# Identifying Inter-subject Difficulties in Norwegian GPA Data

Tony C. A. Tan

Centre for Educational Measurement, University of Oslo Continuous Draft

Prof Rolf V. Olsen & Dr Astrid M. J. Sandsør Høst 2022

# Abstract

[Follow publisher’s instructions.]

*Keywords:* grade point average, Norway, item response theory, selection fairness, group invariance

# Identifying Inter-subject Difficulties in Norwegian GPA Data

The grade point average (GPA, *skolepoeng* in Norwegian) plays a key role in Norway’s educational assessment process. From Year 8 onwards, Norwegian high schoolers receive formal grades from both their teachers (*standpunktkarakter* ) and year-end exams (Ræder

et al., [2020](#_bookmark8)), which are used for high-stake decisions such as graduate certifications (Year 10) and university admissions (VG3). Since different subjects are treated *equally* in its calculation (Directorate for Higher Education and Skills, [2021](#_bookmark1)), GPA implicitly assumes that grades across different specialities are *equivalent* indicators of students’ preparedness for the next stage of education — an assumption that remains untested and questioned by descriptive statistics (Directorate of Education, [2022](#_bookmark2)).

Concerns for the comparability of subject difficulties are further deepened by prior studies from education systems similar to that of Norway’s. He et al. ([2018](#_bookmark3)) in the UK and Korobko et al. ([2008](#_bookmark4)) in the Netherlands both reported persistent disagreements among subject difficulties, which may lead to differential treatment of students with different specialisations. Besides fairness concerns, the lack of difficulty comparability also leads to a lack of construct validity (Messick, [1989](#_bookmark6)) in GPA, as the construct-irrelevant variance related to subject characteristics, in addition to candidates’ competencies, have been included in the GPA calculation. Understanding the presence, directions, and magnitudes of inter-subject difficulties therefore becomes a key issue for assessing the validity of GPA. By analysing Norway’s education record data, this study aims to test GPA’s validity as a measurement scale in mapping candidate competence into numeric scores, as well as the fairness consequences subsequent to its use in high-stake situations. More specifically, the study will address the following research questions:

**RQ1:** Do Norwegian Year 10 subjects differ in their difficulty levels?

**RQ2:** Do subject difficulties differ by source such as between teachers and external examiners?

**RQ3:** Do subject difficulties differ across achievement levels?

**RQ4:** Do subject difficulties differ across demographic attributes such as socioeconomic status, gender and immigration background?

# Conceptual Framework The Norwegian Education and Assessment System

The Norwegian education system is organised into three levels: primary school (Year 1–7) where formal grading is not practised, lower secondary school (Year 8–10) and upper secondary school (Year 11–13). During the first ten years of schooling (*grunnskole*), students follow centralised national curricula with largely compulsory subjects plus a few electives.

Upon successful completion of Year 10, students may choose between vocational and academic tracks for their upper secondary schools. The former is a two-year program (*fagskole*) that prepares students for employment in a specific field, whereas the latter is a three-year program (*videregående opplæring*, VG1–3) that prepares students for university studies. In preparation for the merit-based tertiary entry requirements, academic track students carefully choose their VG subjects in order to maximise their final GPA scores. This study focuses on Year 10 GPA data where the impact of subject choice is minimal.

The GPA aims to provide a sum-score measure of a student’s overall competency. For *grunnskole* graduation purposes, the GPA is calculated as a weighted average of students’ grades from all subjects in Year 10. The weights are determined by the number of hours spent on each subject, with the exception of Norwegian and English, which are weighted equally.

Both teacher-assigned grades and exam grades are included in the GPA calculation, with each subject ranging from 1 (low competency) to 6 (outstanding). While every subject a student undertook receives a teacher-assigned grade, Year 10 students are randomly assigned into participating in *one* of the three written exams, Norwegian, English, and mathematics, hence receives only one written exam grade. Similarly, students are randomly assigned into participating in *one* of the many oral exam options, including Norwegian, English, mathematics, and electives. A candidate’s GPA is then computed by averaging the grades they received, multiplying by 10 and rounding to two decimal places.

# Methods

**Population**

This study retains the entire cohort of Norway’s Year 10 students graduating in 2019 as its targeted population. Students’ GPA (*grunnskolepoeng*), teacher-assigned grades (*standpunktkarakter* ), as well as written (*SKR*) and oral (*MUN* ) exam grades were extracted from the national register. This data source is unique because it is the *population*, rather than

samples, that forms our unit of analysis. Academic attainment records were then re-shaped into the format that each candidate occupies one row and each subject is represented by one column. This process led to a preliminary data set of 64,918 students and 200 subjects. Next, 4*,* 300 students without valid GPA records were excluded from subsequent analyses, representing a loss rate of 6*.*62%. Seventeen subjects were retained based on these inclusion criteria:

## *Teacher-assigned Grades (12 subjects)*

Under the Norwegian education system, Year 10 students shall complete 13 compulsory subjects as well as electives. This study included all compulsory subjects except for Sidemål.1 We applied equal treatment to courses instructed in Norwegian and in Sami language by merging these records.2 Twelve teacher-assigned grades were included for our analysis: Written Norwegian (NORW), Oral Norwegian (NORO), Written English (ENGW), Oral English (ENGO), Mathematics (MATH), Natural Sciences (NATS), Social Sciences (SOCS), Religion (RELI), Music (MUSI), Arts and Handcraft (HAND), Food and Health (FOOD), and Physical Education (PHED).

