Self-Assessment Works! Paper Analyze Data

Steven J. Pierce & Xiaowan Zhang

Contents

1 Purpose

This file reproduces the results reported in our manuscript (Winke, Zhang, & Pierce, 2022), which was based on a presentation (Winke, Pierce, & Zhang 2018). It analyzes data on Spanish language learners who took a Can-Do self-assessment test, along with the more authoritative OPIc language proficiency test (Winke & Zhang, 2022). We did both correlation analyses and continuation-ratio models that examine the effect of course and OPIc speaking proficiency scores on the passing rate for each level of the Can-Do statements self-assessment. The objective was to validate the Can-Do test results.

1.1 Target Journal

We submitted this as a "Research Report" to a journal called Studies in Second Language Acquisition (SSLA), https://www.cambridge.org/core/journals/studies-in-second-language-acquisition. The author instructions are at https://www.cambridge.org/core/journals/studies-in-second-language-acquisition/information/instructions-contributors.

We applied for the SSLA *Open Data Badge* and the *Open Materials Badge* described at https://www.cambridge.org/core/journals/studies-in-second-language-acquisition/open-science-badges . Hosting a public Github repository should meet the requirements for both badges because GitHub is on the Registry of Research Data Repositories. However, we deposited the data in another public repository (Winke & Zhang, 2022) and just use GitHub for the code.

1.2 Comment on Statistical Methodology

We experimented with applying ideas discussed in Wasserstein, Schirm, and Lazar (2019) about abandoning declarations of statistical significance. This landmark editorial paper produced by the American Statistical Association discussed various ways to supplement p-values with additional statistics or replace them altogether. We computed some supplemental statistics and include them in this set of materials but omitted them from the manuscript.

We computed Shannon information values (s-values; Greenland, 2019; Wasserstein, Schirm, & Lazar, 2019). S-values are a rescaling of p-values, such that $s = -log_2(p)$. They can range from $s = 0$ when $p = 1$ to $s = \infty$ when $p = 0$. Larger s-values correspond to greater evidence against the null hypothesis. An s-value can be interpreted as how many bits of information there are against the hypothesis. To make that easier to understand, suppose you want to flip a coin repeatedly to determine whether it is a fair coin rather than one biased toward landing on heads. Each flip provides one bit of information, but only those resulting in heads are information against the null hypothesis. A fair coin should yield heads half of the time and tails the other half (independent with probability = .50 for each outcome on each flip). The null hypothesis is that the coin is fair. Seeing 2 consecutive heads come up in a set of 2 flips ($p = 0.5^2 = 0.25, s = 2$) would be weak evidence against fairness but an s-value of 10 would be more persuasive, like getting 10 heads from a set of 10 coin flips ($p = 0.5^{10} = 0.0009765625, s = 10$).

Other ideas that look viable to use are reporting minimum false positive risk (mFPR; Colquhoun, 2009), Bayes Factor Bounds (BFB; Benjamin & Berger, 2019), and the upper bound for the posterior probability that that alternate hypothesis (H1) is true (Benjamin & Berger, 2019). We have opted to use the latter two instead of mFPR.

Taking the Can-Do self-assessment yields an ordinal score ranging from 1-5 that represents the proficiency level of the learner who took the test. Higher levels indicate greater proficiency. We can conceptualize the process that yields that score as a sequential selection process comprised of a set of four level transition testlets (LTTs) that must be passed in strict order. Every learner starts at level 1 and only advances to level 2 by passing the first LTT. The learner's proficiency level is incremented for each LTT passed. The self-assessment ends as soon as the learner fails to pass an LTT, or reaches the maximum proficiency level (level 5). For a sample of N learners, there will be between N and $4N$ binary LTT results (0 = fail, 1 = pass) depending on how many learners passed each LTT.

We use continuation-ratio models to examine the proficiency levels achieved by the learners. These are simply logistic regression models applied to a reorganized dataset with one row of data per LTT attempted by each learner. We are using a predictive modeling approach here that tests how well OPIc scores and course predict self-assessed proficiency levels. Therefore, we evaluate the models according to both discrimination and calibration criteria relevant to predictive models (Fenlon, O'Grady, Doherty, & Dunnion, 2018).

2 Setup

Set global R chunk options (local chunk options will over-ride global options).

```
# Create a custom chunk hook/option for controlling font size in chunk & output.
def.chunk.hook  <- knitr::knit_hooks$get("chunk")
knitr::knit_hooks$set(chunk = function(x, options) {
  x <- def.chunk.hook(x, options)
  ifelse(options$cfsize != "normalsize", paste0("\n \\", options$cfsize,"\n\n",
                                              x, "\n\n \\normalsize"), x)
  })

# Global chunk options (over-ridden by local chunk options)
knitr::opts_chunk$set(include  = TRUE, echo = TRUE, error = TRUE,
                      message = TRUE, warning = TRUE, fig.pos = "!ht",
                      cfsize = "footnotesize")

# Declare location of this script relative to the project root directory.
here::i_am(path = "inst/SAW_Paper_Analyze_Data.Rmd")
```

```
## here() starts at P:/Consulting/FY18/Winke_Paula/18-009/SAWpaper
```

2.1 Load Packages and Set Package Options

Load contributed R packages that we need to get additional functions.

```
library(here)              # for here()
library(polycor)           # for hetcor()
library(car)               # for residualPlots(), influenceIndexPlot(), outlierTest()
```

```
## Loading required package: carData
```

```
library(multcomp)          # for glht()
```

```
## Loading required package: mvtnorm

## Loading required package: survival

## Loading required package: TH.data
## Loading required package: MASS
```

```
##
## Attaching package: 'TH.data'
```

```
## The following object is masked from 'package:MASS':
##
##     geyser
```

```
library(visreg)            # for visreg()
# Set package options.
# options(knitr.kable.NA = '0.00')
library(rmarkdown)         # for render(), pandoc_version().
library(knitr)             # for kable()
library(texreg)            # for texreg()
```

```
## Version:  1.38.5
## Date:     2022-03-03
## Author:   Philip Leifeld (University of Essex)
##
## Consider submitting praise using the praise or praise_interactive functions.
## Please cite the JSS article in your publications -- see citation("texreg").
```

```
library(pROC)              # for roc()
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(ggplot2)           # for ggplot()
library(tidyr)             # for unite()
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:texreg':
##
##     extract
```

```
library(dplyr)             # for filter(), select(), etc.
```

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

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following object is masked from 'package:car':
##
##     recode
```

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

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

```
library(directlabels)      # for direct.label()
library(lattice)           # for strip.custom()
library(modEvA)            # for HLfit()
# Set package options.
options(kableExtra.latex.load_packages = FALSE)
library(kableExtra)        # for kable_styling(), add_header_above(),
```

```
##
## Attaching package: 'kableExtra'

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

```
                           # column_spec(), collapse_rows(), and landscape()
library(Hmisc)             # for rcorr()
```

```
## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(piercer)           # for p2s(), p2bfb(), p2pp(), convertp(), lrcm(),
                           # brier(), r.p(), r.pc(), r.pc(), ci.rpc(), ci.rp(),
                           # git_report(), which_latex(), extract_glm()
library(stringr)           # for word()
library(broom)             # for tidy()
library(SAWpaper)          # for package version number via session_info()
```

2.2 Load Data

The file `SAW_Paper_Data.RData` contains four data frames: `SData`, `TAData`, `VSData`, and `VTAData`. Table 1 shows the number of observations (rows) and variables in each of these datasets.

```
load(file = here::here("data/SAW_Paper_Data.RData"))
```

```
data.frame(Dataset = c("SData", "VSData", "TAData", "VTAData"),
           N.Rows = c(nrow(SData), nrow(VSData), nrow(TAData), nrow(VTAData)),
           N.Vars = c(ncol(SData), ncol(VSData), ncol(TAData), ncol(VTAData))) %>%
  kable(format = "latex", booktabs = TRUE, digits = 0,
        col.names = c("Data Frame", "Rows", "Variables"),
        caption = "Data Frame Sizes") %>%
  add_header_above(header = c("", "Number of ..." = 2)) %>%
  group_rows(group_label = "Student data", start_row = 1, end_row = 2) %>%
  group_rows(group_label = "Testlet attempt data", start_row = 3, end_row = 4)
```

The `SData` data frame contains the cleaned student-level data for all Spanish students in the study, including those with invalid OPIc scores, while the `VSData` data frame is the subset of data from `SData` containing only data from students who had valid OPIc scores. We use the latter for our correlation analyses. Meanwhile, the `TAData` data frame contains data about the same set of students shown in `SData` but has been expanded to one row per student per self-assessment testlet attempted. The `VTAData` data frame is the subset of `TAData` containing only data for the students included in `VSDdata`. Therefore, we use `VTAData` in our continuation-ratio models.

	Number of ...	
Data Frame	Rows	Variables
Student data		
SData	871	69
VSData	807	69
Testlet attempt data		
TAData	1513	10
VTAData	1320	10

Table 1: Data Frame Sizes

2.3 Data Structure Comments

In SData (and thus also in VSData), Level (and LevelF, which is just a copy of Level that is stored as a factor instead of a numeric variable) denote the five levels of Can-Do statements on the self-assessment, whereas in TAData (and thus also in VTAData) the Testlet variable represents the transition testlets that control passage from one level to the next. The relationship between these variables is illustrated in this diagram: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$. Level in SData is represented by the numbers, and Testlet in TAData is represented by the arrows, which can be numbered 1-4 sequentially from left to right.

Unlike SData and VSData where there is one row per learner with Level indicating the maximum level a learner reached on the self-assessment, TAData and VTAData have been expanded such that there is one row per learner per transition testlet attempted. The binary Pass variable in VTAData indexes whether a learner passed/failed to pass a given transition testlet. For instance, a learner who reached Level 3 on the self-assessment would have one row in VSData with Level equal to 3. The same learner would have three rows in VTAData, having a 1 on Pass for the rows representing testlets 1 and 2, and a 0 on Pass for the row representing testlet 3, which was highest transition attempted.

The data in VTAData is set up as described above for the application of continuation-ratio modeling, which models the probabilities of passing the four transitions (represented by Testlet in VTAData) as a function of predictors (which, in the case of our study, are testlet, course and OPIc speaking scores). Each of those transitions will eventually yield an estimate of the conditional pass rate, which is the probability that an individual who reached the level on the left end of the arrow succeeds in passing on to the level on the right end. Continuation-ratio models can be conducted as logistic regressions where the data frame has one row of data per person for each level transition that the individual actually attempted.

We treat OPIc speaking test proficiency scores as a continuous covariate that ranges from Novice-low to Superior. The variable OPICA in VSData shows the actual OPIc ratings. Numerically transformed OPIc scores are captured by OPICV in VSData (NL = 1, NM = 2, NH = 3, …, AM = 8, AH = 9, Superior = 10). We removed the cases for which the OPIc ratings were not meaningful for this study (AR, BR, and UR).

As readers may notice, some students with OPIc scores did not have any numerically transformed OPIc values (i.e., OPICV = NA in SData). This is because these students did not receive a valid score on the test. Specifically, these students received either above range (AR), below range (BR), or unratable (UR) on the OPIc. AR was most likely awarded when a student selected an test form that was well below their oral proficiency level. BR was most likely awarded when a student selected a test form that was well above their proficiency level. And UR was given when a student's oral response was not ratable due to one of a variety of reasons (e.g., no response, technology failure). For transparency purposes, we include those students with non-scored OPIc tests in our datasets, but we have to exclude them from the main analyses because their proficiency levels as measured by the OPIc were unknown. See the output from our inst/SAW_Paper_Import_Explore_Data.Rmd script for descriptive data on the students with invalid OPIc scores.

For analysis purposes, we treat the variables of Course, Level, and Testlet as factors that are effectively ordinal variables and we treat OPICV as a continuous variable. To make model coefficients more interpretable, we included the centered form of OPIc scores stored in COPIC instead of OPICV in the continuation-ratio models. A score of zero in COPIC represents the Intermediate-mid level on the OPIc scale.

3 Data Visualization

We visualized the relationship of SA total scores with OPIc scores and course level in the VSData dataset using scatterplots. We put students' OPIc scores on the x-axis and their total SA scores (ranging from 1 to 200) on the Y-axis. We assigned different colors to individual data points based on course level. We also jittered the data points so that overlapping points are nudged apart and can be seen more clearly. We added fit lines to the scatterplots to examine the shape of the relationship between SA level and OPI scores. These results are not discussed in the manuscript but were part of our preliminary examination of the data.

3.1 Overall Relationship

In Figure 1, we start with a linear regression line to visually check how well a simple parametric form represents the overall relationship. Then in Figure 2, we added a loess curve instead. A loess curve is a smooth fit line based on local regression of a dependent variable (Y-variable) on an independent variable (X- variable). It helps one to see non-linearity in the relationship between variables. Here we substantial evidence if non-linearity because the loess fit line is quite bowed rather than mostly straight.

```r
FCap <- paste("\\label{fig:plot-SAScore-OPICV-linear}",
              "Self-Assessment Total Score as a function of OPIc Score,",
              "With Linear Fit Line.")

# Objects to store settings for plots.
opic.breaks <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
opic.labels <- c("NL\n1", "NM\n2", "NH\n3", "IL\n4", "IM\n5", "IH\n6",
                 "AL\n7", "AM\n8", "AH\n9")
sata.breaks <- c(40, 80, 120, 160, 200)
sata.labels <- c("40\n(Level 1)", "80 \n(Level 2)", "120\n(Level 3)",
                 "160\n(Level 4)", "200\n(Level 5)")

# Linear regression lines added to jittered and colored-coded plots
ggplot(VSData, aes(OPICV, Item1_50))+
  geom_jitter(aes(color=Course))+
  geom_smooth(method="lm", se=FALSE)+
  xlim(1,9)+
  scale_x_continuous(breaks = opic.breaks, labels = opic.labels)+
  scale_y_continuous(breaks = sata.breaks, labels = sata.labels)+
  xlab("OPIc score")+
  ylab("Self-assessment total score")+
  guides(color = guide_legend(title="Course"))
```

```r
FCap <- paste("\\label{fig:plot-SAScore-OPICV-loess}",
              "Self-Assessment Total Score as a function of OPIc Score,",
              "With Loess Fit Line.")

# Smooth loess lines added to jittered and colored-coded scatterplots
ggplot(VSData, aes(OPICV, Item1_50))+
  geom_jitter(aes(color=Course))+
  geom_smooth(method="loess", se=FALSE, span = .77)+
  xlim(1,9)+
  scale_x_continuous(breaks = opic.breaks, labels = opic.labels)+
  scale_y_continuous(breaks = sata.breaks, labels = sata.labels)+
  xlab("OPIc score")+
  ylab("Self-assessment total score")+
  guides(color = guide_legend(title="Course"))
```

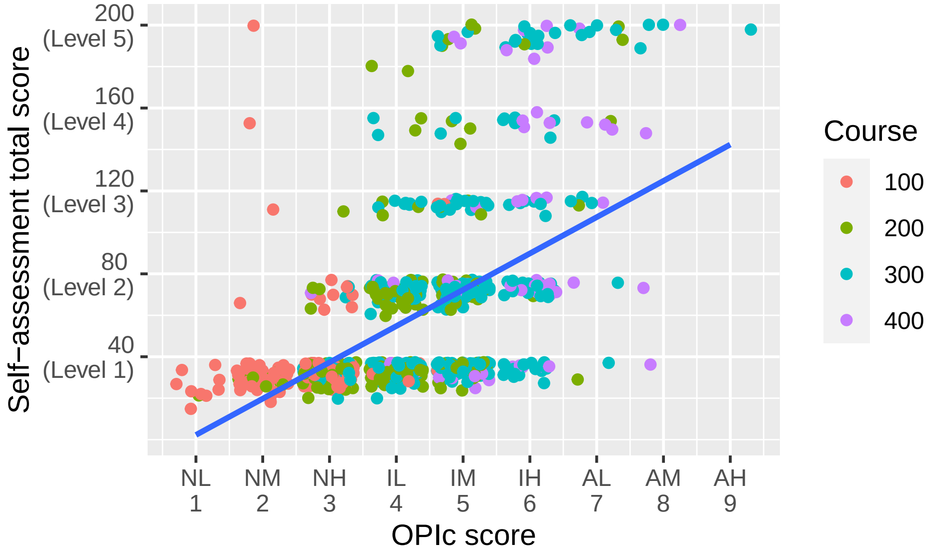

Figure 1: Self-Assessment Total Score as a function of OPIc Score, With Linear Fit Line.

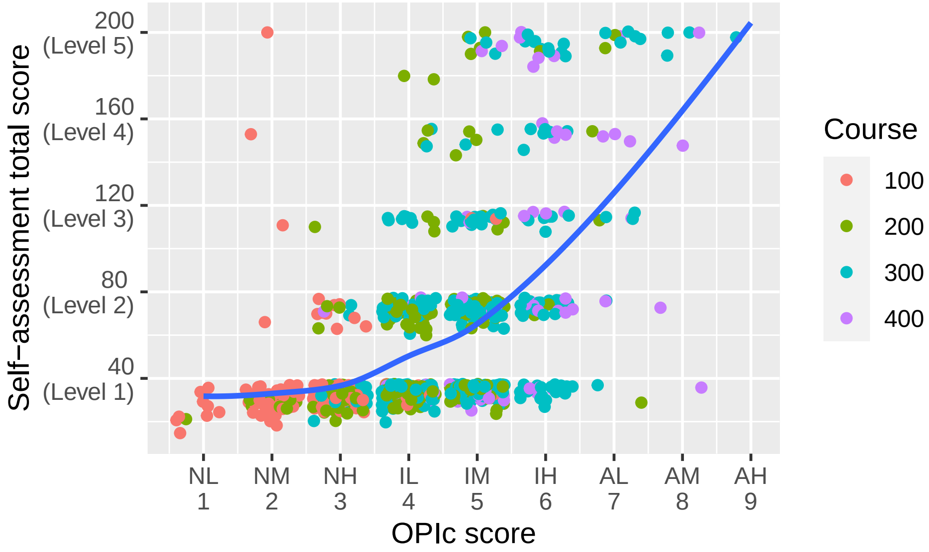

Figure 2: Self-Assessment Total Score as a function of OPIc Score, With Loess Fit Line.

3.2 Relationships Stratified by Course

Next, we tried stratifying the relationships by course. Figure 3 shows four linear regression lines: one for each course level. Compare that to Figure 4 where we show separate loess curves for each of the four course levels instead.

```
FCap <- paste("\\label{fig:plot-SAScore-OPICV-by-Course-linear}",
              "Self-Assessment Total Score as a function of OPIc Score and",
              "Course, With Linear Fit Lines.")

ggplot(VSData, aes(OPICV, Item1_50, color=Course))+
  geom_jitter()+
  geom_smooth(method="lm", se=FALSE)+
  xlim(1,9)+
  scale_x_continuous(breaks = opic.breaks, labels = opic.labels)+
  scale_y_continuous(breaks = sata.breaks, labels = sata.labels)+
  xlab("OPIc score")+
  ylab("Self-assessment total score")+
  guides(color = guide_legend(title="Course"))
```

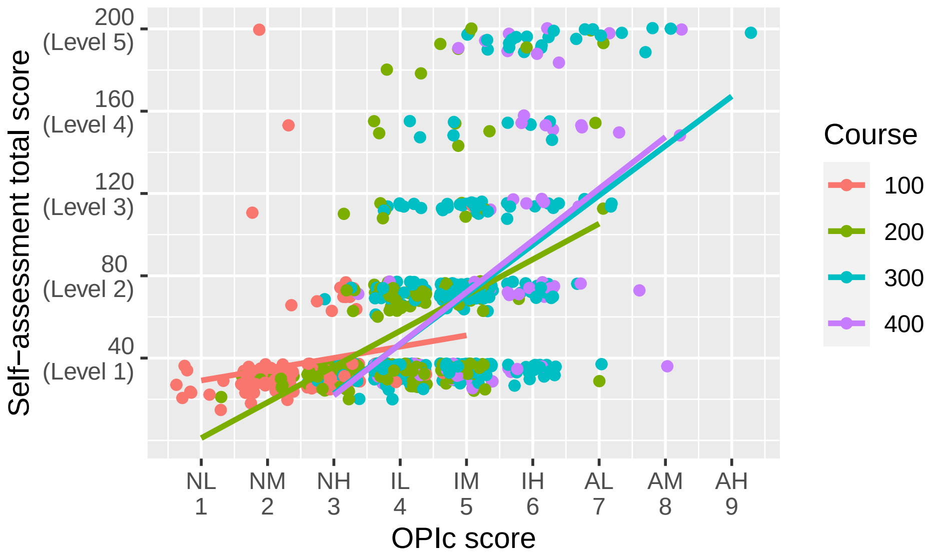

Figure 3: Self-Assessment Total Score as a function of OPIc Score and Course, With Linear Fit Lines.

```
FCap <- paste("\\label{fig:plot-SAScore-OPICV-by-Course-loess}",
              "Self-Assessment Total Score as a function of OPIc Score and",
              "Course, With Loess Fit Lines.")

ggplot(VSData, aes(OPICV, Item1_50, color=Course))+
  geom_jitter()+
  geom_smooth(method="loess", se=FALSE, span = .93)+
  xlim(1,9)+
  scale_x_continuous(breaks = opic.breaks, labels = opic.labels)+
  scale_y_continuous(breaks = sata.breaks, labels = sata.labels)+
  xlab("OPIc score")+
  ylab("Self-assessment total score")+
  guides(color = guide_legend(title="Course"))
```

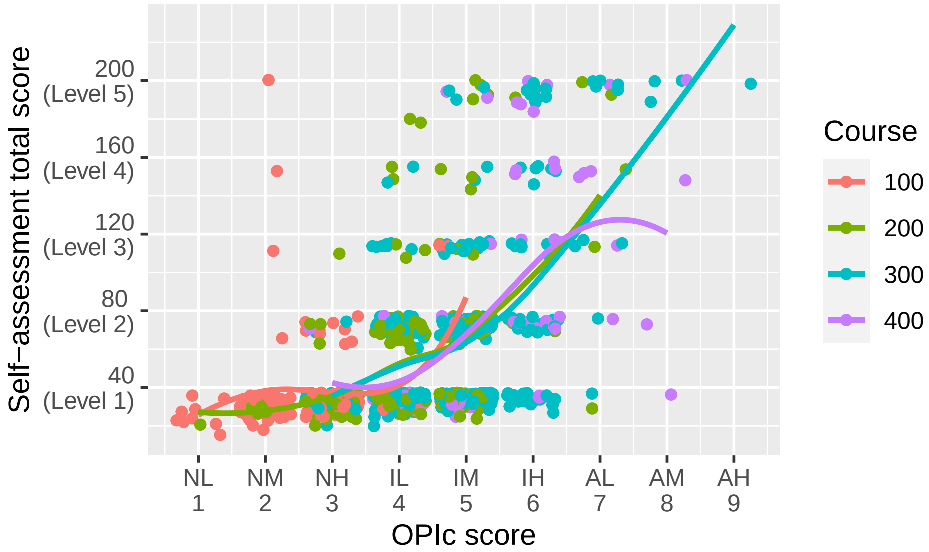

Figure 4: Self-Assessment Total Score as a function of OPIc Score and Course, With Loess Fit Lines.

4 Correlations

We examine the correlations among course level, OPIc scores, and SA level using the student-level dataset (`VSData`). As we mentioned earlier, `Level` represents the highest level that a student reached on the SA. Polyserial correlation is calculated for `OPICV` and `Course` and for `OPICV` and `Level`, whereas polychoric correlation is calculated for the two ordinal variables, `Course` and `Level`. We included both ordinal and continuous versions of selected variables to simplify getting both polychoric and polyserial correlations and (for comparison purposes only) Pearson correlations that are less appropriate for measuring some relationships. The names of continuous versions of variables that should otherwise be treated as ordinal have ".n" suffixes. We also estimated selected Spearman correlations for a similar comparative purpose.

```
# Create numeric versions of variables to meet hetcor() function requirements.
VSData <- VSData %>%
  mutate(OPIC.n = as.numeric(OPICV),
         Level.n = as.numeric(Level),
         Course.n = as.numeric(Course))

# cvars = continuous variables
cvars <- c("OPIC.n", "Item1_50", "Level.n", "Course.n")

# ovars = ordinal variables
ovars <- c("LevelF", "Course")

# Use the hetcor (heterogenous correlations) in "polycor" package to get a
# matrix with Pearson correlations (if both variables are continuous),
# polyserial correlations (if one is continuous and the other is ordinal), and
# polychoric correlations (if both are ordinal). Warning messages from hetcor()
# occur because polyserial correlations between ordinal and continuous versions
# of the same variable are not sensible. It was just faster to estimate them as
# part of a single larger matrix than individually get the subset we want.
# So, we disabled reporting warnings from this chunk.

HC <- hetcor(as.data.frame(VSData[, c(cvars, ovars)]), ML=TRUE,
             use="pairwise.complete.obs")
```

Table 2 shows the polyserial and polychoric correlations, along with confidence intervals and associated statistics.

```
rps.rpc <- rbind(r.ps(HC, cont = cvars, ord = ovars, digits = 2,
                      pdigits = NULL),
               r.pc(HC, ord = ovars, digits = 2, pdigits = NULL)) %>%
  mutate(Variables = word(rownames(.), 2, sep = "\\:"),
         Type = word(rownames(.), 1, sep = "\\:"),
         Pval = format(Pval, digits = 3)) %>%
  select(Variables, Type, Cor, SE, CI.LL, CI.UL, Z, Pval, Sval, BFB, PPH1)

kable(rps.rpc, format = "latex", booktabs = TRUE, row.names = FALSE,
      digits = c(Inf, Inf, rep(x = 2, times = 5), Inf, c(2, 2, 2)),
      caption = "Polyserial and Polychoric Correlations") %>%
kable_styling(latex_options = c("repeat_header"))
```

Variables	Type	Cor	SE	CI.LL	CI.UL	Z	Pval	Sval	BFB	PPH1
OPIC.n, LevelF	r.ps	0.61	0.03	0.56	0.66	23.68	0	Inf	NaN	NaN
OPIC.n, Course	r.ps	0.68	0.02	0.64	0.72	35.82	0	Inf	NaN	NaN
Item1_50, LevelF	r.ps	NA	NA	NA	NA	NA	NA	NA	NA	NA
Item1_50, Course	r.ps	0.32	0.03	0.25	0.38	9.37	0	Inf	NaN	NaN
Level.n, LevelF	r.ps	NA	NA	NA	NA	NA	NA	NA	NA	NA
Level.n, Course	r.ps	0.30	0.03	0.23	0.36	8.68	0	Inf	NaN	NaN
Course.n, LevelF	r.ps	0.36	0.04	0.29	0.43	9.93	0	Inf	NaN	NaN
Course.n, Course	r.ps	NA	NA	NA	NA	NA	NA	NA	NA	NA
LevelF, Course	r.pc	0.38	0.04	0.30	0.45	9.64	0	Inf	NaN	NaN

Table 2: Polyserial and Polychoric Correlations

Obtain Pearson correlations among two continuous variables, OPIC and Item1_50 (SA sum scores ranging from 1 to 200), and two ordinal variables, Course and Level. Note that it is inappropriate to use Pearson's r for ordinal variables. We include them here only for comparison purposes.

```
rp <- r.p(HC, cont = cvars, digits = 2, pdigits = NULL)
```

Now obtain Spearman correlations for comparison purposes even though they are not technically appropriate for our purpose.

```
rsa <- ci.rp(r = cor(x = VSData$Level.n, y = VSData$Course.n,
                     method = "spearman"),
             n = nrow(VSData), rn = "r.s: Level and Course")
rsb <- ci.rp(r = cor(x = VSData$Level.n, y = VSData$OPIC.n,
                     method = "spearman"),
             n = nrow(VSData), rn = "r.s: Level and OPIC")
rsc <- ci.rp(r = cor(x = VSData$Course.n, y = VSData$OPIC.n,
                     method = "spearman"),
             n = nrow(VSData), rn = "r.s: Course and OPIC")
```

Table 3 shows the Pearson and Spearman correlations, along with confidence intervals and associated statistics.

```
# Bind into a data frame
rp.rs <- rbind(rp, rsa, rsb, rsc) %>%
  mutate(Variables = word(rownames(.), 2, sep = "\\:"),
         Type = word(rownames(.), 1, sep = "\\:"),
         Pval = format(Pval, digits = 3),
         BFB = format(BFB, digits = 3))  %>%
  select(Variables, Type, Cor, SE, CI.LL, CI.UL, t, df, Pval, Sval, BFB,
         PPH1)

kable(rp.rs, format = "latex", booktabs = TRUE, row.names = FALSE,
      digits = c(rep(x = 2, times = 8), Inf, c(2, 2, 2)),
      caption = "Pearson and Spearman Correlations") %>%
kable_styling(latex_options = c("repeat_header"))
```

Notice that the t and z-statistics are all really large, so the p-values are zero. That causes all the s-values to become infinite, and the BFB values to become NaN (not-a-number) because we are dividing by zero

Variables	Type	Cor	SE	CI.LL	CI.UL	t	df	Pval	Sval	BFB	PPH1
OPIC.n, Item1_50	r.p	0.53	0.03	0.47	0.57	17.53	805	0.00e+00	Inf	NaN	NaN
OPIC.n, Level.n	r.p	0.50	0.03	0.45	0.55	16.45	805	0.00e+00	Inf	NaN	NaN
OPIC.n, Course.n	r.p	0.65	0.03	0.60	0.68	23.99	805	0.00e+00	Inf	NaN	NaN
Item1_50, Level.n	r.p	1.00	0.00	0.99	1.00	299.99	805	0.00e+00	Inf	NaN	NaN
Item1_50, Course.n	r.p	0.29	0.03	0.23	0.36	8.71	805	0.00e+00	Inf	NaN	NaN
Level.n, Course.n	r.p	0.28	0.03	0.21	0.34	8.13	805	1.78e-15	49	6.1e+12	1
Level and Course	r.s	0.29	0.03	0.23	0.36	8.71	805	0.00e+00	Inf	NaN	NaN
Level and OPIC	r.s	0.50	0.03	0.45	0.55	16.41	805	0.00e+00	Inf	NaN	NaN
Course and OPIC	r.s	0.63	0.03	0.59	0.67	22.99	805	0.00e+00	Inf	NaN	NaN

Table 3: Pearson and Spearman Correlations

in the denominator. You can read more about interpreting s-values and BFB values in Greenland (2019), Benjamin & Berger (2019), and Wasserstein, Schirm, & Lazar (2019).

