Self-Assessment Works! Import Data

Steven J. Pierce & Xiaowan Zhang

Contents

1 Purpose

This file facilitates reproducing the results reported in our manuscript (Winke, Zhang, & Pierce, 2021), which was based on a presentation (Winke, Pierce, & Zhang 2018). It imports data on Spanish language students from SPSS data files (Winke & Zhang, 2022) into R, runs some initial data analyses to explore key variables, and saves out an `.RData` file for use by subsequent scripts.

2 Setup

Set global R chunk options (local chunk options will over-ride global options). The method for creating a size option that controls font size in code chunks and their text output is based on an answer to a question posted on
stackoverflow.com.

```
# 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_Import_Explore_Data.Rmd")
```

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

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

```
library(here)              # for here()
library(devtools)          # for session_info()
```

```
## Loading required package: usethis
```

```
# Set package options.
# options(knitr.kable.NA = '0.00')
library(rmarkdown)         # for render(), pandoc_version().
library(knitr)             # for kable()
library(tidyr)             # for unite()
library(dplyr)             # for filter(), select(), etc.
```

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

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

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

```
library(sjlabelled)        # For set_label(), get_label()
```

```
##
## Attaching package: 'sjlabelled'

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

## The following object is masked from 'package:usethis':
##
##     tidy_labels
```

```
library(haven)             # for read_spss()
```

```
##
## Attaching package: 'haven'

## The following objects are masked from 'package:sjlabelled':
##
##     as_factor, read_sas, read_spss, read_stata, write_sas, zap_labels
```

```
# 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(piercer)           # for git_report(), which_latex()
library(SAWpaper)          # for package version number via session_info()
```

3 Import Data

This section of the script imports the required data from SPSS data files (Winke & Zhang, 2022) into R data frames and saves out an .RData file.

Please see the README.Rmd or README.md files in this *SAWpaper* package for details on obtaining the SPSS data files referenced by the code below. The code chunks show the relative file paths for where you need to put the files in order for the script to work as intended.

3.1 Student-Level Data (for Correlation Analyses)

The file Students_Data-WinkeZhang.sav has the cleaned self-assessment data on a set of Spanish students, with one row per student. We prepared an SPSS file to simplify archiving and attaching meta-data such as variable and value labels, level of measurement, and so on. The chunk below reads that file into a data frame called SData, then converts a couple variables to proper factors.

```
SData <- read_sav(here::here("inst/extdata/Students_Data-WinkeZhang.sav"),
                  user_na = TRUE) %>%
  mutate(Gender = as_factor(Gender),
         Course = as_factor(Course),
         LevelF = as_factor(Level, levels = "values"))
```

Next, we create a smaller data frame called VSData containing only students who had valid OPIC scores (i.e., that were not above or below range or unratable). The latter is the set we used for most analyses.

```
VSData <- SData %>% filter(OPICA >= 1)
```

The SData data frame contains 871 rows and 69 variables, while the VSData data frame contains 807 rows and 69 variables.

3.2 Testlet-Level Data (for Continuation-Rato Modeling)

The file Testlet_Attempts_Data-WinkeZhang.sav contains data about the same set of students (those identified in Students_Data-WinkeZhang.sav) that has been expanded to one row per student per self-assessment testlet attempted. This data structure is conceptually similar to a person-period file for longitudinal analysis. The chunk below reads that file into a data frame called TAData, then converts a couple variables to proper factors.

```
TAData <- read_sav(here::here("inst/extdata/Testlet_Attempts_Data-WinkeZhang.sav"),
                   user_na = TRUE) %>%
  mutate(Gender = as_factor(Gender),
         Course = as_factor(Course),
         Testlet = as_factor(Testlet, levels = "values"),
         MaxLevel = as_factor(MaxLevel, levels = "values"))
```

Next, we create a smaller data frame called VTAData containing only data from students who had valid OPIC scores (i.e., that were not above or below range or unratable). The latter is the set we used for most analyses. We need the VTAData data frame specifically to run continuation-ratio models. Most other analyses are performed based on the VSData data frame.

```
VTAData <- TAData %>% filter(OPICA >= 1)
```

The TAData data frame contains 1513 rows and 10 variables, while the VTAData data frame contains 1320 rows and 10 variables.