## *Written Exam Grades (3 subjects)*

Norway uses a lottery draw to randomly assign Year 10 students to participate in *one* of the following three written exams: Norwegian (E-NORW), English (E-ENGW) and mathematics (E-MATH). This “planned missingness” implies that although numeral in quantity, the unobserved exam grades can be safely modelled under the missing completely at random (MCAR) assumption (Little & Rubin, [2019](#_bookmark5)).3 Rasch models have a major advantage for handling missing values thanks to the sufficient overlap across subjects in the score matrix (He et al., [2018](#_bookmark3)).

1 The Norwegian language has two official written forms: Bokmål and Nynorsk, with the former being more prevalent in the media. Students growing up in one written form must enroll the other as their Sidemål, unless Norwegian is not their native language. Nynorsk users tend to have easier time in Sidemål due to existing exposure to Bokmål. Bokmål users, on the other hand, find Nynorsk more challenging while fulfilling Sidemål. Since Sidemål contains two sub-cohorts with distinct difficulty profiles, we opt not to include this subject in our analyses.

2 For example, [NAT0010 Naturfag 10. årstrinn](https://www.udir.no/kl06/nat0010) and [NAT0020 Naturfag, samisk plan, 10. årstrinn](https://www.udir.no/kl06/nat0020) were merged into one subject Natural Sciences. If academic results were available from both instruction languages, we retained the higher grades during merging.

3 Even if the lottery is less than perfectly random, Rasch models are still valid under the weaker assumption of missing at random (MAR) hence “ignorable” (Molenaar, [1995](#_bookmark7)), as long as one is satisfied that missing propensities are unrelated to item or person parameters.

## *Oral Exam Grades (2 subjects)*

Year 10’s oral exams consist of the same three subjects as in written exams, plus a wide selection such as natural and social sciences, with students being randomly assigned into *one* oral exam by lottery. In order to better match teacher-assigned grades, only Oral Norwegian (E-NORO) and Oral English (E-ENGO) were included in this study. Since students are spread thinly across many oral exam subjects, E-NORO and E-ENGO appeared more sparse than their teacher-assigned counterparts, leading to larger confidence intervals in subsequent analyses.

# References

Directorate for Higher Education and Skills. (2021). *Slik beregner du poeng* [How to calculate your GPA]. Direktoratet for høyere utdanning og kompetanse. <https://utdanning.no/tema/soknad_og_opptak/slik_beregner_du_poeng>

Directorate of Education. (2022). *Karakterstatistikk for videregående skole* [Grade statistics for upper secondary school]. Utdanningsdirektoratet. [https://www.udir.no/tall-og-](https://www.udir.no/tall-og-forskning/statistikk/statistikk-videregaende-skole/karakterer-vgs/) [forskning/statistikk/statistikk-videregaende-skole/karakterer-vgs/](https://www.udir.no/tall-og-forskning/statistikk/statistikk-videregaende-skole/karakterer-vgs/)

He, Q., Stockford, I., & Meadows, M. (2018). Inter-subject comparability of examination standards in GCSE and GCE in England. *Oxford Review of Education*, *44* (4), 494–513. <https://doi.org/10.1080/03054985.2018.1430562>

Korobko, O. B., Glas, C. A. W., Bosker, R. J., & Luyten, J. W. (2008). Comparing the difficulty of examination subjects with item response theory. *Journal of Educational* *Measurement*, *45* (2), 139–157. <https://doi.org/10.1111/j.1745-3984.2007.00057.x>

Little, R. J. A., & Rubin, D. B. (2019). *Statistical analysis with missing data* (3rd ed.). Wiley. <https://doi.org/10.1002/9781119482260>

Messick, S. (1989). Validity. In R. L. Linn (Ed.), *Educational measurement* (3rd ed., pp. 10–103). American Councile on Education; Macmillan.

Molenaar, I. W. (1995). Estimation of item parameters. In G. H. Fischer & I. W. Molenaar (Eds.), *Rasch models: Foundations, recent developments, and applications* (pp. 39–51). Springer-Verlag. <https://doi.org/10.1007/978-1-4612-4230-7_3>

Ræder, H. G., Olsen, R. V., & Blömeke, S. (2020). Large-scale assessments in the Norwegian context. In H. Harju-Luukkainen, N. McElvany, & J. Stang (Eds.), *Monitoring student achievement in the 21st Century: European policy perspectives and assessment strategies* (pp. 195–206). Springer. <https://doi.org/10.1007/978-3-030-38969-7_16>

# Analysis Code, Additional Tables and Figures Register Data Re-format

1 *# Only keep 2019 data*

*# STP (Teacher assigned marks)*

teacher\_mk <- gpa[**which**(gpa**$**AVGDATO == 201906), **c**(1:4, 7)]

*# Save the total number of students*

5 n\_student <- **dim**(teacher\_mk)[1] *# Should be 1,073,204 obs*

**if** (**interactive**()) {n\_student}

*# Inspect unusual marks in the "STP" column*

**if** (**interactive**()) {

10 **table**(**unlist**(teacher\_mk**$**STP))

}

*# These marks are not usable: # ’’ empty [n = 20,042],*

*# 7 [n = 33],*

15 *# D [n = 58,182], # F [n = 37,273], # GK [n = 55],*

*# IM [n = 2],*

*# IV [n = 12,576].*

20

*# Recode un-usable STP into NA*

teacher\_mk**$**STP <- car::recode(teacher\_mk**$**STP, " c(’’, ’7’, ’D’, ’F’, ’GK’, ’IM’, ’IV’) = NA

