U3Paper_Option_B

youngwoo Kwon

2021 4 21

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)
library(arm)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loading required package: lme4

##
## arm (Version 1.11-2, built: 2020-7-27)

## Working directory is C:/Users/FOS_15/Desktop/WS21/Stats 485/Paper3
```

```r
library(ggplot2)
library(matrixStats)
```

```
##
## Attaching package: 'matrixStats'

## The following object is masked from 'package:dplyr':
##
##     count
```

Summary : We analysed the $F. - S.$ model. We first reproduced their results with logistic regression using glm() and bayesglm(). Then, we used different variables to find a better model compare to the $F. - S.$'s model. As a result, we got a better model which showed lower AIC and lower deviance than the original model. We plotted the order of original model and preffered model and get the correlation value. Although two models have overlapping variables, the result states that the order of the models might be distinct from each other. We tried to replace the missing values. The first model replaced the missing value using mean value for the variable form whole data set, and the second model replaced the missing value using mean value for the variabl from the data set that has equal 'BYURBAN' value with the school contained missing value. Results are represented as visual graphs.

A : reconstruct measurement models similar to the one used by Finn and Servoss

```r
rawdata <- read.csv("http://dept.stat.lsa.umich.edu/~bbh/s485/data/security_wide.csv")
rawdata %>% complete.cases() %>% table()
```

```
## .
## FALSE  TRUE
##    44   612
```

We have to compare the result of the models later. To facilitate comparison (to avoid the absence of index on one side) I will remove missing values in advance.

```r
rawdata = rawdata %>% na.omit()
security <- rawdata[c(1,4,5,9,8,15,18,7)]
head(security)
```

```
##   school q38c q38d q38h q38g q38n q39a q38f
## 1 id1011    0    0    1    0    0    1    0
## 2 id1021    1    1    0    1    1    1    0
## 3 id1022    0    0    0    0    0    0    0
## 4 id1031    0    0    1    1    1    1    0
## 5 id1033    0    0    0    0    1    1    0
## 6 id1041    0    0    0    1    1    1    0
```

We don't have to worry about the missing datas now.

```r
sum_index <- security %>%
  dplyr::select(-school) %>% rowSums()
table(sum_index)
```

```
## sum_index
##   0   1   2   3   4   5   6
##  81 146 184 119  64  16   2
```

```r
col_index <- security %>%
  dplyr::select(-school) %>% colSums()
col_index
```

```
## q38c q38d q38h q38g q38n q39a q38f
##   16   65   95  141  289  344  269
```

```r
q_38c_data <- security %>% filter(q38c == 1)
sum(q_38c_data$q38d)/sum(q_38c_data$q38c)
```

```
## [1] 0.6875
```

```r
q_38f_data <- security %>% filter(q38f == 1)
sum(q_38f_data$q38g)/sum(q_38f_data$q38f)
```

```
## [1] 0.3457249
```

```r
security0 <- security %>% gather("item", "response", -school)
head(security0)
```

```
##   school item response
## 1 id1011 q38c        0
## 2 id1021 q38c        1
## 3 id1022 q38c        0
## 4 id1031 q38c        0
## 5 id1033 q38c        0
## 6 id1041 q38c        0
```

```r
rasch0 <- glm(response ~ item + school -1, family = "binomial", data = security0)
head(coef(rasch0), 10)
```

```
##     itemq38c     itemq38d     itemq38f     itemq38g     itemq38h     itemq38n
##  -4.24872351  -2.58132594  -0.14751400  -1.44274198  -2.05607672   0.03936898
##     itemq39a schoolid1021 schoolid1022 schoolid1031
##   0.56660141   2.73227774 -17.76083191   1.76881758
```

```r
rasch1 <- bayesglm(response ~ item + school - 1, data = security0)
```

```
## Warning in model.matrixBayes(object = mt, data = data, contrasts.arg =
## contrasts, : variable 'item' converted to a factor
```

```r
head(coef(rasch1), 10)
```

```
##     itemq38c     itemq38d     itemq38f     itemq38g     itemq38h     itemq38n
##   0.02335348   0.10341088   0.43671255   0.22758172   0.15242565   0.46938940
##     itemq39a schoolid1021 schoolid1022 schoolid1031
##   0.55925101   0.42896082  -0.27934468   0.28724367
```

B : consider whether simple variations of the model specification might improve it

In a new data, I added the question Q38j 'Enforce a strict dress code'. Question 38c and 38d seem similar. So I removed the Q38d from the data. Q38f 'Use one or more random dog sniffs to check for drugs' did not make sense for me. So I removed Q38f.

