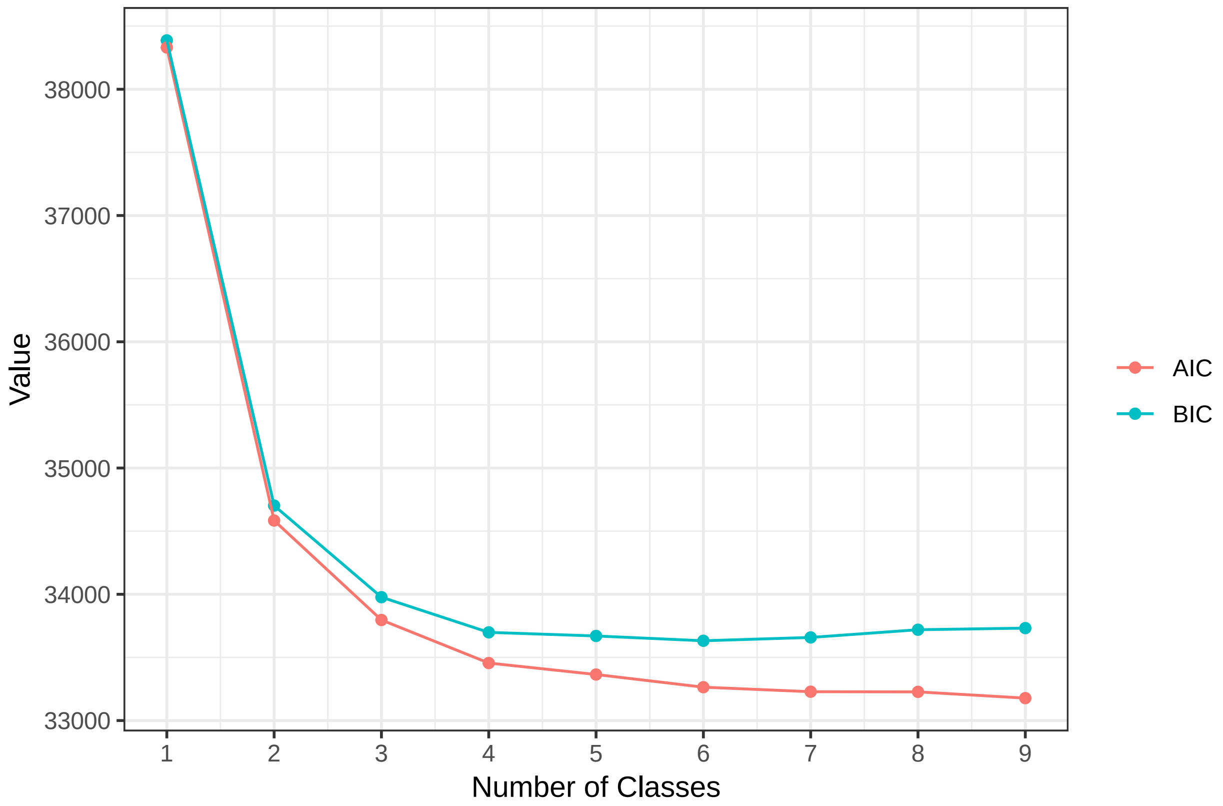
Using latent class analysis through the **poLCA** R package.

```
set.seed(20180925)

f <- cbind(sInvented, sProcedural, sConceptual, tInitSelect, tCompare, tDiscussQ, tConnectBigIdeas, tCon
#2341
```

```
od <- map(1:9, poLCA, formula = f, data = dds, maxiter = 5000, verbose = FALSE, graphs = FALSE) %>%
    map_df(broom::glance)
```

```
od %>%
    mutate(n_classes = 1:9) %>%
    gather(key, val, BIC, AIC) %>%
    ggplot(aes(x = n_classes, y = val, color = key, group = key)) +
    geom_point() +
    geom_line() +
    scale_x_continuous(breaks = 1:9, labels = 1:9) +
    theme_bw() +
    labs(caption = "Lower values of the AIC & BIC suggest preferred model(s); generally, BIC is more co
    xlab("Number of Classes") +
    ylab("Value") +
    scale_color_discrete("")
```



wer values of the AIC & BIC suggest preferred model(s); generally, BIC is more conservative than AIC