3.2.1 Contrast Coding

Tables 1 and 2 below show the default contrast coding associated with key factor variables in TAData so readers know how to interpret model coefficients later. The default for factors should be dummy coding (treatment contrasts) with the first level set as the reference group.

```
# Table caption and footnote text.
TCap <- "Default Contrast Dummy Coding for Testlet"
FN   <- "The reference value is the one always coded as zero."

# Generate a table showing the default contrast coding.
contrasts(TAData$Testlet) %>%
  kable(format = "latex", booktabs = TRUE, row.names = TRUE, caption = TCap) %>%
  kable_styling(latex_options = c("repeat_header")) %>%
  add_header_above(c("Testlet", "Contrast Coding" = 3)) %>%
  column_spec(1:3, width = ".5cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
           threeparttable = TRUE)
```

Testlet	Contrast Coding		
	2	3	4
1	0	0	0
2	1	0	0
3	0	1	0
4	0	0	1

Note: The reference value is the one always coded as zero.

Table 1: Default Contrast Dummy Coding for Testlet

```
# Table caption and footnote text.
TCap <- "Default Contrast Dummy Coding for Course"
FN   <- "The reference value is the one always coded as zero."

# Generate a table showing the default contrast coding.
contrasts(TAData$Course) %>%
  kable(format = "latex", booktabs = TRUE, row.names = TRUE, caption = TCap) %>%
  kable_styling(latex_options = c("repeat_header")) %>%
  add_header_above(c("Course", "Contrast Coding" = 3)) %>%
  column_spec(1:3, width = ".5cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
           threeparttable = TRUE)
```

Course	Contrast Coding		
	200	300	400
100	0	0	0
200	1	0	0
300	0	1	0
400	0	0	1

Note:
The reference value is the one always coded as zero.

Table 2: Default Contrast Dummy Coding for Course

3.2.2 OPIC Scores

We treat OPIc speaking test proficiency scores as a continuous covariate that ranges from Novice-low to Superior. Table 3 shows the coding for the OPICA, OPICV, and COPIC variables in SData. The OPICA variable contains all possible OPIc ratings, including those for individuals who were identified as above or below range or unratable (AR, BR, or UR) in numeric form and has value labels attached that show the meaning of scores. Meanwhile, OPICV is a transformation of OPICA that replaced the AR, BR, and UR values with missing values. To make model coefficients more interpretable, we computed centered OPIc scores from OPICV and stored them in a variable named COPIC. A score of zero in COPIC represents the Intermediate-mid level on the OPIc scale.

```
Total.N <- nrow(SData)
Valid.N <- nrow(VSData)

# Table caption and footnote text.
TCap <- paste0("OPIc Score Variables: Coding and Frequency Distribution (",
              "Total N = ", Total.N, ", Valid N = ", Valid.N, ")")
FN   <- paste("COPIC is a centered form of OPICV (COPIC = OPICV - 5) used to",
              "make model intercept terms more interpretable.",
              "OPICA, numeric ACTFL Oral Proficiency Interview - computer score",
              "OPICV, valid numeric ACTFL Oral Proficiency Interview - computer score.",
              "NA, not available (missing data due to invalid score)")

data.frame(OPIC_Label = get_labels(SData$OPICA),
           OPICA = c(as.numeric(sort(unique(SData$OPICA))), 10),
           OPICV = c(NA, NA, NA, as.numeric(sort(unique(SData$OPICV))), 10),
           COPIC = c(NA, NA, NA, as.numeric(sort(unique(SData$COPIC))), 5),
           Frequency = as.numeric(c(xtabs(~OPICA, addNA = TRUE, data = SData),
                                    0))) %>%
  mutate(Percent = 100*Frequency/Total.N,
         Valid = if_else(OPICA >= 1, true = 100*Frequency/Valid.N, false = 0)) %>%
  kable(format = "latex", booktabs = TRUE, digits = 2,
        col.names = c("Score Label", "OPICA", "OPICV", "COPIC", "Frequency",
                      "Percent", "Valid Percent"),
        caption = TCap) %>%
  kable_styling(latex_options = c("repeat_header")) %>%
  column_spec(1, width = "4cm") %>%
  column_spec(2:4, width = "1.5cm") %>%
  group_rows(group_label = "Invalid scores", bold = FALSE, start_row = 1,
             end_row = 3) %>%
  group_rows(group_label = "Valid scores", bold = FALSE, start_row = 4,
             end_row = 13) %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
           threeparttable = TRUE)
```