5 Fit Continuation-Ratio Models

Here we fit a series of continuation-ratio models to `VTAData` to see whether course level and OPIc scores significantly predict the conditional pass rate for each transition on the SA. A parallel effect for a predictor is one where the predictor's effect is constant across the transition testlets while a non-parallel effect allows it to vary across testlets. Models 1a and 1b focus primarily on whether there is a parallel or non-parallel course effect. Models 2a and 2b primarily test whether there is a parallel or non-parallel effect of OPIc score *and are the models reported in our published manuscript*. Finally, Models 3a and 3b include both course and OPIc score as predictors.

```
# Model 1a: Parallel course effect
m1a <- glm(Pass ~ Testlet + Course - 1, data = VTAData, family = "binomial")

# Model 1b: Non-parallel course effect
m1b <- glm(Pass ~ Testlet + Course + Testlet*Course - 1, data = VTAData,
           family = "binomial")

# Model 2a: Parallel OPIC effect
m2a <- glm(Pass ~ Testlet + COPIC - 1, data = VTAData, family = "binomial")

# Model 2b: Non-parallel OPIC effect
m2b <- glm(Pass ~ Testlet + COPIC + Testlet*COPIC - 1, data = VTAData,
           family = "binomial")

# Model 3a: Parallel OPIC + parallel course effect
m3a <- glm(Pass ~ Testlet + COPIC + Course - 1, data = VTAData,
           family = "binomial")

# Model 3b: Non-parallel OPIC + parallel course effect
m3b <- glm(Pass ~ Testlet + COPIC + Testlet:COPIC + Course - 1, data = VTAData,
           family = "binomial")

# Create a list of the model fit objects.
TModels <- list(m1a, m1b, m2a, m2b, m3a, m3b)

# Create a list of texreg objects for the models.
TModels.TR <- lapply(X = TModels, FUN = extract_glm)
```

Table 4 shows the model parameters for the whole set of models, along with various goodness of fit statistics. The Brier score is an overall measure of accuracy suitable for logistic regression models with binary outcomes (Fenlon et al., 2018). It is the average prediction error (Steyerberg et al., 2001; Steyerberg et al., 2010). The scaled Brier score adjusts the unscaled version to have a range of from 0 to 1, thereby making it similar to an R^2 statistic. Values close to 1 are desirable and indicate good calibration. We report the scaled Brier score below. Pseudo-R^2 (R_p^2) is the squared Pearson correlation between observed and predicted values, as suggested in Hosmer, Lemeshow, & Sturdivant (2013, p. 182). Meanwhile, R_{Dev}^2 is a measure based on deviance residuals (Fox, 1997, p. 451; Cameron & Windmeijer, 1997).

	Model 1a	Model 1b	Model 2a	Model 2b	Model 3a	Model 3b
Testlet1	−1.72 (0.23)***	−2.05 (0.27)***	0.13 (0.09)	0.19 (0.09)*	0.22 (0.28)	0.43 (0.30)
Testlet2	−2.15 (0.26)***	−0.69 (0.55)	−0.70 (0.12)***	−0.69 (0.13)***	−0.62 (0.30)*	−0.46 (0.31)
Testlet3	−1.34 (0.30)***	−0.41 (0.91)	−0.15 (0.21)	−0.01 (0.21)	−0.06 (0.33)	0.22 (0.35)
Testlet4	−1.05 (0.35)**	0.00 (1.41)	−0.06 (0.29)	0.31 (0.30)	0.01 (0.38)	0.52 (0.40)
Course200	1.24 (0.25)***	1.55 (0.30)***			0.01 (0.28)	−0.14 (0.29)
Course300	1.66 (0.24)***	2.05 (0.29)***			−0.13 (0.29)	−0.29 (0.30)
Course400	2.08 (0.28)***	2.43 (0.36)***			−0.14 (0.34)	−0.28 (0.35)
Testlet2:Course200		−2.00 (0.67)**				
Testlet3:Course200		−0.52 (1.06)				
Testlet4:Course200		−1.14 (1.54)				
Testlet2:Course300		−1.86 (0.64)**				
Testlet3:Course300		−1.67 (0.99)				
Testlet4:Course300		−1.26 (1.49)				
Testlet2:Course400		−1.69 (0.72)*				
Testlet3:Course400		−1.14 (1.08)				
Testlet4:Course400		−2.07 (1.54)				
COPIC			0.84 (0.06)***	0.96 (0.08)***	0.86 (0.07)***	1.01 (0.10)***
Testlet2:COPIC				−0.17 (0.16)		−0.19 (0.16)
Testlet3:COPIC				−0.41 (0.20)*		−0.44 (0.20)*
Testlet4:COPIC				−0.65 (0.22)**		−0.66 (0.23)**
Null model deviance	1829.91	1829.91	1829.91	1829.91	1829.91	1829.91
Null model df	1320	1320	1320	1320	1320	1320
Log Likelihood	−844.79	−836.11	−761.34	−756.22	−760.91	−755.51
AIC	1703.58	1704.21	1532.68	1528.43	1537.81	1533.02
BIC	1739.88	1787.18	1558.61	1569.91	1579.29	1590.06
Deviance	1689.58	1672.21	1522.68	1512.43	1521.81	1511.02
Residual df	1313	1304	1315	1312	1312	1309
No. observations	1320	1320	1320	1320	1320	1320
Brier score	0.08	0.09	0.20	0.20	0.20	0.20
Pseudo-R^2 (R_p^2)	0.08	0.09	0.20	0.20	0.20	0.20
R^2_{Dev}	0.08	0.09	0.17	0.17	0.17	0.17

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$
Only models 2a and 2b were reported in our paper.

Table 4: Continuation-Ratio Model Parameter Estimates, Standard Errors, and Goodness of Fit Statistics

In subsequent sections, we examine each of the fitted models in detail by extracting, post-processing, and displaying model parameters, predicted values, and supplemental statistics, hypothesis tests, and model diagnostics. For example, we examine sequential tests of model terms based on Type I sums of squares. Those test the significance of unique additional variance explained by the term on that line after controlling for all previously entered terms (but ignoring terms that enter the model later). Significant results mean adding that term improved the model.

We also examine simultaneous tests of predictors, which are the effects of the indicated terms after controlling for all other terms in the model. They are only computed for terms that are not part of a higher-order interaction because it makes no sense to test for a main effect when the variable is involved in an interaction. These should be functionally equivalent to the results you get with *anova()* when you feed it a pair of nested models that differ only in that one model includes a term that is absent from the other model. They are likelihood ratio tests (LRTs).

We used odds-ratios to quantify effect sizes for both specific model parameters and for contrasts derived from combining model parameters. When using contrasts to do things like pairwise comparisons, we obtain simultaneous 95% CIs that are adjusted for multiple testing via Westfall's (1997) method.

We used the inverse logit transformation to convert fitted values and associated confidence intervals into the conditional probability of passing a particular level transition given that specific values on the predictors. We computed the unconditional pass rates from the conditional pass rates as sets of cumulative products. Tabulating and plotting these conditional and unconditional rates provides deeper understanding of the model results.

Since we are running the continuation-ration models as logistic regressions, all the standard methods for assessing goodness of fit for logistic regresion models apply. We applied the Hosmer-Lemeshow test (HLT), wherein the null hypothesis is that the data fit the model. Thus, for the HLT a significant result is actually undesirable because it says that the data don't fit the model. This is a measure of calibration (Fenlon et al., 2018), which can also be examined with calibration plots. These plot split the data into the same set of bins used for the HLT, by grouping the predicted probabilities into bins with a minimum width (typically 0.10 because that way all the values grouped together are quite similar). The confidence interval for each bin is labeled with the bin's sample size. Ideally, the point estimate for each bin is close to the dashed line, but at a minimum we want all the confidence intervals to overlap that dashed line.

We also use functions from the pROC package to identify the optimal threshold (i.e., cutpoint) c to use when converting fitted probabilities of passing a testlet into a binary prediction of whether the individual will pass it. The naieve cutpoint would be $c = 0.5$, but that is not necessarily the optimal cutpoint. Choosing a cutpoint via use of the Youden index (Youden, 1950) is optimal with respect to overall misclassification rate for a given weighting of sensitivity and specificity (Perkins & Schisterman, 2006). Thus, we show the results of various classification measures given that we use the optimal cutpoint based on the Youden index. Here are short definitions of what some of these statistics measure.

- Accuracy is the model's overall ability to correctly classify learners according to whether they pass versus fail the level transitions.
- Sensitivity is the ability of the model to correctly identify learners who passed level transitions.
- Specificity is the ability of the model to correctly identify learners who fail the level transitions.

Table 5 shows a list of the classification measures computed for each model, along with the label used in tables later to identify each measure.

```r
data.frame(Measure = c("Threshold value", "Specificity", "Sensitivity",
                       "Accuracy", "True negative count", "True positive count",
                       "False negative count", "False positive count",
                       "Negative predictive value", "Positive predictive value",
                       "False discovery rate", "False postivie rate",
                       "True positive rate", "True negative rate",
                       "False negative rate", "1 - Specificity",
                       "1 - Sensitivity", "1 - Accuracy",
                       "1 - Negative predictive value",
                       "1 - Positive predictive value", "Precision", "Recall",
                       "Youden Index",
                       "Distance to Top Left Corner of the ROC space"),
           Label = c("threshold", "specificity", "sensitivity", "accuracy",
                     "tn", "tp", "fn", "fp", "npv", "ppv", "fdr", "fpr", "tpr",
                     "tnr", "fnr", "1-specificity", "1-sensitivity",
                     "1-accuracy", "1-npv", "1-ppv", "precision", "recall",
                     "youden", "closest.topleft")) %>%
  kable(format = "latex", booktabs = TRUE, row.names = FALSE,
        caption = "Classification Measures Reported for Each Model")
```

Measure	Label
Threshold value	threshold
Specificity	specificity
Sensitivity	sensitivity
Accuracy	accuracy
True negative count	tn
True positive count	tp
False negative count	fn
False positive count	fp
Negative predictive value	npv
Positive predictive value	ppv
False discovery rate	fdr
False postivie rate	fpr
True positive rate	tpr
True negative rate	tnr
False negative rate	fnr
1 - Specificity	1-specificity
1 - Sensitivity	1-sensitivity
1 - Accuracy	1-accuracy
1 - Negative predictive value	1-npv
1 - Positive predictive value	1-ppv
Precision	precision
Recall	recall
Youden Index	youden
Distance to Top Left Corner of the ROC space	closest.topleft

Table 5: Classification Measures Reported for Each Model

The area under the receiver operating characteristic curve (AUC) measures the model's ability to discriminate those who pass from those who fail to pass the level transitions. It can range from 0.50 to 1.00; values of 0.50–0.69 are poor, 0.70-0.79 are acceptable, .80-.89 are excellent, and ≥ 0.90 are outstanding. These interpretive heuristics come from Hosmer, Lemeshow, & Sturdivant (2013, p. 177). Thus, for each model we plot the ROC curve for the model and annotate it with the best classification threshold, plus the corresponding values of sensitivity and specificity, the AUC, and the 95% confidence interval for the AUC. To compare ROC curves between selected models, we use two-sided, stratified bootstrap tests for the difference between the AUC values (Robin et al., 2011).

```
# Save model predicted values to facilitate computing classification measures.
VTAData <- VTAData %>%
  mutate(pred.m1a = predict(m1a, type = "response"),
         pred.m1b = predict(m1b, type = "response"),
         pred.m2a = predict(m2a, type = "response"),
         pred.m2b = predict(m2b, type = "response"),
         pred.m3a = predict(m3a, type = "response"),
         pred.m3b = predict(m3b, type = "response"))
```

Finally, we run model diagnostics for the best candidate models to see if there are any obvious problems with each model. First we check for outliers, examine residual plots and index plots for a couple influence measures to look for observations with values exceeding the cutoffs.

6 Model 1a: Parallel Course Effect

We first fit a basic model that omits the intercept term in order to simplify post-processing of the model results into interpretable estimates. We focus here on a model examining whether the course a learner is taking affects the pass rate for each level transition testlet. This model assumes that the higher level courses have a constant effect on the pass rates across all level transition testlets. Table 6 below shows the raw parameter estimates, confidence intervals, s-values, BFBs, and posterior probabilities of H1 corresponding to the p-values.

```r
m1a %>%
  tidy(., conf.int = TRUE, conf.level = .95) %>%
  cbind(., convertp(.$p.value, digits = 2)) %>%
  kable(format = "latex", booktabs = TRUE, format.args = list(digits = 3),
        digits = c(2, 2, 2, 2, Inf, 2, 2, 2, 2, 2),
        col.names = c("Term", "Estimate", "SE", "z-value", "p-value", "CI.LL",
                      "CI.UL", "S", "BFB", "PPH1"),
        caption = "Model 1a Coefficients")
```

Term	Estimate	SE	z-value	p-value	CI.LL	CI.UL	S	BFB	PPH1
Testlet1	-1.72	0.23	-7.57	3.63e-14	-2.19	-1.30	44.65	3.27e+11	1.00
Testlet2	-2.15	0.26	-8.28	1.26e-16	-2.68	-1.66	52.81	7.96e+13	1.00
Testlet3	-1.34	0.30	-4.47	7.82e-06	-1.94	-0.76	16.97	4.00e+03	1.00
Testlet4	-1.05	0.35	-3.01	2.61e-03	-1.74	-0.37	8.58	2.37e+01	0.96
Course200	1.24	0.25	4.92	8.78e-07	0.76	1.75	20.12	3.01e+04	1.00
Course300	1.66	0.24	6.82	8.81e-12	1.20	2.16	36.72	1.64e+09	1.00
Course400	2.08	0.28	7.41	1.27e-13	1.54	2.65	42.84	9.76e+10	1.00

Table 6: Model 1a Coefficients

6.1 Sequential Tests (Type I SS)

Each row in Table 7 tests the significance of unique additional variance explained by the term on that line after controlling for all previously entered terms. Significant results mean adding that term improved the model.

```r
m1a %>%
 anova(., test = "Chisq") %>%
 cbind(., convertp(.[,"Pr(>Chi)"])) %>%
 kable(format = "latex", booktabs = TRUE,
       digits = c(0, 2, 0, 2, Inf, 2, 2, 2),
       col.names = c("DF", "Deviance", "Resid. DF", "Resid. Dev", "p-value",
                     "S", "BFB", "PPH1"),
       caption = "Model 1a Sequential Tests (Type I SS): Analysis of Deviance")
```

	DF	Deviance	Resid. DF	Resid. Dev	p-value	S	BFB	PPH1
NULL	NA	NA	1320	1829.91	NA	NA	NA	NA
Testlet	4	62.16	1316	1767.75	1.018257e-12	39.84	1.308385e+10	1
Course	3	78.17	1313	1689.58	7.594483e-17	53.55	1.305089e+14	1

Table 7: Model 1a Sequential Tests (Type I SS): Analysis of Deviance

6.2 Simultaneous Tests of Main Effects via LRT (Type III SS)

The simultaneous tests in Table 8 are the effects of the indicated terms after controlling for all other terms in the model.

```
m1a %>%
  drop1(., test = "Chisq") %>%
  cbind(., convertp(.[,"Pr(>Chi)"])) %>%
  kable(format = "latex", booktabs = TRUE,
       digits = c(0, 2, 2, 2, Inf, 2, 2, 2),
       col.names = c("DF", "Deviance", "AIC", "LRT", "p-value", "S", "BFB",
                     "PPH1"),
       caption = "Model 1a Simultaneous Tests (Type III SS)")
```

	DF	Deviance	AIC	LRT	p-value	S	BFB	PPH1
<none>	NA	1689.58	1703.58	NA	NA	NA	NA	NA
Testlet	4	1796.84	1802.84	107.26	2.795496e-22	71.60	2.651627e+19	1
Course	3	1767.75	1775.75	78.17	7.594483e-17	53.55	1.305089e+14	1

Table 8: Model 1a Simultaneous Tests (Type III SS)

6.3 Conditional and Unconditional Pass Rates

Table 9 shows the conditional and unconditional pass rates estimated by Model 1a as a function of level transition testlet and course.

```
# Create a new data frame object for use with predict()
m1a.ND <- data.frame(Testlet = gl(n = 4, k = 1, length = 16,
                                  labels = c("1", "2", "3", "4")),
                     Course = gl(n = 4, k = 4, length = 16,
                                 labels = c("100", "200", "300", "400")))

# Compute predicted mean passing rate at each combination of Level & Course
m1a.pred <- predict(m1a, newdata = m1a.ND, type = "link", se.fit = TRUE)

# Add fitted values and CIs to the new data frame & display it.
critval       <- qnorm(0.975)  # For Wald 95% CIs
m1a.ND$fit    <- m1a.pred$fit
m1a.ND$se.fit <- m1a.pred$se.fit
m1a.ND$fit.LL <- with(m1a.ND, fit - (critval * se.fit))
m1a.ND$fit.UL <- with(m1a.ND, fit + (critval * se.fit))

# Convert fitted values and CIs to probabilities.
m1a.ND$Pass.Rate <- invlogit(m1a.ND$fit)
m1a.ND$Pass.LL   <- invlogit(m1a.ND$fit.LL)
m1a.ND$Pass.UL   <- invlogit(m1a.ND$fit.UL)

# Compute unconditional pass rates.
m1a.ND$Pass.URate <- c(cumprod(m1a.ND[m1a.ND$Course == "100", "Pass.Rate"]),
                       cumprod(m1a.ND[m1a.ND$Course == "200", "Pass.Rate"]),
                       cumprod(m1a.ND[m1a.ND$Course == "300", "Pass.Rate"]),
                       cumprod(m1a.ND[m1a.ND$Course == "400", "Pass.Rate"]))

ShowVars <- c("Testlet", "Course", "Pass.Rate", "Pass.LL", "Pass.UL",
              "Pass.URate")
kable(m1a.ND[, ShowVars], format = "latex", booktabs = TRUE, digits = 2,
      format.args = list(nsmall = 2),
      caption = paste("Model 1a Conditional Pass Rates with 95 percent CIs",
                      "and Unconditional Pass Rates by Testlet and Course")) %>%
kable_styling(latex_options = c("repeat_header"))
```

Testlet	Course	Pass.Rate	Pass.LL	Pass.UL	Pass.URate
1	100	0.15	0.10	0.22	0.15
2	100	0.10	0.07	0.16	0.02
3	100	0.21	0.13	0.32	0.00
4	100	0.26	0.15	0.41	0.00
1	200	0.38	0.33	0.44	0.38
2	200	0.29	0.23	0.35	0.11
3	200	0.48	0.37	0.58	0.05
4	200	0.55	0.41	0.68	0.03
1	300	0.49	0.44	0.53	0.49
2	300	0.38	0.32	0.44	0.18
3	300	0.58	0.48	0.67	0.11
4	300	0.65	0.52	0.76	0.07
1	400	0.59	0.51	0.67	0.59
2	400	0.48	0.39	0.57	0.28
3	400	0.68	0.57	0.77	0.19
4	400	0.74	0.61	0.83	0.14

Table 9: Model 1a Conditional Pass Rates with 95 percent CIs and Unconditional Pass Rates by Testlet and Course

6.4 Odds-Ratios for Course Effect

Table 10 shows the Model 1a odds-ratios for the effect of being in a second-, third-, or fourth-year Spanish course instead of a first-year course. They quantify the course effect sizes, which this model assumes are equal across level transition testlets. Model 1b tests whether that assumption is reasonable.

```r
# Run multiple comparisons examine the course effect & get adjusted 95% CIs.
m1a.ct <- glht(m1a, linfct = mcp(Course = "Tukey"))
m1a.mc <- summary(m1a.ct, test = adjusted("Westfall"))
m1a.ci <- confint(m1a.ct, calpha = adjusted_calpha(test = "Westfall"))
data.frame(Est   = m1a.mc$test$coefficients,
           SE    = m1a.mc$test$sigma,
           CI.LL = m1a.ci$confint[, "lwr"],
           CI.UL = m1a.ci$confint[, "upr"],
           OR    = exp(m1a.mc$test$coefficients),
           OR.LL = exp(m1a.ci$confint[, "lwr"]),
           OR.UL = exp(m1a.ci$confint[, "upr"]),
           z     = m1a.mc$test$tstat,
           p     = m1a.mc$test$pvalues,
           Sval  = p2s(m1a.mc$test$pvalues),
           BFB   = p2bfb(m1a.mc$test$pvalues),
           PPH1  = p2pp(m1a.mc$test$pvalues)) %>%
  kable(., format = "latex", booktabs = TRUE, format.args = list(digits = 3),
        digits = c(rep(x = 2, times = 8), Inf, 2, 2, 2),
        caption = paste("Model 1a Multiple Comparisons for Course Effect,",
                        "With Westfall (1997) Adjustment for Multiplicity")) %>%
  kable_styling(latex_options = c("repeat_header"))
```

	Est	SE	CI.LL	CI.UL	OR	OR.LL	OR.UL	z	p	Sval	BFB	PPH1
200 - 100	1.24	0.25	0.60	1.88	3.45	1.82	6.53	4.92	1.76e-06	19.12	1.58e+04	1.00
300 - 100	1.66	0.24	1.04	2.28	5.27	2.84	9.78	6.82	2.77e-11	35.07	5.46e+08	1.00
400 - 100	2.08	0.28	1.37	2.79	8.01	3.93	16.36	7.41	3.49e-13	41.38	3.67e+10	1.00
300 - 200	0.42	0.13	0.08	0.77	1.53	1.09	2.15	3.15	1.64e-03	9.25	3.50e+01	0.97
400 - 200	0.84	0.19	0.35	1.33	2.33	1.42	3.80	4.37	3.60e-05	14.76	1.00e+03	1.00
400 - 300	0.42	0.18	-0.04	0.88	1.52	0.96	2.40	2.34	1.94e-02	5.69	4.82e+00	0.83

Table 10: Model 1a Multiple Comparisons for Course Effect, With Westfall (1997) Adjustment for Multiplicity

6.5 Assessing Goodness of Fit, Discrimination, and Calibration

6.5.1 Hosmer-Lemeshow Goodness of Fit Test

Figure 5 shows a calibration plot for this model. The plot is based on the bins summarized in Table 11, while the Hosmer-Lemeshow test is shown in Table 12.

```
FCap <- paste("\\label{fig:m1a-calib-plot}",
              "Model 1a Calibration Plot with Bin Sample Sizes and",
              "Hosmer-Lemeshow Test")
HLT.m1a <- HLfit(m1a, bin.method = "prob.bins", min.prob.interval = 0.1,
                 xlab = "Predicted Probability",
                 ylab = "Observed Probability")
```

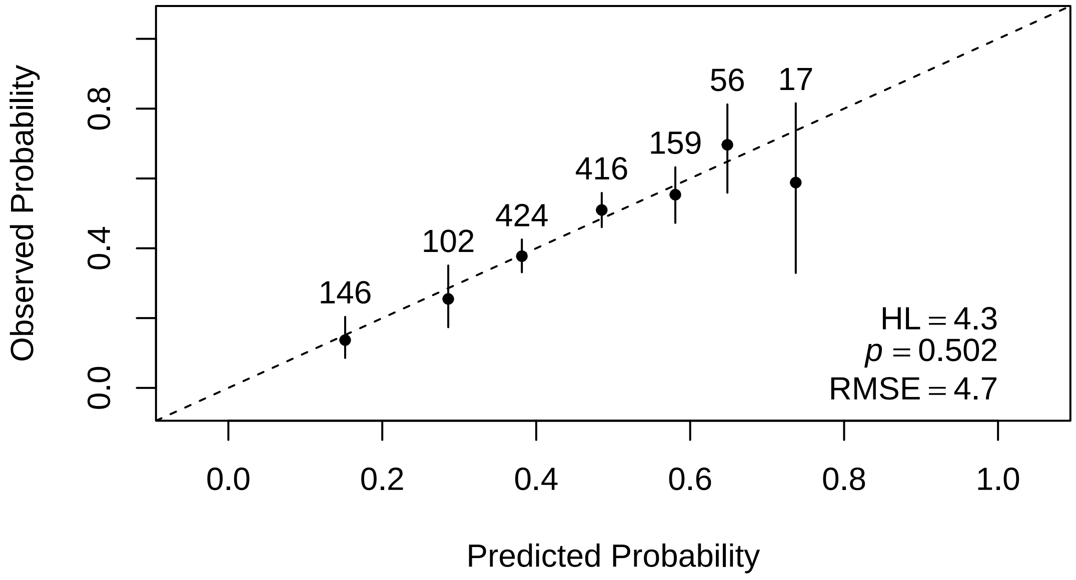

Figure 5: Model 1a Calibration Plot with Bin Sample Sizes and Hosmer-Lemeshow Test

```
HLT.bin.vnames <- c("Bin Center (Median)", "Bin N",
                    "Observed Proportion", "Predicted Proportion",
                    "Lower Limit", "Upper Limit")
FN <- "Minimum bin interval width = 0.10."
```

```
HLT.m1a$bins.table %>%
  kable(., format = "latex", booktabs = TRUE, digits = 3, row.names = FALSE,
        col.names = HLT.bin.vnames,
        caption = "Calibration Bins Used for Model 1a Hosmer-Lemeshow Test") %>%
  add_header_above(., header = c(" " = 4, "Obs. Prop. 95% CI" = 2)) %>%
  column_spec(column = 1, width = "1.75cm") %>%
  column_spec(column = 3:4, width = "1.75cm") %>%
  column_spec(column = 5:6, width = "1.25cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

				Obs. Prop. 95% CI	
Bin Center (Median)	Bin N	Observed Proportion	Predicted Proportion	Lower Limit	Upper Limit
0.152	146	0.137	0.147	0.086	0.204
0.286	102	0.255	0.281	0.174	0.351
0.381	424	0.377	0.381	0.331	0.425
0.485	416	0.510	0.484	0.460	0.559
0.581	159	0.553	0.582	0.473	0.632
0.648	56	0.696	0.661	0.559	0.812
0.737	17	0.588	0.737	0.329	0.816

Note: Minimum bin interval width = 0.10.

Table 11: Calibration Bins Used for Model 1a Hosmer-Lemeshow Test

```
HLT.col.vnames <- c("Chi-square", "df", "p", "RMSE", "S", "BFB", "PPH1")
FN <- "Minimum bin interval width = 0.10."

HLT.m1a %>%
  as_tibble() %>%
  select(chi.sq, DF, p.value, RMSE) %>%
  unique() %>%
  cbind(., convertp(p = .$p.value, digits = 2)) %>%
  kable(., format = "latex", booktabs = TRUE,
        digits = c(2, 0, Inf, 2, 2, 2, 2),
        col.names = HLT.col.vnames,
        caption = "Hosmer-Lemeshow Test for Goodness of Fit of Model 1a") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

Chi-square	df	p	RMSE	S	BFB	PPH1
4.34	5	0.5018586	4.68	0.99	1.06	0.52

Note: Minimum bin interval width = 0.10.

Table 12: Hosmer-Lemeshow Test for Goodness of Fit of Model 1a

6.5.2 Classification

Table 13 presents the classification measures for this model.

```
set.seed(4921) # For reproducibility of bootstrap estimates.
roc.m1a <- roc(Pass ~ pred.m1a, data = VTAData, ci = TRUE, direction = "<",
               ci.method = "bootstrap", boot.n = 10000)
```

```
## Setting levels: control = 0, case = 1
```

```
round(lrcm(roc.m1a, seed = 4684), digits = 3) %>%
  kable(format = "latex", booktabs = TRUE,
        caption = "Classification Measures for Model 1a") %>%
  add_header_above(header = c(" " = 2, "Bootstrapped Quantiles" = 3))
```

		Bootstrapped Quantiles		
	threshold	2.5%	50%	97.5%
threshold	0.428	0.380	0.428	0.430
specificity	0.609	0.407	0.609	0.646
sensitivity	0.629	0.589	0.631	0.814
accuracy	0.617	0.573	0.617	0.643
tn	466.000	310.975	466.000	494.000
tp	349.000	326.975	350.000	452.000
fn	206.000	103.000	205.000	228.025
fp	299.000	271.000	299.000	454.025
npv	0.693	0.670	0.695	0.754
ppv	0.539	0.496	0.539	0.566
fdr	0.461	0.434	0.461	0.504
fpr	0.391	0.354	0.391	0.593
tpr	0.629	0.589	0.631	0.814
tnr	0.609	0.407	0.609	0.646
fnr	0.371	0.186	0.369	0.411
1-specificity	0.391	0.354	0.391	0.593
1-sensitivity	0.371	0.186	0.369	0.411
1-accuracy	0.383	0.357	0.383	0.427
1-npv	0.307	0.246	0.305	0.330
1-ppv	0.461	0.434	0.461	0.504
precision	0.539	0.496	0.539	0.566
recall	0.629	0.589	0.631	0.814
youden	1.238	1.188	1.238	1.290
closest.topleft	0.291	0.253	0.291	0.391

Table 13: Classification Measures for Model 1a

6.5.3 Area Under the Curve (AUC)

Figure 6 shows the ROC curve for the model, annotated with the best classification threshold for balancing sensitivity versus specificity and the area under the curve (AUC).

```
FCap <- paste("\\label{fig:m1a-auc-plot}",
              "Model 1a Receiver Operating Characteristic (ROC) Curve.",
              "The dot marks the best classification threshold.",
              "AUC 95% confidence interval obtained via stratified bootstrap",
              "with 10,000 replicates.")
print(roc.m1a)
```

```
##
## Call:
## roc.formula(formula = Pass ~ pred.m1a, data = VTAData, ci = TRUE,     direction = "<", ci.method = "bootstrap", boot.n = 1000
##
## Data: pred.m1a in 765 controls (Pass 0) < 555 cases (Pass 1).
## Area under the curve: 0.6537
## 95% CI: 0.624-0.6819 (10000 stratified bootstrap replicates)
```

```
plot.roc(roc.m1a, print.auc = TRUE, print.auc.cex = .8, print.thres = "best",
         print.thres.cex = .8)
```

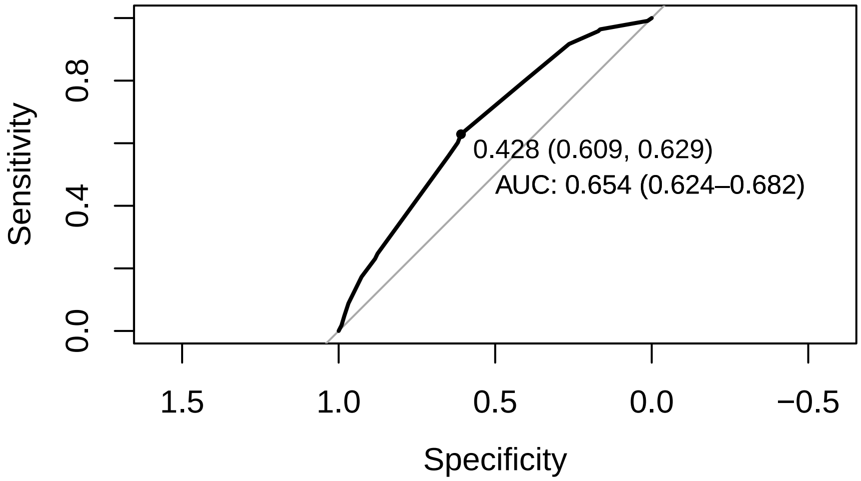

Figure 6: Model 1a Receiver Operating Characteristic (ROC) Curve. The dot marks the best classification threshold. AUC 95% confidence interval obtained via stratified bootstrap with 10,000 replicates.