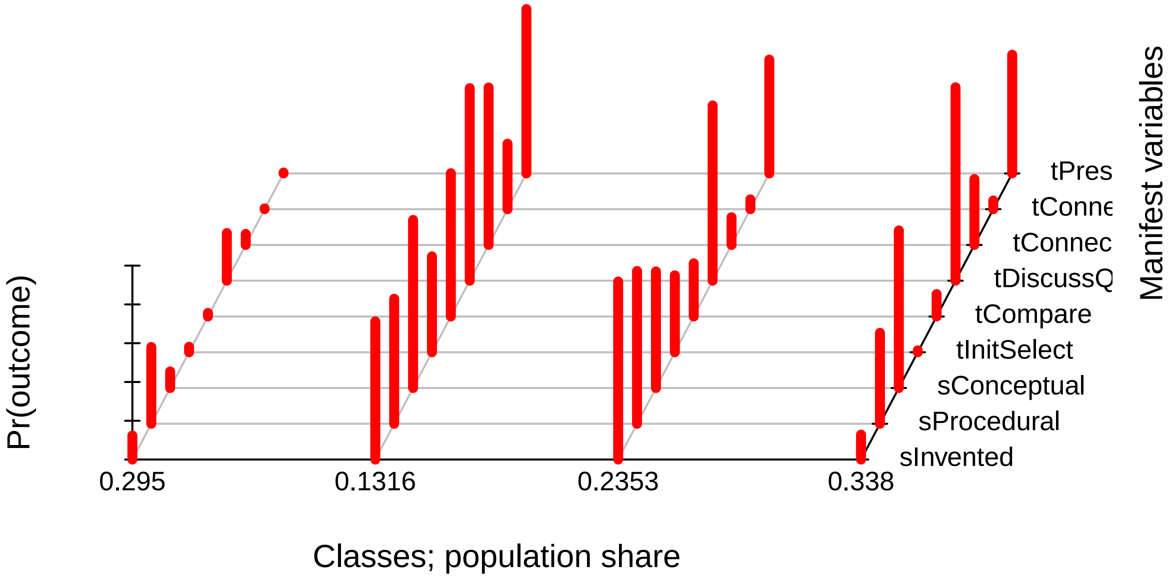
Based on these fit statistics, a three or four class solution seems to exhibit the best fit, though a three-class solution may also be suitable; for comparison, a two-class solution is also explored.

4. Examining 2, 3, 4, and 5 class solutions

```
set.seed(20180925)
```

```
f <- cbind(sInvented, sProcedural, sConceptual, tInitSelect, tCompare, tDiscussQ, tConnectBigIdeas, tCon

#m2 <- poLCA(f, dds, nclass = 2, maxiter = 10000, graphs = TRUE)
#m3 <- poLCA(f, dds, nclass = 3, maxiter = 10000, graphs = TRUE)
m4 <- poLCA(f, dds, nclass = 4, maxiter = 10000, graphs = TRUE)
```



```
## Conditional item response (column) probabilities,
##  by outcome variable, for each class (row)
##
## $sInvented
##            Pr(1)  Pr(2)
## class 1:  0.8761 0.1239
## class 2:  0.2874 0.7126
## class 3:  0.0818 0.9182
## class 4:  0.8716 0.1284
##
## $sProcedural
##            Pr(1)  Pr(2)
## class 1:  0.6039 0.3961
## class 2:  0.3553 0.6447
## class 3:  0.2120 0.7880
## class 4:  0.5310 0.4690
##
## $sConceptual
##            Pr(1)  Pr(2)
## class 1:  0.9144 0.0856
## class 2:  0.1331 0.8669
## class 3:  0.3986 0.6014
## class 4:  0.1874 0.8126
##
## $tInitSelect
##            Pr(1)  Pr(2)
## class 1:  0.9720 0.0280
## class 2:  0.5053 0.4947
## class 3:  0.6033 0.3967
## class 4:  0.9903 0.0097
##
## $tCompare
##            Pr(1)  Pr(2)
## class 1:  0.9814 0.0186
## class 2:  0.2609 0.7391
## class 3:  0.7255 0.2745
## class 4:  0.8842 0.1158
##
## $tDiscussQ
##            Pr(1)  Pr(2)
## class 1:  0.7540 0.2460
## class 2:  0.0066 0.9934
## class 3:  0.0959 0.9041
## class 4:  0.0025 0.9975
##
## $tConnectBigIdeas
##            Pr(1)  Pr(2)
## class 1:  0.9428 0.0572
## class 2:  0.1879 0.8121
## class 3:  0.8569 0.1431
## class 4:  0.6602 0.3398
##
## $tConnectOthers
##            Pr(1)  Pr(2)
## class 1:  0.9959 0.0041
## class 2:  0.6622 0.3378
## class 3:  0.9493 0.0507
## class 4:  0.9552 0.0448
##
## $tPressExplain
##            Pr(1)  Pr(2)
## class 1:  0.9956 0.0044
## class 2:  0.1523 0.8477
## class 3:  0.4128 0.5872
## class 4:  0.3876 0.6124
##
## Estimated class population shares
##  0.295 0.1316 0.2353 0.338
##
## Predicted class memberships (by modal posterior prob.)
##  0.2963 0.1164 0.2443 0.3429
##
## =========================================================
## Fit for 4 latent classes:
## =========================================================
## number of observations: 3753
## number of estimated parameters: 39
## residual degrees of freedom: 472
## maximum log-likelihood: -16688.75
##
## AIC(4): 33455.5
## BIC(4): 33698.49
## G^2(4): 832.2073 (Likelihood ratio/deviance statistic)
## X^2(4): 1870.21 (Chi-square goodness of fit)
##
```

```
#m5 <- poLCA(f, dds, nclass = 5, maxiter = 10000, graphs = TRUE)
```

```
data.frame(dds, class = m4$predclass) %>%
    dplyr::select(class, condition) %>%
    count(class, condition)
```