")

25

*# Part 1: Re-shape teacher-assigned marks columns: one subject per column*

30 *# How many subjects there are? (Answer: 200 different subjects in total)*

*# How many times each subject name appeared (with or without valid score)?* subject\_frequency <- **sort**(**table**(**unlist**(teacher\_mk**$**FAGKODE)), decreasing = T) **if** (**interactive**()) {subject\_frequency}

*# Save subject list*

35 subject\_list <- **as**.**character**(**data**.**frame**(subject\_frequency)[, 1])

*# Save total number of subjects*

n\_subject <- **length**(subject\_list)

**if** (**interactive**()) {n\_subject} *# Should be 200 subjects in total*

40 *# Create a placeholder spreadsheet*

stp\_spreadsheet <- **data**.**frame**(**matrix**(NA, **nrow** = n\_student, **ncol** = n\_subject))

**colnames**(stp\_spreadsheet) <- subject\_list

*# Stitch STP and this empty placeholder spreadsheet together*

45 teacher\_reshape <- **cbind**(teacher\_mk, stp\_spreadsheet)

**if** (**interactive**()) {**names**(teacher\_reshape)}

*# Set up a progress bar*

n\_iter <- **dim**(teacher\_reshape)[2] *# Set the progress bar’s end point*

50 pb <- progress::progress\_bar**$new**( *# Refresh progress bar’s internal definition*

**format** = "(:spin) [:bar] :percent [Elapsed time: :elapsedfull || Estimated time remaining: :

*'→* eta]",

total = n\_iter, **complete** = "=", incomplete = "-",

55 current = ">", clear = F, width = 100

)

60 **for** (j in 6:n\_iter) { *# 200 cycles # Insert progress bar here*

progress::pb**$**tick() *# Update progress bar*

*# Create a placeholder list*

65 temp <- **rep**(**names**(teacher\_reshape)[j], n\_student)

*# Test whether subject names match*

equal\_test <- temp == teacher\_reshape[, 4]

*# Turn FALSE***/***TRUE to 0***/***1*

equal\_test <- equal\_test + 0

70

*# If subject name matches, copy-paste teacher-assign marks # into the temp\_subject column*

temp\_subject <- equal\_test **\*** teacher\_reshape[, 5]

*# Turn off list property (in order to recode)*

75 temp\_subject <- **as**.**numeric**(**unlist**(temp\_subject))

*# Recode 0 to NA*

teacher\_reshape[, j] <- car::recode(temp\_subject, "0 = NA")

}

**cat**("\n") *# Start a new line once progress bar is full*

80

*# Remove subject name and STP columns* teacher\_reshaped <- teacher\_reshape[, -**c**(4, 5)] *# Inspect the newly shaped data set*

**if** (**interactive**()) {head(teacher\_reshaped, 20)}

85

*# Save to external file.*

**if** (Sys.info()["sysname"] == "Windows") {

**data**.**table**::fwrite(teacher\_reshaped, "M:**/**p1708-tctan**/**Documents**/**teacher0.csv",

90 **row**.**names** = F

)

} **else** {

**data**.**table**::fwrite(teacher\_reshaped, "**/**tsd**/**p1708**/**home**/**p1708-tctan**/**Documents**/**teacher0.csv",

95 **row**.**names** = F

)

}

*# Should be 239,329 KB in size*

100

*# Part 2: Re-shape rows: one student per row*

*# How many (unique) students there are? (Answer: 64,918 unique students)*

105 *# How many times each student ID appeared (with or without valid score)?*

student\_frequency <- **data**.**frame**(**sort**( **table**(**unlist**(teacher\_reshaped**$**w21\_4952\_lopenr\_person)), decreasing = T

))

110 *# Display the top 20 students who took the most number of subjects*

head(student\_frequency, 20)

*# Display the bottom 20 students who took the least number of subjects*

tail(student\_frequency, 20)

*# Save student list*

115 student\_list <- **as**.**character**(student\_frequency[, 1])

*# Save total number of unique students*

(n\_unique\_student <- **length**(student\_list)) *# 64,918 unique students*

*# Set up a placeholder spreadsheet*

120 teacher\_reshaped\_final <- **matrix**(

**nrow** = n\_unique\_student, **ncol** = **dim**(teacher\_reshaped)[2]

125

130

)

**colnames**(teacher\_reshaped\_final) <- **names**(teacher\_reshaped) teacher\_reshaped\_final <- **data**.**frame**(teacher\_reshaped\_final)

*# Prepare multi-core processing*

**if** (Sys.info()["sysname"] == "Windows") { *# Windows can only use single core*

n\_cores <- 1

} **else** { *# Both Linux and Mac can implement multicore*

n\_cores <- parallel::detectCores() *# Count the total number of CPU cores*

n\_cores <- n\_cores - 1 *# Reserve one core for system admin*

}

*# Set up a progress bar*

135 n\_iter <- n\_unique\_student *# Set the progress bar’s end point*

pb <- progress::progress\_bar**$new**( *# Refresh progress bar’s internal definition*

**format** = "(:spin) [:bar] :percent [Elapsed time: :elapsedfull || Estimated time remaining: :

*'→* eta]",

total = n\_iter,

140

)

145

**complete** = "=", incomplete = "-", current = ">", clear = F,

width = 100

150

**for**(i in 1:n\_iter) {

*# Insert progress bar here*

progress::pb**$**tick()