Based on this AUC result, we conclude that Model 1a is generally poor at discriminating people who are passing versus failing the level transitions. That means course is not a particularly good predictor regardless of what hypothesis testing results show.

6.5.4 R^2 Measures

The low values for $R^2_p = 0.08$ and $R^2_{Dev} = 0.08$ are not very encouraging about this model.

6.6 Diagnostics

We need to run model diagnostics to see if there are any obvious problems with the model. First we check for outliers and find that there are none of note.

```
outlierTest(m1a)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferroni p
## 182 2.140342           0.032327           NA
```

Figure 7 shows residual plots for this model.

```
FCap <- paste("\\label{fig:m1a-resid-plots}",
              "Model 1a Residual Plots.")
residualPlots(m1a, layout = c(1,3), tests = FALSE)
```

Now we look at some index plots for influence measures. We are looking for observations with values exceeding the cutoffs. Figure 8 shows an index plot of Cook's D for this model.

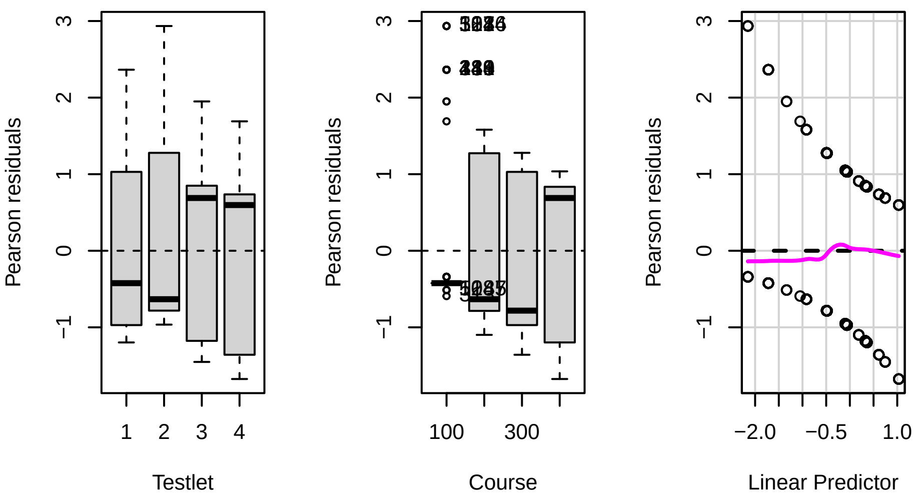

Figure 7: Model 1a Residual Plots.

```
FCap <- paste("\\label{fig:m1a-plotCookD}",
              "Model 1a Index Plot of Coook's D.",
              "The", sum(cooks.distance(m1a) > CookDco(m1a)), "observations",
              "with values exceeding the cutoff are shown in red.")
PlotCookD(m1a)
```

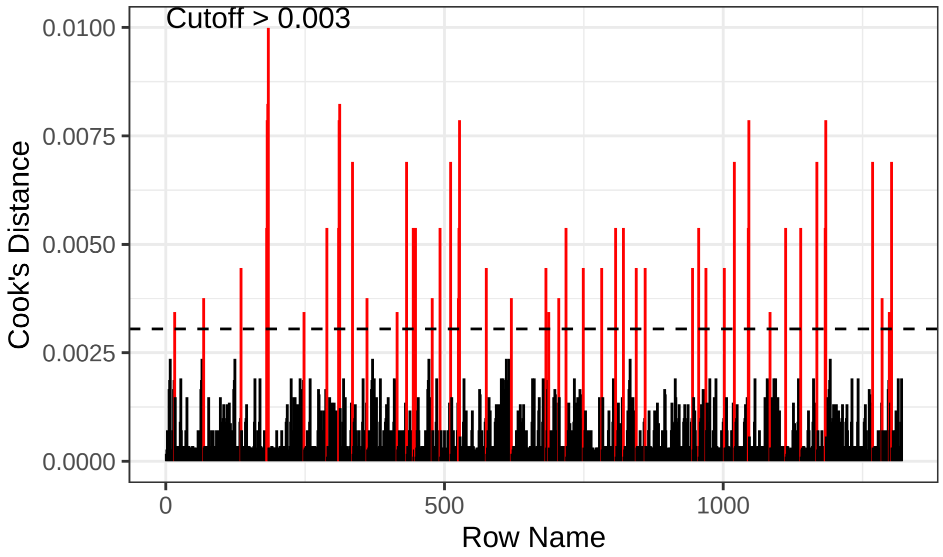

Figure 8: Model 1a Index Plot of Coook's D. The 53 observations with values exceeding the cutoff are shown in red.

Figure 9 shows an index plot of leverage values for this model, while Figure 10 shows the standardized residuals versus the leverage values, with contours for Cook's D.

```
FCap <- paste("\\label{fig:m1a-plot-leverage}",
              "Model 1a Index Plot of Leverage Values.",
              "The", sum(hatvalues(m1a) > hatco(m1a)), "observations",
              "with values exceeding the cutoff are shown in red.")
PlotHat(m1a)
```

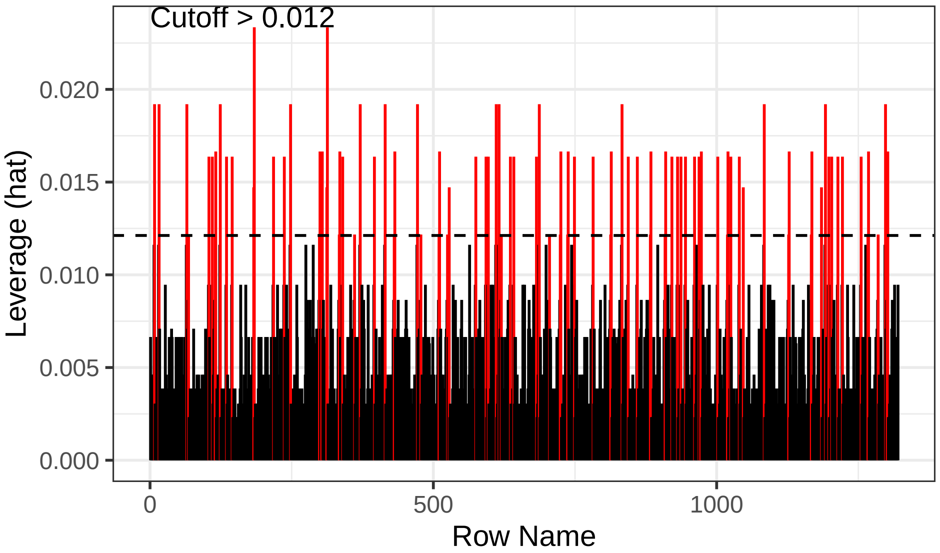

Figure 9: Model 1a Index Plot of Leverage Values. The 95 observations with values exceeding the cutoff are shown in red.

```
FCap <- paste("\\label{fig:m1a-plot-Pearson-leverage}",
              "Model 1a Plot of Standardized Pearson Residuals Versus",
              "Leverage Values, with Cook's D Contours.")
plot(m1a, which = 5, id.n = 10, cex.id = .6, caption = "", sub.caption = "",
     cook.levels = round(CookDco(m1a), digits = 3))
abline(v = hatco(m1a), lty = 2, col = "blue")
text(x = hatco(m1a), y = 3, pos = 4, col = "blue", cex = .75,
     labels = paste("Leverage cutoff >", round(hatco(m1a), digits = 3)))
```

Table 14 shows a list of influential cases for Model 1a. While there are a fair number of cases with both high Cook's distance and high leverage, only a few also have large standardized Pearson residuals.

```
FN <- paste0("All cases shown have high leverage (hat) and Cook's D values, ",
            "defined by hat > ", round(hatco(m1a), digits = 3),
            " and Cook's D > ", round(CookDco(m1a), digits = 3),
            ". Standardized Pearson residuals with absolute values > 1.96 are ",
            "flagged.")

# Identify cases w/ high leverage and high Cook's D.
m1a %>% InfCases(.) %>%
  mutate(Flag = if_else(abs(StdPearson) > 1.96, true = "x", false = "")) %>%
  kable(format = "latex", booktabs = TRUE, longtable = TRUE, digits = 3,
        caption = "Model 1a Influential Cases") %>%
  kable_styling(latex_options = c("repeat_header")) %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

Table 14: Model 1a Influential Cases

	Pass	Testlet	Course	hat	CookD	StdPearson	Flag
16	0	4	200	0.019	0.003	-1.109	
68	0	3	400	0.012	0.004	-1.461	
135	0	4	300	0.016	0.004	-1.369	
183	1	3	100	0.015	0.008	1.964	x
184	1	4	100	0.023	0.010	1.710	
248	0	4	200	0.019	0.003	-1.109	
312	1	3	100	0.015	0.008	1.964	x
335	0	4	400	0.017	0.007	-1.689	
361	0	3	400	0.012	0.004	-1.461	
415	0	4	200	0.019	0.003	-1.109	

Table 14: Model 1a Influential Cases *(continued)*

	Pass	Testlet	Course	hat	CookD	StdPearson	Flag
432	0	4	400	0.017	0.007	-1.689	
478	0	3	400	0.012	0.004	-1.461	
511	0	4	400	0.017	0.007	-1.689	
525	0	3	400	0.012	0.004	-1.461	
575	0	4	300	0.016	0.004	-1.369	
620	0	3	400	0.012	0.004	-1.461	
682	0	4	300	0.016	0.004	-1.369	
687	0	4	200	0.019	0.003	-1.109	
705	0	3	400	0.012	0.004	-1.461	
749	0	4	300	0.016	0.004	-1.369	
782	0	4	300	0.016	0.004	-1.369	
844	0	4	300	0.016	0.004	-1.369	
860	0	4	300	0.016	0.004	-1.369	
945	0	4	300	0.016	0.004	-1.369	
969	0	4	300	0.016	0.004	-1.369	
1002	0	4	300	0.016	0.004	-1.369	
1020	0	4	400	0.017	0.007	-1.689	
1084	0	4	200	0.019	0.003	-1.109	
1168	0	4	400	0.017	0.007	-1.689	
1268	0	4	400	0.017	0.007	-1.689	
1285	0	3	400	0.012	0.004	-1.461	
1298	0	4	200	0.019	0.003	-1.109	
1302	0	4	400	0.017	0.007	-1.689	

Note: All cases shown have high leverage (hat) and Cook's D values, defined by hat > 0.012 and Cook's D > 0.003. Standardized Pearson residuals with absolute values > 1.96 are flagged.

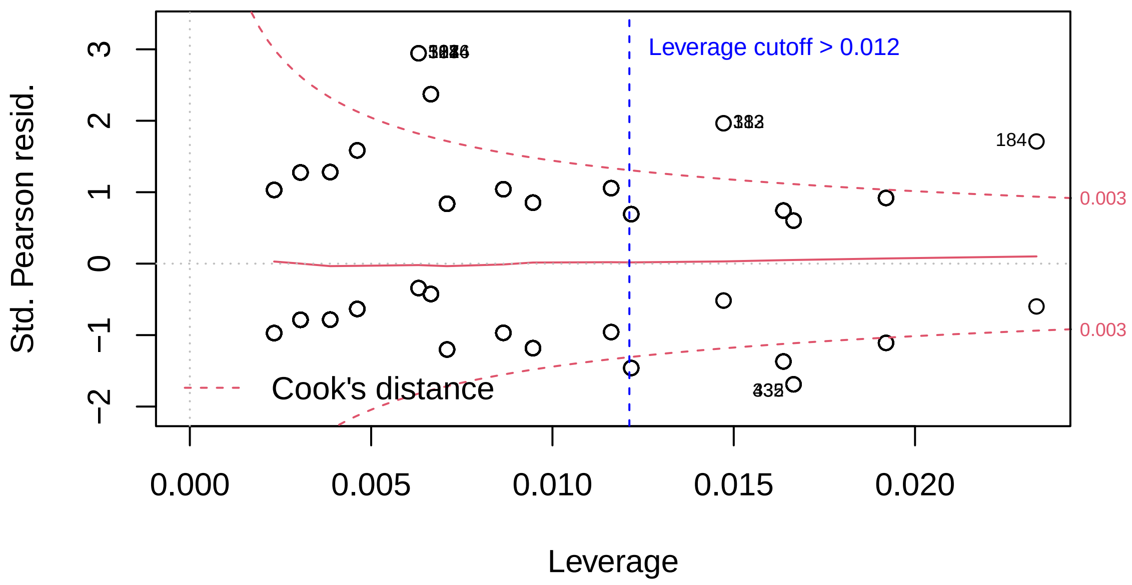

Figure 10: Model 1a Plot of Standardized Pearson Residuals Versus Leverage Values, with Cook's D Contours.

6.7 Graphs

6.7.1 Main Effect of Testlet

Figures 11 and 12 respectively plot the main effect of the self-assessment level transition testlet on the scale of the linear predictor (log-odds) and on the response scale (probability or conditional pass rates).

```
FCap <- paste("\\label{fig:m1a-plot-Testlet-LogOdds}",
              "Model 1a Main Effect of Testlet on Log-Odds Scale.")
visreg(m1a, xvar = "Testlet", ylab = "Log odds (Pass)")
```

```
FCap <- paste("\\label{fig:m1a-plot-Testlet-CPR}",
              "Model 1a Main Effect of Testlet on Probability Scale.")
visreg(m1a, xvar = "Testlet", ylab = "Conditional Pass Rate",
       scale = "response", rug = 2, ylim = c(0, 1))
```

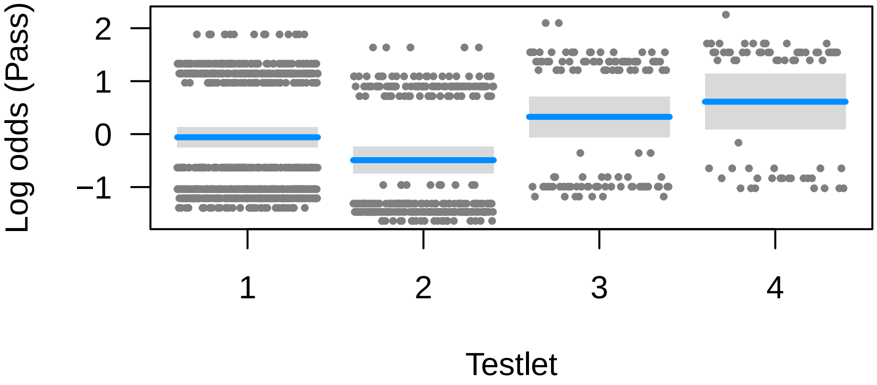

Figure 11: Model 1a Main Effect of Testlet on Log-Odds Scale.

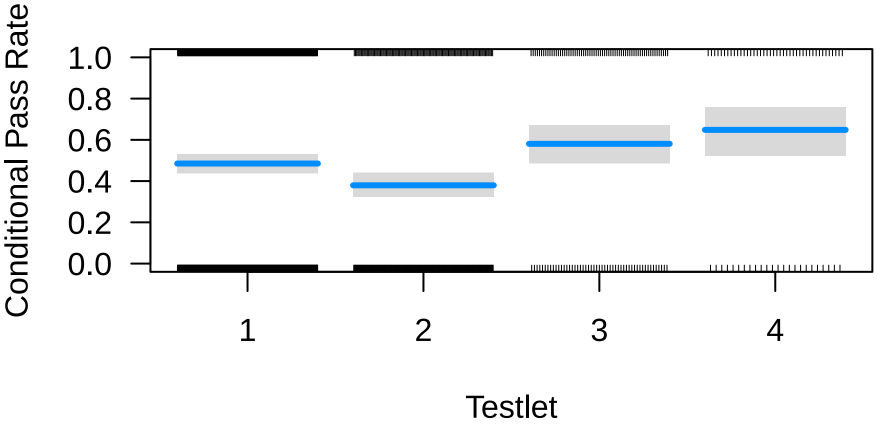

Figure 12: Model 1a Main Effect of Testlet on Probability Scale.

6.7.2 Main Effect of Course

Figures 13 and 14 respectively plot the main effect of course on the scale of the linear predictor (log-odds) and on the response scale (probability or conditional pass rates).

```
FCap <- paste("\\label{fig:m1a-plot-Course-LogOdds}",
              "Model 1a Main Effect of Course on Log-Odds Scale.")
visreg(m1a, xvar = "Course", ylab = "Log odds (Pass)")
```

```
FCap <- paste("\\label{fig:m1a-plot-Course-CPR}",
              "Model 1a Main Effect of Course on Probability Scale.")
visreg(m1a, xvar = "Course", ylab = "Conditional Pass Rate",
       scale = "response", rug = 2, ylim = c(0, 1))
```

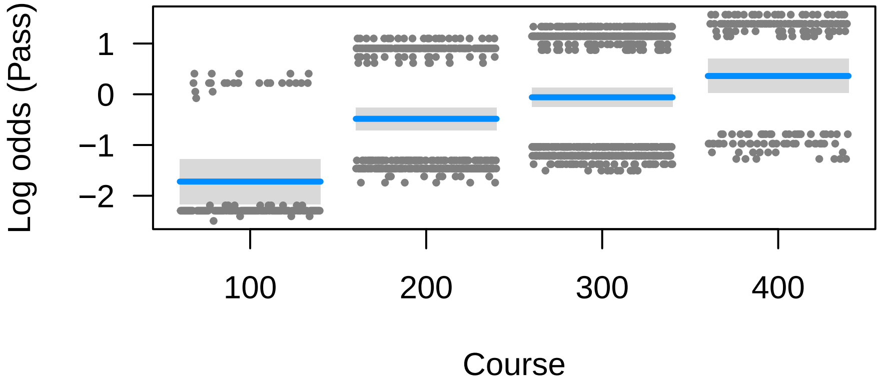

Figure 13: Model 1a Main Effect of Course on Log-Odds Scale.

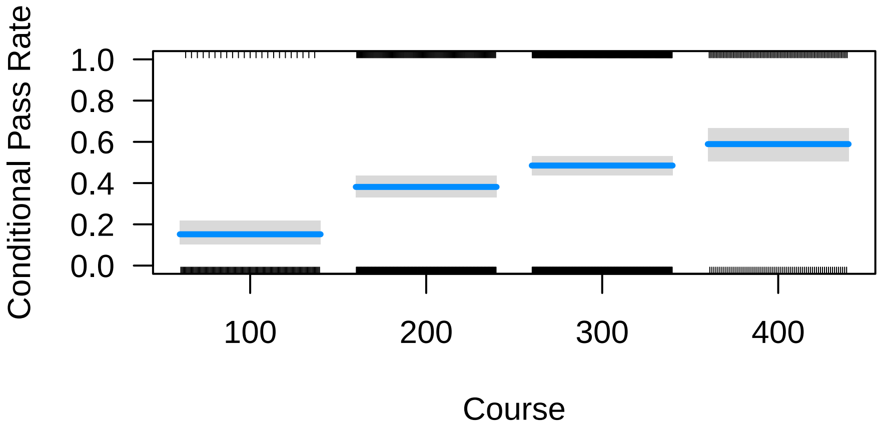

Figure 14: Model 1a Main Effect of Course on Probability Scale.

6.7.3 Conditional Pass Rates

Figure 15 plots the conditional pass rates derived from Model 1a as a function of both testlet and course.

```
FCap <- paste("\\label{fig:m1a-plot-CPR}",
              "Model 1a Conditional Pass Rates by Self-Assessment Testlet and",
              "Course.",
              "Lines are labeled with Course.")
m1a.FCPR <- ggplot(m1a.ND[, ShowVars],
                   aes(Testlet, Pass.Rate, group=Course, color=Course))+
      geom_line()+ coord_cartesian(ylim = c(0, 0.8))+
      xlab("Self-Assessment Testlet")+
      ylab("Conditional Pass Rate")
direct.label(m1a.FCPR, "first.points")
```

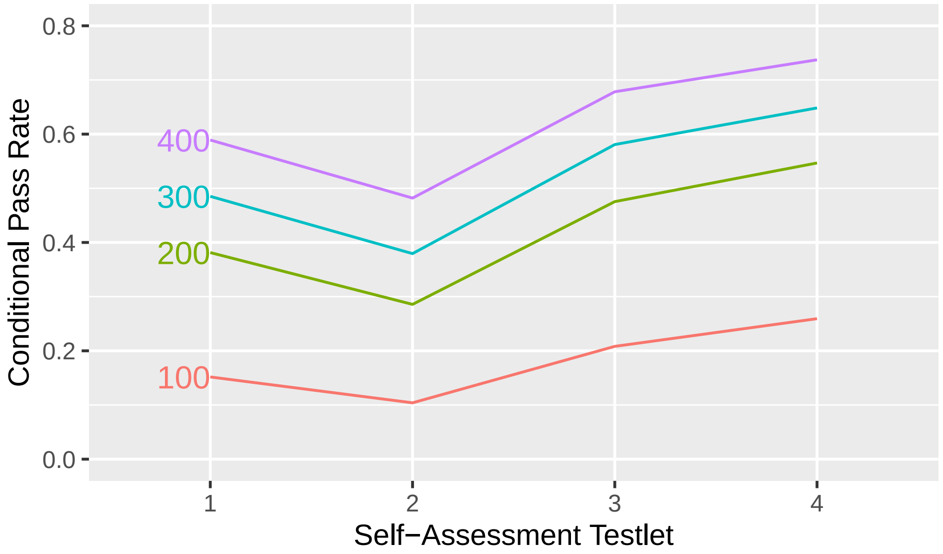

Figure 15: Model 1a Conditional Pass Rates by Self-Assessment Testlet and Course. Lines are labeled with Course.

6.7.4 Unconditional Pass Rates

Figure 16 plots the unconditional pass rates derived from Model 1a as a function of both testlet and course.

```
FCap <- paste("\\label{fig:m1a-plot-UPR}",
              "Model 1a Unconditional Pass Rates by Self-Assessment Testlet",
              "and Course.",
              "Lines are labeled with Course.")
m1a.FUPR <- ggplot(m1a.ND[, ShowVars],
                   aes(Testlet, Pass.URate, group=Course, color=Course))+
      geom_line()+ coord_cartesian(ylim = c(0, 0.65))+
      xlab("Self-Assessment Testlet")+
      ylab("Unconditional Pass Rate")
direct.label(m1a.FUPR, "first.points")
```

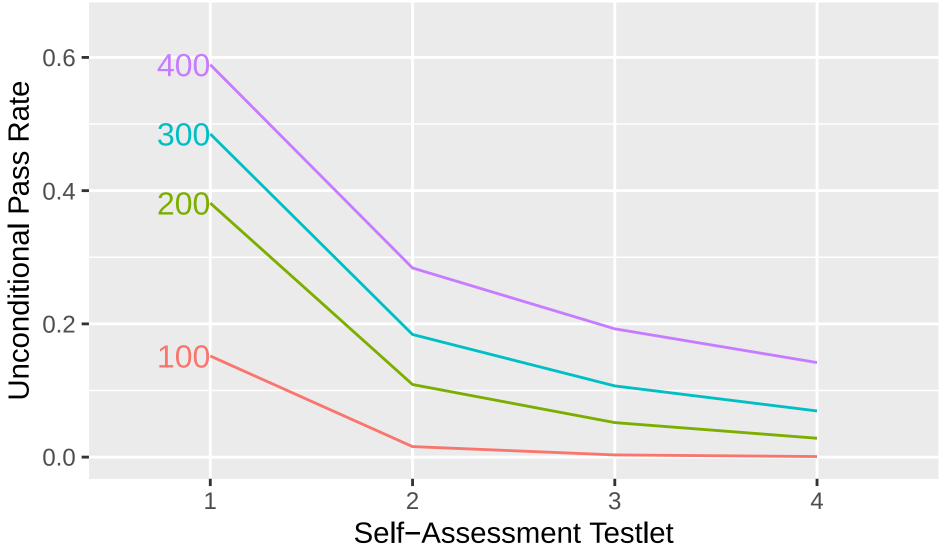

Figure 16: Model 1a Unconditional Pass Rates by Self-Assessment Testlet and Course. Lines are labeled with Course.

6.8 Conclusion

Learners taking second-, third-, and fourth-year courses are more likely to pass each of the level transitions than learners in the first-year courses. The increasing trend in the odds-ratios across the more advanced courses indicate that the more advanced the course, the larger the difference in the passing rates is relative to first-year courses. This makes intuitive sense: we expect learners with more training to do better on self-assessments if the instrument is a valid measure of Spanish proficiency.

The course effects in this model are assumed to be parallel (constant across the different level transitions), but we can test whether that assumption needs to be relaxed by running another model and comparing it to this one.

Despite the significant effects, the model doesn't discriminate well between those who pass vs. fail (it has inadequate accuracy and low AUC). It explains only a small percentage of the variance. We can probably do better with a different model.

7 Model 1b: Non-parallel Course Effect

We again omit the intercept term in order to simplify post-processing of the model results into interpretable estimates. We continue to focus on a model examining whether the course a learner is taking affects the pass rate, but relax the assumption that it has a constant effect across all level transitions. This model allows course to have a level-specific effect on the pass rates. Table 15 below shows the raw parameter estimates, confidence intervals, s-values, BFBs, and posterior probabilities of H1 corresponding to the p-values.

```
m1b %>%
  tidy(., conf.int = TRUE, conf.level = .95) %>%
  cbind(., convertp(.$p.value, digits = 2)) %>%
  kable(format = "latex", booktabs = TRUE, format.args = list(digits = 3),
        digits = c(2, 2, 2, 2, Inf, 2, 2, 2, 2, 2),
        col.names = c("Term", "Estimate", "SE", "z-value", "p-value", "CI.LL",
                      "CI.UL", "S", "BFB", "PPH1"),
        caption = "Model 1b Coefficients")
```

Term	Estimate	SE	z-value	p-value	CI.LL	CI.UL	S	BFB	PPH1
Testlet1	-2.05	0.27	-7.46	8.98e-14	-2.62	-1.54	43.34	1.36e+11	1.00
Testlet2	-0.69	0.55	-1.27	2.06e-01	-1.86	0.34	2.28	1.13e+00	0.53
Testlet3	-0.41	0.91	-0.44	6.57e-01	-2.43	1.39	0.61	1.33e+00	0.57
Testlet4	0.00	1.41	0.00	1.00e+00	-3.23	3.23	0.00	1.07e+13	1.00
Course200	1.55	0.30	5.10	3.35e-07	0.98	2.18	21.51	7.37e+04	1.00
Course300	2.05	0.29	6.94	3.89e-12	1.50	2.66	37.90	3.60e+09	1.00
Course400	2.43	0.36	6.80	1.07e-11	1.75	3.16	36.45	1.36e+09	1.00
Testlet2:Course200	-2.00	0.67	-2.98	2.89e-03	-3.29	-0.63	8.43	2.18e+01	0.96
Testlet3:Course200	-0.52	1.06	-0.49	6.26e-01	-2.58	1.74	0.68	1.25e+00	0.56
Testlet4:Course200	-1.14	1.54	-0.74	4.57e-01	-4.54	2.26	1.13	1.03e+00	0.51
Testlet2:Course300	-1.86	0.64	-2.90	3.73e-03	-3.10	-0.54	8.06	1.76e+01	0.95
Testlet3:Course300	-1.67	0.99	-1.69	9.17e-02	-3.62	0.47	3.45	1.68e+00	0.63
Testlet4:Course300	-1.26	1.49	-0.84	4.00e-01	-4.59	2.08	1.32	1.00e+00	0.50
Testlet2:Course400	-1.69	0.72	-2.37	1.80e-02	-3.08	-0.24	5.80	5.09e+00	0.84
Testlet3:Course400	-1.14	1.08	-1.05	2.92e-01	-3.24	1.15	1.78	1.02e+00	0.51
Testlet4:Course400	-2.07	1.54	-1.35	1.78e-01	-5.47	1.33	2.49	1.20e+00	0.54

Table 15: Model 1b Coefficients

7.1 Sequential Tests (Type I SS)

Table 16 presents the sequential test for the Model 1b. Here, we want to focus on the result for the interaction effect. That tells us whether allowing course to have a level-specific effect instead of a constant effect across levels improved the model.

```
m1b %>%
 anova(., test = "Chisq") %>%
 cbind(., convertp(.[,"Pr(>Chi)"])) %>%
 kable(format = "latex", booktabs = TRUE,
       digits = c(0, 2, 0, 2, Inf, 2, 2, 2),
       col.names = c("DF", "Deviance", "Resid. DF", "Resid. Dev", "p-value",
                     "S", "BFB", "PPH1"),
       caption = "Model 1b Sequential Tests (Type I SS): Analysis of Deviance")
```

	DF	Deviance	Resid. DF	Resid. Dev	p-value	S	BFB	PPH1
NULL	NA	NA	1320	1829.91	NA	NA	NA	NA
Testlet	4	62.16	1316	1767.75	1.018257e-12	39.84	1.308385e+10	1.00
Course	3	78.17	1313	1689.58	7.594483e-17	53.55	1.305089e+14	1.00
Testlet:Course	9	17.37	1304	1672.21	4.322566e-02	4.53	2.710000e+00	0.73

Table 16: Model 1b Sequential Tests (Type I SS): Analysis of Deviance

7.2 Simultaneous Tests of Interaction Effects via LRT (Type III SS)

Table 17 shows the result of another way of testing the interaction: a likelihood ratio test (LRT) for comparing nested models.

```
anova(m1a, m1b, test = "Chisq") %>%
  cbind(., convertp(.[,"Pr(>Chi)"])) %>%
  kable(format = "latex", booktabs = TRUE,
       digits = c(0, 2, 0, 2, Inf, 2, 2, 2),
       col.names = c("Resid. Df", "Resid. Deviance", "Df", "Deviance",
                     "p-value", "S", "BFB", "PPH1"),
       caption = "Model 1a vs 1b: Likelihood Ratio Test for Interaction")
```

Resid. Df	Resid. Deviance	Df	Deviance	p-value	S	BFB	PPH1
1313	1689.58	NA	NA	NA	NA	NA	NA
1304	1672.21	9	17.37	0.04322566	4.53	2.71	0.73

Table 17: Model 1a vs 1b: Likelihood Ratio Test for Interaction

Both methods demonstrated for testing the interaction term yield the same p-value, so we will only use one of them from here on out.

7.3 Conditional and Unconditional Pass Rates (Omitted)

We omit these rates because Model 1b is not better than Model 1a.

7.4 Odds-Ratios for Course Effect (Omitted)

We omit these odds-ratios because Model 1b is not better than Model 1a.

7.5 Assessing Goodness of Fit, Discrimination, and Calibration

7.5.1 Hosmer-Lemeshow Goodness of Fit Test

Figure 17 shows a calibration plot for this model. The plot is based on the bins summarized in Table 18, while the Hosmer-Lemeshow test is shown in Table 19.