```r
security_new <- rawdata[c(1,4,9,8,15,18,11)]
head(security_new)
```

```
##   school q38c q38h q38g q38n q39a q38j
## 1 id1011    0    1    0    0    1    0
## 2 id1021    1    0    1    1    1    1
## 3 id1022    0    0    0    0    0    0
## 4 id1031    0    1    1    1    1    1
## 5 id1033    0    0    0    1    1    0
## 6 id1041    0    0    1    1    1    1
```

```r
security_new0 <- security_new %>% gather("item", "response", -school)
head(security_new0)
```

```
##   school item response
## 1 id1011 q38c        0
## 2 id1021 q38c        1
## 3 id1022 q38c        0
## 4 id1031 q38c        0
## 5 id1033 q38c        0
## 6 id1041 q38c        0
```

```r
rasch2 <- glm(response ~ item + school -1, family = "binomial", data = security_new0)
head(coef(rasch2), 10)
```

```
##      itemq38c      itemq38g      itemq38h      itemq38j      itemq38n
## -4.334269e+00 -1.522525e+00 -2.127519e+00  5.124067e-01 -9.160632e-02
##      itemq39a  schoolid1021  schoolid1022  schoolid1031  schoolid1033
##  4.014357e-01  3.669851e+00 -1.796160e+01  3.669851e+00  2.760727e-13
```

```r
rasch3 <- bayesglm(response ~ item + school - 1, data = security_new0)
```

```
## Warning in model.matrixBayes(object = mt, data = data, contrasts.arg =
## contrasts, : variable 'item' converted to a factor
```

```r
head(coef(rasch3), 10)
```

```
##     itemq38c     itemq38g     itemq38h     itemq38j     itemq38n     itemq39a
##  0.022594494  0.226822488  0.151666517  0.578097352  0.468629760  0.558491173
## schoolid1021 schoolid1022 schoolid1031 schoolid1033
##  0.494106489 -0.331062465  0.494106489 -0.001039664
```

C : compare the alternative Rasch models you will have fit, arriving at a recommended model

```r
rasch0$aic
```

```
## [1] 4471.554
```

```r
rasch1$aic
```

```
## [1] 4846.643
```

```r
rasch2$aic
```

```
## [1] 4245.735
```

```r
rasch3$aic
```

```
## [1] 4628.027
```

The model Rasch2 shows the lowest AIC. Rasch0 shows the second lowest AIC.

```r
rasch0$deviance
```

```
## [1] 3235.554
```

```r
rasch1$deviance
```

```
## [1] 570.5367
```

```r
rasch2$deviance
```

```
## [1] 3011.735
```

```r
rasch3$deviance
```

```
## [1] 528.6911
```

Rasch4 has the lowest deviance. Rasch2 has the second lowest deviance. Comparing the results, it seems the second data set performs the better result. between glm() model and bayesglm() model, I choose the glm() model, the one with lower AIC, since I prefer AIC than deviance when selecting the model.

D : compare Finn and Servoss. The school security measurement to your preferred alternative.