*# Pull out lines that share the same Student ID*

student\_temp <- teacher\_reshaped[**which**( teacher\_reshaped[, 1] == student\_list[i]

155

160

), ]

*# Collapse multiple lines into one line*

student\_temp\_teacher <- parallel::mclapply(student\_temp[, -**c**(1:3)],

**function**(x) **max**(x, **na**.**rm** = T), mc.cores = n\_cores)

*# In cases where, same person, same subject, but multiple marks,*

*# take the maximum, because I do not know which score was given first. # When I asked R to compute max from a column containing NA only,*

*# R produced -Inf and a warning.*

*# Safe to ignore these warnings and turn -Inf to NA. # Recode 0 and -Inf to NA*

student\_temp\_teacher <- car::recode(student\_temp\_teacher, " c(’0’, ’-Inf’) = NA

165

")

*# Stitch admin variables to student\_temp\_teacher (need transpose)*

teacher\_reshaped\_final[i, ] <- **data**.**frame**(**cbind**( student\_temp[1, **c**(1:3)], **t**(student\_temp\_teacher)

))

170 }

175

**cat**("\n") *# Start a new line once progress bar is full*

*# Save the standard Student ID list for subsequent work*

**if** (Sys.info()["sysname"] == "Windows") {

**write**.**table**(teacher\_reshaped\_final[, 1], "M:**/**p1708-tctan**/**Documents**/**student\_id.csv", **row**.**names** = F, **col**.**names** = **c**("student\_id")

180

)

} **else** {

**write**.**table**(teacher\_reshaped\_final[, 1], "**/**tsd**/**p1708**/**home**/**p1708-tctan**/**Documents**/**student\_id.csv", **row**.**names** = F, **col**.**names** = "student\_id"

)

}

185 *# Should be 888 KB in size*

*# Save teacher-assigned marks*

**if** (Sys.info()["sysname"] == "Windows") {

**data**.**table**::fwrite(teacher\_reshaped\_final,

190

)

} **else** {

"M:**/**p1708-tctan**/**Documents**/**teacher1.csv",

**row**.**names** = F

195

**data**.**table**::fwrite(teacher\_reshaped\_final, "**/**tsd**/**p1708**/**home**/**p1708-tctan**/**Documents**/**teacher1.csv", **row**.**names** = F

)

}

*# Should be 15,345 KB in size*

# Retain Students with Valid GPA and Sufficient Teacher-assigned Grades

1 *# Full data set: N = 64,918*

*# Drop students without valid GPAs*

teacher\_gpa <- teacher[**!is**.**na**(teacher**$**GRUNNSKOLEPOENG), ]

5 **if** (**interactive**()) {**dim**(teacher\_gpa)} *# 60,618 students remain*

*# Compute data loss rate (n = 4,300, % = 6.62)*

**if** (**interactive**()) {

**dim**(teacher)[1] - **dim**(teacher\_gpa)[1]

10 **round**((**dim**(teacher)[1] - **dim**(teacher\_gpa)[1]) **/ dim**(teacher)[1] **\*** 100, 2)

}

*# Sort columns by number of valid entries*

teacher\_sorted <- teacher\_gpa[ , **order**(colSums(**is**.**na**(teacher\_gpa)))]

15

*# Keep the order of admin variables*

teacher\_export <- **cbind**(teacher\_gpa[, **c**(1:8)], teacher\_sorted[, -**c**(1:8)])

*# Save the subject list*

20 n\_valid <- **dim**(teacher\_export)[1] - colSums(**is**.**na**(teacher\_export[,-**c**(1:8)])) r\_valid <- **round**(n\_valid **/ dim**(teacher\_export)[1] **\*** 100, 2)

teacher\_valid <- **cbind**(n\_valid, r\_valid)

*# Preserve subject list to an external file*

25 **data**.**table**::fwrite(**data**.**frame**(teacher\_valid), "subject\_list.csv", **row**.**names** = T)

*# Save data set containing anyone with valid GPA*

**data**.**table**::fwrite(teacher\_export, ".**/**Rolf**/**stp\_valid\_gpa.csv", **row**.**names** = F)

30 *# Save "the 12 subjects" including both Norwegian and Sami as instruction lang*

teacher\_export\_subj\_12 <- teacher\_export[, **c**(

*# English x 2 # ENGW:*

’ENG0012’, *# 1 English (written)*

35 *# ENGO:*

’ENG0013’, *# 2 English (oral) # HAND: Handcraft*

’KHV0010’, *# 3 Handcraft*

’KHV0020’, *# 4 Duoji (Sami handcraft)*

40 *# PHED: Physical Education*

’KRO0020’, *# 5 P.E.*

*# MATH: Mathematics* ’MAT0010’, *# 6 Mathematics # FOOD: Food and Health*

45 ’MHE0010’, *# 7 Food and Health*

’MHE0020’, *# 8 Food and Health (instructed in Sami) # MUSI: Music*

’MUS0010’, *# 9 Music*

’MUS0020’, *# 10 Music (instructed in Sami)*

50 *# NATS: Natural Sciences*

’NAT0010’, *# 11 Natural Sciences*

’NAT0020’, *# 12 Natural Sciences (instructed in Sami) # Norwegian x 2*

*# NORW:*

55 ’NOR0214’, *# 13 Norwegian (written)*

’NOR0041’, *# 14 Norwegian (written, native language Sami) # NORO:*

’NOR0216’, *# 15 Norwegian (oral)*

’NOR0042’, *# 16 Norwegian (oral, native language Sami)*

60 *# RELI: Religion*

’RLE0030’, *# 17 Religion*

’RLE0040’, *# 18 Religion (instructed in Sami) # SOCS: Social Sciences*

’SAF0010’, *# 19 Social Sciences*

65 ’SAF0020’ *# 20 Social Sciences (instructed in Sami)*

)]