```
FCap <- paste("\\label{fig:m1b-calib-plot}",
              "Model 1b Calibration Plot with Bin Sample Sizes and",
              "Hosmer-Lemeshow Test")
HLT.m1b <- HLfit(m1b, bin.method = "prob.bins", min.prob.interval = 0.1,
                 xlab = "Predicted Probability",
                 ylab = "Observed Probability")
```

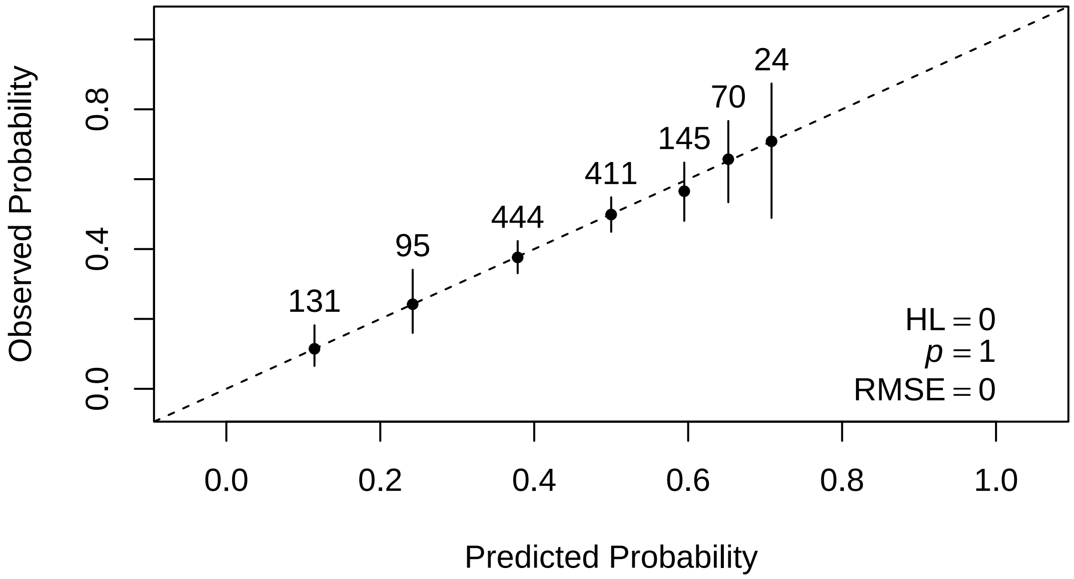

Figure 17: Model 1b Calibration Plot with Bin Sample Sizes and Hosmer-Lemeshow Test

```
FN <- "Minimum bin interval width = 0.10."

HLT.m1b$bins.table %>%
  kable(., format = "latex", booktabs = TRUE, digits = 3, row.names = FALSE,
        col.names = HLT.bin.vnames,
        caption = "Calibration Bins Used for Model 1b Hosmer-Lemeshow Test") %>%
  add_header_above(., header = c(" " = 4, "Obs. Prop. 95% CI" = 2)) %>%
  column_spec(column = 1, width = "1.75cm") %>%
  column_spec(column = 3:4, width = "1.75cm") %>%
  column_spec(column = 5:6, width = "1.25cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

```
FN <- "Minimum bin interval width = 0.10."

HLT.m1b %>%
  as_tibble() %>%
  select(chi.sq, DF, p.value, RMSE) %>%
  unique() %>%
```

				Obs. Prop. 95% CI	
Bin Center (Median)	Bin N	Observed Proportion	Predicted Proportion	Lower Limit	Upper Limit
0.115	131	0.115	0.115	0.066	0.182
0.242	95	0.242	0.242	0.160	0.341
0.378	444	0.376	0.376	0.331	0.423
0.500	411	0.499	0.499	0.449	0.548
0.595	145	0.566	0.566	0.481	0.648
0.652	70	0.657	0.657	0.534	0.767
0.708	24	0.708	0.708	0.489	0.874

Note: Minimum bin interval width = 0.10.

Table 18: Calibration Bins Used for Model 1b Hosmer-Lemeshow Test

```
  cbind(., convertp(p = .$p.value, digits = 2)) %>%
  kable(., format = "latex", booktabs = TRUE,
        digits = c(2, 0, Inf, 2, 2, 2, 2),
        col.names = HLT.col.vnames,
        caption = "Hosmer-Lemeshow Test for Goodness of Fit of Model 1b") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

Chi-square	df	p	RMSE	S	BFB	PPH1
0	5	1	0	0	-Inf	NaN

Note: Minimum bin interval width = 0.10.

Table 19: Hosmer-Lemeshow Test for Goodness of Fit of Model 1b

7.5.2 Classification

Table 20 presents the classification measures for this model.

```
set.seed(1574) # For reproducible bootstrap estimates.
roc.m1b <- roc(Pass ~ pred.m1b, data = VTAData, ci = TRUE, direction = "<",
               ci.method = "bootstrap", boot.n = 10000)
```

```
## Setting levels: control = 0, case = 1
```

```
round(lrcm(roc.m1b, seed = 6711), digits = 3) %>%
  kable(format = "latex", booktabs = TRUE,
        caption = "Classification Measures for Model 1b") %>%
  add_header_above(header = c(" " = 2, "Bootstrapped Quantiles" = 3))
```

		Bootstrapped Quantiles		
	threshold	2.5%	50%	97.5%
threshold	0.446	0.377	0.446	0.496
specificity	0.608	0.409	0.609	0.664
sensitivity	0.631	0.571	0.632	0.811
accuracy	0.617	0.578	0.618	0.644
tn	465.000	313.000	466.000	508.000
tp	350.000	317.000	351.000	450.000
fn	205.000	105.000	204.000	238.000
fp	300.000	257.000	299.000	452.000
npv	0.694	0.670	0.695	0.750
ppv	0.538	0.499	0.539	0.568
fdr	0.462	0.432	0.461	0.501
fpr	0.392	0.336	0.391	0.591
tpr	0.631	0.571	0.632	0.811
tnr	0.608	0.409	0.609	0.664
fnr	0.369	0.189	0.368	0.429
1-specificity	0.392	0.336	0.391	0.591
1-sensitivity	0.369	0.189	0.368	0.429
1-accuracy	0.383	0.356	0.382	0.422
1-npv	0.306	0.250	0.305	0.330
1-ppv	0.462	0.432	0.461	0.501
precision	0.538	0.499	0.539	0.568
recall	0.631	0.571	0.632	0.811
youden	1.238	1.190	1.239	1.291
closest.topleft	0.290	0.252	0.291	0.386

Table 20: Classification Measures for Model 1b

7.5.3 Area Under the Curve (AUC)

Figure 18 shows the ROC curve for the model, annotated with the best classification threshold for balancing sensitivity versus specificity and the area under the curve (AUC).

```
FCap <- paste("\\label{fig:m1b-auc-plot}",
              "Model 1b Receiver Operating Characteristic (ROC) Curve.",
              "The dot marks the best classification threshold.",
              "AUC 95% confidence interval obtained via stratified bootstrap",
              "with 10,000 replicates.")
print(roc.m1b)
```

```
##
## Call:
## roc.formula(formula = Pass ~ pred.m1b, data = VTAData, ci = TRUE,     direction = "<", ci.method = "bootstrap", boot.n = 1000
##
## Data: pred.m1b in 765 controls (Pass 0) < 555 cases (Pass 1).
## Area under the curve: 0.6613
## 95% CI: 0.633-0.6894 (10000 stratified bootstrap replicates)
```

```
plot.roc(roc.m1b, print.auc = TRUE, print.auc.cex = .8, print.thres = "best",
         print.thres.cex = .8)
```

Based on this AUC result, we conclude that Model 1b does a poor job of discriminating between those who pass vs fail the level transitions. That means course is still not a particularly good predictor even after we allow it to have a non-parallel effect. Table 21 compares the ROC curves for Models 1a and 1b and shows that they do not differ meaningfully with respect to discrimination performance.

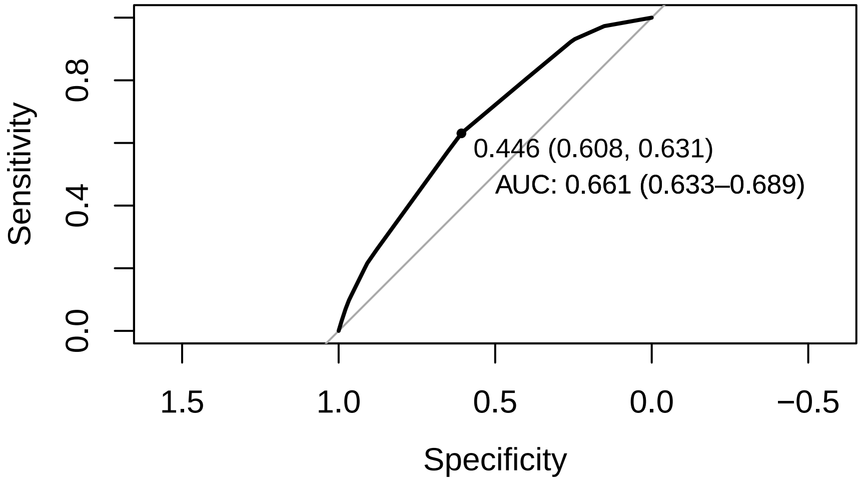

Figure 18: Model 1b Receiver Operating Characteristic (ROC) Curve. The dot marks the best classification threshold. AUC 95% confidence interval obtained via stratified bootstrap with 10,000 replicates.

```
set.seed(9537) # For reproducible bootstrap estimates.
roct.m1a.m1b <- roc.test(roc.m1a, roc.m1b, method = "bootstrap", boot.n = 10000)
glance(roct.m1a.m1b) %>%
  cbind(., convertp(p = .$p.value, digits = 2)) %>%
  select(boot.n, estimate1, estimate2, statistic, p.value, S, BFB, PPH1) %>%
  kable(format = "latex", booktabs = TRUE,
       digits = c(0, 3, 3, 2, Inf, 2, 2, 2),
       col.names = c("Bootstrap N", "Model 1a", "Model 1b", "D", "p", "S",
                     "BFB", "PPH1"),
       caption = paste("Comparing Models 1a and 1b Via Two-Sided, Stratified",
                       "Bootstrap Test for Correlated ROC Curves")) %>%
  add_header_above(header = c(" ", "AUC Estimates" = 2, " " = 5))
```

	AUC Estimates						
Bootstrap N	Model 1a	Model 1b	D	p	S	BFB	PPH1
10000	0.654	0.661	-1.15	0.2504989	2	1.06	0.51

Table 21: Comparing Models 1a and 1b Via Two-Sided, Stratified Bootstrap Test for Correlated ROC Curves

7.5.4 R^2 Measures

The $R^2_p = 0.09$ and $R^2_{Dev} = 0.09$ for Model 1b are only marginally better than those from Model 1a.

7.6 Diagnostics (Omitted)

We omit detailed diagnostics because Model 1b is not better than Model 1a.

7.7 Graphs (Omitted)

We omit graphs because Model 1b is not better than Model 1a. See Model 1a graphs instead.

7.8 Conclusion

The interaction is not improving the model. We should interpret Model 1a instead of Model 1b. That may change if we build in other covariates. However, we should also test COPIC as an alternative to course. We turn to that next in Model 2a.

8 Model 2a: Parallel OPIc Effect

We now want to look at a different covariate, namely OPIc speaking proficiency scores. We again fit a model that omits the intercept term in order to simplify post-processing of the model results into interpretable estimates. We start by assuming that the OPIc scores have a constant effect on the pass rates across all level transitions. We are using the centered version of OPIc in the model. Table 22 below shows the raw parameter estimates, confidence intervals, s-values, BFBs, and posterior probabilities of H1 corresponding to the p-values.

```
m2a %>%
  tidy(., conf.int = TRUE, conf.level = .95) %>%
  cbind(., convertp(.$p.value, digits = 2)) %>%
  kable(format = "latex", booktabs = TRUE, format.args = list(digits = 3),
        digits = c(2, 2, 2, 2, Inf, 2, 2, 2, 2, 2),
        col.names = c("Term", "Estimate", "SE", "z-value", "p-value", "CI.LL",
                      "CI.UL", "S", "BFB", "PPH1"),
        caption = "Model 2a Coefficients")
```

Term	Estimate	SE	z-value	p-value	CI.LL	CI.UL	S	BFB	PPH1
Testlet1	0.13	0.09	1.51	1.30e-01	-0.04	0.30	2.94	1.39e+00	0.58
Testlet2	-0.70	0.12	-5.62	1.88e-08	-0.95	-0.46	25.66	1.10e+06	1.00
Testlet3	-0.15	0.21	-0.70	4.82e-01	-0.55	0.26	1.05	1.05e+00	0.51
Testlet4	-0.06	0.29	-0.22	8.24e-01	-0.62	0.51	0.28	2.30e+00	0.70
COPIC	0.84	0.06	13.50	1.67e-41	0.72	0.96	135.46	2.34e+38	1.00

Table 22: Model 2a Coefficients

8.1 Sequential Tests (Type I SS)

Each row in Table 23 tests the significance of unique additional variance explained by the term on that line after controlling for all previously entered terms. Significant results mean adding that term improved the model.

```
m2a %>%
 anova(., test = "Chisq") %>%
 cbind(., convertp(.[,"Pr(>Chi)"])) %>%
 kable(format = "latex", booktabs = TRUE,
       digits = c(0, 2, 0, 2, Inf, 2, 2, 2),
       col.names = c("DF", "Deviance", "Resid. DF", "Resid. Dev", "p-value",
                     "S", "BFB", "PPH1"),
       caption = "Model 2a Sequential Tests (Type I SS): Analysis of Deviance")
```

	DF	Deviance	Resid. DF	Resid. Dev	p-value	S	BFB	PPH1
NULL	NA	NA	1320	1829.91	NA	NA	NA	NA
Testlet	4	62.16	1316	1767.75	1.018257e-12	39.84	1.308385e+10	1
COPIC	1	245.07	1315	1522.68	3.089707e-55	181.08	9.486274e+51	1

Table 23: Model 2a Sequential Tests (Type I SS): Analysis of Deviance

8.2 Simultaneous Tests of Main Effects via LRT (Type III SS)

The simultaneous tests in Table 24 are the effects of the indicated terms after controlling for all other terms in the model.

```r
m2a %>%
  drop1(., test = "Chisq") %>%
  cbind(., convertp(.[,"Pr(>Chi)"])) %>%
  kable(format = "latex", booktabs = TRUE,
       digits = c(0, 2, 2, 2, Inf, 2, 2, 2),
       col.names = c("DF", "Deviance", "AIC", "LRT", "p-value", "S", "BFB",
                     "PPH1"),
       caption = "Model 2a Simultaneous Tests (Type III SS)")
```

	DF	Deviance	AIC	LRT	p-value	S	BFB	PPH1
<none>	NA	1522.68	1532.68	NA	NA	NA	NA	NA
Testlet	4	1558.33	1560.33	35.65	3.411467e-07	21.48	7.241722e+04	1
COPIC	1	1767.75	1775.75	245.07	3.089707e-55	181.08	9.486274e+51	1

Table 24: Model 2a Simultaneous Tests (Type III SS)

8.3 Conditional and Unconditional Pass Rates

Table 25 shows the conditional and unconditional pass rates estimated by Model 2a as a function of level transition testlet and COPIC scores.

```r
# Create a new data frame object for use with predict()
m2a.ND <- data.frame(Testlet = gl(n = 4, k = 1, length = 36,
                                  labels = c("1", "2", "3", "4")),
                     COPIC = rep(-4:4, each = 4), OPIC = rep(1:9, each = 4))

# Compute predicted mean passing rate at each combination of Level & Course
m2a.pred <- predict(m2a, newdata = m2a.ND, type = "link", se.fit = TRUE)

# Add fitted values and CIs to the new data frame & display it.
critval       <- qnorm(0.975) # For Wald 95% CIs
m2a.ND$fit    <- m2a.pred$fit
m2a.ND$se.fit <- m2a.pred$se.fit
m2a.ND$fit.LL <- with(m2a.ND, fit - (critval * se.fit))
m2a.ND$fit.UL <- with(m2a.ND, fit + (critval * se.fit))

# Convert fitted values and CIs to probabilities.
m2a.ND$Pass.Rate <- invlogit(m2a.ND$fit)
m2a.ND$Pass.LL   <- invlogit(m2a.ND$fit.LL)
m2a.ND$Pass.UL   <- invlogit(m2a.ND$fit.UL)

# Compute unconditional pass rates.
m2a.ND$Pass.URate <- c(cumprod(m2a.ND[m2a.ND$COPIC == -4, "Pass.Rate"]),
                       cumprod(m2a.ND[m2a.ND$COPIC == -3, "Pass.Rate"]),
                       cumprod(m2a.ND[m2a.ND$COPIC == -2, "Pass.Rate"]),
                       cumprod(m2a.ND[m2a.ND$COPIC == -1, "Pass.Rate"]),
                       cumprod(m2a.ND[m2a.ND$COPIC == 0, "Pass.Rate"]),
                       cumprod(m2a.ND[m2a.ND$COPIC == 1, "Pass.Rate"]),
                       cumprod(m2a.ND[m2a.ND$COPIC == 2, "Pass.Rate"]),
                       cumprod(m2a.ND[m2a.ND$COPIC == 3, "Pass.Rate"]),
                       cumprod(m2a.ND[m2a.ND$COPIC == 4, "Pass.Rate"]))

ShowVars <- c("Testlet", "COPIC", "OPIC", "Pass.Rate", "Pass.LL", "Pass.UL",
              "Pass.URate")

kable(m2a.ND[, ShowVars], format = "latex", booktabs = TRUE, digits = 2,
      format.args = list(nsmall = 2),
      caption = paste("Model 2a Conditional Pass Rates (with 95 percent CIs)",
                      "and Unconditional Pass Rates by Testlet and OPIc",
                    "Score")) %>%
kable_styling(latex_options = c("repeat_header"))
```

Testlet	COPIC	OPIC	Pass.Rate	Pass.LL	Pass.UL	Pass.URate
1	-4.00	1.00	0.04	0.02	0.06	0.04
2	-4.00	1.00	0.02	0.01	0.03	0.00
3	-4.00	1.00	0.03	0.02	0.06	0.00
4	-4.00	1.00	0.03	0.01	0.07	0.00
1	-3.00	2.00	0.08	0.06	0.12	0.08
2	-3.00	2.00	0.04	0.02	0.06	0.00
3	-3.00	2.00	0.07	0.04	0.11	0.00
4	-3.00	2.00	0.07	0.04	0.13	0.00
1	-2.00	3.00	0.18	0.14	0.21	0.18
2	-2.00	3.00	0.09	0.06	0.12	0.02
3	-2.00	3.00	0.14	0.09	0.21	0.00
4	-2.00	3.00	0.15	0.08	0.25	0.00
1	-1.00	4.00	0.33	0.29	0.37	0.33
2	-1.00	4.00	0.18	0.14	0.22	0.06
3	-1.00	4.00	0.27	0.19	0.37	0.02
4	-1.00	4.00	0.29	0.18	0.42	0.00
1	0.00	5.00	0.53	0.49	0.57	0.53
2	0.00	5.00	0.33	0.28	0.39	0.18
3	0.00	5.00	0.46	0.37	0.56	0.08
4	0.00	5.00	0.48	0.35	0.62	0.04
1	1.00	6.00	0.72	0.67	0.77	0.72
2	1.00	6.00	0.53	0.47	0.60	0.39
3	1.00	6.00	0.67	0.57	0.75	0.26
4	1.00	6.00	0.68	0.55	0.79	0.18
1	2.00	7.00	0.86	0.81	0.89	0.86
2	2.00	7.00	0.73	0.66	0.79	0.62
3	2.00	7.00	0.82	0.75	0.88	0.51
4	2.00	7.00	0.83	0.74	0.90	0.43
1	3.00	8.00	0.93	0.90	0.96	0.93
2	3.00	8.00	0.86	0.80	0.90	0.80
3	3.00	8.00	0.91	0.86	0.95	0.73
4	3.00	8.00	0.92	0.86	0.96	0.67
1	4.00	9.00	0.97	0.95	0.98	0.97
2	4.00	9.00	0.93	0.89	0.96	0.91
3	4.00	9.00	0.96	0.93	0.98	0.87
4	4.00	9.00	0.96	0.93	0.98	0.84

Table 25: Model 2a Conditional Pass Rates (with 95 percent CIs) and Unconditional Pass Rates by Testlet and OPIc Score

8.4 Odds-Ratio for COPIC Effect

Table 26 shows the odds-ratio associated with an extra point on the `COPIC` (i.e., a score of 6 vs 5 in the raw OPIc score). This quantifies the effect of COPIC, which this model assumes is equal across level transitions. We will test whether that assumption is reasonable in Model 2b later.

```
m2a.OR <- cbind(OR = exp(coef(m2a)[5]),
                LL = exp(confint(m2a, level = 0.95)[5, 1]),
                UL = exp(confint(m2a, level = 0.95)[5, 2]))
dimnames(m2a.OR)[[1]] <- "COPIC (0 vs 1)"
kable(m2a.OR, format = "latex", booktabs = TRUE, digits = 2,
      caption = paste("Model 2a Odds-Ratios for COPIC Effect (Estimate and ",
                      "95 Percent Confidence Interval)")) %>%
  kable_styling(latex_options = c("repeat_header"))
```

	OR	LL	UL
COPIC (0 vs 1)	2.31	2.05	2.61

Table 26: Model 2a Odds-Ratios for COPIC Effect (Estimate and 95 Percent Confidence Interval)

8.5 Assessing Goodness of Fit, Discrimination, and Calibration

8.5.1 Hosmer-Lemeshow Goodness of Fit Test

Figure 19 shows a calibration plot for this model. The plot is based on the bins summarized in Table 27, while the Hosmer-Lemeshow test is shown in Table 28.

```
FCap <- paste("\\label{fig:m2a-calib-plot}",
              "Model 2a Calibration Plot with Bin Sample Sizes and",
              "Hosmer-Lemeshow Test")
HLT.m2a <- HLfit(m2a, bin.method = "prob.bins", min.prob.interval = 0.1,
                 xlab = "Predicted Probability",
                 ylab = "Observed Probability")
```

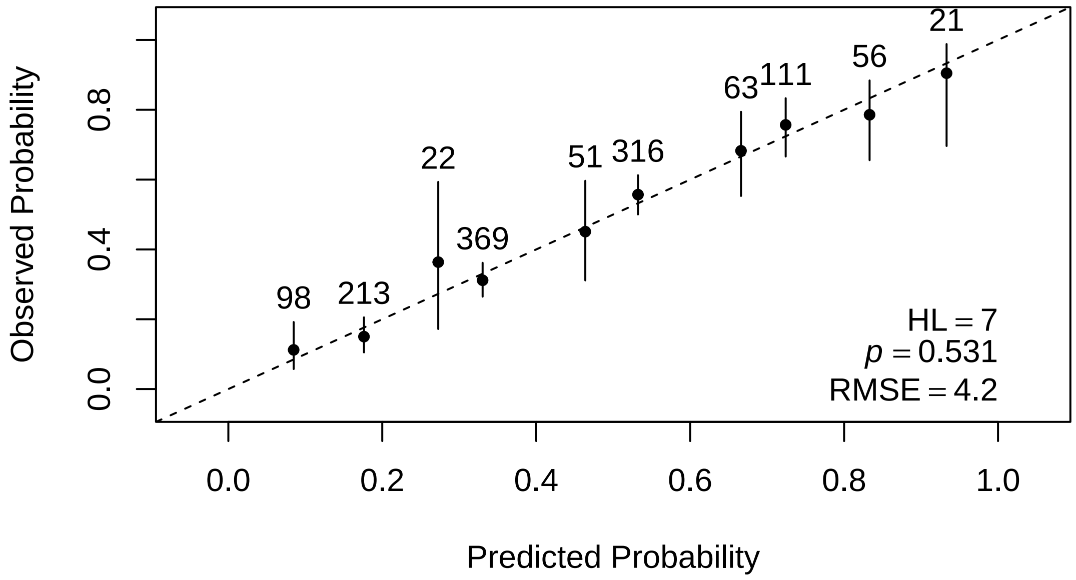

Figure 19: Model 2a Calibration Plot with Bin Sample Sizes and Hosmer-Lemeshow Test

```
FN <- "Minimum bin interval width = 0.10."

HLT.m2a$bins.table %>%
  kable(., format = "latex", booktabs = TRUE, digits = 3, row.names = FALSE,
        col.names = HLT.bin.vnames,
        caption = "Calibration Bins Used for Model 2a Hosmer-Lemeshow Test") %>%
  add_header_above(., header = c(" " = 4, "Obs. Prop. 95% CI" = 2)) %>%
  column_spec(column = 1, width = "1.75cm") %>%
  column_spec(column = 3:4, width = "1.75cm") %>%
  column_spec(column = 5:6, width = "1.25cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

				Obs. Prop. 95% CI	
Bin Center (Median)	Bin N	Observed Proportion	Predicted Proportion	Lower Limit	Upper Limit
0.085	98	0.112	0.077	0.057	0.192
0.176	213	0.150	0.176	0.105	0.205
0.273	22	0.364	0.277	0.172	0.593
0.330	369	0.312	0.331	0.265	0.362
0.464	51	0.451	0.469	0.311	0.597
0.532	316	0.557	0.533	0.500	0.613
0.666	63	0.683	0.673	0.553	0.794
0.724	111	0.757	0.724	0.666	0.833
0.833	56	0.786	0.842	0.656	0.884
0.933	21	0.905	0.930	0.696	0.988

Note: Minimum bin interval width = 0.10.

Table 27: Calibration Bins Used for Model 2a Hosmer-Lemeshow Test

```
FN <- "Minimum bin interval width = 0.10."

HLT.m2a %>%
  as_tibble() %>%
  select(chi.sq, DF, p.value, RMSE) %>%
  unique() %>%
  cbind(., convertp(p = .$p.value, digits = 2)) %>%
  kable(., format = "latex", booktabs = TRUE,
        digits = c(2, 0, Inf, 2, 2, 2, 2),
        col.names = HLT.col.vnames,
        caption = "Hosmer-Lemeshow Test for Goodness of Fit of Model 2a") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

Chi-square	df	p	RMSE	S	BFB	PPH1
7.05	8	0.5314931	4.24	0.91	1.1	0.52

Note: Minimum bin interval width = 0.10.

Table 28: Hosmer-Lemeshow Test for Goodness of Fit of Model 2a

8.5.2 Classification

Table 29 presents the classification measures for this model.

```
set.seed(5378) # For reproducible bootstrap estimates.
roc.m2a <- roc(Pass ~ pred.m2a, data = VTAData, ci = TRUE, direction = "<",
               ci.method = "bootstrap", boot.n = 10000)
```

```
## Setting levels: control = 0, case = 1
```

```
round(lrcm(roc.m2a, seed = 9825), digits = 3) %>%
  kable(format = "latex", booktabs = TRUE,
        caption = "Classification Measures for Model 2a") %>%
  add_header_above(header = c(" " = 2, "Bootstrapped Quantiles" = 3))
```

		Bootstrapped Quantiles		
	threshold	2.5%	50%	97.5%
threshold	0.474	0.398	0.474	0.508
specificity	0.731	0.681	0.724	0.761
sensitivity	0.676	0.640	0.683	0.730
accuracy	0.708	0.681	0.707	0.732
tn	559.000	520.975	554.000	582.000
tp	375.000	355.000	379.000	405.000
fn	180.000	150.000	176.000	200.000
fp	206.000	183.000	211.000	244.025
npv	0.756	0.736	0.759	0.786
ppv	0.645	0.610	0.643	0.675
fdr	0.355	0.325	0.357	0.390
fpr	0.269	0.239	0.276	0.319
tpr	0.676	0.640	0.683	0.730
tnr	0.731	0.681	0.724	0.761
fnr	0.324	0.270	0.317	0.360
1-specificity	0.269	0.239	0.276	0.319
1-sensitivity	0.324	0.270	0.317	0.360
1-accuracy	0.292	0.268	0.293	0.319
1-npv	0.244	0.214	0.241	0.264
1-ppv	0.355	0.325	0.357	0.390
precision	0.645	0.610	0.643	0.675
recall	0.676	0.640	0.683	0.730
youden	1.406	1.357	1.407	1.456
closest.topleft	0.178	0.148	0.177	0.208

Table 29: Classification Measures for Model 2a

8.5.3 Area Under the Curve (AUC)

Figure 20 shows the ROC curve for the model, annotated with the best classification threshold for balancing sensitivity versus specificity and the area under the curve (AUC).

```
FCap <- paste("\\label{fig:m2a-auc-plot}",
              "Model 2a Receiver Operating Characteristic (ROC) Curve.",
              "The dot marks the best classification threshold.",
              "AUC 95% confidence interval obtained via stratified bootstrap",
              "with 10,000 replicates.")
print(roc.m2a)
```

```
##
## Call:
## roc.formula(formula = Pass ~ pred.m2a, data = VTAData, ci = TRUE,     direction = "<", ci.method = "bootstrap", boot.n = 1000
##
## Data: pred.m2a in 765 controls (Pass 0) < 555 cases (Pass 1).
## Area under the curve: 0.7524
## 95% CI: 0.7258-0.7784 (10000 stratified bootstrap replicates)
```

```
plot.roc(roc.m2a, print.auc = TRUE, print.auc.cex = .8, print.thres = "best",
         print.thres.cex = .8)
```

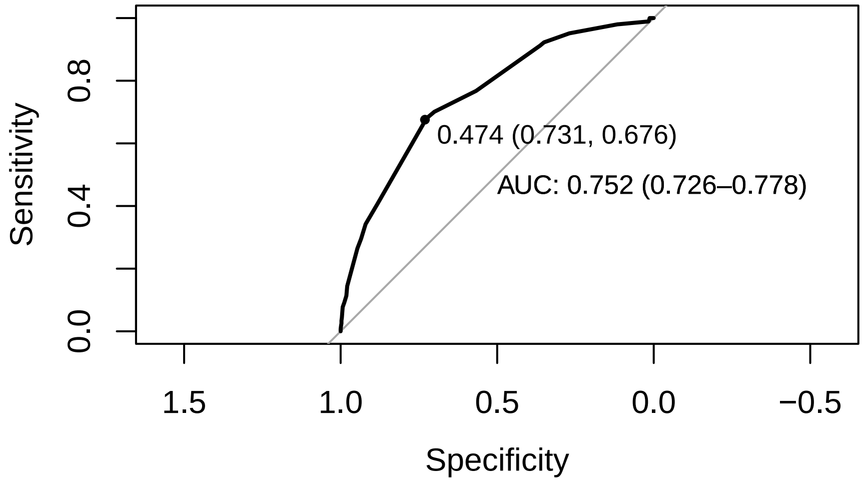

Figure 20: Model 2a Receiver Operating Characteristic (ROC) Curve. The dot marks the best classification threshold. AUC 95% confidence interval obtained via stratified bootstrap with 10,000 replicates.

Model 2a shows acceptable ability to discriminate those who pass from those who fail to pass the level transition testlets. Table 30 compares the ROC curves for Models 1a and 2a and shows that Model 2a performs better than Model 1a with respect to that.