Score Label	OPICA	OPICV	COPIC	Frequency	Percent	Valid Percent
Invalid scores						
AR, above range	-2	NA	NA	4	0.46	0.00
BR, below range	-1	NA	NA	59	6.77	0.00
UR, unratable	0	NA	NA	1	0.11	0.00
Valid scores						
NL, novice-low	1	1	-4	10	1.15	1.24
NM, novice-mid	2	2	-3	63	7.23	7.81
NH, novice-high	3	3	-2	134	15.38	16.60
IL, intermediate-low	4	4	-1	230	26.41	28.50
IM, intermediate-mid	5	5	0	249	28.59	30.86
IH, intermediate-high	6	6	1	92	10.56	11.40
AL, advanced-low	7	7	2	21	2.41	2.60
AM, advanced-mid	8	8	3	7	0.80	0.87
AH, advanced-high	9	9	4	1	0.11	0.12
S, superior	10	10	5	0	0.00	0.00

Note: COPIC is a centered form of OPICV (COPIC = OPICV - 5) used to make model intercept terms more interpretable. OPICA, numeric ACTFL Oral Proficiency Interview – computer score OPICV, valid numeric ACTFL Oral Proficiency Interview – computer score. NA, not available (missing data due to invalid score)

Table 3: OPIc Score Variables: Coding and Frequency Distribution (Total N = 871, Valid N = 807)

4 Explore Data

This section of the script performs some initial data analyses that explore variable distributions and provide descriptive summaries we may use in the manuscript.

4.1 Student-Level Data (All Students)

We start by examining the student-level data in `SData`. Table 4 below summarizes the univariate distribution for maximum self-assessment level achieved by the students.

```
# Table caption and footnote text.
TCap <- "Frequency Table for Maximum Self-Assesment Level Achieved (All Students)"
FN   <- paste("Remaining N is the number of students who reached a level above",
              "the one associated with the current row in the table.")

# Frequency table for maximum level achieved (All Students).
TLevelF <- SData %>%
  group_by(LevelF) %>%
  summarise(N = n(),
            Pct = 100*N/Total.N) %>%
  ungroup() %>%
  mutate(CumN = cumsum(N),
         CumPct = cumsum(Pct),
         RemainN = sum(N) - CumN,
         RemainPct = 100 - CumPct)

# Display formatted table.
kable(TLevelF, format = "latex", booktabs = TRUE, digits = 0,
      col.names = c("Level", "N", "%", "Cumulative N", "Cumulative %",
                    "Remaining N", "Remaining %"),
      caption = TCap) %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
           threeparttable = TRUE)
```

Level	N	%	Cumulative N	Cumulative %	Remaining N	Remaining %
1	492	56	492	56	379	44
2	221	25	713	82	158	18
3	53	6	766	88	105	12
4	31	4	797	92	74	8
5	74	8	871	100	0	0

Note: Remaining N is the number of students who reached a level above the one associated with the current row in the table.