```
rasch2_index <- as.vector(c(schoolid1011=0, coef(rasch2)[-(1:6)]))

rasch2_matrix <- matrix(c(as.vector(security$school), rasch2_index), ncol = 2)
head(rasch2_matrix)
```

```
##      [,1]     [,2]
## [1,] "id1011" "0"
## [2,] "id1021" "3.66985115967302"
## [3,] "id1022" "-17.9616040402437"
## [4,] "id1031" "3.66985115967313"
## [5,] "id1033" "2.76072654316916e-13"
## [6,] "id1041" "2.14003632437597"
```

```
rasch2_ordered0 <- rasch2_matrix[order(rasch2_matrix[,2]),]
head(rasch2_ordered0)
```

```
##      [,1]     [,2]
## [1,] "id1781" "-1.14468502860331"
## [2,] "id1661" "-1.14468502860333"
## [3,] "id2081" "-1.14468502860333"
## [4,] "id1702" "-1.14468502860334"
## [5,] "id1732" "-1.14468502860334"
## [6,] "id1791" "-1.14468502860334"
```

This is the order of the school using rasch2, glm() model with security new data.

```
rasch2_ordered1 = cbind(rasch2_ordered0, c(1:612))
rasch2_ordered2 = rasch2_ordered1[order(rasch2_ordered1[,1]),]
head(rasch2_ordered2)
```

```
##      [,1]     [,2]                   [,3]
## [1,] "id1011" "0"                    "205"
## [2,] "id1021" "3.66985115967302"     "609"
## [3,] "id1022" "-17.9616040402437"    "202"
## [4,] "id1031" "3.66985115967313"     "612"
## [5,] "id1033" "2.76072654316916e-13" "573"
## [6,] "id1041" "2.14003632437597"     "402"
```

The first column of this matrix is school names. The second column is their glm() coefficients. The third column is their order.

Let's repeat this procedure for the $F. - S.$ model, rasch0.

```
rasch0_index <- as.vector(c(schoolid1011=0, coef(rasch0)[-(1:7)]))
rasch0_matrix <- matrix(c(as.vector(security$school), rasch0_index), ncol = 2)
rasch0_ordered0 <- rasch0_matrix[order(rasch0_matrix[,2]),]
rasch0_ordered1 = cbind(rasch0_ordered0, c(1:612))
rasch0_ordered2 = rasch0_ordered1[order(rasch0_ordered1[,1]),]
head(rasch0_ordered2)
```

```
##      [,1]     [,2]                   [,3]
## [1,] "id1011" "0"                    "228"
## [2,] "id1021" "2.73227773914541"     "412"
## [3,] "id1022" "-17.7608319146448"    "153"
## [4,] "id1031" "1.76881757903712"     "363"
## [5,] "id1033" "3.68920363618418e-13" "435"
## [6,] "id1041" "0.892167222603639"    "237"
```

Let's combine the order.