*# Add admin variables to "the 12 subjects" list*

teacher\_export\_12 <- **cbind**(teacher\_export[, **c**(1:8)], teacher\_export\_subj\_12)

*# Save "the 12 subjects"*

70 **data**.**table**::fwrite(teacher\_export\_12, ".**/**Rolf**/**stp\_12.csv", **row**.**names** = F)

*# Merge Norwegian- and Sami-instructed marks*

75

*# Create a placeholder matrix*

subj\_12 <- **data**.**frame**(**matrix**(NA, **nrow**= **dim**(teacher\_export\_12)[1], **ncol** = 12))

**names**(subj\_12) <- **c**( "ENGW", "ENGO", *# 1, 2*

80 "HAND", *# 3*

"PHED", *# 4*

"MATH", *# 5*

"FOOD", *# 6*

"MUSI", *# 7*

85 "NATS", *# 8*

"NORW", "NORO", *# 9, 10*

"RELI", *# 11*

"SOCS" *# 12*

)

90

*# Copy-paste subjects that do not need merges*

subj\_12[, 1] <- teacher\_export\_subj\_12[, 1] *# ENGW: English (written)* subj\_12[, 2] <- teacher\_export\_subj\_12[, 2] *# ENGO: English (oral)* subj\_12[, 4] <- teacher\_export\_subj\_12[, 5] *# PHED: Physical education*

95 subj\_12[, 5] <- teacher\_export\_subj\_12[, 6] *# MATH: Mathematics*

100

105

*# Set up a progress bar*

**library**(progress)

n\_iter <- **dim**(teacher\_export\_12)[1] *# Set the progress bar’s end point*

pb <- progress\_bar**$new**( *# Refresh progress bar’s internal definition*

**format** = "(:spin) [:bar] :percent [Elapsed time: :elapsedfull || Estimated time remaining: :

*'→* eta]",

total = n\_iter, **complete** = "=", incomplete = "-", current = ">", clear = F,

width = 100

110 )

*# Merge HAND*

**for** (i in 1:n\_iter) {

*# Insert progress bar here*

115

120

}

pb**$**tick() *# Update progress bar*

subj\_12[i, 3] <- **max**( teacher\_export\_subj\_12[i, 3],

teacher\_export\_subj\_12[i, 4],

**na**.**rm** = T

)

**cat**("\n") *# Start a new line once progress bar is full*

125 *# Reset progress bar*

n\_iter <- **dim**(teacher\_export\_12)[1] *# Set the progress bar’s end point*

pb <- progress\_bar**$new**( *# Refresh progress bar’s internal definition*

**format** = "(:spin) [:bar] :percent [Elapsed time: :elapsedfull || Estimated time remaining: :

*'→* eta]",

total = n\_iter,

130

135 )

**complete** = "=", incomplete = "-", current = ">", clear = F,

width = 100

140

145

*# Merge FOOD*

**for** (i in 1:n\_iter) {

*# Insert progress bar here*

pb**$**tick() *# Update progress bar*

subj\_12[i, 6] <- **max**( teacher\_export\_subj\_12[i, 7],

teacher\_export\_subj\_12[i, 8],

**na**.**rm** = T

)

}

**cat**("\n") *# Start a new line once progress bar is full*

150 *# Reset progress bar*

n\_iter <- **dim**(teacher\_export\_12)[1] *# Set the progress bar’s end point*

155

pb <- progress\_bar**$new**( *# Refresh progress bar’s internal definition*

**format** = "(:spin) [:bar] :percent [Elapsed time: :elapsedfull || Estimated time remaining: :

*'→* eta]",

total = n\_iter, **complete** = "=", incomplete = "-", current = ">", clear = F,

width = 100

160 )

*# Merge MUSI*

**for** (i in 1:n\_iter) {

*# Insert progress bar here*

165

170

}

pb**$**tick() *# Update progress bar*

subj\_12[i, 7] <- **max**( teacher\_export\_subj\_12[i, 9],

teacher\_export\_subj\_12[i, 10],

**na**.**rm** = T

)

**cat**("\n") *# Start a new line once progress bar is full*

175 *# Reset progress bar*

n\_iter <- **dim**(teacher\_export\_12)[1] *# Set the progress bar’s end point*

pb <- progress\_bar**$new**( *# Refresh progress bar’s internal definition*

**format** = "(:spin) [:bar] :percent [Elapsed time: :elapsedfull || Estimated time remaining: :

*'→* eta]",

total = n\_iter,

180

185 )

**complete** = "=", incomplete = "-", current = ">", clear = F,

width = 100

190

195

*# Merge NATS*

**for** (i in 1:n\_iter) {

*# Insert progress bar here*

pb**$**tick() *# Update progress bar*

subj\_12[i, 8] <- **max**( teacher\_export\_subj\_12[i, 11],

teacher\_export\_subj\_12[i, 12],

**na**.**rm** = T

)

}

**cat**("\n") *# Start a new line once progress bar is full*

200 *# Reset progress bar*

n\_iter <- **dim**(teacher\_export\_12)[1] *# Set the progress bar’s end point*

pb <- progress\_bar**$new**( *# Refresh progress bar’s internal definition*

**format** = "(:spin) [:bar] :percent [Elapsed time: :elapsedfull || Estimated time remaining: :

*'→* eta]",

total = n\_iter,

205

210 )