Table 4: Frequency Table for Maximum Self-Assesment Level Achieved (All Students)

We now examine the relationships among Level, Course, and OPIC by creating crosstabulations. Table 5 shows the two-way contingency table for Level and Course.

```
# Table caption and footnote text.
TCap <- paste("Crosstab Maximum Self-Assessment Level by Course, With Marginal",
              "Sums (All Students)")
FN   <- paste("Course, type of Spanish course the student was enrolled in;",
              "Level, maximum self-assessment level achieved by the student.")

# Examine the relationship between SA level and course level.
addmargins(xtabs(~LevelF + Course, data = SData)) %>%
  kable(format = "latex", booktabs = TRUE, caption = TCap) %>%
  kable_styling(latex_options = c("repeat_header")) %>%
  add_header_above(c("Level", "Course" = 4, " ")) %>%
  row_spec(row = 6, italic = TRUE) %>%
  column_spec(column = 6, italic = TRUE) %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
           threeparttable = TRUE)
```

Level	Course				
	100	200	300	400	Sum
1	119	158	181	34	*492*
2	10	74	113	24	*221*
3	3	8	35	7	*53*
4	1	6	16	8	*31*
5	1	13	42	18	*74*
Sum	*134*	*259*	*387*	*91*	871

Note: Course, type of Spanish course the student was enrolled in; Level, maximum self-assessment level achieved by the student.

Table 5: Crosstab Maximum Self-Assessment Level by Course, With Marginal Sums (All Students)

Table 6 summarizes the cross-tabulations between OPIc scores and two other variables: maximum self-assessment level achieved and course.

```
# Table caption and footnote text.
TCap <- paste("Crosstabs of Maximum Self-Assessment Level and Course by OPIc",
              "Score, With Marginal  (All Students)")
FN   <- paste("Course, type of Spanish course the student was enrolled in;",
              "Level, maximum self-assessment level achieved by the student;",
              "OPIc, ACTFL Oral Proficiency Interview - computer.",
              "The invalid OPIc scores are: -2 = above range (AR), -1 = below",
              "range (BR), and 0 = unratable (UR).")

# Combine two crosstab outputs into a single table:
# Examine the relationship between OPIc scores and SA level.
# Examine the relationship between OPIc scores and course level.
rbind(addmargins(xtabs(~LevelF + OPICA, addNA = TRUE, data = SData)),
      addmargins(xtabs(~Course + OPICA, addNA = TRUE, data = SData))) %>%
  kable(format = "latex", booktabs = TRUE,
      caption = TCap) %>%
  kable_styling(latex_options = c("repeat_header")) %>%
  add_header_above(header = c(" ", "Invalid" = 3, "Valid" = 9, " ")) %>%
  add_header_above(header = c(" ", "OPIc Score" = 12, " ")) %>%
  group_rows(group_label = "Level", bold = FALSE, start_row = 1,
            end_row = 6) %>%
  group_rows(group_label = "Course", bold = FALSE, start_row = 7,
            end_row = 11) %>%
  row_spec(row = c(6, 11), italic = TRUE) %>%
  column_spec(column = 14, italic = TRUE) %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
          threeparttable = TRUE)
```

	OPIc Score												
	Invalid			Valid									
	-2	-1	0	1	2	3	4	5	6	7	8	9	Sum
Level													
1	2	13	0	10	59	118	152	110	25	2	1	0	*492*
2	1	6	1	0	1	15	62	102	30	2	1	0	*221*
3	0	2	0	0	1	1	10	23	11	5	0	0	*53*
4	0	7	0	0	1	0	4	5	9	4	1	0	*31*
5	1	31	0	0	1	0	2	9	17	8	4	1	*74*
Sum	*4*	*59*	*1*	*10*	*63*	*134*	*230*	*249*	*92*	*21*	*7*	*1*	871
Course													
100	0	3	0	9	50	61	8	3	0	0	0	0	*134*
200	1	7	0	1	13	49	103	77	3	5	0	0	*259*
300	3	37	1	0	0	22	104	145	61	10	3	1	*387*
400	0	12	0	0	0	2	15	24	28	6	4	0	*91*
Sum	*4*	*59*	*1*	*10*	*63*	*134*	*230*	*249*	*92*	*21*	*7*	*1*	871

Note: Course, type of Spanish course the student was enrolled in; Level, maximum self-assessment level achieved by the student; OPIc, ACTFL Oral Proficiency Interview – computer. The invalid OPIc scores are: -2 = above range (AR), -1 = below range (BR), and 0 = unratable (UR).