```
rasch_comp <- cbind(Rasch0 = as.numeric(rasch0_ordered2[,3]), Rasch2 = as.numeric(rasch2_ordered2[,3]))
rasch_comp1 <- as.data.frame(rasch_comp[order(rasch_comp[,1]),])
head(rasch_comp)
```

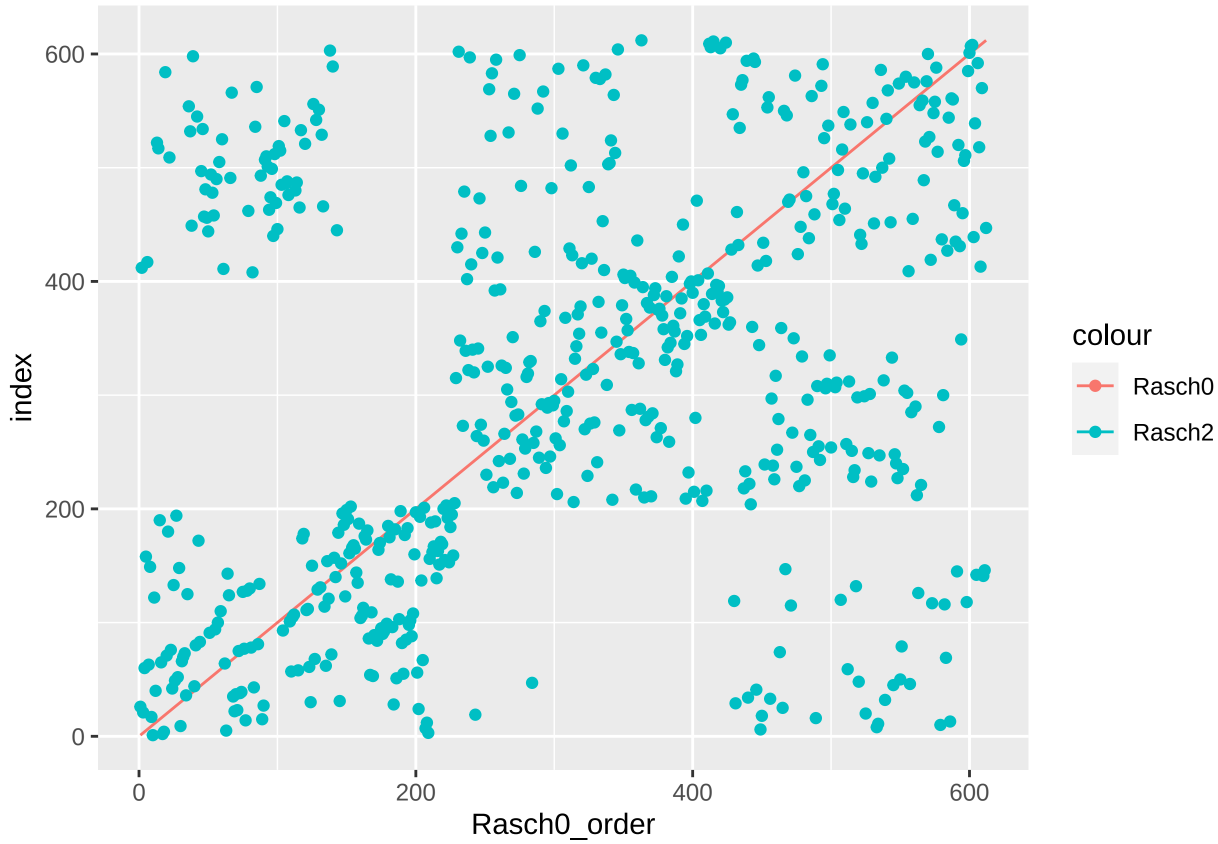
```
##      Rasch0 Rasch2
## [1,]    228    205
## [2,]    412    609
## [3,]    153    202
## [4,]    363    612
## [5,]    435    573
## [6,]    237    402
```

```
head(rasch_comp1)
```

```
##   Rasch0 Rasch2
## 1      1     26
## 2      2    412
## 3      3     21
## 4      4     60
## 5      5    158
## 6      6    417
```

The first row is index from the $F. - S.$ model and the second row is index from the fixed (preffered) model.

```
  ggplot(rasch_comp1, aes(x = Rasch0)) +
  geom_line(aes(y = Rasch0, colour = "Rasch0")) +
  geom_point(aes(y = Rasch2, colour = "Rasch2")) +
  labs(y = "index", x = "Rasch0_order")
```




```
cor(x = rasch_comp1$Rasch0, y = rasch_comp1$Rasch2)
```

```
## [1] 0.3130139
```

E : For the Version 2 of the paper

```
rawdata0 <- as.data.frame(read.csv("http://dept.stat.lsa.umich.edu/~bbh/s485/data/security_wide.csv"))
rawdata1 <- as.data.frame(read.csv("http://dept.stat.lsa.umich.edu/~bbh/s485/data/security_school_vars.c
```

```
rawdata0_new = rawdata0 %>% dplyr::left_join(rawdata1)
```

```
## Joining, by = "school"
```

```
head(rawdata0_new)
```

```
##   school q38a q38b q38c q38d q38e q38f q38g q38h q38i q38j q38k q38l q38m q38n
## 1 id1011    1    1    0    0    0    0    0    1    0    0    0    0    0    0
## 2 id1021    1    1    1    1    1    0    1    0    1    1    0    1    1    1
## 3 id1022    1    0    0    0    0    0    0    0    0    0    0    0    1    0
## 4 id1031    1    1    0    0    1    0    1    1    0    1    0    1    1    1
## 5 id1033    1    1    0    0    1    0    0    0    0    0    0    0    0    1
```