5. Examining predictors of the 4-class solution - does not work well for 3 class solution

Moving forward with four-class solution

```
post_probs <- m4$posterior %>% as.data.frame() %>% setNames(paste0("C", 1:4, "_prob"))
df <- bind_cols(dds, post_probs)
df$class <- m4$predclass
df <- df %>% dplyr::select(-`Duplicate Condition`) %>% mutate_if(is.numeric, round, 3)
write_csv(df, "2019-02-10-data-with-class-probs.csv")
```

Plots

```
t <- df %>%
    arrange(teacher, unit, seg_num) %>%
    group_by(unit) %>%
    summarize(max_seg_num = max(seg_num),
              max_unit = max(unit))

the_seqqer <- function(x) {
    seq(1, t$max_seg_num[x])
}

l <- list()
for (i in seq(t$max_seg_num)) {
    l[[i]] <- seq(1, t$max_seg_num[i])
}
```

```r
t$seq_l <- l

dtm <- dplyr::select(df, teacher, condition) %>% distinct() %>% arrange(teacher)

df$class <- as.factor(df$class)

df$class<- forcats::fct_recode(df$class,
                               `Low Activity` = "1",
                               `Inventing & Connecting` = "2",
                               `Inventing & Discussing` = "3",
                               `Discussing Ideas` = "4")

teacher_ID = dtm$teacher
condition = dtm$condition
map2(teacher_ID, condition, f, df)
```

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
##
## [[7]]
## NULL
##
## [[8]]
## NULL
##
## [[9]]
## NULL
##
## [[10]]
## NULL
##
## [[11]]
## NULL
##
## [[12]]
## NULL
##
## [[13]]
## NULL
##
## [[14]]
## NULL
##
## [[15]]
## NULL
##
## [[16]]
## NULL
##
## [[17]]
## NULL
##
## [[18]]
## NULL
##
## [[19]]
## NULL
##
## [[20]]
## NULL
##
## [[21]]
## NULL
##
## [[22]]
## NULL
##
## [[23]]
## NULL
##
## [[24]]
## NULL
##
## [[25]]
## NULL
##
## [[26]]
## NULL
##
## [[27]]
## NULL
##
## [[28]]
## NULL
##
## [[29]]
## NULL
##
## [[30]]
## NULL
##
## [[31]]
## NULL
##
## [[32]]
## NULL
##
## [[33]]
## NULL
##
## [[34]]
## NULL
##
## [[35]]
## NULL
##
## [[36]]
## NULL
##
## [[37]]
## NULL
##
## [[38]]
## NULL
##
## [[39]]
## NULL
##
## [[40]]
## NULL
##
## [[41]]
## NULL
##
## [[42]]
## NULL
##
## [[43]]
## NULL
##
## [[44]]
## NULL
##
## [[45]]
## NULL
##
## [[46]]
## NULL
##
## [[47]]
## NULL
##
## [[48]]
## NULL
##
## [[49]]
## NULL
##
## [[50]]
## NULL
##
## [[51]]
## NULL
##
## [[52]]
## NULL
##
## [[53]]
## NULL
##
## [[54]]
## NULL
##
## [[55]]
## NULL
##
## [[56]]
## NULL
##
## [[57]]
## NULL
##
## [[58]]
## NULL
##
## [[59]]
## NULL
##
## [[60]]
## NULL
##
## [[61]]
## NULL
##
## [[62]]
## NULL
##
## [[63]]
## NULL
##
## [[64]]
## NULL
##
## [[65]]
## NULL
##
## [[66]]
## NULL
##
## [[67]]
## NULL
##
## [[68]]
## NULL
##
## [[69]]
## NULL
##
## [[70]]
## NULL
##
## [[71]]
## NULL
##
## [[72]]
## NULL
##
## [[73]]
## NULL
##
## [[74]]
## NULL
##
## [[75]]
## NULL
##
## [[76]]
## NULL
##
## [[77]]
## NULL
```

Analysis

```
dm1 <- df %>% count(class, condition) %>%
    spread(condition, n) %>%
    mutate(`0` = replace_na(`0`, 0))
names_dm1 <- dm1$class
mat1 <- as.matrix(dm1[, -1])
cs1 <- chisq.test(mat1)
write_csv(dm1, "tab1.csv")
write_csv(as.data.frame(cs1$stdres), "mat1.csv")
#clipr::write_clip(cs1$stdres)

dm2 <- df %>% count(class, groups) %>%
    spread(groups, n)
names_dm2 <- dm2$class
mat2 <- as.matrix(dm2[, -c(1, 5)])
cs2 <- chisq.test(mat2)
write_csv(dm2, "tab2.csv")
write_csv(as.data.frame(cs2$stdres), "mat2.csv")
# clipr::write_clip(cs2$stdres)
```