```
set.seed(7436) # For reproducible bootstrap estimates.
roct.m1a.m2a <- roc.test(roc.m1a, roc.m2a, method = "bootstrap", boot.n = 10000)
glance(roct.m1a.m2a) %>%
  cbind(., convertp(p = .$p.value, digits = 2)) %>%
  select(boot.n, estimate1, estimate2, statistic, p.value, S, BFB, PPH1) %>%
  kable(format = "latex", booktabs = TRUE,
      digits = c(0, 3, 3, 2, Inf, 2, 2, 2),
      col.names = c("Bootstrap N", "Model 1a", "Model 2a", "D", "p", "S",
                    "BFB", "PPH1"),
      caption = paste("Comparing Models 1a and 2a Via Two-Sided, Stratified",
                      "Bootstrap Test for Correlated ROC Curves")) %>%
  add_header_above(header = c(" ", "AUC Estimates" = 2, " " = 5))
```

	AUC Estimates						
Bootstrap N	Model 1a	Model 2a	D	p	S	BFB	PPH1
10000	0.654	0.752	-7	2.524361e-12	38.53	5457087180	1

Table 30: Comparing Models 1a and 2a Via Two-Sided, Stratified Bootstrap Test for Correlated ROC Curves

8.5.4 R^2 Measures

The values for $R^2_p = 0.2$ and $R^2_{Dev} = 0.17$ are more encouraging than those from Models 1a and 1b.

8.6 Diagnostics

We need to run model diagnostics to see if there are any obvious problems with the model. First we check for outliers and find that there are none of note.

```
outlierTest(m2a)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferroni p
## 182 2.558714           0.010506           NA
```

Figure 21 shows residual plots for this model.

```
FCap <- paste("\\label{fig:m2a-resid-plots}",
              "Model 2a Residual Plots.")
residualPlots(m2a, layout = c(1,3), tests = FALSE)
```

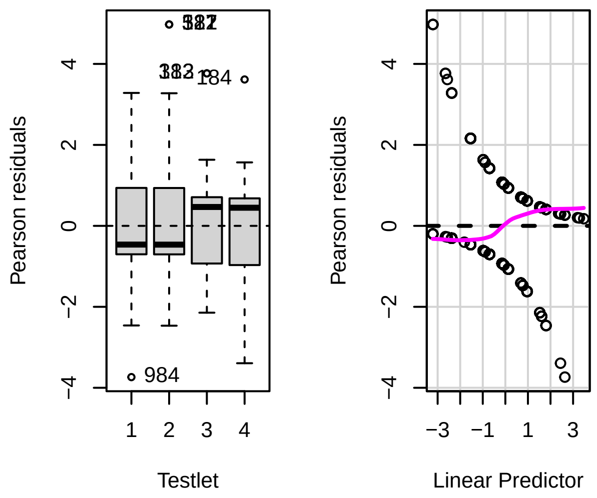

Figure 21: Model 2a Residual Plots.

Now we look at some index plots for influence measures. We are looking for observations with values exceeding the cutoffs. Figure 22 shows an index plot of Cook's D for this model.

```
FCap <- paste("\\label{fig:m2a-plotCookD}",
              "Model 2a Index Plot of Coook's D.",
              "The", sum(cooks.distance(m2a) > CookDco(m2a)), "observations",
              "with values exceeding the cutoff are shown in red.")
PlotCookD(m2a)
```

Figure 23 shows an index plot of leverage values for this model, while Figure 24 shows the standardized residuals versus the leverage values, with contours for Cook's D.

```
FCap <- paste("\\label{fig:m2a-plot-leverage}",
              "Model 2a Index Plot of Leverage Values.",
              "The", sum(hatvalues(m2a) > hatco(m2a)), "observations",
              "with values exceeding the cutoff are shown in red.")
PlotHat(m2a)
```

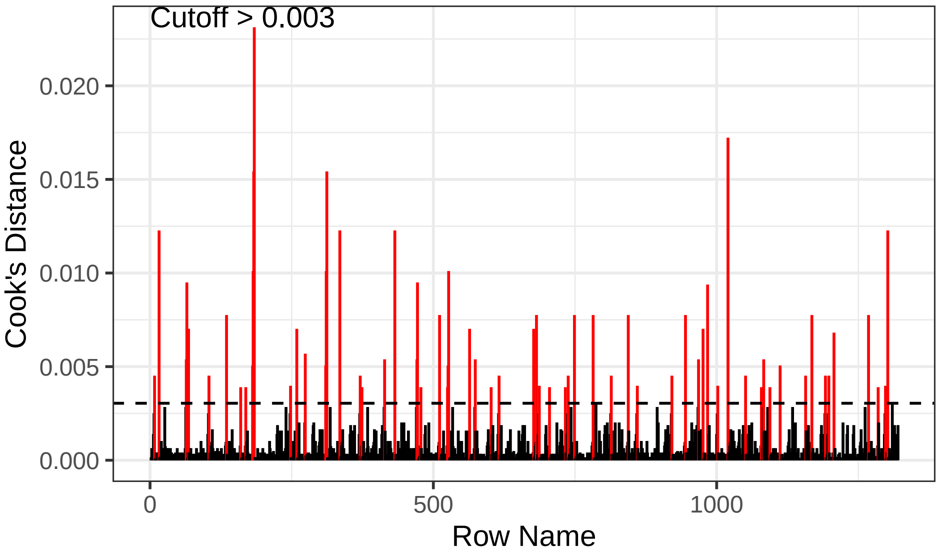

Figure 22: Model 2a Index Plot of Coook's D. The 67 observations with values exceeding the cutoff are shown in red.

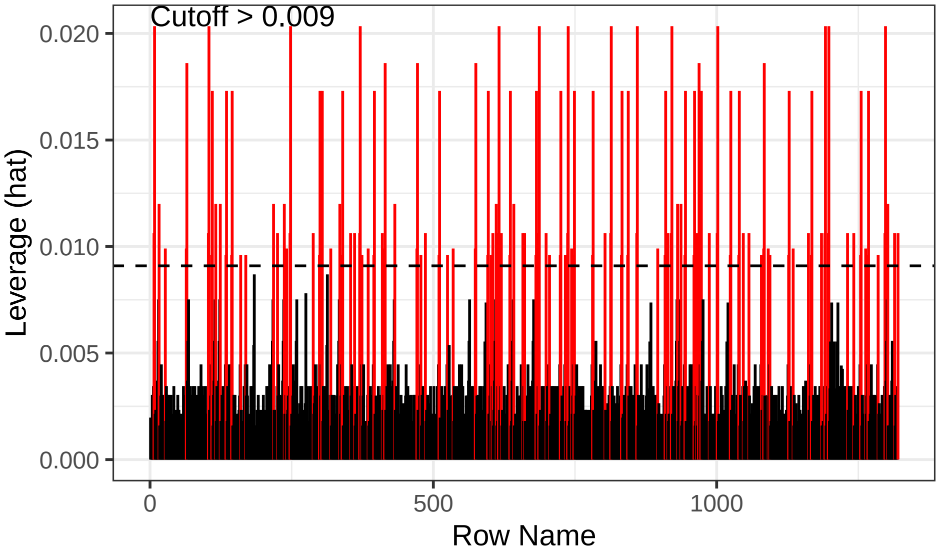

Figure 23: Model 2a Index Plot of Leverage Values. The 148 observations with values exceeding the cutoff are shown in red.

```r
FCap <- paste("\\label{fig:m2a-plot-Pearson-leverage}",
              "Model 2a Plot of Standardized Pearson Residuals Versus",
              "Leverage Values, with Cook's D Contours.")
plot(m2a, which = 5, id.n = 10, cex.id = .6, caption = "", sub.caption = "",
     cook.levels = round(CookDco(m2a), digits = 3))
abline(v = hatco(m2a), lty = 2, col = "blue")
text(x = hatco(m2a), y = 5, pos = 4, col = "blue", cex = .75,
     labels = paste("Leverage cutoff >", round(hatco(m2a), digits = 3)))
```

Table 31 shows a list of influential cases for Model 2a. While there are a fair number of cases with both high Cook's distance and high leverage, only a few also have large standardized Pearson residuals.

```r
FN <- paste0("All cases shown have high leverage (hat) and Cook's D values, ",
            "defined by hat > ", round(hatco(m2a), digits = 3),
```

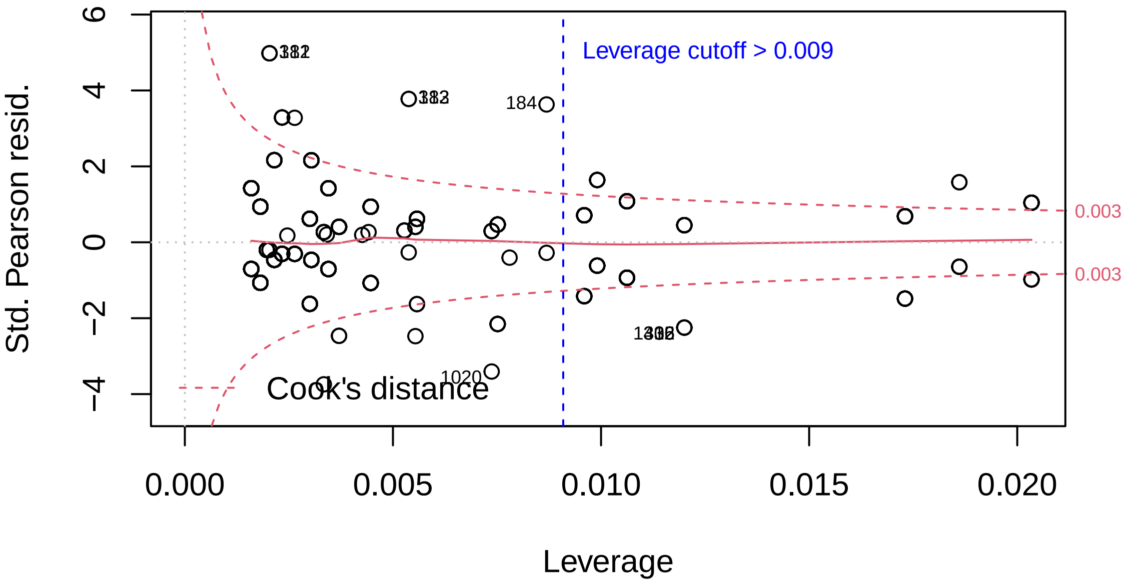

Figure 24: Model 2a Plot of Standardized Pearson Residuals Versus Leverage Values, with Cook's D Contours.

```
             " and Cook's D > ", round(CookDco(m2a), digits = 3),
             ". Standardized Pearson residuals with absolute values > 1.96 are ",
             "flagged.")

# Identify cases w/ high leverage and high Cook's D.
m2a %>% InfCases(.) %>%
  mutate(Flag = if_else(abs(StdPearson) > 1.96, true = "x", false = "")) %>%
  kable(format = "latex", booktabs = TRUE, longtable = TRUE, digits = 3,
        caption = "Model 2a Influential Cases") %>%
  kable_styling(latex_options = c("repeat_header")) %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

Table 31: Model 2a Influential Cases

	Pass	Testlet	COPIC	hat	CookD	StdPearson	Flag
8	1	4	0	0.020	0.005	1.043	
16	0	4	2	0.012	0.012	-2.248	x
64	1	3	-1	0.010	0.005	1.641	
65	1	4	-1	0.019	0.009	1.583	
104	1	4	0	0.020	0.005	1.043	
135	0	4	1	0.017	0.008	-1.484	
160	0	3	1	0.010	0.004	-1.419	
169	0	3	1	0.010	0.004	-1.419	
248	0	4	0	0.020	0.004	-0.979	
335	0	4	2	0.012	0.012	-2.248	x
371	1	4	0	0.020	0.005	1.043	
374	0	3	1	0.010	0.004	-1.419	
414	1	3	-1	0.010	0.005	1.641	
432	0	4	2	0.012	0.012	-2.248	x
471	1	3	-1	0.010	0.005	1.641	
472	1	4	-1	0.019	0.009	1.583	
478	0	3	1	0.010	0.004	-1.419	

Table 31: Model 2a Influential Cases *(continued)*

	Pass	Testlet	COPIC	hat	CookD	StdPearson	Flag
511	0	4	1	0.017	0.008	-1.484	
525	0	3	1	0.010	0.004	-1.419	
574	1	3	-1	0.010	0.005	1.641	
602	0	3	1	0.010	0.004	-1.419	
616	1	4	0	0.020	0.005	1.043	
682	0	4	1	0.017	0.008	-1.484	
687	0	4	0	0.020	0.004	-0.979	
705	0	3	1	0.010	0.004	-1.419	
733	0	3	1	0.010	0.004	-1.419	
738	1	4	0	0.020	0.005	1.043	
749	0	4	1	0.017	0.008	-1.484	
782	0	4	1	0.017	0.008	-1.484	
814	1	4	0	0.020	0.005	1.043	
844	0	4	1	0.017	0.008	-1.484	
860	0	4	0	0.020	0.004	-0.979	
921	1	4	0	0.020	0.005	1.043	
945	0	4	1	0.017	0.008	-1.484	
968	1	3	-1	0.010	0.005	1.641	
1002	0	4	0	0.020	0.004	-0.979	
1079	0	3	1	0.010	0.004	-1.419	
1083	1	3	-1	0.010	0.005	1.641	
1094	0	3	1	0.010	0.004	-1.419	
1168	0	4	1	0.017	0.008	-1.484	
1192	1	4	0	0.020	0.005	1.043	
1198	1	4	0	0.020	0.005	1.043	
1268	0	4	1	0.017	0.008	-1.484	
1285	0	3	1	0.010	0.004	-1.419	
1298	0	4	0	0.020	0.004	-0.979	
1302	0	4	2	0.012	0.012	-2.248	x

Note: All cases shown have high leverage (hat) and Cook's D values, defined by hat > 0.009 and Cook's D > 0.003. Standardized Pearson residuals with absolute values > 1.96 are flagged.

8.7 Graphs

8.7.1 Main Effect of Testlet

Figures 25 and 26 respectively plot the main effect of the self-assessment level transition testlet on the scale of the linear predictor (log-odds) and on the response scale (probability or conditional pass rates).

```
FCap <- paste("\\label{fig:m2a-plot-Testlet-LogOdds}",
              "Model 2a Main Effect of Testlet on Log-Odds Scale.")
visreg(m2a, xvar = "Testlet", ylab = "Log odds (Pass)")
```

```
FCap <- paste("\\label{fig:m2a-plot-Testlet-CPR}",
              "Model 2a Main Effect of Testlet on Probability Scale.")
visreg(m2a, xvar = "Testlet", ylab = "Conditional Pass Rate",
       scale = "response", rug = 2, ylim = c(0, 1))
```

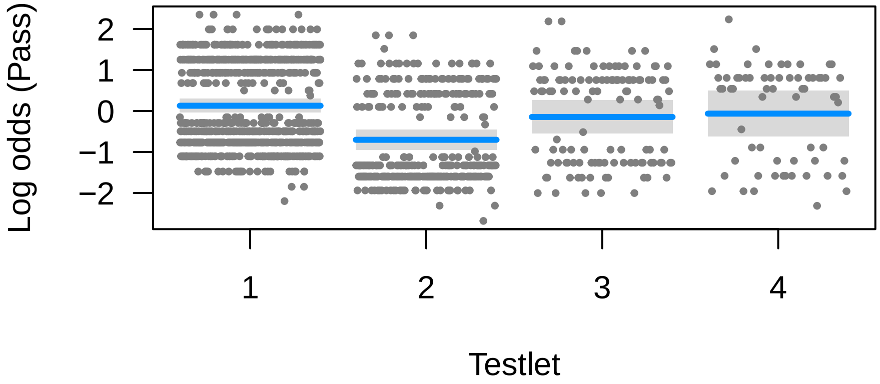

Figure 25: Model 2a Main Effect of Testlet on Log-Odds Scale.

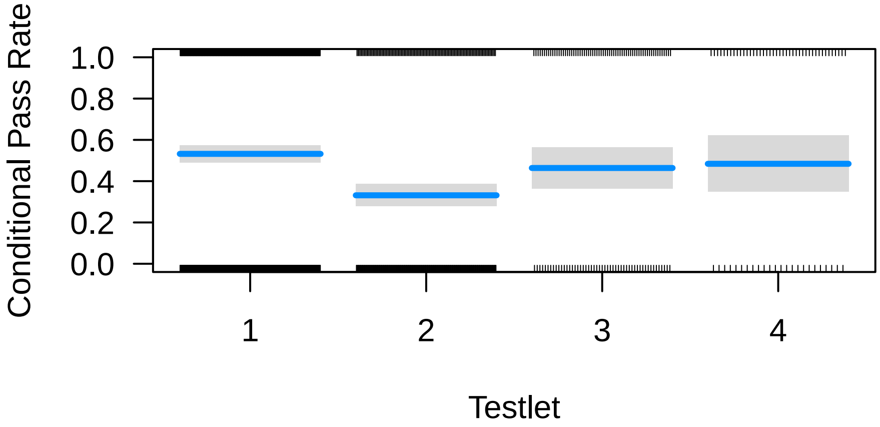

Figure 26: Model 2a Main Effect of Testlet on Probability Scale.

8.7.2 Main Effect of COPIC

Figures 27 and 28 respectively plot the main effect of COPIC on the scale of the linear predictor (log-odds) and on the response scale (probability or conditional pass rates).

```
FCap <- paste("\\label{fig:m2a-plot-COPIC-LogOdds}",
              "Model 2a Main Effect of Centered OPIc Score on Log-Odds Scale.")
visreg(m2a, xvar = "COPIC", jitter = TRUE, scale = "linear",
       xlab = "Centered OPIC Score (OPIC - 5)",
       ylab = "Log odds (Pass)")
```

```
FCap <- paste("\\label{fig:m2a-plot-COPIC-CPR}",
              "Model 2a Main Effect of Centered OPIc Score on Probability Scale.")
visreg(m2a, xvar = "COPIC", jitter = TRUE, scale = "response", rug = 2,
       xlab = "Centered OPIC Score (OPIC - 5)",
       ylab = "Conditional Pass Rate")
```

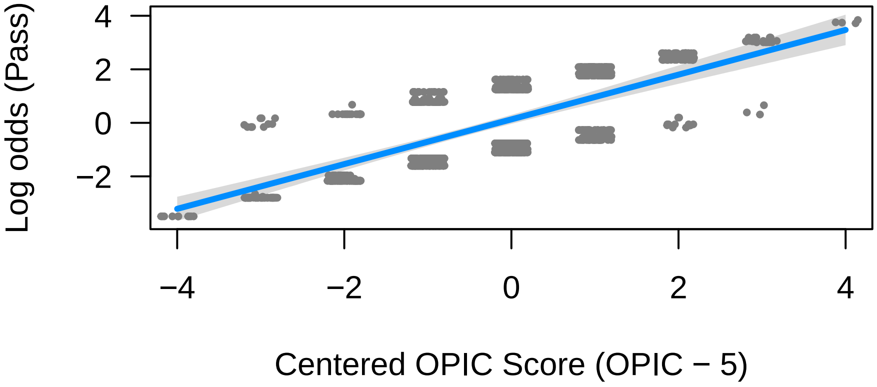

Figure 27: Model 2a Main Effect of Centered OPIc Score on Log-Odds Scale.

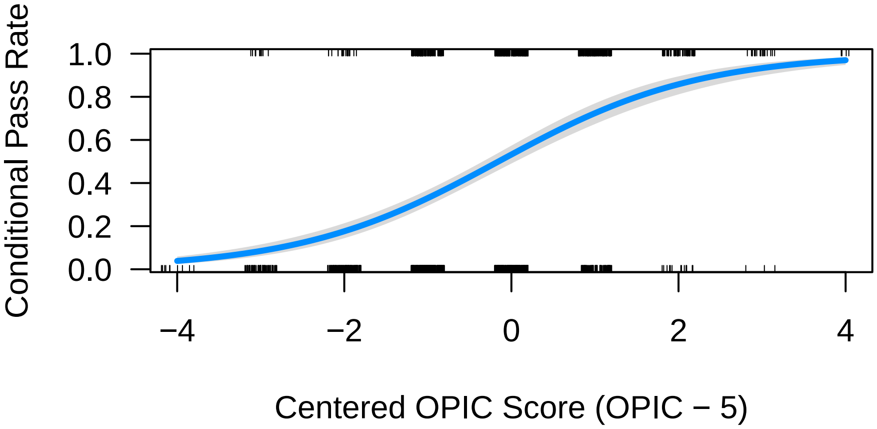

Figure 28: Model 2a Main Effect of Centered OPIc Score on Probability Scale.

8.7.3 Conditional Pass Rates

Figure 29 plots the conditional pass rates derived from Model 2a as a function of both testlet and OPIc score.

```
FCap <- paste("\\label{fig:m2a-plot-CPR}",
              "Model 2a Conditional Pass Rates by OPIc Score and",
              "Self-Assessment Testlet.",
              "Lines are labeled with OPIc score.")
m2a.FCPR <- ggplot(m2a.ND[, ShowVars],
                   aes(Testlet, Pass.Rate, group=OPIC, color=OPIC))+
  geom_line()+ coord_cartesian(ylim = c(0, 1))+
  xlab("Self-assessment Testlet")+
  ylab("Conditional Pass Rate")
  direct.label(m2a.FCPR, "first.points")
```

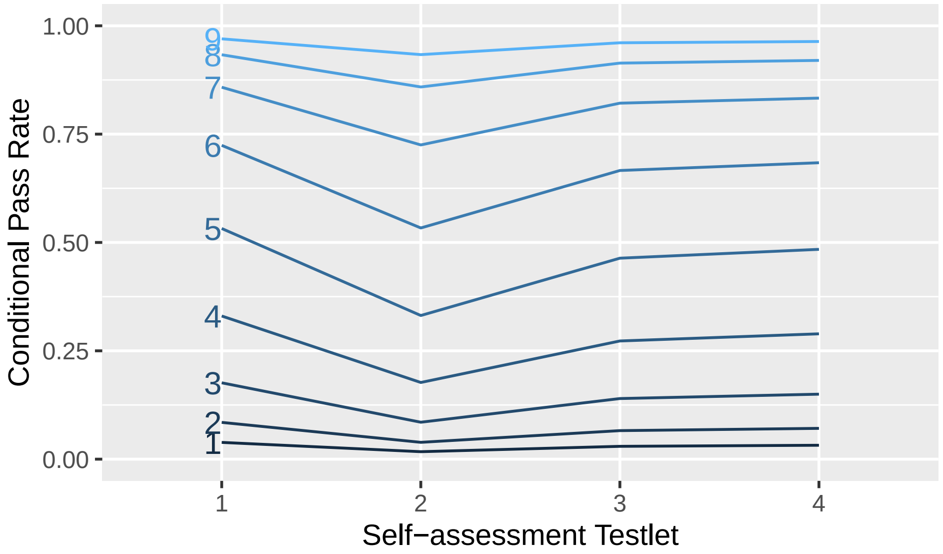

Figure 29: Model 2a Conditional Pass Rates by OPIc Score and Self-Assessment Testlet. Lines are labeled with OPIc score.

```
# for the manuscript
ggsave("F4.png", plot=direct.label(m2a.FCPR, "first.points"),
       width = 6.5, height = 8.5, dpi = 300)
```

8.7.4 Unconditional Pass Rates

Figure 30 plots the unconditional pass rates derived from Model 2a as a function of both testlet and OPIc score.

```
FCap <- paste("\\label{fig:m2a-plot-UPR}",
              "Model 2a Unconditional Pass Rates by OPIc Score and",
              "Self-Assessment Testlet.",
              "Lines are labeled with OPIc score.")
m2a.FUPR <- ggplot(m2a.ND[, ShowVars],
                   aes(Testlet, Pass.URate, group=OPIC, color=OPIC))+
  geom_line()+ coord_cartesian(ylim = c(0, 1))+
  xlab("Self-assessment Testlet")+
  ylab("Unconditional Pass Rate")
  direct.label(m2a.FUPR, "first.points")
```

```
# for the manuscript
ggsave("F5.png", plot=direct.label(m2a.FUPR, "first.points"),
       width = 9, height = 5.5, dpi = 300)
```

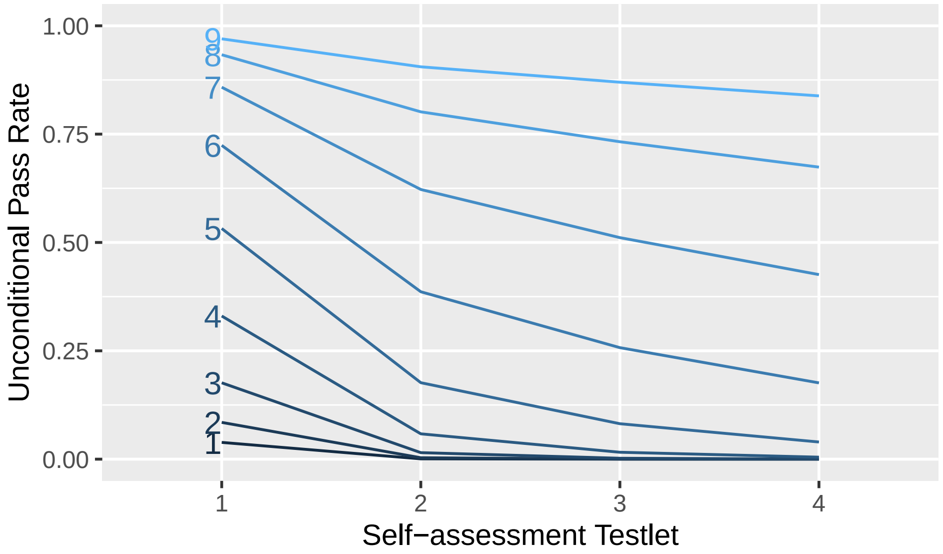

Figure 30: Model 2a Unconditional Pass Rates by OPIc Score and Self-Assessment Testlet. Lines are labeled with OPIc score.

8.8 Conclusion

Model 2a suggests that higher OPIC scores are associated with higher passing rates across the level transitions. Furthermore, Model 2a is clearly better than Model 1a. However, we still need to run another model to see whether relaxing the assumption that the OPIC effect is constant across testlets is viable. We do that in Model 2b.

9 Model 2b: Non-parallel OPIc Effect

We now relax the assumption that the OPIC effect is constant across level transitions. We again fit a model that omits the intercept term in order to simplify post-processing of the model results into interpretable estimates. We are still using the centered version of OPIC in the model. Table 32 below shows the raw parameter estimates, confidence intervals, s-values, BFBs, and posterior probabilities of H1 corresponding to the p-values.

```
m2b %>%
  tidy(., conf.int = TRUE, conf.level = .95) %>%
  cbind(., convertp(.$p.value, digits = 2)) %>%
  kable(format = "latex", booktabs = TRUE, format.args = list(digits = 3),
        digits = c(2, 2, 2, 2, Inf, 2, 2, 2, 2, 2),
        col.names = c("Term", "Estimate", "SE", "z-value", "p-value", "CI.LL",
                      "CI.UL", "S", "BFB", "PPH1"),
        caption = "Model 2b Coefficients")
```

We can see from Table 4 that the difference in AICs ($\Delta AIC = AIC_{2a} - AIC_{2b} = 4.247$) favors Model 2b, which has a lower AIC value than Model 2a, but not by a large margin. Meanwhile, the difference in BICs ($\Delta BIC = BIC_{2a} - BIC_{2b} = -11.309$) favors Model 2a by a larger margin because it imposes a stronger penalty for model complexity (lack of parsimony).