```
## 6 id1041    1    0    0    0    0    0    1    0    0    1    0    0    1    1
##   q38o q38p q39a q39b q39c q40a q40b q40c q40d q40e BYG10EP BYURBAN BYREGION
## 1    1    1    1    0    0    1    1    0    0    0       6       1        1
## 2    1    0    1    1    1    1    1    1    0    1       3       1        1
## 3    1    0    0    0    0    0    0    1    0    0       4       1        1
## 4    1    1    1    1    1    1    1    1    1    1       5       1        1
## 5    1    0    1    0    0    1    1    1    1    0       6       1        1
## 6    1    0    1    1    0    1    1    1    0    0       5       2        1
##   BYSPANP BY10FLP BYFTTP
## 1       3       5      6
## 2       3       7      6
## 3       3       3      6
## 4       3      -9      6
## 5       3       4      6
## 6       3       2      6
```

We can use new variables to estimate the missing values. We will now generate two models. The first model is similar to our preferred model. It uses the glm() for our new data set, but it replaces the missing varaibles to the mean value. (For example, if itemq38c is missed for some response, instead of just remove the data like rasch2 model, its itemq38c value would be total mean value of itemq38c).

```r
rawdata0_new_df = as.data.frame(rawdata0_new[c(1,4,9,8,15,18,11)])
newdata_mean_1 <-  cbind(rawdata0_new[,1], as_tibble(sapply(rawdata0_new_df[,-1], function(x) ifelse(is

names(newdata_mean_1)[1] <- "school"
head(newdata_mean_1)
```

```
##   school q38c q38h q38g q38n q39a q38j
## 1 id1011    0    1    0    0    1    0
## 2 id1021    1    0    1    1    1    1
## 3 id1022    0    0    0    0    0    0
## 4 id1031    0    1    1    1    1    1
## 5 id1033    0    0    0    1    1    0
## 6 id1041    0    0    1    1    1    1
```

```r
newdata_mean_2 <- newdata_mean_1 %>% gather("item", "response", -school)
head(newdata_mean_2)
```

```
##   school item response
## 1 id1011 q38c        0
## 2 id1021 q38c        1
## 3 id1022 q38c        0
## 4 id1031 q38c        0
## 5 id1033 q38c        0
## 6 id1041 q38c        0
```

```r
nrow(rawdata0_new_df %>% na.omit())
```

```
## [1] 632
```

```r
nrow(newdata_mean_1 %>% na.omit())
```

```
## [1] 656
```

So there are 24 values that have been replaced into the mean value.

```r
select_na  <-  cbind(rawdata0_new[,1], as_tibble(sapply(rawdata0_new_df[,-1], function(x) ifelse(is.na(
names(select_na)[1] <- "school"
select_na  %>% gather("item", "response", -school) %>% filter(response == -20)
```

```
##    school item response
## 1  id3572 q38c      -20
## 2  id2241 q38h      -20
## 3  id3361 q38h      -20
## 4  id3572 q38h      -20
## 5  id3592 q38h      -20
## 6  id4251 q38h      -20
## 7  id1071 q38g      -20
## 8  id1452 q38g      -20
## 9  id1502 q38g      -20
## 10 id2342 q38g      -20
## 11 id2572 q38g      -20
## 12 id3361 q38g      -20
## 13 id3572 q38g      -20
## 14 id3592 q38g      -20
## 15 id1502 q38n      -20
## 16 id2372 q38n      -20
## 17 id2441 q38n      -20
## 18 id3181 q38n      -20
## 19 id3842 q38n      -20
## 20 id1182 q39a      -20
## 21 id1882 q39a      -20
## 22 id2991 q39a      -20
## 23 id3082 q39a      -20
## 24 id3151 q39a      -20
## 25 id1071 q38j      -20
## 26 id1621 q38j      -20
## 27 id2372 q38j      -20
## 28 id3243 q38j      -20
## 29 id3461 q38j      -20
## 30 id3521 q38j      -20
## 31 id3572 q38j      -20
## 32 id4261 q38j      -20
```

Those are the schools and items that had missing responses. There are 32 rows, because, some schools like id3572 had more than 2 items that are missing.

Now let's make an data set that replaces the the responses based on the additional independent variables. We will use urban as a new independent variable that to weaken the MAR assumption