**complete** = "=", incomplete = "-", current = ">", clear = F,

width = 100

215

220

*# Merge NORW*

**for** (i in 1:n\_iter) {

*# Insert progress bar here*

pb**$**tick() *# Update progress bar*

subj\_12[i, 9] <- **max**( teacher\_export\_subj\_12[i, 13],

teacher\_export\_subj\_12[i, 14],

**na**.**rm** = T

)

}

**cat**("\n") *# Start a new line once progress bar is full*

225 *# Reset progress bar*

n\_iter <- **dim**(teacher\_export\_12)[1] *# Set the progress bar’s end point*

pb <- progress\_bar**$new**( *# Refresh progress bar’s internal definition*

**format** = "(:spin) [:bar] :percent [Elapsed time: :elapsedfull || Estimated time remaining: :

*'→* eta]",

total = n\_iter,

230

235 )

**complete** = "=", incomplete = "-", current = ">", clear = F,

width = 100

240

245

*# Merge NORO*

**for** (i in 1:n\_iter) {

*# Insert progress bar here*

pb**$**tick() *# Update progress bar*

subj\_12[i, 10] <- **max**( teacher\_export\_subj\_12[i, 15],

teacher\_export\_subj\_12[i, 16],

**na**.**rm** = T

)

}

**cat**("\n") *# Start a new line once progress bar is full*

250 *# Reset progress bar*

n\_iter <- **dim**(teacher\_export\_12)[1] *# Set the progress bar’s end point*

pb <- progress\_bar**$new**( *# Refresh progress bar’s internal definition*

**format** = "(:spin) [:bar] :percent [Elapsed time: :elapsedfull || Estimated time remaining: :

*'→* eta]",

total = n\_iter,

255

260 )

**complete** = "=", incomplete = "-", current = ">", clear = F,

width = 100

265

270

*# Merge RELI*

**for** (i in 1:n\_iter) {

*# Insert progress bar here*

pb**$**tick() *# Update progress bar*

subj\_12[i, 11] <- **max**( teacher\_export\_subj\_12[i, 17],

teacher\_export\_subj\_12[i, 18],

**na**.**rm** = T

)

}

**cat**("\n") *# Start a new line once progress bar is full*

275 *# Reset progress bar*

n\_iter <- **dim**(teacher\_export\_12)[1] *# Set the progress bar’s end point*

pb <- progress\_bar**$new**( *# Refresh progress bar’s internal definition*

**format** = "(:spin) [:bar] :percent [Elapsed time: :elapsedfull || Estimated time remaining: :

*'→* eta]",

total = n\_iter,

280

285 )

**complete** = "=", incomplete = "-", current = ">", clear = F,

width = 100

290

*# Merge SOCS*

**for** (i in 1:n\_iter) {

*# Insert progress bar here*

pb**$**tick() *# Update progress bar*

295

}

subj\_12[i, 12] <- **max**( teacher\_export\_subj\_12[i, 19],

teacher\_export\_subj\_12[i, 20],

**na**.**rm** = T

)

**cat**("\n") *# Start a new line once progress bar is full*

300 *# Turn -Inf to NA column-by-column # Create a placeholder matrix*

subj\_12\_clean <- **matrix**(NA, **nrow** = **dim**(subj\_12)[1], **ncol** = **dim**(subj\_12)[2]) n\_iter <- **dim**(subj\_12\_clean)[2] *# Set the progress bar’s end point*

pb <- progress\_bar**$new**( *# Refresh progress bar’s internal definition*

305

310

)

**format** = "(:spin) [:bar] :percent [Elapsed time: :elapsedfull || Estimated time remaining: :

*'→* eta]",

total = n\_iter, **complete** = "=", incomplete = "-", current = ">", clear = F,

width = 100

315

**for** (j in 1:n\_iter) { *# 12 cycles # Insert progress bar here* pb**$**tick() *# Update progress bar*

subj\_12\_clean[, j] <- car::recode(subj\_12[, j], " ’-Inf’ = NA

")

320 }

subj\_12\_clean <- **data**.**frame**(subj\_12\_clean) **names**(subj\_12\_clean) <- **names**(subj\_12)

*# Re-order subjects*

325 subj\_12\_clean <- subj\_12\_clean[, **c**(9,10,1,2,5,8,12,4,7,6,3,11)]

*# New order:*

*# NORW: Norwegian (written) # NORO: Norwegian (oral)*

*# ENGW: English (wrItten)*

330 *# ENGO: English (oral) # MATH: Mathematics*

*# NATS: Natural Sciences # SOCS: Social Sciences*

*# PHED: Physical Education*

335 *# MUSI: Music*

*# FOOD: Food and Health*

*# HAND: Arts and Handcraft # RELI: Religion*

340 *# Count the number of missings for each student*

missing\_12 <- rowSums(**is**.**na**(subj\_12\_clean)) *# Total number of missings* missing\_7 <- rowSums(**is**.**na**(subj\_12\_clean[, **c**(1:7)])) *# 7 major subjects* missing\_5 <- rowSums(**is**.**na**(subj\_12\_clean[, **c**(8:12)])) *# 5 minor subjects*

345 *# Stitch admin, missing counts and marks together*

teacher\_final <- **cbind**(teacher\_export\_12[, **c**(1:8)], *# Admin variables*

missing\_12, missing\_7, missing\_5, *# Missing counts*

subj\_12\_clean *# Teacher-assigned marks*

350

355

)