Term	Estimate	SE	z-value	p-value	CI.LL	CI.UL	S	BFB	PPH1
Testlet1	0.19	0.09	2.11	3.51e-02	0.01	0.37	4.83	3.13e+00	0.76
Testlet2	-0.69	0.13	-5.51	3.64e-08	-0.94	-0.45	24.71	5.89e+05	1.00
Testlet3	-0.01	0.21	-0.06	9.52e-01	-0.44	0.40	0.07	7.85e+00	0.89
Testlet4	0.31	0.30	1.02	3.09e-01	-0.29	0.91	1.69	1.01e+00	0.50
COPIC	0.96	0.08	11.65	2.25e-31	0.80	1.13	101.81	2.31e+28	1.00
Testlet2:COPIC	-0.17	0.16	-1.08	2.80e-01	-0.47	0.14	1.84	1.03e+00	0.51
Testlet3:COPIC	-0.41	0.20	-2.12	3.43e-02	-0.79	-0.01	4.87	3.18e+00	0.76
Testlet4:COPIC	-0.65	0.22	-2.89	3.87e-03	-1.08	-0.18	8.01	1.71e+01	0.94

Table 32: Model 2b Coefficients

9.1 Sequential Tests (Type I SS)

Each row in Table 33 tests the significance of unique additional variance explained by the term on that line after controlling for all previously entered terms. Significant results mean adding that term improved the model. Here, we want to focus on the result for the interaction effect. That tells us whether allowing OPIc to have a level-specific effect instead of a constant effect across levels improved the model.

```
m2b %>%
 anova(., test = "Chisq") %>%
 cbind(., convertp(.[,"Pr(>Chi)"])) %>%
 kable(format = "latex", booktabs = TRUE,
       digits = c(0, 2, 0, 2, Inf, 2, 2, 2),
       col.names = c("DF", "Deviance", "Resid. DF", "Resid. Dev", "p-value",
                     "S", "BFB", "PPH1"),
       caption = "Model 2b Sequential Tests (Type I SS): Analysis of Deviance")
```

	DF	Deviance	Resid. DF	Resid. Dev	p-value	S	BFB	PPH1
NULL	NA	NA	1320	1829.91	NA	NA	NA	NA
Testlet	4	62.16	1316	1767.75	1.018257e-12	39.84	1.308385e+10	1.00
COPIC	1	245.07	1315	1522.68	3.089707e-55	181.08	9.486274e+51	1.00
Testlet:COPIC	3	10.25	1312	1512.43	1.657570e-02	5.91	5.410000e+00	0.84

Table 33: Model 2b Sequential Tests (Type I SS): Analysis of Deviance

9.2 Simultaneous Tests of Interaction Effects via LRT (Type III SS)

Table 34 shows the result of another way of testing the interaction: a likelihood ratio test (LRT) for comparing nested models.

```
anova(m2a, m2b, test = "Chisq") %>%
  cbind(., convertp(.[,"Pr(>Chi)"])) %>%
  kable(format = "latex", booktabs = TRUE,
       digits = c(0, 2, 0, 2, Inf, 2, 2, 2),
       col.names = c("Resid. Df", "Resid. Deviance", "Df", "Deviance",
                     "p-value", "S", "BFB", "PPH1"),
       caption = "Model 2a vs 2b: Likelihood Ratio Test for Interaction")
```

Resid. Df	Resid. Deviance	Df	Deviance	p-value	S	BFB	PPH1
1315	1522.68	NA	NA	NA	NA	NA	NA
1312	1512.43	3	10.25	0.0165757	5.91	5.41	0.84

Table 34: Model 2a vs 2b: Likelihood Ratio Test for Interaction

9.3 Conditional and Unconditional Pass Rates

Table 35 shows the conditional and unconditional pass rates estimated by Model 2b as a function of level transition testlet and COPIC scores.

```r
# Create a new data frame object for use with predict()
m2b.ND <- data.frame(Testlet = gl(n = 4, k = 1, length = 36,
                                  labels = c("1", "2", "3", "4")),
                     COPIC = rep(-4:4, each = 4),
                     OPIC = rep(1:9, each = 4))

# Compute predicted mean passing rate at each combination of Level & Course
m2b.pred <- predict(m2b, newdata = m2b.ND, type = "link", se.fit = TRUE)

# Add fitted values and CIs to the new data frame & display it.
critval      <- qnorm(0.975)  # For Wald 95% CIs
m2b.ND$fit    <- m2b.pred$fit
m2b.ND$se.fit <- m2b.pred$se.fit
m2b.ND$fit.LL <- with(m2b.ND, fit - (critval * se.fit))
m2b.ND$fit.UL <- with(m2b.ND, fit + (critval * se.fit))

# Convert fitted values and CIs to probabilities.
m2b.ND$Pass.Rate <- invlogit(m2b.ND$fit)
m2b.ND$Pass.LL   <- invlogit(m2b.ND$fit.LL)
m2b.ND$Pass.UL   <- invlogit(m2b.ND$fit.UL)

# Compute unconditional pass rates.
m2b.ND$Pass.URate <- c(cumprod(m2b.ND[m2b.ND$COPIC == -4, "Pass.Rate"]),
                       cumprod(m2b.ND[m2b.ND$COPIC == -3, "Pass.Rate"]),
                       cumprod(m2b.ND[m2b.ND$COPIC == -2, "Pass.Rate"]),
                       cumprod(m2b.ND[m2b.ND$COPIC == -1, "Pass.Rate"]),
                       cumprod(m2b.ND[m2b.ND$COPIC == 0, "Pass.Rate"]),
                       cumprod(m2b.ND[m2b.ND$COPIC == 1, "Pass.Rate"]),
                       cumprod(m2b.ND[m2b.ND$COPIC == 2, "Pass.Rate"]),
                       cumprod(m2b.ND[m2b.ND$COPIC == 3, "Pass.Rate"]),
                       cumprod(m2b.ND[m2b.ND$COPIC == 4, "Pass.Rate"]))

ShowVars <- c("Testlet", "COPIC", "OPIC", "Pass.Rate", "Pass.LL", "Pass.UL",
              "Pass.URate")
kable(m2b.ND[, ShowVars], format = "latex", booktabs = TRUE, digits = 2,
      format.args = list(nsmall = 2),
      caption = paste("Conditional Pass Rates (with 95 percent CIs) and",
                      "Unconditional Pass Rates by Level Transition and Course")) %>%
kable_styling(latex_options = c("repeat_header"))
```

Testlet	COPIC	OPIC	Pass.Rate	Pass.LL	Pass.UL	Pass.URate
1	-4.00	1.00	0.03	0.01	0.04	0.03
2	-4.00	1.00	0.02	0.01	0.06	0.00
3	-4.00	1.00	0.10	0.02	0.36	0.00
4	-4.00	1.00	0.28	0.05	0.74	0.00
1	-3.00	2.00	0.06	0.04	0.10	0.06
2	-3.00	2.00	0.04	0.02	0.10	0.00
3	-3.00	2.00	0.16	0.05	0.41	0.00
4	-3.00	2.00	0.35	0.10	0.73	0.00
1	-2.00	3.00	0.15	0.12	0.19	0.15
2	-2.00	3.00	0.09	0.05	0.16	0.01
3	-2.00	3.00	0.25	0.11	0.46	0.00
4	-2.00	3.00	0.42	0.18	0.72	0.00
1	-1.00	4.00	0.32	0.28	0.36	0.32
2	-1.00	4.00	0.18	0.13	0.25	0.06
3	-1.00	4.00	0.36	0.23	0.52	0.02
4	-1.00	4.00	0.50	0.29	0.71	0.01
1	0.00	5.00	0.55	0.50	0.59	0.55
2	0.00	5.00	0.33	0.28	0.39	0.18
3	0.00	5.00	0.50	0.39	0.60	0.09
4	0.00	5.00	0.58	0.43	0.71	0.05
1	1.00	6.00	0.76	0.70	0.81	0.76
2	1.00	6.00	0.52	0.44	0.60	0.40
3	1.00	6.00	0.63	0.53	0.72	0.25
4	1.00	6.00	0.65	0.53	0.76	0.16
1	2.00	7.00	0.89	0.84	0.93	0.89
2	2.00	7.00	0.71	0.59	0.81	0.63
3	2.00	7.00	0.75	0.60	0.85	0.47
4	2.00	7.00	0.72	0.55	0.84	0.34
1	3.00	8.00	0.96	0.92	0.97	0.96
2	3.00	8.00	0.84	0.71	0.92	0.81
3	3.00	8.00	0.83	0.66	0.93	0.67
4	3.00	8.00	0.78	0.55	0.91	0.52
1	4.00	9.00	0.98	0.96	0.99	0.98
2	4.00	9.00	0.92	0.81	0.97	0.91
3	4.00	9.00	0.90	0.71	0.97	0.81
4	4.00	9.00	0.83	0.53	0.95	0.67

Table 35: Conditional Pass Rates (with 95 percent CIs) and Unconditional Pass Rates by Level Transition and Course

9.4 Odds-Ratios for COPIC Effect

Table 36 contains the odds-ratios showing simple effect of COPIC at each level transition testlet, which vary because this model relaxed the assumption of parallel effect across testlets. It also shows contrasts that estimate pairwise differences in the simple slope of COPIC between testlets.

```
# Create objects to hold row names and a contrast matrix (K).
RN <- c("Slope @ T1", "Slope @ T2", "Slope @ T3", "Slope @ T4",
        "Slope diff: T1 - T2", "Slope diff: T1 - T3",
        "Slope diff: T1 - T4", "Slope diff: T2 - T3",
        "Slope diff: T2 - T4", "Slope diff: T3 - T4")
K <- matrix(c(0, 0, 0, 0, 1, 0, 0, 0,
              0, 0, 0, 0, 1, 1, 0, 0,
              0, 0, 0, 0, 1, 0, 1, 0,
              0, 0, 0, 0, 1, 0, 0, 1,
              0, 0, 0, 0, 0, -1, 0, 0,
              0, 0, 0, 0, 0, 0, -1, 0,
              0, 0, 0, 0, 0, 0, 0, -1,
              0, 0, 0, 0, 0, 1, -1, 0,
              0, 0, 0, 0, 0, 1, 0, -1,
              0, 0, 0, 0, 0, 0, 1, -1),
            nrow=10, byrow = TRUE,
            dimnames = list(RN, names(coef(m2b))))

# Run multiple comparisons examine the COPIC effect & get adjusted 95% CIs.
m2b.ct <- glht(m2b, linfct = K)
m2b.mc <- summary(m2b.ct, test = adjusted("Westfall"))
m2b.ci <- confint(m2b.ct, calpha = adjusted_calpha(test = "Westfall"))
m2b.OR <- data.frame(Est   = m2b.mc$test$coefficients,
                  SE    = m2b.mc$test$sigma,
                  CI.LL = m2b.ci$confint[, "lwr"],
                  CI.UL = m2b.ci$confint[, "upr"],
                  OR    = exp(m2b.mc$test$coefficients),
                  OR.LL = exp(m2b.ci$confint[, "lwr"]),
                  OR.UL = exp(m2b.ci$confint[, "upr"]),
                  z     = m2b.mc$test$tstat,
                  p     = m2b.mc$test$pvalues,
                  Sval  = p2s(m2b.mc$test$pvalues),
                  BFB   = p2bfb(m2b.mc$test$pvalues),
                  PPH1  = p2pp(m2b.mc$test$pvalues))
m2b.OR$OR[5:10]    <- NA
m2b.OR$OR.LL[5:10] <- NA
m2b.OR$OR.UL[5:10] <- NA
kable(m2b.OR, format = "latex", booktabs = TRUE, format.args = list(digits = 3),
      digits = c(rep(x = 2, times = 8), Inf, 2, 2, 2),
      caption = paste("Simple Slopes of COPIC Effect, Westfall (1997)",
                      "Adjustment for Multiplicity")) %>%
kable_styling(latex_options = c("repeat_header", "scale_down"))
```

	Est	SE	CI.LL	CI.UL	OR	OR.LL	OR.UL	z	p	Sval	BFB	PPH1
Slope @ T1	0.96	0.08	0.74	1.18	2.61	2.10	3.25	11.65	0.00e+00	Inf	NaN	NaN
Slope @ T2	0.79	0.13	0.44	1.14	2.21	1.55	3.14	5.98	8.07e-09	26.88	2.45e+06	1.00
Slope @ T3	0.54	0.18	0.07	1.02	1.72	1.07	2.77	3.06	8.11e-03	6.95	9.42e+00	0.90
Slope @ T4	0.31	0.21	-0.24	0.87	1.37	0.78	2.38	1.49	2.52e-01	1.99	1.06e+00	0.51
Slope diff: T1 - T2	0.17	0.16	-0.25	0.58	NA	NA	NA	1.08	4.81e-01	1.06	1.04e+00	0.51
Slope diff: T1 - T3	0.41	0.20	-0.11	0.94	NA	NA	NA	2.12	1.15e-01	3.11	1.48e+00	0.60
Slope diff: T1 - T4	0.65	0.22	0.05	1.25	NA	NA	NA	2.89	1.92e-02	5.70	4.85e+00	0.83
Slope diff: T2 - T3	0.25	0.22	-0.35	0.84	NA	NA	NA	1.11	2.66e-01	1.91	1.04e+00	0.51
Slope diff: T2 - T4	0.48	0.25	-0.18	1.14	NA	NA	NA	1.94	1.26e-01	2.99	1.41e+00	0.58
Slope diff: T3 - T4	0.23	0.27	-0.50	0.96	NA	NA	NA	0.85	4.81e-01	1.06	1.04e+00	0.51

Table 36: Simple Slopes of COPIC Effect, Westfall (1997) Adjustment for Multiplicity

9.5 Assessing Assessing Goodness of Fit, Discrimination, and Calibration

9.5.1 Hosmer-Lemeshow Goodness of Fit Test

Figure 31 shows a calibration plot for this model. The plot is based on the bins summarized in Table 37, while the Hosmer-Lemeshow test is shown in Table 38.

```
FCap <- paste("\\label{fig:m2b-calib-plot}",
              "Model 2b Calibration Plot with Bin Sample Sizes and",
              "Hosmer-Lemeshow Test.")
HLT.m2b <- HLfit(m2b, bin.method = "prob.bins", min.prob.interval = 0.17,
                 xlab = "Predicted Probability",
                 ylab = "Observed Probability")
```

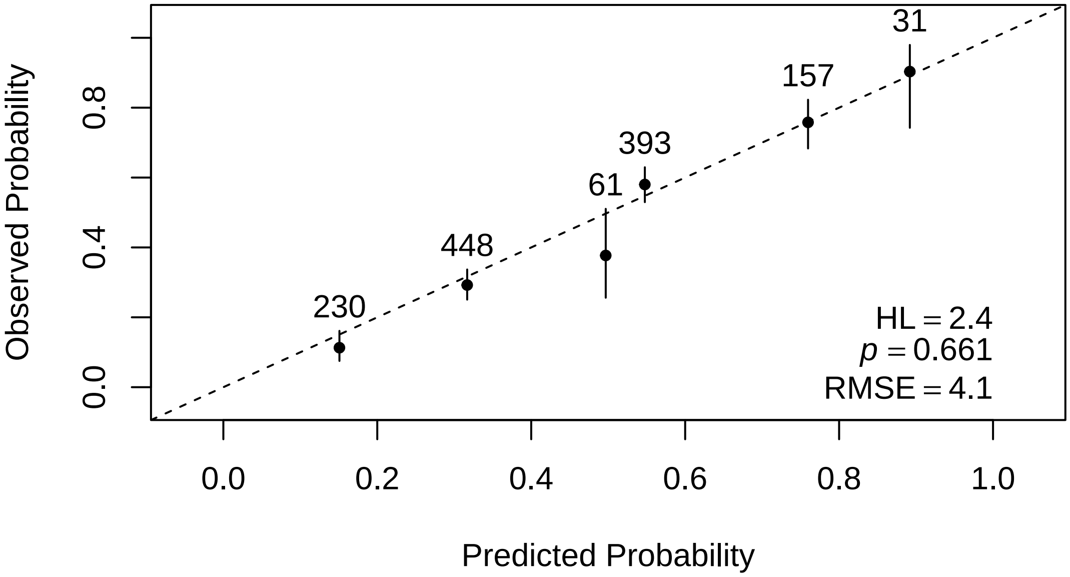

Figure 31: Model 2b Calibration Plot with Bin Sample Sizes and Hosmer-Lemeshow Test.

```
FN <- "Minimum bin interval width = 0.17."

HLT.m2b$bins.table %>%
  kable(., format = "latex", booktabs = TRUE, digits = 3, row.names = FALSE,
        col.names = HLT.bin.vnames,
        caption = "Calibration Bins Used for Model 2b Hosmer-Lemeshow Test") %>%
  add_header_above(., header = c(" " = 4, "Obs. Prop. 95% CI" = 2)) %>%
  column_spec(column = 1, width = "1.75cm") %>%
  column_spec(column = 3:4, width = "1.75cm") %>%
  column_spec(column = 5:6, width = "1.25cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

				Obs. Prop. 95% CI	
Bin Center (Median)	Bin N	Observed Proportion	Predicted Proportion	Lower Limit	Upper Limit
0.151	230	0.113	0.116	0.075	0.161
0.317	448	0.292	0.299	0.251	0.337
0.497	61	0.377	0.457	0.256	0.510
0.548	393	0.580	0.559	0.530	0.629
0.760	157	0.758	0.755	0.683	0.823
0.892	31	0.903	0.910	0.742	0.980

Note: Minimum bin interval width = 0.17.

Table 37: Calibration Bins Used for Model 2b Hosmer-Lemeshow Test

```
FN <- "Minimum bin interval width = 0.17."

HLT.m2b %>%
  as_tibble() %>%
  select(chi.sq, DF, p.value, RMSE) %>%
  unique() %>%
  cbind(., convertp(p = .$p.value, digits = 2)) %>%
  kable(., format = "latex", booktabs = TRUE,
        digits = c(2, 0, Inf, 2, 2, 2, 2),
        col.names = HLT.col.vnames,
        caption = "Hosmer-Lemeshow Test for Goodness of Fit of Model 2b") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

Chi-square	df	p	RMSE	S	BFB	PPH1
2.41	4	0.6606655	4.09	0.6	1.34	0.57

Note: Minimum bin interval width = 0.17.

Table 38: Hosmer-Lemeshow Test for Goodness of Fit of Model 2b

9.5.2 Classification

Table 39 presents the classification measures for this model.

```
set.seed(3179) # For reproducible bootstrap estimates.
roc.m2b <- roc(Pass ~ pred.m2b, data = VTAData, ci = TRUE, direction = "<",
               ci.method = "bootstrap", boot.n = 10000)
```

```
## Setting levels: control = 0, case = 1
```

```
round(lrcm(roc.m2b, seed = 7146), digits = 3) %>%
  kable(format = "latex", booktabs = TRUE,
        caption = "Classification Measures for Model 2b") %>%
  add_header_above(header = c(" " = 2, "Bootstrapped Quantiles" = 3))
```

		Bootstrapped Quantiles		
	threshold	2.5%	50%	97.5%
threshold	0.512	0.341	0.512	0.512
specificity	0.731	0.668	0.723	0.761
sensitivity	0.676	0.641	0.688	0.741
accuracy	0.708	0.681	0.708	0.732
tn	559.000	511.000	553.000	582.000
tp	375.000	356.000	382.000	411.000
fn	180.000	144.000	173.000	199.000
fp	206.000	183.000	212.000	254.000
npv	0.756	0.737	0.762	0.787
ppv	0.645	0.608	0.642	0.675
fdr	0.355	0.325	0.358	0.392
fpr	0.269	0.239	0.277	0.332
tpr	0.676	0.641	0.688	0.741
tnr	0.731	0.668	0.723	0.761
fnr	0.324	0.259	0.312	0.359
1-specificity	0.269	0.239	0.277	0.332
1-sensitivity	0.324	0.259	0.312	0.359
1-accuracy	0.292	0.268	0.292	0.319
1-npv	0.244	0.213	0.238	0.263
1-ppv	0.355	0.325	0.358	0.392
precision	0.645	0.608	0.642	0.675
recall	0.676	0.641	0.688	0.741
youden	1.406	1.359	1.411	1.456
closest.topleft	0.178	0.149	0.175	0.207

Table 39: Classification Measures for Model 2b

9.5.3 Area Under the Curve (AUC)

Figure 32 shows the ROC curve for the model, annotated with the best classification threshold for balancing sensitivity versus specificity and the area under the curve (AUC).

```
FCap <- paste("\\label{fig:m2b-auc-plot}",
              "Model 2b Receiver Operating Characteristic (ROC) Curve.",
              "The dot marks the best classification threshold.",
              "AUC 95% confidence interval obtained via stratified bootstrap",
              "with 10,000 replicates.")
print(roc.m2b)
```

```
##
## Call:
## roc.formula(formula = Pass ~ pred.m2b, data = VTAData, ci = TRUE,     direction = "<", ci.method = "bootstrap", boot.n = 1000
##
## Data: pred.m2b in 765 controls (Pass 0) < 555 cases (Pass 1).
## Area under the curve: 0.7557
## 95% CI: 0.7295-0.7809 (10000 stratified bootstrap replicates)
```

```
plot.roc(roc.m2b, print.auc = TRUE, print.auc.cex = .8, print.thres = "best",
         print.thres.cex = .8)
```

Model 2b shows acceptable ability to discriminate those who pass from those who fail to pass the level transition testlets, but Table 40 shows that it does not perform better at that than Model 2a.

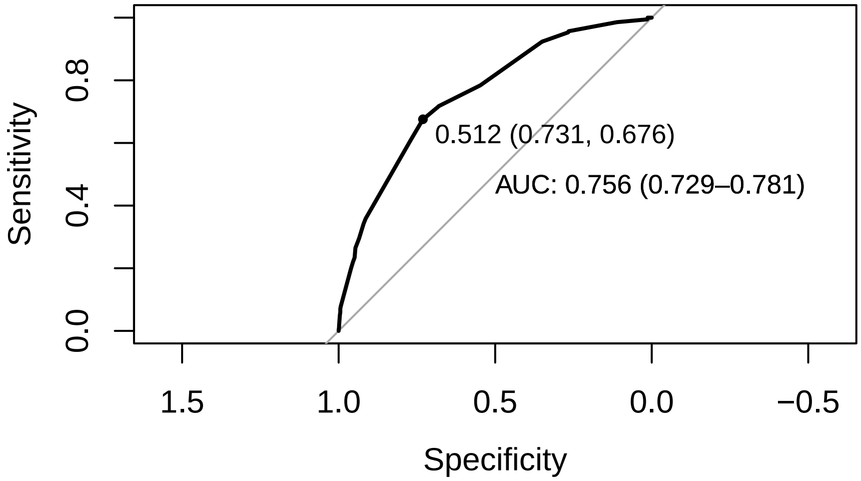

Figure 32: Model 2b Receiver Operating Characteristic (ROC) Curve. The dot marks the best classification threshold. AUC 95% confidence interval obtained via stratified bootstrap with 10,000 replicates.

```
set.seed(1128) # For reproducible bootstrap estimates.
roct.m2a.m2b <- roc.test(roc.m2a, roc.m2b, method = "bootstrap", boot.n = 10000)
glance(roct.m2a.m2b) %>%
  cbind(., convertp(p = .$p.value, digits = 2)) %>%
  select(boot.n, estimate1, estimate2, statistic, p.value, S, BFB, PPH1) %>%
  kable(format = "latex", booktabs = TRUE,
        digits = c(0, 3, 3, 2, Inf, 2, 2, 2),
        col.names = c("Bootstrap N", "Model 2a", "Model 2b", "D", "p", "S",
                      "BFB", "PPH1"),
        caption = paste("Comparing Models 2a and 2b Via Two-Sided, Stratified",
                        "Bootstrap Test for Correlated ROC Curves")) %>%
  add_header_above(header = c(" ", "AUC Estimates" = 2, " " = 5))
```

	AUC Estimates						
Bootstrap N	Model 2a	Model 2b	D	p	S	BFB	PPH1
10000	0.752	0.756	-0.82	0.4099252	1.29	1.01	0.5

Table 40: Comparing Models 2a and 2b Via Two-Sided, Stratified Bootstrap Test for Correlated ROC Curves

9.5.4 R^2 Measures

The values for $R^2_p = 0.2$ and $R^2_{Dev} = 0.17$ are more encouraging than those from Models 1a and 1b, but identical to those from Model 2a.

9.6 Diagnostics

We need to run model diagnostics to see if there are any obvious problems with the model. First we check for outliers and find that there are none of note.

```
outlierTest(m2b)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferroni p
## 182 2.530127           0.011402           NA
```

Figure 33 shows residual plots for this model.

```
FCap <- paste("\\label{fig:m2b-resid-plots}",
              "Model 2b Residual Plots.")
residualPlots(m2b, layout = c(1,3), tests = FALSE)
```

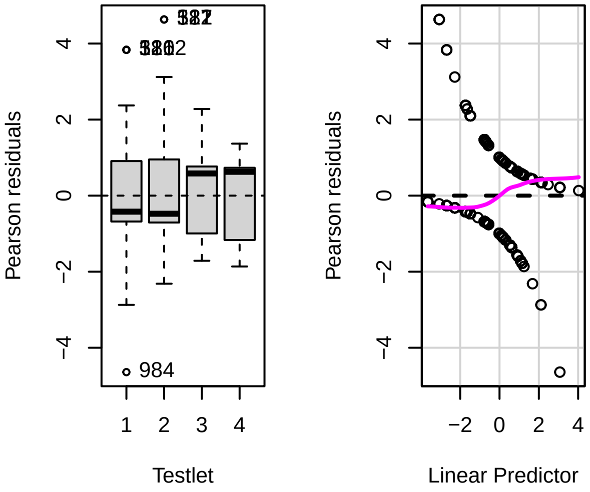

Figure 33: Model 2b Residual Plots.

Now we look at some index plots for influence measures. We are looking for observations with values exceeding the cutoffs. Figure 34 shows an index plot of Cook's D for this model.

```
FCap <- paste("\\label{fig:m2b-plotCookD}",
              "Model 2b Index Plot of Coook's D.",
              "The", sum(cooks.distance(m2b) > CookDco(m2b)), "observations",
              "with values exceeding the cutoff are shown in red.")
PlotCookD(m2b)
```

Figure 35 shows an index plot of leverage values for this model, while Figure 36 shows the standardized residuals versus the leverage values, with contours for Cook's D.

```
FCap <- paste("\\label{fig:m2b-plot-leverage}",
              "Model 2b Index Plot of Leverage Values.",
              "The", sum(hatvalues(m2b) > hatco(m2b)), "observations",
              "with values exceeding the cutoff are shown in red.")
PlotHat(m2b)
```

```
FCap <- paste("\\label{fig:m2b-plot-Pearson-leverage}",
              "Model 2b Plot of Standardized Pearson Residuals Versus",
              "Leverage Values, with Cook's D Contours.")
plot(m2b, which = 5, id.n = 10, cex.id = .6, caption = "", sub.caption = "",
     cook.levels = round(CookDco(m2b), digits = 3))
abline(v = hatco(m2b), lty = 2, col = "blue")
text(x = hatco(m2b), y = 5, pos = 4, col = "blue", cex = .75,
     labels = paste("Leverage cutoff >", round(hatco(m2b), digits = 3)))
```

Table 41 shows a list of influential cases for Model 2b. While there are a fair number of cases with both high Cook's distance and high leverage, only a few also have large standardized Pearson residuals.

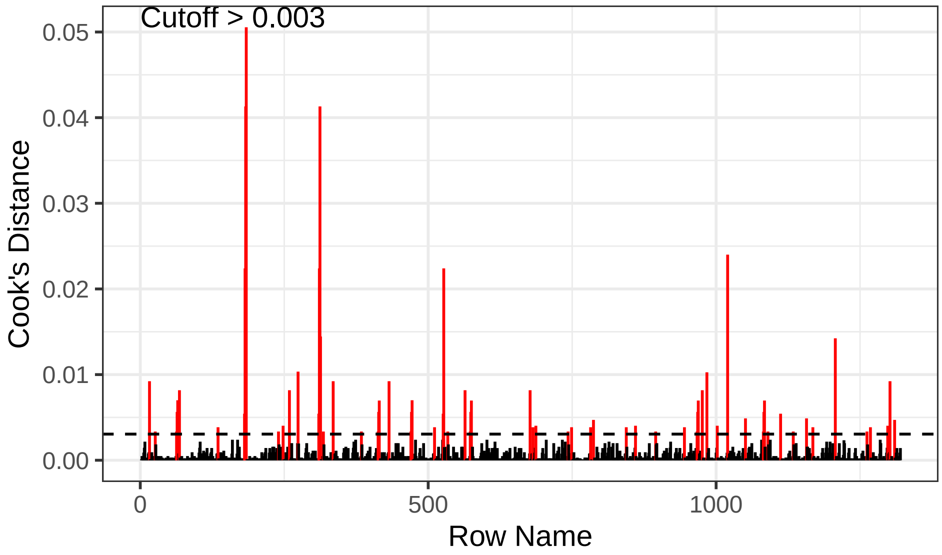

Figure 34: Model 2b Index Plot of Coook’s D. The 70 observations with values exceeding the cutoff are shown in red.

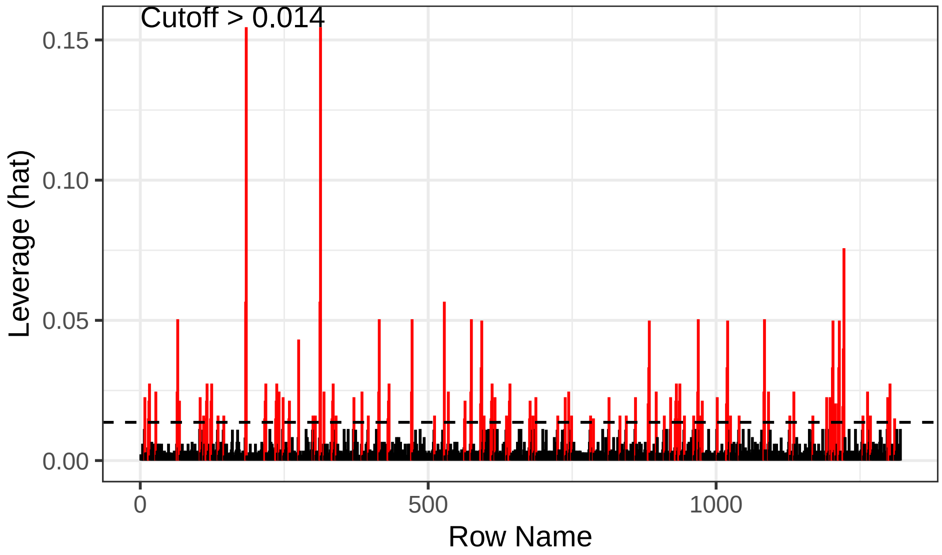

Figure 35: Model 2b Index Plot of Leverage Values. The 135 observations with values exceeding the cutoff are shown in red.

```
FN <- paste0("All cases shown have high leverage (hat) and Cook's D values, ",
            "defined by hat > ", round(hatco(m2b), digits = 3),
            " and Cook's D > ", round(CookDco(m2b), digits = 3),
            ". Standardized Pearson residuals with absolute values > 1.96 are ",
            "flagged.")

# Identify cases w/ high leverage and high Cook's D.
m2b %>% InfCases(.) %>%
  mutate(Flag = if_else(abs(StdPearson) > 1.96, true = "x", false = "")) %>%
  kable(format = "latex", booktabs = TRUE, longtable = TRUE, digits = 3,
        caption = "Model 2b Influential Cases") %>%
  kable_styling(latex_options = c("repeat_header")) %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

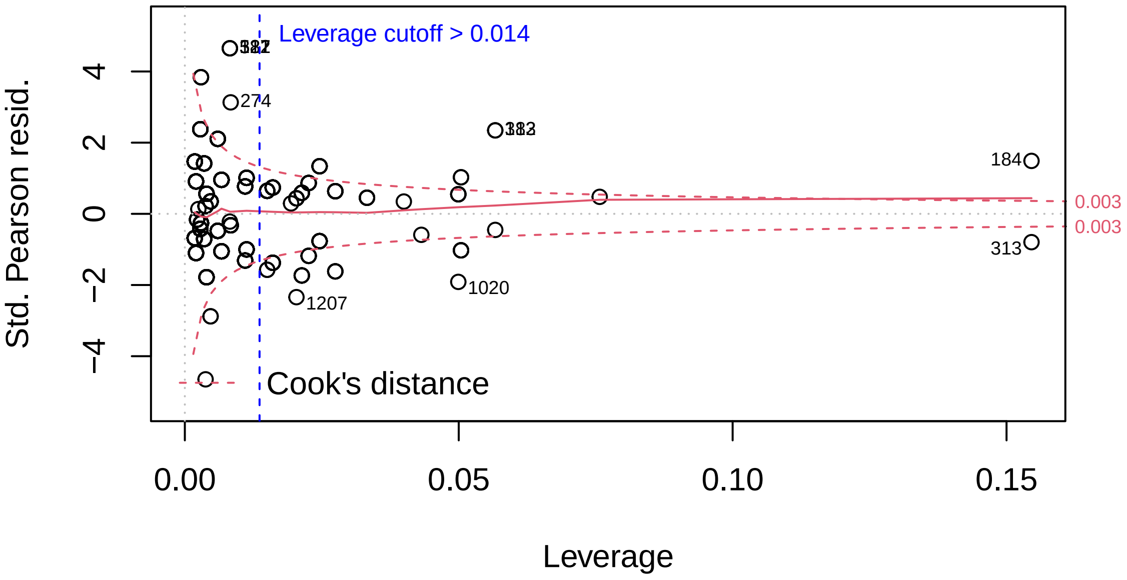

Figure 36: Model 2b Plot of Standardized Pearson Residuals Versus Leverage Values, with Cook's D Contours.

Table 41: Model 2b Influential Cases

	Pass	Testlet	COPIC	hat	CookD	StdPearson	Flag
16	0	4	2	0.027	0.009	-1.617	
64	1	3	-1	0.025	0.006	1.338	
65	1	4	-1	0.050	0.007	1.027	
68	0	3	2	0.021	0.008	-1.732	
135	0	4	1	0.016	0.004	-1.375	
183	1	3	-3	0.057	0.041	2.346	x
184	1	4	-3	0.155	0.051	1.487	
248	0	4	0	0.023	0.004	-1.181	
259	0	3	2	0.021	0.008	-1.732	
312	1	3	-3	0.057	0.041	2.346	x
313	0	4	-3	0.155	0.014	-0.795	
335	0	4	2	0.027	0.009	-1.617	
414	1	3	-1	0.025	0.006	1.338	
415	0	4	-1	0.050	0.007	-1.025	
432	0	4	2	0.027	0.009	-1.617	
471	1	3	-1	0.025	0.006	1.338	
472	1	4	-1	0.050	0.007	1.027	
511	0	4	1	0.016	0.004	-1.375	
564	0	3	2	0.021	0.008	-1.732	
574	1	3	-1	0.025	0.006	1.338	
575	0	4	-1	0.050	0.007	-1.025	
677	0	3	2	0.021	0.008	-1.732	
682	0	4	1	0.016	0.004	-1.375	
687	0	4	0	0.023	0.004	-1.181	
749	0	4	1	0.016	0.004	-1.375	
782	0	4	1	0.016	0.004	-1.375	
787	0	2	2	0.015	0.005	-1.572	
844	0	4	1	0.016	0.004	-1.375	
860	0	4	0	0.023	0.004	-1.181	
945	0	4	1	0.016	0.004	-1.375	

Table 41: Model 2b Influential Cases *(continued)*

	Pass	Testlet	COPIC	hat	CookD	StdPearson	Flag
968	1	3	-1	0.025	0.006	1.338	
969	0	4	-1	0.050	0.007	-1.025	
976	0	3	2	0.021	0.008	-1.732	
1002	0	4	0	0.023	0.004	-1.181	
1020	0	4	3	0.050	0.024	-1.912	
1083	1	3	-1	0.025	0.006	1.338	
1084	0	4	-1	0.050	0.007	-1.025	
1168	0	4	1	0.016	0.004	-1.375	
1207	0	2	3	0.020	0.014	-2.340	x
1268	0	4	1	0.016	0.004	-1.375	
1298	0	4	0	0.023	0.004	-1.181	
1302	0	4	2	0.027	0.009	-1.617	
1310	0	2	2	0.015	0.005	-1.572	

Note: All cases shown have high leverage (hat) and Cook's D values, defined by hat > 0.014 and Cook's D > 0.003. Standardized Pearson residuals with absolute values > 1.96 are flagged.

9.7 Graphs

9.7.1 Interaction Effect of COPIC by Testlet

We visualize below the transition testlet-specific OPIc effect. Figures 37 and 38 respectively plot the centered OPIC score by testlet interaction effect by showing the simple effect of centered OPIc score separately for each self-assessment transition testlet on the scale of the linear predictor (log-odds) and on the response scale (probability or conditional pass rates). The effect of OPIc decreases as we go from the first to the fourth testlet, as evidenced by the shallower slope from the left-most to the right-most panel.