```r
nrow(rawdata0_new %>% filter(BYURBAN == 1))
```

```
## [1] 207
```

```r
nrow(rawdata0_new %>% filter(BYURBAN == 2))
```

```
## [1] 322
```

```r
nrow(rawdata0_new %>% filter(BYURBAN == 3))
```

```
## [1] 127
```

```r
nrow(rawdata0_new %>% filter(BYURBAN != 1 & BYURBAN != 2 & BYURBAN != 3))
```

```
## [1] 0
```

So we can check that all the BYURBAN value in the dataset are 1,2 or 3.

We will seperate the data into three parts. One per BYURBAN value.

```r
data_urban1 = (rawdata0_new %>% filter(BYURBAN == 1))
data_urban2 = (rawdata0_new %>% filter(BYURBAN == 2))
data_urban3 = (rawdata0_new %>% filter(BYURBAN == 3))
```

And replace NA into mean values.

```r
data_urban1_df = as.data.frame(data_urban1)
data_urban1_mean <-  cbind(data_urban1[,1], as_tibble(sapply(data_urban1_df[,-1], function(x) ifelse(is
names(data_urban1_mean)[1] <- "school"
```

```r
data_urban2_df = as.data.frame(data_urban2)
data_urban2_mean <-  cbind(data_urban2[,1], as_tibble(sapply(data_urban2_df[,-1], function(x) ifelse(is
names(data_urban2_mean)[1] <- "school"
```

```r
data_urban3_df = as.data.frame(data_urban3)
data_urban3_mean <-  cbind(data_urban3[,1], as_tibble(sapply(data_urban3_df[,-1], function(x) ifelse(is
names(data_urban3_mean)[1] <- "school"
```

Combine them together

```r
newdata_mean_3 = rbind(data_urban1_mean[c(1,4,9,8,15,18,11)], data_urban2_mean[c(1,4,9,8,15,18,11)], da

newdata_mean_4 <- newdata_mean_3 %>% gather("item", "response", -school)
head(newdata_mean_4)
```

```
##   school item response
## 1 id1011 q38c        0
## 2 id1021 q38c        1
## 3 id1022 q38c        0
## 4 id1031 q38c        0
## 5 id1033 q38c        0
## 6 id1301 q38c        0
```

So now we have two data sets, newdatamean2 and newdatamean4.

Based on these data, we will generate models.

```r
rasch2_1 <- glm(response ~ item + school -1, family = "binomial", data = newdata_mean_2)
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```r
head(coef(rasch2_1), 10)
```

```
##      itemq38c      itemq38g      itemq38h      itemq38j      itemq38n
## -4.263466e+00 -1.480592e+00 -2.110410e+00  4.618481e-01 -6.524349e-02
##      itemq39a  schoolid1021  schoolid1022  schoolid1031  schoolid1033
##  3.869463e-01  3.627786e+00 -1.795240e+01  3.627786e+00  4.038911e-13
```

```r
rasch2_2 <- glm(response ~ item + school -1, family = "binomial", data = newdata_mean_4)
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```r
head(coef(rasch2_2), 10)
```

```
##      itemq38c      itemq38g      itemq38h      itemq38j      itemq38n
## -4.260922e+00 -1.480843e+00 -2.110476e+00  4.620121e-01 -6.545223e-02
##      itemq39a  schoolid1021  schoolid1022  schoolid1031  schoolid1033
##  3.870403e-01  3.626760e+00 -1.795244e+01  3.626760e+00  2.937097e-13
```

```r
rasch2_1$aic
```

```
## [1] 4579.353
```

```r
rasch2_2$aic
```

```
## [1] 4579.4
```

```r
rasch2_1$deviance
```

```
## [1] 3228.912
```

```r
rasch2_2$deviance
```

```
## [1] 3228.977
```

The only difference between two models is how we replaced the NA values. That is, most of the data are not affected. So the AIC and deviance between models are almost equal.