Table 6: Crosstabs of Maximum Self-Assessment Level and Course by OPIc Score, With Marginal (All Students)

4.2 Student-Level Data (Students with Valid OPIc Scores)

We start by examining the student-level data in `VSData` because those are the data we used in most of the analyses reported in our published manuscript. Table 7 below summarizes the univariate distribution for maximum self-assessment level achieved by the students who had valid OPIC scores.

```
# Table caption and footnote text.
TCap <- paste("Frequency Table for Maximum Self-Assesment Level Achieved",
              "(Students with Valid OPIc Scores)")
FN   <- paste("Only students with valid OPIc scores (OPICA > 0) were included.",
              "Remaining N is the number of students who reached a level above",
              "the one associated with the current row in the table.")

# Frequency table for maximum level achieved (Students with Valid OPIC).
TLevelFV <- VSData %>%
  group_by(LevelF) %>%
  summarise(N = n(),
            Pct = 100*N/Valid.N) %>%
  ungroup() %>%
  mutate(CumN = cumsum(N),
         CumPct = cumsum(Pct),
         RemainN = sum(N) - CumN,
         RemainPct = 100 - CumPct)

# Extract N and percent of students with self-assessed level > 1 (i.e., >= 2).
N_SA_GE_2   <- TLevelFV$RemainN[1]
Pct_SA_GE_2 <- TLevelFV$RemainPct[1]

# Extract N and percent of students with self-assessed level > 2 (i.e., >= 3).
N_SA_GE_3   <- TLevelFV$RemainN[2]
Pct_SA_GE_3 <- TLevelFV$RemainPct[2]


# Display formatted table.
kable(TLevelFV, format = "latex", booktabs = TRUE, digits = 0,
      col.names = c("Level", "N", "%", "Cumulative N", "Cumulative %",
                    "Remaining N", "Remaining %"),
      caption = TCap) %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
           threeparttable = TRUE)
```

Level	N	%	Cumulative N	Cumulative %	Remaining N	Remaining %
1	477	59	477	59	330	41
2	213	26	690	86	117	14
3	51	6	741	92	66	8
4	24	3	765	95	42	5
5	42	5	807	100	0	0

Note: Only students with valid OPIc scores (OPICA > 0) were included. Remaining N is the number of students who reached a level above the one associated with the current row in the table.

Table 7: Frequency Table for Maximum Self-Assesment Level Achieved (Students with Valid OPIc Scores)

Only $N = 330$ (41%) of the students reached level 2 or higher on the self-assessment. Only $N = 117$ (14%) students reached level 3 or higher on the self-assessment.

We now examine the relationships among Level, Course, and OPIC by creating crosstabulations. Table 8 shows the two-way contingency table for Level and Course.

```
# Table caption and footnote text.
TCap <- paste("Crosstab Maximum Self-Assessment Level by Course, With Marginal",
              "Sums (Students with Valid OPIc Scores)")
FN   <- paste("Only students with valid OPIc scores (OPICA > 0) were included.",
              "Course, type of Spanish course the student was enrolled in;",
              "Level, maximum self-assessment level achieved by the student.")

# Examine the relationship between SA level and course level.
addmargins(xtabs(~LevelF + Course, data = VSData)) %>%
  kable(format = "latex", booktabs = TRUE, caption = TCap) %>%
  kable_styling(latex_options = c("repeat_header")) %>%
  add_header_above(c("Level", "Course" = 4, " ")) %>%
  row_spec(row = 6, italic = TRUE) %>%
  column_spec(column = 6, italic = TRUE) %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
           threeparttable = TRUE)
```