*# Save teacher\_final*

**data**.**table**::fwrite(teacher\_final, ".**/**Rolf**/**60618.csv", **row**.**names** = F)

*# Keep students with 4 or more of the 7-major subjects*

major\_4\_plus <- teacher\_final[**which**(teacher\_final**$**missing\_7 < 4), ]

**if** (**interactive**()) {**dim**(major\_4\_plus)} *# 59,517 students remain*

*# Compute data loss rate (n = 1,101, % = 1.82)*

360 **if** (**interactive**()) {

**dim**(teacher\_final)[1] - **dim**(major\_4\_plus)[1]

**round**((**dim**(teacher\_final)[1] - **dim**(major\_4\_plus)[1]) **/ dim**(teacher\_final)[1] **\*** 100, 2)

}

365 *# Save major\_4\_plus to an external file*

**data**.**table**::fwrite(major\_4\_plus[, -**c**(9:11)], ".**/**Rolf**/**59517.csv", **row**.**names** = F)

*# Keep students with 3 or more of the 5-minor subjects*

370 minor\_3\_plus <- major\_4\_plus[**which**(major\_4\_plus**$**missing\_5 < 3), ]

**dim**(minor\_3\_plus) *# 57,730 students remain*

*# Compute data loss rate (n = 1,787, % = 3)*

**if** (**interactive**()) {

375

}

**dim**(major\_4\_plus)[1] - **dim**(minor\_3\_plus)[1]

**round**((**dim**(major\_4\_plus)[1] - **dim**(minor\_3\_plus)[1]) **/ dim**(major\_4\_plus)[1] **\*** 100, 2)

*# Save minor\_3\_plus to an external file*

380 **data**.**table**::fwrite(minor\_3\_plus[, -**c**(9:11)], ".**/**Rolf**/**57730.csv", **row**.**names** = F)

# Subject Difficulty Analysis using GPCM

1 *# Load R package ‘mirt‘*

suppressWarnings(suppressMessages(**library**(mirt)))

*# Generalised partial credit model*

5 gpcm <- mirt(difficulty[,**c**(9:20)], itemtype = "gpcm", SE = T)

**coef**(gpcm, printSE = T, IRTpars = T)

**data**.**table**::fwrite(**coef**(gpcm, printSE = T, IRTpars = T, **as**.**data**.**frame** = T), ".**/**Rolf**/**parameter.csv",

now.**names** = T

10 )

*# Save subjects’ codes and names* subj\_code <- **names**(difficulty)[-**c**(1:8)] subj\_name <- **c**(

15 "Written Norwegian",

"Oral Norwegian", "Written English", "Oral English", "Mathematics",

20 "Natural Sciences",

"Social Sciences", "Physical Education", "Music",

"Food and Health",

25 "Arts and Handcraft", "Religion"

)

*# Item characteristic curves*

30 *# Auto-print is off in loops, causing corrupted PDFs. Insert print().*

**for** (i in 1:12) {

pdf(**file** = paste0(".**/**Rolf**/**trace**/**trace\_", subj\_code[i], ".pdf"))

**print**(directlabels::direct.label( itemplot(gpcm, item = i, type = ’trace’,

35 xlim = **c**(-6.5,6.5),

main = paste0(

"Trace Plot for ", subj\_code[i], " (", subj\_name[i], ")"

40 ))

)

), ’top.points’

**dev**.**off**()

}

*# Expected scores*

45 **for** (i in 1:12) {

pdf(**file** = paste0(".**/**Rolf**/**score**/**score\_", subj\_code[i], ".pdf"))

**print**(itemplot(gpcm, item = i, type = ’score’, CE = T, xlim = **c**(-6.5,6.5),

main = paste0(

50 "Expected Score for ", subj\_code[i], " (", subj\_name[i], ")"

)

))

**dev**.**off**()

}

55

*# Information and standard errors*

**for** (i in 1:12) {

pdf(**file** = paste0(".**/**Rolf**/**info**/**infoSE\_", subj\_code[i], ".pdf"))

**print**(itemplot(gpcm, item = i, type = ’infoSE’, CE = T,

60 xlim = **c**(-6.5,6.5),

main = paste0(

"Information and SE for ", subj\_code[i], " (", subj\_name[i], ")"

)

))

65 **dev**.**off**()

}

# IRT Analysis Output

**Table 1**

*Partial Credit Model (PCM) Parameter Estimates*

Subject Code Subject Name *b*1 *b*2 *b*3 *b*4 *b*5

ENGW Written English *−*5*.*189

NORW

NORO

Written Norwegian

Oral Norwegian

*−*5*.*953 *−*3*.*062 *−*0*.*805

(0*.*065) (0*.*021) (0*.*015)

*−*6*.*313 *−*3*.*724 *−*1*.*633

(0*.*087) (0*.*026) (0*.*016)

(0*.*015)

1*.*205

(0*.*024)

3*.*605

(0*.*014)

0*.*242

(0*.*018)

2*.*622

(0*.*049)

ENGO Oral English *−*5*.*655

(0*.*067)

MATH

NATS

Mathematics

Natural Sciences

*−*4*.*934 *−*1*.*856 *−*0*.*213

(0*.*037) (0*.*017) (0*.*015)

*−*5*.*724 *−*3*.*085 *−*1*.*226

(0*.*059) (0*.*022) (0*.*016)

(0*.*016)