```r
rasch2_1_index <- as.vector(c(schoolid1011=0, coef(rasch2_1)[-(1:6)]))
rasch2_1_matrix <- matrix(c(as.vector(rawdata0_new$school), rasch2_1_index), ncol = 2)
rasch2_1_ordered0 <- rasch2_1_matrix[order(rasch2_1_matrix[,2]),]
rasch2_1_ordered1 = cbind(rasch2_1_ordered0, c(1:656))
rasch2_1_ordered2 = rasch2_1_ordered1[order(rasch2_1_ordered1[,1]),]
```

```r
rasch2_2_index <- as.vector(c(schoolid1011=0, coef(rasch2_2)[-(1:6)]))
rasch2_2_matrix <- matrix(c(as.vector(rawdata0_new$school), rasch2_2_index), ncol = 2)
rasch2_2_ordered0 <- rasch2_2_matrix[order(rasch2_2_matrix[,2]),]
rasch2_2_ordered1 = cbind(rasch2_2_ordered0, c(1:656))
rasch2_2_ordered2 = rasch2_2_ordered1[order(rasch2_2_ordered1[,1]),]
```

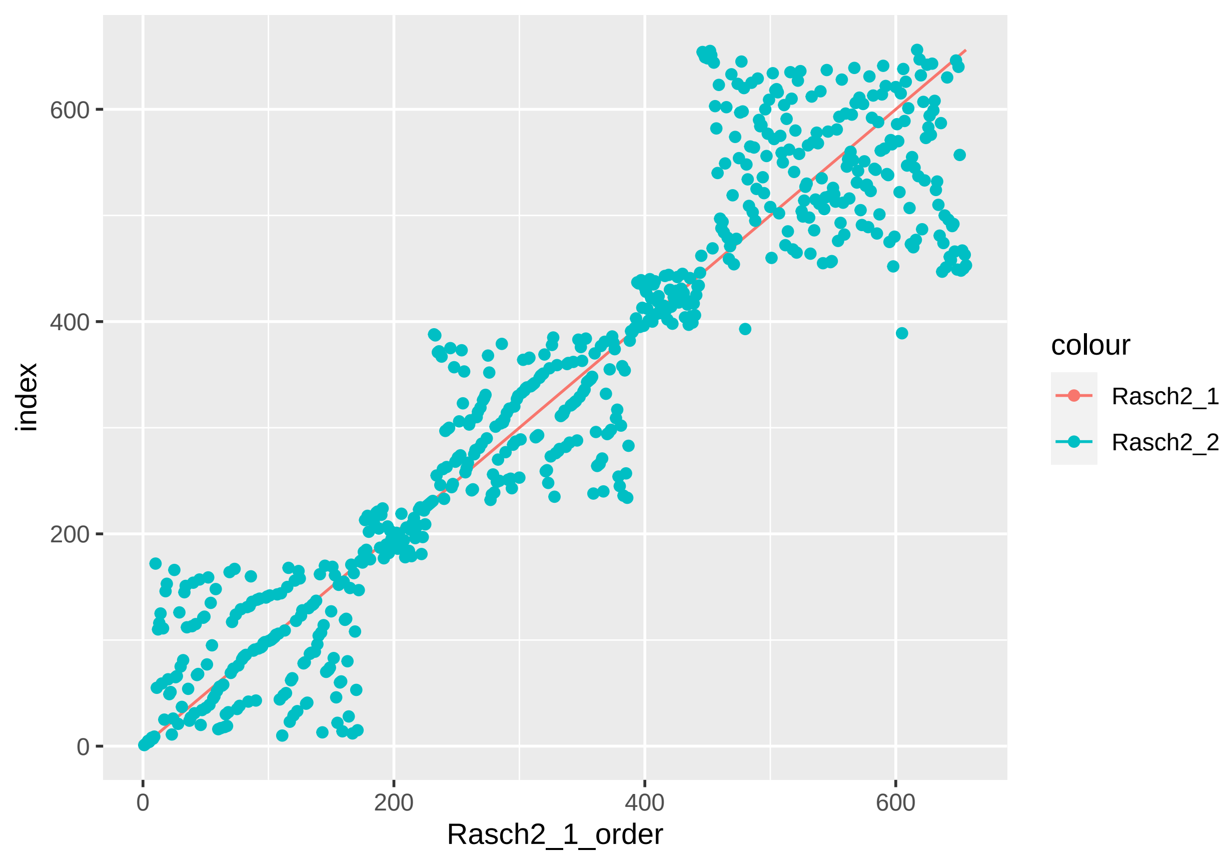
```r
rasch_comp2 <- cbind(Rasch2_1 = as.numeric(rasch2_1_ordered2[,3]), Rasch2_2 = as.numeric(rasch2_2_order
rasch_comp3 <- as.data.frame(rasch_comp2[order(rasch_comp2[,1]),])
head(rasch_comp2)
```