Level	Course				
	100	200	300	400	Sum
1	116	156	173	32	*477*
2	10	72	108	23	*213*
3	3	8	33	7	*51*
4	1	6	10	7	*24*
5	1	9	22	10	*42*
Sum	*131*	*251*	*346*	*79*	807

Note: Only students with valid OPIc scores (OPICA > 0) were included. Course, type of Spanish course the student was enrolled in; Level, maximum self-assessment level achieved by the student.

Table 8: Crosstab Maximum Self-Assessment Level by Course, With Marginal Sums (Students with Valid OPIc Scores)

Table 9 summarizes the cross-tabulations between OPIc scores and two other variables: maximum self-assessment level achieved and course.

```
# Table caption and footnote text.
TCap <- paste("Crosstabs of Maximum Self-Assessment Level and Course by OPIc",
              "Score, With Marginal Sums (Students with Valid OPIc Scores)")
FN   <- paste("Only students with valid OPIc scores (OPICA > 0) were included.",
              "Course, type of Spanish course the student was enrolled in;",
              "Level, maximum self-assessment level achieved by the student;",
              "OPIc, ACTFL Oral Proficiency Interview - computer.")

# Combine two crosstab outputs into a single table:
# Examine the relationship between OPIc scores and SA level.
# Examine the relationship between OPIc scores and course level.
rbind(addmargins(xtabs(~LevelF + OPICV, data = VSData)),
      addmargins(xtabs(~Course + OPICV, data = VSData))) %>%
  kable(format = "latex", booktabs = TRUE,
      caption = TCap) %>%
  kable_styling(latex_options = c("repeat_header")) %>%
  add_header_above(header = c(" ", "OPIc Score" = 9, " ")) %>%
  group_rows(group_label = "Level", bold = FALSE, start_row = 1,
             end_row = 6) %>%
  group_rows(group_label = "Course", bold = FALSE, start_row = 7,
             end_row = 11) %>%
  row_spec(row = c(6, 11), italic = TRUE) %>%
  column_spec(column = 11, italic = TRUE) %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
           threeparttable = TRUE)
```

	OPIc Score									
	1	2	3	4	5	6	7	8	9	Sum
Level										
1	10	59	118	152	110	25	2	1	0	*477*
2	0	1	15	62	102	30	2	1	0	*213*
3	0	1	1	10	23	11	5	0	0	*51*
4	0	1	0	4	5	9	4	1	0	*24*
5	0	1	0	2	9	17	8	4	1	*42*
Sum	*10*	*63*	*134*	*230*	*249*	*92*	*21*	*7*	*1*	807
Course										
100	9	50	61	8	3	0	0	0	0	*131*
200	1	13	49	103	77	3	5	0	0	*251*
300	0	0	22	104	145	61	10	3	1	*346*
400	0	0	2	15	24	28	6	4	0	*79*
Sum	*10*	*63*	*134*	*230*	*249*	*92*	*21*	*7*	*1*	807

Note: Only students with valid OPIc scores (OPICA > 0) were included. Course, type of Spanish course the student was enrolled in; Level, maximum self-assessment level achieved by the student; OPIc, ACTFL Oral Proficiency Interview – computer.

Table 9: Crosstabs of Maximum Self-Assessment Level and Course by OPIc Score, With Marginal Sums (Students with Valid OPIc Scores)

4.3 Testlet-Level Data

Now we examine some additional crosstabulations based on the testlet-level data frame `VTAData`. Note that the Testlet variable in `VTAData` records the level transition testlet used to determine whether a student could move on from one self-assessment level to the next. Thus, Testlet $= 1$ pertains to the attempt to transition from level 1 to level 2, Testlet $= 2$ pertains to the transition from level 2 to level 3, and so on.