```
FCap <- paste("\\label{fig:m2b-plot-Interaction-LogOdds}",
              "Model 2b Interaction Effect of COPIC by Testlet on Log-Odds",
              "Scale.")
visreg(m2b, xvar = "COPIC", by = "Testlet", layout = c(4, 1), jitter = TRUE,
       strip.names = TRUE, scale = "linear",
       xlab = "Centered OPIC Score (OPIC - 5)",
       ylab = "Log odds (Pass)")
```

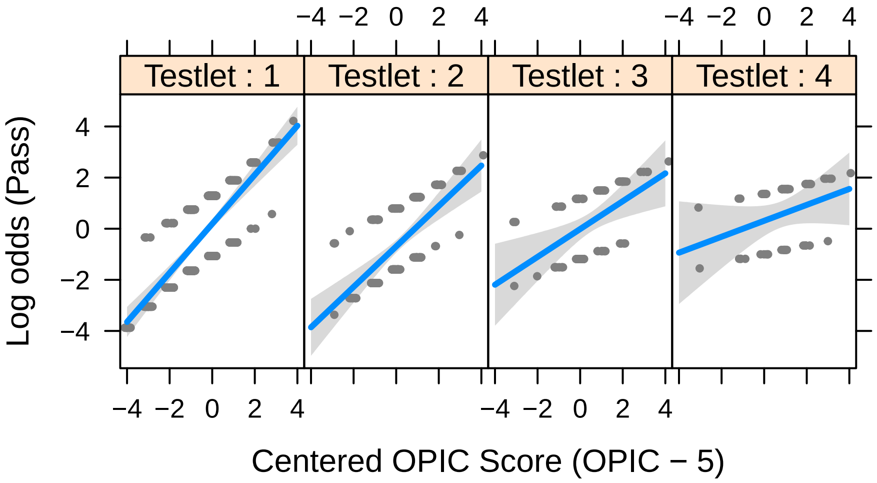

Figure 37: Model 2b Interaction Effect of COPIC by Testlet on Log-Odds Scale.

```
FCap <- paste("\\label{fig:m2b-plot-Interaction-CPR}",
              "Model 2b Interaction Effect of COPIC by Testlet on",
              "Probability Scale.")
visreg(m2b, xvar = "COPIC", by = "Testlet", layout = c(4, 1), jitter = TRUE,
       strip.names = TRUE, scale = "response", rug = 2,
       xlab = "Centered OPIC Score (OPIC - 5)",
       ylab = "Conditional Pass Rate")
```

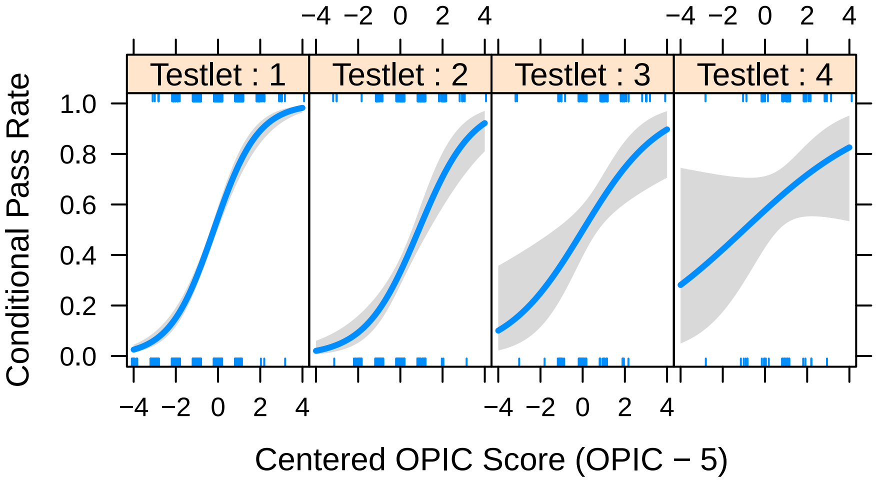

Figure 38: Model 2b Interaction Effect of COPIC by Testlet on Probability Scale.

9.7.2 Conditional Pass Rates

Figure 39 plots the conditional pass rates derived from Model 2b as a function of both testlet and OPIc score.

```
FCap <- paste("\\label{fig:m2b-plot-CPR}",
              "Model 2b Conditional Pass Rates by OPIc Score and",
              "Self-Assessment Testlet.",
              "Lines are labeled with OPIc score.")
m2b.FCPR <- ggplot(m2b.ND[, ShowVars],
                   aes(Testlet, Pass.Rate, group=OPIC, color=OPIC))+
      geom_line()+ coord_cartesian(ylim = c(0, 1))+
      xlab("Self-Assessment Testlet")+
      ylab("Conditional Pass Rate")
direct.label(m2b.FCPR, "first.points")
```

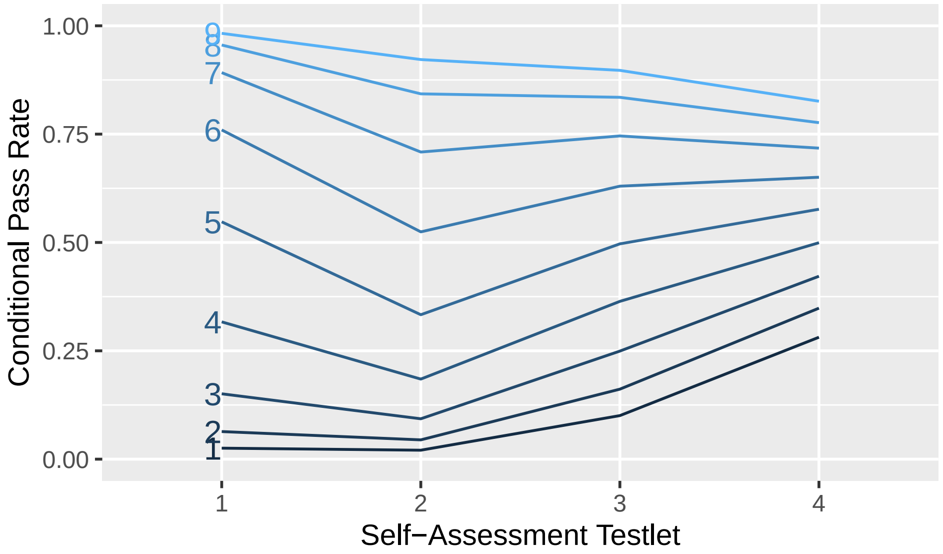

Figure 39: Model 2b Conditional Pass Rates by OPIc Score and Self-Assessment Testlet. Lines are labeled with OPIc score.

9.7.3 Unconditional Pass Rates

Figure 40 plots the unconditional pass rates derived from Model 2b as a function of both testlet and OPIc score.

```
FCap <- paste("\\label{fig:m2b-plot-UPR}",
              "Model 2b Unconditional Pass Rates by OPIc Score and",
              "Self-Assessment Testlet.",
              "Lines are labeled with OPIc score.")
m2b.FUPR <- ggplot(m2b.ND[, ShowVars],
                   aes(Testlet, Pass.URate, group=OPIC, color=OPIC))+
      geom_line()+ coord_cartesian(ylim = c(0, 1))+
      xlab("Self-Assessment Testlet")+
      ylab("Unconditional Pass Rate")
direct.label(m2b.FUPR, "first.points")
```

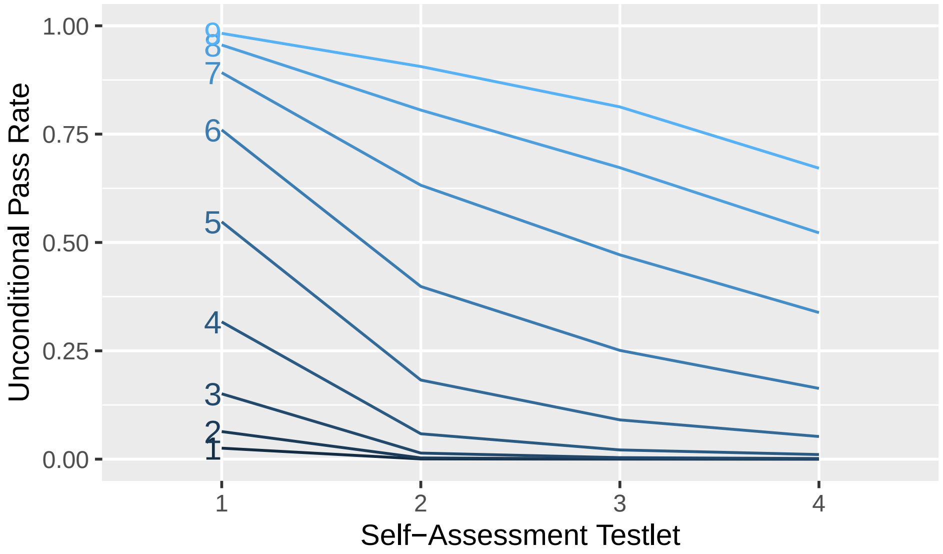

Figure 40: Model 2b Unconditional Pass Rates by OPIc Score and Self-Assessment Testlet. Lines are labeled with OPIc score.

9.8 Conclusion

Model 2b is not really any better than Model 2a even though the Testlet x COPIC interaction is significant. It has better accuracy and sensitivity than Model 1a, but we need to run models to decide whether adding a course effect to Model 2a or 2b will improve the situation. That will be done in Models 3a and 3b.

10 Model 3a: Parallel Course + Parallel OPIc Effects

We now test a model with a parallel course effect and a parallel OPIc effect. We want to see if course and OPIc are still significant predictors of the conditional pass rates when they are both in the same model. Model 3a tells us to what extent OPIc and Course each predict the conditional pass rates, after controlling for the other predictor's effect. Table 42 below shows the raw parameter estimates, confidence intervals, s-values, BFBs, and posterior probabilities of H1 corresponding to the p-values.

```
m3a %>%
  tidy(., conf.int = TRUE, conf.level = .95) %>%
  cbind(., convertp(.$p.value, digits = 2)) %>%
  kable(format = "latex", booktabs = TRUE, format.args = list(digits = 3),
        digits = c(2, 2, 2, 2, Inf, 2, 2, 2, 2, 2),
        col.names = c("Term", "Estimate", "SE", "z-value", "p-value", "CI.LL",
                      "CI.UL", "S", "BFB", "PPH1"),
        caption = "Model 3a Coefficients")
```

Term	Estimate	SE	z-value	p-value	CI.LL	CI.UL	S	BFB	PPH1
Testlet1	0.22	0.28	0.77	4.43e-01	-0.35	0.76	1.17	1.02e+00	0.50
Testlet2	-0.62	0.30	-2.08	3.75e-02	-1.21	-0.05	4.74	2.99e+00	0.75
Testlet3	-0.06	0.33	-0.17	8.62e-01	-0.72	0.59	0.21	2.87e+00	0.74
Testlet4	0.01	0.38	0.03	9.72e-01	-0.74	0.77	0.04	1.34e+01	0.93
COPIC	0.86	0.07	11.63	3.00e-31	0.72	1.01	101.40	1.75e+28	1.00
Course200	0.01	0.28	0.02	9.85e-01	-0.53	0.57	0.02	2.40e+01	0.96
Course300	-0.13	0.29	-0.44	6.63e-01	-0.69	0.46	0.59	1.35e+00	0.57
Course400	-0.14	0.34	-0.42	6.77e-01	-0.81	0.54	0.56	1.39e+00	0.58

Table 42: Model 3a Coefficients

10.1 Sequential Tests (Type I SS)

Each row in Table 43 tests the significance of unique additional variance explained by the term on that line after controlling for all previously entered terms. Significant results mean adding that term improved the model.

```
m3a %>%
 anova(., test = "Chisq") %>%
 cbind(., convertp(.[,"Pr(>Chi)"])) %>%
 kable(format = "latex", booktabs = TRUE,
       digits = c(0, 2, 0, 2, Inf, 2, 2, 2),
       col.names = c("DF", "Deviance", "Resid. DF", "Resid. Dev", "p-value",
                     "S", "BFB", "PPH1"),
       caption = "Model 3a Sequential Tests (Type I SS): Analysis of Deviance")
```

	DF	Deviance	Resid. DF	Resid. Dev	p-value	S	BFB	PPH1
NULL	NA	NA	1320	1829.91	NA	NA	NA	NA
Testlet	4	62.16	1316	1767.75	1.018257e-12	39.84	1.308385e+10	1.00
COPIC	1	245.07	1315	1522.68	3.089707e-55	181.08	9.486274e+51	1.00
Course	3	0.87	1312	1521.81	8.329399e-01	0.26	2.420000e+00	0.71

Table 43: Model 3a Sequential Tests (Type I SS): Analysis of Deviance

Note that course entered the model last and it was not significant. This indicates that course did not explain a significant amount of the variance in the conditional pass rates after controlling for OPIc scores.

10.2 Simultaneous Tests of Main/Interaction Effects via LRT (Type III SS)

The simultaneous tests in Table 44 are the effects of the indicated terms after controlling for all other terms in the model.

```
m3a %>%
  drop1(., test = "Chisq") %>%
  cbind(., convertp(.[,"Pr(>Chi)"])) %>%
  kable(format = "latex", booktabs = TRUE,
       digits = c(0, 2, 2, 2, Inf, 2, 2, 2),
       col.names = c("DF", "Deviance", "AIC", "LRT", "p-value", "S", "BFB",
                     "PPH1"),
       caption = "Model 3a Simultaneous Tests (Type III SS)")
```

Results of the simultaneous tests are consistent with those of the sequential tests in showing that course was no longer a significant predictor of the conditional pass rates after controlling for OPIc scores. This suggests that we should consider reporting Model 2a instead of Model 3a.

	DF	Deviance	AIC	LRT	p-value	S	BFB	PPH1
<none>	NA	1521.81	1537.81	NA	NA	NA	NA	NA
Testlet	4	1552.77	1560.77	30.96	3.119467e-06	18.29	9.302070e+03	1.00
COPIC	1	1689.58	1703.58	167.77	2.270603e-38	125.05	1.869194e+35	1.00
Course	3	1522.68	1532.68	0.87	8.329399e-01	0.26	2.420000e+00	0.71

Table 44: Model 3a Simultaneous Tests (Type III SS)

10.3 Conditional and Unconditional Pass Rates (Omitted)

We omit calculating these rates because Model 3a is not better than Model 2a.

10.4 Odds-Ratio for COPIC Effect (Omitted)

We omit calculating odds-ratios because Model 3a is not better than Model 2a.

10.5 Assessing Goodness of Fit, Discrimination, and Calibration

10.5.1 Hosmer-Lemeshow Goodness of Fit Test

Figure 41 shows a calibration plot for this model. The plot is based on the bins summarized in Table 45, while the Hosmer-Lemeshow test is shown in Table 46.

```
FCap <- paste("\\label{fig:m3a-calib-plot}",
              "Model 3a Calibration Plot with Bin Sample Sizes and",
              "Hosmer-Lemeshow Test.")
HLT.m3a <- HLfit(m3a, bin.method = "prob.bins", min.prob.interval = 0.1,
                 xlab = "Predicted Probability",
                 ylab = "Observed Probability")
```

```
FN <- "Minimum bin interval width = 0.10."

HLT.m3a$bins.table %>%
  kable(., format = "latex", booktabs = TRUE, digits = 3, row.names = FALSE,
        col.names = HLT.bin.vnames,
        caption = "Calibration Bins Used for Model 3a Hosmer-Lemeshow Test") %>%
  add_header_above(., header = c(" " = 4, "Obs. Prop. 95% CI" = 2)) %>%
  column_spec(column = 1, width = "1.75cm") %>%
  column_spec(column = 3:4, width = "1.75cm") %>%
  column_spec(column = 5:6, width = "1.25cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

```
FN <- "Minimum bin interval width = 0.10."

HLT.m3a %>%
  as_tibble() %>%
  select(chi.sq, DF, p.value, RMSE) %>%
  unique() %>%
  cbind(., convertp(p = .$p.value, digits = 2)) %>%
  kable(., format = "latex", booktabs = TRUE,
        digits = c(2, 0, Inf, 2, 2, 2, 2),
        col.names = HLT.col.vnames,
        caption = "Hosmer-Lemeshow Test for Goodness of Fit of Model 3a") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

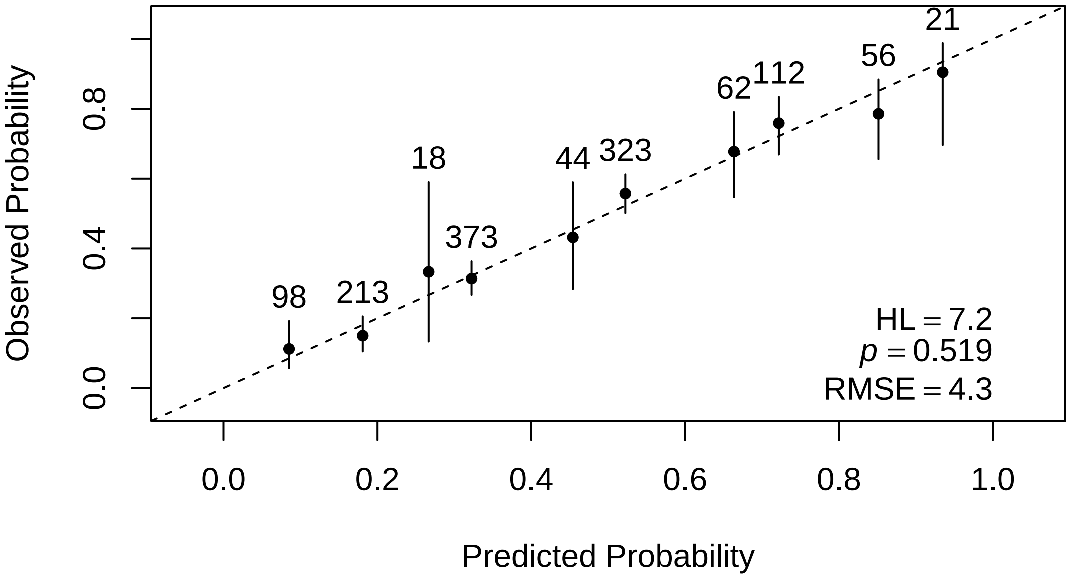

Figure 41: Model 3a Calibration Plot with Bin Sample Sizes and Hosmer-Lemeshow Test.

				Obs. Prop. 95% CI	
Bin Center (Median)	Bin N	Observed Proportion	Predicted Proportion	Lower Limit	Upper Limit
0.085	98	0.112	0.078	0.057	0.192
0.181	213	0.150	0.177	0.105	0.205
0.267	18	0.333	0.271	0.133	0.590
0.322	373	0.314	0.330	0.267	0.363
0.454	44	0.432	0.465	0.283	0.590
0.522	323	0.557	0.531	0.501	0.612
0.664	62	0.677	0.669	0.547	0.791
0.722	112	0.759	0.723	0.669	0.835
0.851	56	0.786	0.847	0.656	0.884
0.935	21	0.905	0.932	0.696	0.988

Note: Minimum bin interval width = 0.10.

Table 45: Calibration Bins Used for Model 3a Hosmer-Lemeshow Test

Chi-square	df	p	RMSE	S	BFB	PPH1
7.16	8	0.5193258	4.28	0.95	1.08	0.52

Note: Minimum bin interval width = 0.10.

Table 46: Hosmer-Lemeshow Test for Goodness of Fit of Model 3a

10.5.2 Classification

Table 47 presents the classification measures for this model.

```
set.seed(3419) # For reproducibility of bootstrap estimates.
roc.m3a <- roc(Pass ~ pred.m3a, data = VTAData, ci = TRUE, direction = "<",
               ci.method = "bootstrap", boot.n = 10000)
```

```
## Setting levels: control = 0, case = 1
```

```
round(lrcm(roc.m3a, seed = 6342), digits = 3) %>%
  kable(format = "latex", booktabs = TRUE,
        caption = "Classification Measures for Model 3a") %>%
  add_header_above(header = c(" " = 2, "Bootstrapped Quantiles" = 3))
```

		Bootstrapped Quantiles		
	threshold	2.5%	50%	97.5%
threshold	0.461	0.401	0.461	0.496
specificity	0.724	0.688	0.725	0.761
sensitivity	0.688	0.647	0.688	0.732
accuracy	0.709	0.684	0.710	0.733
tn	554.000	525.975	555.000	582.000
tp	382.000	359.000	382.000	406.000
fn	173.000	149.000	173.000	196.000
fp	211.000	183.000	210.000	239.025
npv	0.762	0.738	0.763	0.788
ppv	0.644	0.613	0.645	0.676
fdr	0.356	0.324	0.355	0.387
fpr	0.276	0.239	0.275	0.312
tpr	0.688	0.647	0.688	0.732
tnr	0.724	0.688	0.725	0.761
fnr	0.312	0.268	0.312	0.353
1-specificity	0.276	0.239	0.275	0.312
1-sensitivity	0.312	0.268	0.312	0.353
1-accuracy	0.291	0.267	0.290	0.316
1-npv	0.238	0.212	0.237	0.262
1-ppv	0.356	0.324	0.355	0.387
precision	0.644	0.613	0.645	0.676
recall	0.688	0.647	0.688	0.732
youden	1.412	1.363	1.414	1.462
closest.topleft	0.173	0.145	0.173	0.206

Table 47: Classification Measures for Model 3a

10.5.3 Area Under the Curve (AUC)

Figure 42 shows the ROC curve for the model, annotated with the best classification threshold for balancing sensitivity versus specificity and the area under the curve (AUC).

```
FCap <- paste("\\label{fig:m3a-auc-plot}",
              "Model 3a Receiver Operating Characteristic (ROC) Curve.",
              "The dot marks the best classification threshold.",
              "AUC 95% confidence interval obtained via stratified bootstrap",
              "with 10,000 replicates.")
print(roc.m3a)
```

```
##
## Call:
```

```
## roc.formula(formula = Pass ~ pred.m3a, data = VTAData, ci = TRUE,     direction = "<", ci.method = "bootstrap", boot.n = 1000
##
## Data: pred.m3a in 765 controls (Pass 0) < 555 cases (Pass 1).
## Area under the curve: 0.7548
## 95% CI: 0.7287-0.7807 (10000 stratified bootstrap replicates)
```

```
plot.roc(roc.m3a, print.auc = TRUE, print.auc.cex = .8, print.thres = "best",
         print.thres.cex = .8)
```

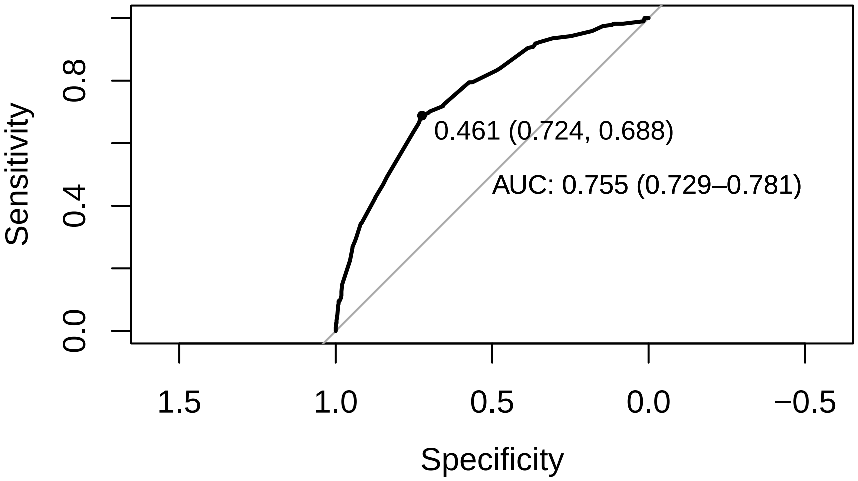

Figure 42: Model 3a Receiver Operating Characteristic (ROC) Curve. The dot marks the best classification threshold. AUC 95% confidence interval obtained via stratified bootstrap with 10,000 replicates.

Model 3a shows acceptable ability to discriminate those who pass from those who fail to pass the level transition testlets, but Table 48 shows that it does not perform better at that than Model 2a.

```
set.seed(1346) # For reproducible bootstrap estimates.
roct.m2a.m3a <- roc.test(roc.m2a, roc.m3a, method = "bootstrap", boot.n = 10000)
glance(roct.m2a.m3a) %>%
  cbind(., convertp(p = .$p.value, digits = 2)) %>%
  select(boot.n, estimate1, estimate2, statistic, p.value, S, BFB, PPH1) %>%
  kable(format = "latex", booktabs = TRUE,
        digits = c(0, 3, 3, 2, Inf, 2, 2, 2),
        col.names = c("Bootstrap N", "Model 2a", "Model 3a", "D", "p", "S",
                      "BFB", "PPH1"),
        caption = paste("Comparing Models 2a and 3a Via Two-Sided, Stratified",
                        "Bootstrap Test for Correlated ROC Curves")) %>%
  add_header_above(header = c(" ", "AUC Estimates" = 2, " " = 5))
```

	AUC Estimates						
Bootstrap N	Model 2a	Model 3a	D	p	S	BFB	PPH1
10000	0.752	0.755	-0.74	0.4563585	1.13	1.03	0.51

Table 48: Comparing Models 2a and 3a Via Two-Sided, Stratified Bootstrap Test for Correlated ROC Curves

10.5.4 R^2 Measures

The values for $R^2_p = 0.2$ and $R^2_{Dev} = 0.17$ are more encouraging than those from Models 1a and 1b, but identical to those from Models 2a and 2b.

10.6 Diagnostics (Omitted)

We omit diagnostics because Model 3a is not better than Model 2a.

10.7 Graphs (Omitted)

We omit graphs because Model 3a is not better than Model 2a.

10.8 Conclusion

What distinguished this model from Model 2a was inclusion of the parallel course effect, but the Type III LRT for that effect is not-significant. We will report Model 2a over this model on the basis of parsimony.

11 Model 3b: Parallel Course + Non-parallel OPIc Effects

We finally test a model with a parallel course effect and a non-parallel OPIc effect. We add the course effect into the model after the interaction so that we can examine whether it adds any value beyond the non-parallel OPIc effect in Model 2b. Table 49 below shows the raw parameter estimates, confidence intervals, s-values, BFBs, and posterior probabilities of H1 corresponding to the p-values.

```
m3b %>%
  tidy(., conf.int = TRUE, conf.level = .95) %>%
  cbind(., convertp(.$p.value, digits = 2)) %>%
  kable(format = "latex", booktabs = TRUE, format.args = list(digits = 3),
        digits = c(2, 2, 2, 2, Inf, 2, 2, 2, 2, 2),
        col.names = c("Term", "Estimate", "SE", "z-value", "p-value", "CI.LL",
                      "CI.UL", "S", "BFB", "PPH1"),
        caption = "Model 3b Coefficients")
```

Term	Estimate	SE	z-value	p-value	CI.LL	CI.UL	S	BFB	PPH1
Testlet1	0.43	0.30	1.43	1.51e-01	-0.17	1.01	2.72	1.29e+00	0.56
Testlet2	-0.46	0.31	-1.50	1.34e-01	-1.08	0.13	2.90	1.37e+00	0.58
Testlet3	0.22	0.35	0.63	5.26e-01	-0.47	0.89	0.93	1.09e+00	0.52
Testlet4	0.52	0.40	1.30	1.94e-01	-0.28	1.31	2.37	1.16e+00	0.54
COPIC	1.01	0.10	10.47	1.13e-25	0.83	1.21	82.87	5.68e+22	1.00
Course200	-0.14	0.29	-0.48	6.29e-01	-0.70	0.44	0.67	1.26e+00	0.56
Course300	-0.29	0.30	-0.95	3.41e-01	-0.87	0.31	1.55	1.00e+00	0.50
Course400	-0.28	0.35	-0.79	4.30e-01	-0.96	0.42	1.22	1.01e+00	0.50
Testlet2:COPIC	-0.19	0.16	-1.19	2.32e-01	-0.49	0.13	2.11	1.08e+00	0.52
Testlet3:COPIC	-0.44	0.20	-2.23	2.60e-02	-0.81	-0.03	5.27	3.88e+00	0.80
Testlet4:COPIC	-0.66	0.23	-2.95	3.17e-03	-1.10	-0.20	8.30	2.01e+01	0.95

Table 49: Model 3b Coefficients

11.1 Sequential Tests (Type I SS)

Each row in Table 50 tests the significance of unique additional variance explained by the term on that line after controlling for all previously entered terms. Significant results mean adding that term improved the model. Note that course actually entered the model last, but is not the last row in the table. That means R's algorithm is testing all the main effects before all the interaction effects, which is not the order we intended. Therefore, we should rely on a simultaneous test (Type III SS) of the course effect instead of the sequential test reported in Table 50.