```
##      Rasch2_1 Rasch2_2
## [1,]      226      226
## [2,]      451      650
## [3,]      221      225
## [4,]      448      649
## [5,]      469      633
## [6,]      394      437
```

```r
head(rasch_comp3)
```

```
##   Rasch2_1 Rasch2_2
## 1        1        1
## 2        2        2
## 3        3        3
## 4        4        5
## 5        5        4
## 6        6        6
```

```r
ggplot(rasch_comp3, aes(x = Rasch2_1)) +
  geom_line(aes(y = Rasch2_1, colour = "Rasch2_1")) +
  geom_point(aes(y = Rasch2_2, colour = "Rasch2_2")) +
  labs(y = "index", x = "Rasch2_1_order")
```




```
cor(x = rasch_comp3$Rasch2_1, y = rasch_comp3$Rasch2_2)
```

```
## [1] 0.9339474
```

```
head(rawdata0_new %>% filter(school == "id2241"))
```

```
##   school q38a q38b q38c q38d q38e q38f q38g q38h q38i q38j q38k q38l q38m q38n
## 1 id2241    0    1    0    1    1    1    1   NA   NA    1    1    1    1    1
##   q38o q38p q39a q39b q39c q40a q40b q40c q40d q40e BYG10EP BYURBAN BYREGION
## 1    0    1    1    0    0    1    1    1    0    0       5       2        3
##   BYSPANP BY10FLP BYFTTP
## 1       3       3      6
```

```
rasch2_1_ordered2 %>% as_tibble() %>% filter(rasch2_1_ordered2[,1] == "id2241")
```

```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if `.name_repair` is
## Using compatibility `.name_repair`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
## # A tibble: 1 x 3
##   V1     V2               V3
##   <chr>  <chr>            <chr>
## 1 id2241 2.31568060796691 444
```

```r
rasch2_2_ordered2 %>% as_tibble() %>% filter(rasch2_1_ordered2[,1] == "id2241")
```

```
## # A tibble: 1 x 3
##   V1     V2               V3
##   <chr>  <chr>            <chr>
## 1 id2241 2.3073346343572 446
```

```r
rasch2_1_ordered2 %>% as_tibble() %>% filter(rasch2_1_ordered2[,1] == "id2342")
```

```
## # A tibble: 1 x 3
##   V1     V2                V3
##   <chr>  <chr>             <chr>
## 1 id2342 1.24940077595383 389
```

```r
rasch2_2_ordered2 %>% as_tibble() %>% filter(rasch2_1_ordered2[,1] == "id2342")
```

```
## # A tibble: 1 x 3
##   V1     V2                V3
##   <chr>  <chr>             <chr>
## 1 id2342 1.25347784380892 391
```