Table 10 shows the contingency table between Testlet and Pass, which we can use to get crude estimates of conditional and unconditional pass rates (albeit without confidence intervals). The conditional pass rate uses a denominator comprised of only students who took the given testlet, whereas the the unconditional pass rate uses the total number of students in the sample as the denominator.

```
# Table caption and footnote text.
TCap <- paste("Crosstab Self-Assessment Level Transition Testlet by Pass",
              "(Students with Valid OPIc Scores)")
FN   <- paste("Only students with valid OPIc scores (OPICA > 0) were included.",
              "CPR, conditional pass rate;",
              "Pass, whether student passed the testlet (0 = No, 1 = Yes);",
              "Testlet, level transition testlet attempted by student;",
              "UPR, unconditional pass rate.")

# Crosstab to get a look at how often people pass each level transition.
addmargins(xtabs(~Testlet + Pass, data = VTAData)) %>%
  as_tibble(.) %>%
  pivot_wider(names_from = Pass, values_from = n) %>%
  mutate(CPR = case_when(Testlet != "Sum" ~ `1`/Sum),
         UPR = case_when(Testlet != "Sum" ~ `1`/n_distinct(VTAData$StudentID))) %>%
  kable(format = "latex", booktabs = TRUE, digits = 3, caption = TCap) %>%
  kable_styling(latex_options = c("repeat_header")) %>%
  add_header_above(header = c(" ", "Pass" = 2, " ", " ", " ")) %>%
  add_header_above(header = c(" ", "Counts" = 3, "Pass Rates" = 2)) %>%
  row_spec(row = 5, italic = TRUE) %>%
  column_spec(column = 4, italic = TRUE) %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
           threeparttable = TRUE)
```

Table 11 shows the contingency table between Testlet and Course.

	Counts			Pass Rates	
	Pass				
Testlet	0	1	Sum	CPR	UPR
1	477	330	*807*	0.409	0.409
2	213	117	*330*	0.355	0.145
3	51	66	*117*	0.564	0.082
4	24	42	*66*	0.636	0.052
Sum	*765*	*555*	1320	*NA*	*NA*

Note: Only students with valid OPIc scores (OPICA > 0) were included. CPR, conditional pass rate; Pass, whether student passed the testlet (0 = No, 1 = Yes); Testlet, level transition testlet attempted by student; UPR, unconditional pass rate.

Table 10: Crosstab Self-Assessment Level Transition Testlet by Pass (Students with Valid OPIc Scores)

```r
# Table caption and footnote text.
TCap <- paste("Crosstab Self-Assessment Level Transition Testlet by Course",
              "(Students with Valid OPIc Scores)")
FN   <- paste("Only students with valid OPIc scores (OPICA > 0) were included.",
              "Course, type of Spanish course the student was enrolled in;",
              "Testlet, level transition testlet attempted by student.")

# Examine the relationship between SA level transition and course level.
addmargins(xtabs(~Testlet + Course, data = VTAData)) %>%
  kable(format = "latex", booktabs = TRUE, caption = TCap) %>%
  kable_styling(latex_options = c("repeat_header")) %>%
  add_header_above(header = c("Testlet ", "Course" = 4, " ")) %>%
  row_spec(row = 5, italic = TRUE) %>%
  column_spec(column = 6, italic = TRUE) %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
           threeparttable = TRUE)
```

Testlet	Course				
	100	200	300	400	Sum
1	131	251	346	79	*807*
2	15	95	173	47	*330*
3	5	23	65	24	*117*
4	2	15	32	17	*66*
Sum	*153*	*384*	*616*	*167*	1320

Note:
Only students with valid OPIc scores (OPICA > 0) were included. Course, type of Spanish course the student was enrolled in; Testlet, level transition testlet attempted by student.