1*.*111

(0*.*021)

3*.*028

(0*.*014)

0*.*345

(0*.*018)

2*.*433

*−*2*.*967

(0*.*021)

*−*3*.*618

(0*.*026)

*−*1*.*017

(0*.*015)

*−*1*.*788

(0*.*016)

0*.*957

(0*.*015)

0*.*310

(0*.*014)

3*.*138

(0*.*021)

2*.*618

(0*.*018)

SOCS Social Sciences *−*6*.*065

(0*.*074)

RELI Religion and Ethics *−*5*.*822

(0*.*066)

MUSI

Music

*−*6*.*305 *−*4*.*288 *−*2*.*471 *−*0*.*198 2*.*557

(0*.*104) (0*.*034) (0*.*018) (0*.*014)

(0*.*017)

*−*3*.*519

(0*.*025)

*−*3*.*374

(0*.*024)

*−*1*.*638

(0*.*016)

*−*1*.*583

(0*.*016)

0*.*071

(0*.*014)

0*.*152

(0*.*014)

2*.*333

(0*.*017)

2*.*325

(0*.*017)

HAND Arts and Handcraft *−*6*.*447

(0*.*117)

FOOD

Food and Health

*−*7*.*768 *−*5*.*455 *−*2*.*791 *−*0*.*336 2*.*485

(0*.*274) (0*.*050) (0*.*019) (0*.*014)

(0*.*017)

*−*4*.*599

(0*.*037)

*−*2*.*522

(0*.*018)

*−*0*.*143

(0*.*014)

2*.*687

(0*.*018)

PHED Physical Education *−*6*.*221

(0*.*114)

*−*4*.*607

(0*.*041)

*−*3*.*035

(0*.*021)

*−*0*.*730

(0*.*014)

2*.*265

(0*.*016)

*Note.* A partial credit model (PCM) computes the difficulty (*b*) parameters for each subject. Standard errors are enclosed in parenthesis below point estimates. All estimates are significant at *.*001 level.

# Table 2

*Generalised Partial Credit Model (GPCM) Parameter Estimates*

Subject Code Subject Name *a b*1 *b*2 *b*3 *b*4 *b*5

ENGW Written English 1*.*790

NORW

NORO

Written Norwegian

Oral Norwegiani

(0*.*025)

3*.*021 *−*2*.*882 *−*1*.*535 *−*0*.*403

(0*.*026) (0*.*010) (0*.*006)

(0*.*007)

0*.*627

(0*.*011)

1*.*789

(0*.*028)

3*.*346 *−*3*.*024 *−*1*.*845 *−*0*.*817

(0*.*031) (0*.*011) (0*.*007)

(0*.*006)

0*.*154

(0*.*008)

1*.*300

(0*.*015)

ENGO Oral English 1*.*689

(0*.*014)

MATH

NATS

Mathematics

Natural Sciences

(0*.*014)

1*.*715 *−*2*.*773 *−*1*.*027 *−*0*.*113

(0*.*024) (0*.*010) (0*.*009)

(0*.*009)

0619

(0*.*013)

1*.*694

(0*.*022)

2*.*656 *−*2*.*877 *−*1*.*602 *−*0*.*640

(0*.*026) (0*.*010) (0*.*007)

(0*.*007)

0*.*206

(0*.*009)

1*.*266

*−*2*.*875

(0*.*029)

*−*3*.*185

(0*.*041)

*−*1*.*638

(0*.*013)

*−*2*.*024

(0*.*016)

*−*0*.*559

(0*.*009)

*−*1*.*001

(0*.*010)

0*.*534

(0*.*008)

0*.*175

(0*.*008)

1*.*741

(0*.*013)

1*.*473

(0*.*011)

SOCS Social Sciences 3*.*397

(0*.*028)

RELI Religion and Ethics 3*.*154

(0*.*026)

MUSI

Music

(0*.*011)

1*.*331 *−*3*.*832 *−*2*.*558 *−*1*.*493 *−*0*.*140 1*.*559

(0*.*077) (0*.*026) (0*.*014) (0*.*009)

(0*.*013)

*−*2*.*920

(0*.*027)

*−*2*.*850

(0*.*025)

*−*1*.*766

(0*.*010)

*−*1*.*715

(0*.*011)

*−*0*.*826

(0*.*007)

*−*0*.*800

(0*.*007)

0*.*074

(0*.*006)

0*.*108

(0*.*006)

1*.*170

(0*.*008)

1*.*181

(0*.*008)

HAND Arts and Handcraft 1*.*138

(0*.*010)

FOOD

Food and Health

(0*.*012)

1*.*429 *−*4*.*565 *−*3*.*263 *−*1*.*644 *−*0*.*208 1*.*479

(0*.*173) (0*.*037) (0*.*014) (0*.*009)

(0*.*012)

*−*4*.*129

(0*.*101)

*−*2*.*924

(0*.*032)

*−*1*.*616

(0*.*016)

*−*0*.*124

(0*.*010)

1*.*751

(0*.*016)

PHED Physical Education 0*.*804

(0*.*008)

*−*4*.*483

(0*.*138)

*−*3*.*264

(0*.*050)

*−*2*.*295

(0*.*026)

*−*0*.*658

(0*.*015)

1*.*750

(0*.*019)

*Note.* A generalised partial credit model (GPCM) computes the discrimination (*a*) and difficulty (*b*) parameters for each subject. Standard errors are enclosed in parenthesis below point estimates. All estimates are significant at *.*001 level.