```
m3b %>%
 anova(., test = "Chisq") %>%
 cbind(., convertp(.[,"Pr(>Chi)"])) %>%
 kable(format = "latex", booktabs = TRUE,
       digits = c(0, 2, 0, 2, Inf, 2, 2, 2),
       col.names = c("DF", "Deviance", "Resid. DF", "Resid. Dev", "p-value",
                     "S", "BFB", "PPH1"),
       caption = "Model 3b Sequential Tests (Type I SS): Analysis of Deviance")
```

	DF	Deviance	Resid. DF	Resid. Dev	p-value	S	BFB	PPH1
NULL	NA	NA	1320	1829.91	NA	NA	NA	NA
Testlet	4	62.16	1316	1767.75	1.018257e-12	39.84	1.308385e+10	1.00
COPIC	1	245.07	1315	1522.68	3.089707e-55	181.08	9.486274e+51	1.00
Course	3	0.87	1312	1521.81	8.329399e-01	0.26	2.420000e+00	0.71
Testlet:COPIC	3	10.79	1309	1511.02	1.290368e-02	6.28	6.550000e+00	0.87

Table 50: Model 3b Sequential Tests (Type I SS): Analysis of Deviance

11.2 Simultaneous Tests of Main/Interaction Effects via LRT (Type III SS)

The simultaneous tests in Table 51 are the effects of the indicated terms after controlling for all other terms in the model. They are only computed for terms that are not part of a higher-order interaction because it makes no sense to test for a main effect when the variable is involved in an interaction. The course effect here effectively compares Models 2b and 3b.

```
m3b %>%
  drop1(., test = "Chisq") %>%
  cbind(., convertp(.[,"Pr(>Chi)"])) %>%
  kable(format = "latex", booktabs = TRUE,
       digits = c(0, 2, 2, 2, Inf, 2, 2, 2),
       col.names = c("DF", "Deviance", "AIC", "LRT", "p-value", "S", "BFB",
                     "PPH1"),
       caption = "Model 3b Simultaneous Tests (Type III SS)")
```

	DF	Deviance	AIC	LRT	p-value	S	BFB	PPH1
<none>	NA	1511.02	1533.02	NA	NA	NA	NA	NA
Course	3	1512.43	1528.43	1.41	0.70232867	0.51	1.48	0.60
Testlet:COPIC	3	1521.81	1537.81	10.79	0.01290368	6.28	6.55	0.87

Table 51: Model 3b Simultaneous Tests (Type III SS)

The fact that the course effect is not significant in Table 51 suggests that we can revert back to Model 2a because course doesn't have an effect after controlling for testlet, COPIC, and their interaction. Model 3b isn't really any better than Model 2b, which itself was no better than Model 2a.

11.3 Conditional and Unconditional Pass Rates (Omitted)

We omit calculating these rates because Model 3b is not better than Model 2b.

11.4 Odds-Ratio for COPIC Effect (Omitted)

We omit calculating odds-ratios because Model 3b is not better than Model 2b.

11.5 Assessing Goodness of Fit, Discrimination, and Calibration

11.5.1 Hosmer-Lemeshow Goodness of Fit Test

Figure 43 shows a calibration plot for this model. The plot is based on the bins summarized in Table 52, while the Hosmer-Lemeshow test is shown in Table 53.

```
FCap <- paste("\\label{fig:m3b-calib-plot}",
              "Model 3b Calibration Plot with Bin Sample Sizes and",
              "Hosmer-Lemeshow Test.")
HLT.m3b <- HLfit(m3b, bin.method = "prob.bins", min.prob.interval = 0.1,
                 xlab = "Predicted Probability",
                 ylab = "Observed Probability")
```

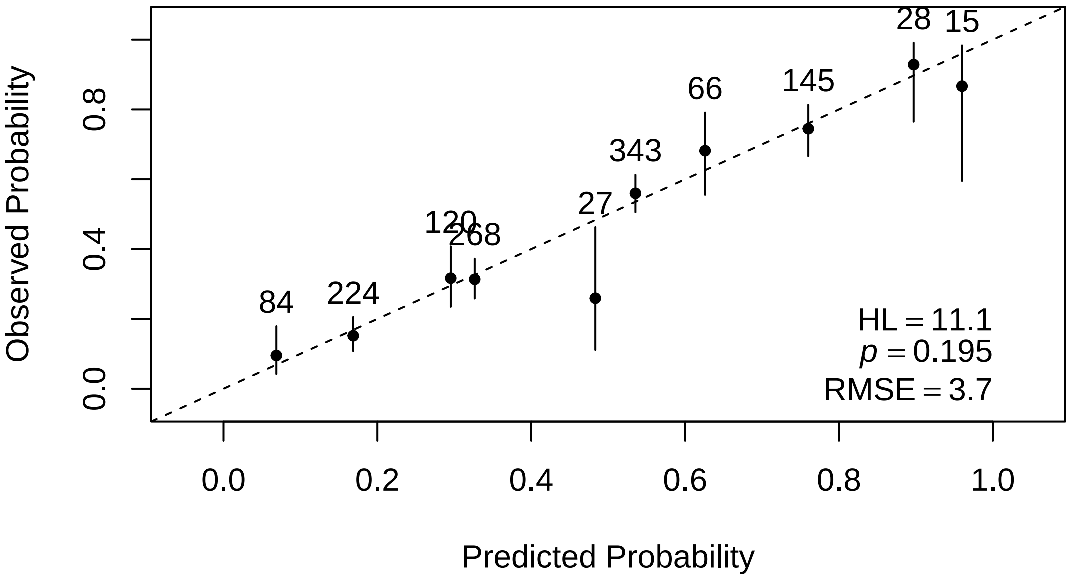

Figure 43: Model 3b Calibration Plot with Bin Sample Sizes and Hosmer-Lemeshow Test.

```
FN <- "Minimum bin interval width = 0.10."

HLT.m3b$bins.table %>%
  kable(., format = "latex", booktabs = TRUE, digits = 3, row.names = FALSE,
        col.names = HLT.bin.vnames,
        caption = "Calibration Bins Used for Model 3b Hosmer-Lemeshow Test") %>%
  add_header_above(., header = c(" " = 4, "Obs. Prop. 95% CI" = 2)) %>%
  column_spec(column = 1, width = "1.75cm") %>%
  column_spec(column = 3:4, width = "1.75cm") %>%
  column_spec(column = 5:6, width = "1.25cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

```
FN <- "Minimum bin interval width = 0.10."

HLT.m3b %>%
  as_tibble() %>%
  select(chi.sq, DF, p.value, RMSE) %>%
  unique() %>%
```

				Obs. Prop. 95% CI	
Bin Center (Median)	Bin N	Observed Proportion	Predicted Proportion	Lower Limit	Upper Limit
0.069	84	0.095	0.063	0.042	0.179
0.169	224	0.152	0.164	0.107	0.206
0.295	120	0.317	0.295	0.235	0.408
0.327	268	0.313	0.333	0.258	0.373
0.483	27	0.259	0.483	0.111	0.463
0.535	343	0.560	0.542	0.505	0.613
0.626	66	0.682	0.632	0.556	0.791
0.760	145	0.745	0.752	0.666	0.814
0.897	28	0.929	0.874	0.765	0.991
0.960	15	0.867	0.939	0.595	0.983

Note: Minimum bin interval width = 0.10.

Table 52: Calibration Bins Used for Model 3b Hosmer-Lemeshow Test

```
  cbind(., convertp(p = .$p.value, digits = 2)) %>%
  kable(., format = "latex", booktabs = TRUE,
        digits = c(2, 0, Inf, 2, 2, 2, 2),
        col.names = HLT.col.vnames,
        caption = "Hosmer-Lemeshow Test for Goodness of Fit of Model 3b") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

Chi-square	df	p	RMSE	S	BFB	PPH1
11.13	8	0.1945699	3.72	2.36	1.16	0.54

Note: Minimum bin interval width = 0.10.

Table 53: Hosmer-Lemeshow Test for Goodness of Fit of Model 3b

11.5.2 Classification

Table 54 presents the classification measures for this model.

```
set.seed(9173) # For reproducibility of bootstrap estimates.
roc.m3b <- roc(Pass ~ pred.m3b, data = VTAData, ci = TRUE, direction = "<",
               ci.method = "bootstrap", boot.n = 10000)
```

```
## Setting levels: control = 0, case = 1
```

```
round(lrcm(roc.m3b, seed = 7146), digits = 3) %>%
  kable(format = "latex", booktabs = TRUE,
        caption = "Classification Measures for Model 3b") %>%
  add_header_above(header = c(" " = 2, "Bootstrapped Quantiles" = 3))
```

		Bootstrapped Quantiles		
	threshold	2.5%	50%	97.5%
threshold	0.485	0.366	0.485	0.519
specificity	0.719	0.676	0.719	0.757
sensitivity	0.695	0.654	0.699	0.742
accuracy	0.709	0.685	0.711	0.733
tn	550.000	517.000	550.000	579.000
tp	386.000	363.000	388.000	412.000
fn	169.000	143.000	167.000	192.000
fp	215.000	186.000	215.000	248.000
npv	0.765	0.742	0.767	0.791
ppv	0.642	0.612	0.643	0.672
fdr	0.358	0.328	0.357	0.388
fpr	0.281	0.243	0.281	0.324
tpr	0.695	0.654	0.699	0.742
tnr	0.719	0.676	0.719	0.757
fnr	0.305	0.258	0.301	0.346
1-specificity	0.281	0.243	0.281	0.324
1-sensitivity	0.305	0.258	0.301	0.346
1-accuracy	0.291	0.267	0.289	0.315
1-npv	0.235	0.209	0.233	0.258
1-ppv	0.358	0.328	0.357	0.388
precision	0.642	0.612	0.643	0.672
recall	0.695	0.654	0.699	0.742
youden	1.414	1.366	1.418	1.464
closest.topleft	0.172	0.144	0.170	0.202

Table 54: Classification Measures for Model 3b

11.5.3 Area Under the Curve (AUC)

Figure 44 shows the ROC curve for the model, annotated with the best classification threshold for balancing sensitivity versus specificity and the area under the curve (AUC).

```
FCap <- paste("\\label{fig:m3b-auc-plot}",
              "Model 3b Receiver Operating Characteristic (ROC) Curve.",
              "The dot marks the best classification threshold.",
              "AUC 95% confidence interval obtained via stratified bootstrap",
              "with 10,000 replicates.")
print(roc.m3b)
```

```
##
## Call:
## roc.formula(formula = Pass ~ pred.m3b, data = VTAData, ci = TRUE,     direction = "<", ci.method = "bootstrap", boot.n = 1000
##
## Data: pred.m3b in 765 controls (Pass 0) < 555 cases (Pass 1).
## Area under the curve: 0.7581
## 95% CI: 0.7321-0.783 (10000 stratified bootstrap replicates)
```

```
plot.roc(roc.m3b, print.auc = TRUE, print.auc.cex = .8, print.thres = "best",
         print.thres.cex = .8)
```

Model 3b shows acceptable ability to discriminate those who pass from those who fail to pass the level transition testlets, but Table 55 shows that it does not perform better at that than Model 2a.

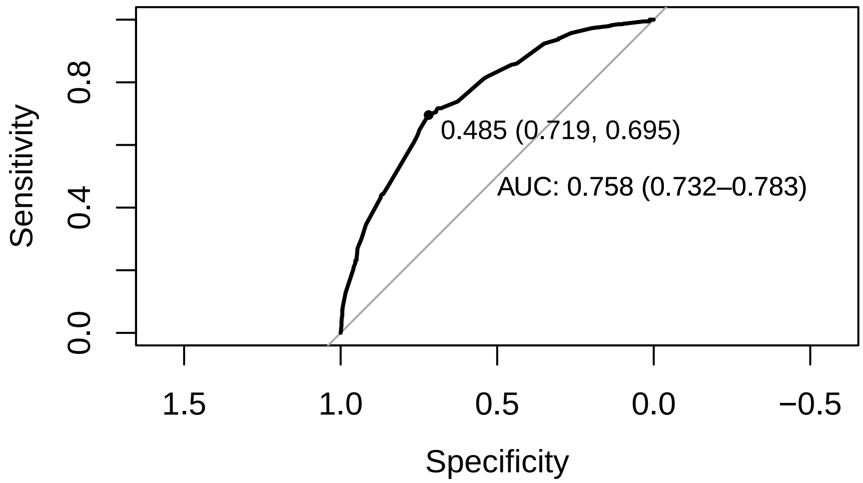

Figure 44: Model 3b Receiver Operating Characteristic (ROC) Curve. The dot marks the best classification threshold. AUC 95% confidence interval obtained via stratified bootstrap with 10,000 replicates.

```
set.seed(1128) # For reproducible bootstrap estimates.
roct.m2a.m3b <- roc.test(roc.m2a, roc.m3b, method = "bootstrap", boot.n = 10000)
glance(roct.m2a.m3b) %>%
  cbind(., convertp(p = .$p.value, digits = 2)) %>%
  select(boot.n, estimate1, estimate2, statistic, p.value, S, BFB, PPH1) %>%
  kable(format = "latex", booktabs = TRUE,
       digits = c(0, 3, 3, 2, Inf, 2, 2, 2),
       col.names = c("Bootstrap N", "Model 2a", "Model 3b", "D", "p", "S",
                     "BFB", "PPH1"),
       caption = paste("Comparing Models 2a and 3b Via Two-Sided, Stratified",
                       "Bootstrap Test for Correlated ROC Curves")) %>%
  add_header_above(header = c(" ", "AUC Estimates" = 2, " " = 5))
```

	AUC Estimates						
Bootstrap N	Model 2a	Model 3b	D	p	S	BFB	PPH1
10000	0.752	0.758	-1.17	0.243002	2.04	1.07	0.52

Table 55: Comparing Models 2a and 3b Via Two-Sided, Stratified Bootstrap Test for Correlated ROC Curves

11.5.4 R^2 Measures

The values for $R^2_p = 0.2$ and $R^2_{Dev} = 0.17$ are more encouraging than those from Models 1a and 1b, but identical to those from Models 2a, 2b, and 3a.

11.6 Diagnostics (Omitted)

We omit diagnostics because Model 3b is not better than Model 2b.

11.7 Graphs (Omitted)

We omit graphs because Model 3b is not better than Models 2a and 2b.

11.8 Conclusion

What distinguished Model 3b from Model 2b was inclusion of the parallel course effect, but the Type III LRT for that effect is not-significant. We will report Model 2a over this model on the basis of parsimony.

12 References

Benjamin, D. J., & Berger, J. O. (2019). Three recommendations for improving the use of p-values. *The American Statistician, 73*(Supplement 1), 186-191. https://doi.org/10.1080/00031305.2018.1543135

Cameron, A. C., & Windmeijer, F. A. G. (1997). An R-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics, 77*(2), 329-342. https://doi.org/10.1016/S0304-4076(96)01818-0

Colquhoun, D. (2019). The false positive risk: A proposal concerning what to do about p-values. *The American Statistician, 73*(Supplement 1), 192-201. https://doi-org/10.1080/00031305.2018.1529622

Fenlon, C., O'Grady, L., Doherty, M. L., & Dunnion, J. (2018). A discussion of calibration techniques for evaluating binary and categorical predictive models. *Preventive Veterinary Medicine, 149*, 107-114. https://doi.org/10.1016/j.prevetmed.2017.11.018

Fox, J. (1997). *Applied regression analysis, linear models, and related methods.* Thousand Oaks, CA: Sage Publications.

Greenland, S. (2019). Valid p-values behave exactly as they should: Some misleading criticisms of p-values and their resolution with s-values. *The American Statistician, 73*(Supplement 1), 106-114. https://doi.org/10.1080/00031305.2018.1529625

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (3rd ed.). Hoboken, NJ: John Wiley & Sons, Inc.

Perkins, N. J., & Schisterman, E. F. (2006). The inconsistency of "optimal" cutpoints obtained using two criteria based on the receiver operating characteristic curve. *American Journal of Epidemiology, 163*(7), 670-675. https://doi-org/10.1093/aje/kwj063

Raykov, T., & Marcoulides, G. A. (2011). *Introduction to psychometric theory.* New York, NY: Routledge.

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., & Müller, M. (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics, 12*, 77. https://doi.org/10.1186/1471-2105-12-77

Steyerberg, E. W., Harrell Jr., F. E., Borsboom, G. J. J. M., Eijkemans, M. J. C., Vergouwe, Y., & Habbema, J. D. F. (2001). Internal validation of predictive models: Efficiency of some procedures for logistic regression analysis. *Journal of Clinical Epidemiology, 54*(8), 774-781. https://doi.org/10.1016/S0895-4356(01)00341-9

Steyerberg, E. W., Vickers, A. J., Cook, N. R., Gerds, T., Gonen, M., Obuchowski, N. A., Pencina, M. J., Kattan, M. W. (2010). Assessing the performance of prediction models: A framework for traditional and novel measures. *Epidemiology, 21*(1), 128-138. http://doi.org/10.1097/EDE.0b013e3181c30fb2

Wasserstein, R. L., Schirm, A. L., & Lazar, N. A. (2019). Moving to a world beyond "p < .05". *The American Statistician, 73*(Supplement 1), 1-19. https://doi.org/10.1080/00031305.2019.1583913

Westfall, P. H. (1997). Multiple testing of general contrasts using logical constraints and correlations. *Journal of the American Statistical Association, 92*(437), 299-306. https://doi.org/10.1080/01621459.1997.10473627

Winke, P., & Zhang, X. (2022, March 14). *Data and codebook for SSLA article: "A closer look at a marginalized test method: Self-assessment as a measure of speaking proficiency."* (Study 164981; Version V1) [Data files and codebooks]. Inter-university Consortium for Political and Social Research. https://doi.org/10.3886/E164981V1

Winke, P., Zhang, X., & Pierce, S. J. (2022). A closer look at a marginalized test method: Self-assessment as a measure of speaking proficiency [Manuscript accepted for publication]. *Studies in Second Language Acquisition.*

Winke, P., Pierce, S. J., & Zhang, X. (2018, October 12-13). *Self-assessment works! Continuation-ratio models for testing course and OPIc score effects on oral proficiency self-assessments.* [Paper presentation] East Coast Organization of Language Testers 2018 conference, Princeton, NJ, United States.

Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer, 3*(1), 32-35. https://doi.org/10.1002/1097-0142(1950)3%3A1%3C32%3A%3AAID-CNCR2820030106%3E3.0.CO%3B2-3

13 Software Information

We use R Markdown to enhance reproducibility. Knitting the source R Markdown script *SAW_Paper_Analyze_Data.Rmd* generates this PDF file.

- We used RStudio to work with R and R markdown files.
- Our software chain looks like this: **Rmd file > RStudio > R > rmarkdown > knitr > md file > pandoc > tex file > TinyTeX > PDF file**.
- We recommend using TinyTeX to compile LaTeX files into PDF files. However, it should be viable to use MiKTeX instead.
- We used pandoc 2.17.1.1 for this document.

This document was generated using the following computational environment and dependencies:

```
# Check and report whether we used TinyTex or other LaTeX software.
which_latex()
```

```
## [1] "is_tinytex = TRUE. We used TinyTeX."
```

```
# Get R and R package version numbers in use.
devtools::session_info()
```

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.1.3 (2022-03-10)
##  os       Windows 10 x64 (build 19042)
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  English_United States.1252
##  ctype    English_United States.1252
##  tz       America/New_York
##  date     2022-03-27
##  pandoc   2.17.1.1 @ C:/Program Files/RStudio/bin/quarto/bin/ (via rmarkdown)
##
## - Packages -------------------------------------------------------------------
##  package      * version   date (UTC) lib source
##  abind          1.4-5     2016-07-21 [1] CRAN (R 4.1.0)
##  admisc         0.26      2022-03-14 [1] CRAN (R 4.1.3)
##  assertthat     0.2.1     2019-03-21 [1] CRAN (R 4.1.0)
##  backports      1.4.1     2021-12-13 [1] CRAN (R 4.1.2)
##  base64enc      0.1-3     2015-07-28 [1] CRAN (R 4.1.0)
##  brio           1.1.3     2021-11-30 [1] CRAN (R 4.1.2)
##  broom        * 0.7.12    2022-01-28 [1] CRAN (R 4.1.2)
##  cachem         1.0.6     2021-08-19 [1] CRAN (R 4.1.1)
##  callr          3.7.0     2021-04-20 [1] CRAN (R 4.1.0)
##  car          * 3.0-12    2021-11-06 [1] CRAN (R 4.1.2)
##  carData      * 3.0-5     2022-01-06 [1] CRAN (R 4.1.2)
##  checkmate      2.0.0     2020-02-06 [1] CRAN (R 4.1.0)
##  cli            3.2.0     2022-02-14 [1] CRAN (R 4.1.2)
##  cluster        2.1.2     2021-04-17 [2] CRAN (R 4.1.3)
##  codetools      0.2-18    2020-11-04 [2] CRAN (R 4.1.3)
##  colorspace     2.0-3     2022-02-21 [1] CRAN (R 4.1.2)
##  crayon         1.5.1     2022-03-26 [1] CRAN (R 4.1.3)
##  data.table     1.14.2    2021-09-27 [1] CRAN (R 4.1.1)
##  DBI            1.1.2     2021-12-20 [1] CRAN (R 4.1.2)
##  desc           1.4.1     2022-03-06 [1] CRAN (R 4.1.2)
##  devtools       2.4.3     2021-11-30 [1] CRAN (R 4.1.2)
##  digest         0.6.29    2021-12-01 [1] CRAN (R 4.1.2)
##  directlabels * 2021.1.13 2021-01-16 [1] CRAN (R 4.1.0)
##  dplyr        * 1.0.8     2022-02-08 [1] CRAN (R 4.1.2)
##  ellipsis       0.3.2     2021-04-29 [1] CRAN (R 4.1.0)
##  evaluate       0.15      2022-02-18 [1] CRAN (R 4.1.2)
##  fansi          1.0.2     2022-01-14 [1] CRAN (R 4.1.3)
##  farver         2.1.0     2021-02-28 [1] CRAN (R 4.1.0)
##  fastmap        1.1.0     2021-01-25 [1] CRAN (R 4.1.0)
##  forcats        0.5.1     2021-01-27 [1] CRAN (R 4.1.0)
##  foreign        0.8-82    2022-01-13 [1] CRAN (R 4.1.2)
##  Formula      * 1.2-4     2020-10-16 [1] CRAN (R 4.1.0)
##  fs             1.5.2     2021-12-08 [1] CRAN (R 4.1.2)
##  generics       0.1.2     2022-01-31 [1] CRAN (R 4.1.2)
##  ggplot2      * 3.3.5     2021-06-25 [1] CRAN (R 4.1.0)
##  git2r          0.30.1    2022-03-16 [1] CRAN (R 4.1.3)
##  glue           1.6.2     2022-02-24 [1] CRAN (R 4.1.2)
##  gridExtra      2.3       2017-09-09 [1] CRAN (R 4.1.0)
##  gtable         0.3.0     2019-03-25 [1] CRAN (R 4.1.0)
##  haven          2.4.3     2021-08-04 [1] CRAN (R 4.1.0)
##  here         * 1.0.1     2020-12-13 [1] CRAN (R 4.1.0)
##  highr          0.9       2021-04-16 [1] CRAN (R 4.1.0)
##  Hmisc        * 4.6-0     2021-10-07 [1] CRAN (R 4.1.1)
##  hms            1.1.1     2021-09-26 [1] CRAN (R 4.1.1)
##  htmlTable      2.4.0     2022-01-04 [1] CRAN (R 4.1.2)
##  htmltools      0.5.2     2021-08-25 [1] CRAN (R 4.1.1)
##  htmlwidgets    1.5.4     2021-09-08 [1] CRAN (R 4.1.1)
##  httr           1.4.2     2020-07-20 [1] CRAN (R 4.1.0)
##  jpeg           0.1-9     2021-07-24 [1] CRAN (R 4.1.0)
##  kableExtra   * 1.3.4     2021-02-20 [1] CRAN (R 4.1.0)
##  knitr        * 1.38      2022-03-25 [1] CRAN (R 4.1.3)
##  labeling       0.4.2     2020-10-20 [1] CRAN (R 4.1.0)
##  lattice      * 0.20-45   2021-09-22 [1] CRAN (R 4.1.1)
##  latticeExtra   0.6-29    2019-12-19 [1] CRAN (R 4.1.0)
##  lifecycle      1.0.1     2021-09-24 [1] CRAN (R 4.1.1)
##  magrittr       2.0.2     2022-01-26 [1] CRAN (R 4.1.2)
##  MASS         * 7.3-56    2022-03-23 [1] CRAN (R 4.1.3)
##  Matrix         1.4-1     2022-03-23 [1] CRAN (R 4.1.3)
##  memoise        2.0.1     2021-11-26 [1] CRAN (R 4.1.2)
##  mgcv           1.8-39    2022-02-24 [1] CRAN (R 4.1.2)
##  modEvA       * 3.0       2021-12-20 [1] CRAN (R 4.1.2)
##  multcomp     * 1.4-18    2022-01-04 [1] CRAN (R 4.1.2)
##  munsell        0.5.0     2018-06-12 [1] CRAN (R 4.1.0)
##  mvtnorm      * 1.1-3     2021-10-08 [1] CRAN (R 4.1.1)
##  nlme           3.1-155   2022-01-13 [1] CRAN (R 4.1.3)
##  nnet           7.3-17    2022-01-13 [1] CRAN (R 4.1.2)
##  pbivnorm       0.6.0     2015-01-23 [1] CRAN (R 4.1.0)
##  piercer      * 0.11.0    2022-03-13 [1] Github (sjpierce/piercer@c92ebeb)
##  pillar         1.7.0     2022-02-01 [1] CRAN (R 4.1.2)
##  pkgbuild       1.3.1     2021-12-20 [1] CRAN (R 4.1.2)
##  pkgconfig      2.0.3     2019-09-22 [1] CRAN (R 4.1.0)
##  pkgload        1.2.4     2021-11-30 [1] CRAN (R 4.1.2)
##  plyr           1.8.6     2020-03-03 [1] CRAN (R 4.1.3)
##  png            0.1-7     2013-12-03 [1] CRAN (R 4.1.0)
##  polycor      * 0.8-1     2022-01-11 [1] CRAN (R 4.1.2)
##  prettyunits    1.1.1     2020-01-24 [1] CRAN (R 4.1.0)
##  pROC         * 1.18.0    2021-09-03 [1] CRAN (R 4.1.1)
##  processx       3.5.2     2021-04-30 [1] CRAN (R 4.1.3)
##  ps             1.6.0     2021-02-28 [1] CRAN (R 4.1.0)
##  purrr          0.3.4     2020-04-17 [1] CRAN (R 4.1.0)
##  quadprog       1.5-8     2019-11-20 [1] CRAN (R 4.1.0)
##  R6             2.5.1     2021-08-19 [1] CRAN (R 4.1.1)
##  RColorBrewer   1.1-2     2014-12-07 [1] CRAN (R 4.1.0)
##  Rcpp           1.0.8.3   2022-03-17 [1] CRAN (R 4.1.3)
##  remotes        2.4.2     2021-11-30 [1] CRAN (R 4.1.2)
##  rlang          1.0.2     2022-03-04 [1] CRAN (R 4.1.2)
##  rmarkdown    * 2.13      2022-03-10 [1] CRAN (R 4.1.3)
##  rpart          4.1.16    2022-01-24 [1] CRAN (R 4.1.2)
##  rprojroot      2.0.2     2020-11-15 [1] CRAN (R 4.1.0)
##  rstudioapi     0.13      2020-11-12 [1] CRAN (R 4.1.0)
##  rvest          1.0.2     2021-10-16 [1] CRAN (R 4.1.1)
##  sandwich       3.0-1     2021-05-18 [1] CRAN (R 4.1.0)
##  SAWpaper     * 0.21.0    2022-03-27 [1] Github (sjpierce/SAWpaper@bf40984)
##  scales         1.1.1     2020-05-11 [1] CRAN (R 4.1.0)
##  sessioninfo    1.2.2     2021-12-06 [1] CRAN (R 4.1.2)
##  stringi        1.7.6     2021-11-29 [1] CRAN (R 4.1.2)
##  stringr      * 1.4.0     2019-02-10 [1] CRAN (R 4.1.0)
##  survival     * 3.3-1     2022-03-03 [1] CRAN (R 4.1.3)
##  svglite        2.1.0     2022-02-03 [1] CRAN (R 4.1.2)
##  systemfonts    1.0.4     2022-02-11 [1] CRAN (R 4.1.2)
##  testthat       3.1.2     2022-01-20 [1] CRAN (R 4.1.2)
##  texreg       * 1.38.5    2022-03-04 [1] CRAN (R 4.1.2)
##  TH.data      * 1.1-0     2021-09-27 [1] CRAN (R 4.1.1)
##  tibble         3.1.6     2021-11-07 [1] CRAN (R 4.1.2)
##  tidyr        * 1.2.0     2022-02-01 [1] CRAN (R 4.1.2)
##  tidyselect     1.1.2     2022-02-21 [1] CRAN (R 4.1.2)
##  tinytex        0.37      2022-02-16 [1] CRAN (R 4.1.2)
##  usethis        2.1.5     2021-12-09 [1] CRAN (R 4.1.2)
##  utf8           1.2.2     2021-07-24 [1] CRAN (R 4.1.0)
##  vctrs          0.3.8     2021-04-29 [1] CRAN (R 4.1.0)
##  viridisLite    0.4.0     2021-04-13 [1] CRAN (R 4.1.0)
##  visreg       * 2.7.0     2020-06-04 [1] CRAN (R 4.1.1)
##  webshot        0.5.2     2019-11-22 [1] CRAN (R 4.1.0)
##  withr          2.5.0     2022-03-03 [1] CRAN (R 4.1.2)
##  xfun           0.30      2022-03-02 [1] CRAN (R 4.1.2)
##  xml2           1.3.3     2021-11-30 [1] CRAN (R 4.1.2)
##  yaml           2.3.5     2022-02-21 [1] CRAN (R 4.1.2)
##  zoo            1.8-9     2021-03-09 [1] CRAN (R 4.1.0)
##
##  [1] C:/Users/pierces1/OneDrive - Michigan State University/CSTATRedirects/Documents/R/win-library/4.1
##  [2] C:/Program Files/R/R-4.1.3/library
##
## ------------------------------------------------------------------------------
```

The current Git commit details and status are:

```
git_report()
```

```
## Local:    master P:/Consulting/FY18/Winke_Paula/18-009/SAWpaper
## Remote:   master @ origin (https://github.com/sjpierce/SAWpaper.git)
## Head:     [bf40984] 2022-03-27: Updated version number, date, and news.
##
```

```
## Untracked files:
##  Untracked:  inst/R_Citations_Published.pdf
##  Untracked:  inst/SAW_Paper_Analyze_Data_Published.pdf
##  Untracked:  inst/SAW_Paper_Analyze_Data_Published_files/
##  Untracked:  inst/SAW_Paper_Import_Explore_Data_Published.pdf
```