Table 11: Crosstab Self-Assessment Level Transition Testlet by Course (Students with Valid OPIc Scores)

Table 12 summarizes the cross-tabulations between Testlet and centered OPIc scores.

```r
# Table caption and footnote text.
TCap <- paste("Crosstab Self-Assessment Level Transition Testlet by Centered",
              "OPIc Score (Students with Valid OPIc Scores)")
```

```
FN   <- paste("Only students with valid OPIc scores (OPICA > 0) were included.",
              "COPIC is a centered form of OPICN (COPIC = OPICN - 5) used to",
              "make model intercept terms more interpretable.",
              "OPICN, numeric ACTFL Oral Proficiency Interview - computer score;",
              "Testlet, level transition testlet attempted by student.")

# Examine the relationship between centered OPIc scores and SA level transition.
addmargins(xtabs(~Testlet + COPIC, data = VTAData)) %>%
  kable(format = "latex", booktabs = TRUE, caption = TCap) %>%
  kable_styling(latex_options = c("repeat_header")) %>%
  add_header_above(header = c("Testlet", "COPIC" = 9,
                              " ")) %>%
  row_spec(row = 5, italic = TRUE) %>%
  column_spec(column = 11, italic = TRUE) %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
           threeparttable = TRUE)
```

Testlet	COPIC									
	-4	-3	-2	-1	0	1	2	3	4	Sum
1	10	63	134	230	249	92	21	7	1	*807*
2	0	4	16	78	139	67	19	6	1	*330*
3	0	3	1	16	37	37	17	5	1	*117*
4	0	2	0	6	14	26	12	5	1	*66*
Sum	*10*	*72*	*151*	*330*	*439*	*222*	*69*	*23*	*4*	1320

Note: Only students with valid OPIc scores (OPICA > 0) were included. COPIC is a centered form of OPICN (COPIC = OPICN - 5) used to make model intercept terms more interpretable. OPICN, numeric ACTFL Oral Proficiency Interview – computer score; Testlet, level transition testlet attempted by student.

Table 12: Crosstab Self-Assessment Level Transition Testlet by Centered OPIc Score (Students with Valid OPIc Scores)

5 Save Data

The chunks below save the `SData`, `VSData`, `TAData`, and `VTAData` data frames to an `.RData` file in the local copy of the repository for use by subsequent scripts.

```
save(SData, VSData, TAData, VTAData,
     file = here::here("data/SAW_Paper_Data.RData"))
```

6 References

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.

7 Software Information

We use R Markdown to enhance reproducibility. Knitting the source R Markdown script *SAW_Paper_Import_Explore_Data.* 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
##  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)
##  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)
##  cli           3.2.0   2022-02-14 [1] CRAN (R 4.1.2)
##  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)
##  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)
##  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)
##  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)
##  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)
##  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)
##  hms           1.1.1   2021-09-26 [1] CRAN (R 4.1.1)
##  htmltools     0.5.2   2021-08-25 [1] CRAN (R 4.1.1)
##  httr          1.4.2   2020-07-20 [1] CRAN (R 4.1.0)
##  insight       0.16.0  2022-02-17 [1] CRAN (R 4.1.2)
##  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)
##  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)
##  memoise       2.0.1   2021-11-26 [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)
##  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)
##  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)
##  R6            2.5.1   2021-08-19 [1] CRAN (R 4.1.1)
##  Rcpp          1.0.8.3 2022-03-17 [1] CRAN (R 4.1.3)
##  readr         2.1.2   2022-01-30 [1] CRAN (R 4.1.2)
##  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)
##  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)
##  SAWpaper    * 1.0.0   2022-03-27 [1] Github (sjpierce/SAWpaper@a44d7ef)
##  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)
##  sjlabelled  * 1.1.8   2021-05-11 [1] CRAN (R 4.1.0)
##  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)
##  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)
##  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)
##  tzdb          0.2.0   2021-10-27 [1] CRAN (R 4.1.1)
##  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)
##  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)
##
##  [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:     [a44d7ef] 2022-03-27: Updated version number.
##
```

```
## Unstaged changes:
##  Modified:   NEWS.md
```