Juntos Project Initial Report

Baseline Assessment Cleaning and Recommendations

Alejandra Garcia Isaza

March 2021

Center for Equity Promotion
Centro para la Promoción de la Equidad

Juntos Project Description

Study and intervention details

The *Juntos* Project was a three-year study led by the University of Oregon's Center for Equity Promotion CEQP.. The project developed a culturally specific family–school partnership intervention, *Conexiones: Families and Schools United for Equity* (hereafter referred to as *Conexiones*), designed to enhance Latino parents' and educators' capacities to effectively support Latino student success.

The *Conexiones* curricula was built on Latino cultural assets, addressed common challenges confronting immigrant students and families in terms of school success, and utilized effective strategies for increasing educators' awareness of Latino cultures and the barriers that exist for Latino immigrant students and families in schools. It also focused on building effective family-school communication and partnerships with the aim of improving Latino students' academic success.

The six participating schools belonged to three different school districts in the state of Oregon and were randomly assigned to either a control group or a intervention group that received the *Conexiones* intervention program. Study participants completed assessments at three different time points (baseline, immediately post-intervention, and 12-month post-intervention). The complete dataset in the project is made of three waves of data with separate assessments for each participant type (parents, students, and educators).

Report details

This report will be focusing only on the baseline assessment and is intended to describe the data cleaning process with the aim of helping CEQP staff replicate these procedures in subsequent waves of data and future projects. The report will also include a brief description of the sociodemographic characteristics of the study participants, the scale

creation process, the average scores of participants' responses in regards to major study constructs, and recommendations for more advanced statistical analyses that link the different types of participants in the study.


Center for
Equity Promotion
Centro para la Promoción de la Equidad

Data Cleaning procedures

The following section describes the data cleaning procedures I performed in each of the participant's type datasets. I performed data cleaning using the R and R Studio softwares, but had in mind that end users of the cleaned datasets will likely be SPSS users.

Educator's dataset

The raw dataset had 43 observations and 202 variables of which 17 were metadata variables created by Qualtrics, the software used to develop the assessment surveys. Of the 43 observations, one case, participant with `id` 153 had incomplete data.

In the following code, I removed all but one of the metadata variables, `response_id`, that is an unique identifier assigned by Qualtrics that resulted handy in dealing with duplicated ids. Other data cleaning procedures are described in the comments marked with a # sign.

```
elt_w1_clean <- w1_raw_elt %>%
 janitor::clean_names() %>% # function that formats variables' names
 select(-1:-8, -10:-17, -202) %>% # selecting out columns with metadata
 rename(c("id" = "pj")) %>% # renaming id variable
 arrange(id) # ordering participants ids in descending order
```

0.0.1 Dealing with duplicated ids

When evaluating if the dataset had duplicated ids, I found that `id` 257 was duplicated and there was no `id` 254. In the table below, I am just showing a few variables and participants from `school` 2.

response_id	id	school	q1	q2	q3
R_1NsKbbg0xSNm9DI	251	2	3	3	2
R_Xvok02kOfilkkV3	252	2	3	3	4
R_294kWxIg2imaph1	253	2	4	3	3
R_3NEywI5hBzdP9Kt	255	2	3	2	3
R_3McjQ3QdB3iSnbT	256	2	4	3	4
R_6EELe7Uuwi9W7zX	257	2	2	2	3
R_3IRUos8weYHpWB1	257	2	4	3	3

After checking with CEQP's research assistant, I corroborated that one of the duplicated cases of `id` 257 in fact was `id` 254. I fixed this mistake with the code below using the `response_id` variable and the mutate and case_when functions.

```
elt_w1_clean <- elt_w1_clean %>%
  mutate(id = case_when(response_id == "R_6EELe7Uuwi9W7zX" ~ "254",
                        TRUE ~ as.character(id))) %>%
  arrange(id)
```

0.0.2 Dealing with survey coding errors

The id protocol followed in CEQP projects is very straightforward. They usually use three digits for each individual participant id and use the first of these three digits to indicate the school id. In this system, ids in the 100's would belong to school 1, ids in the 200's to school 2, and so on.

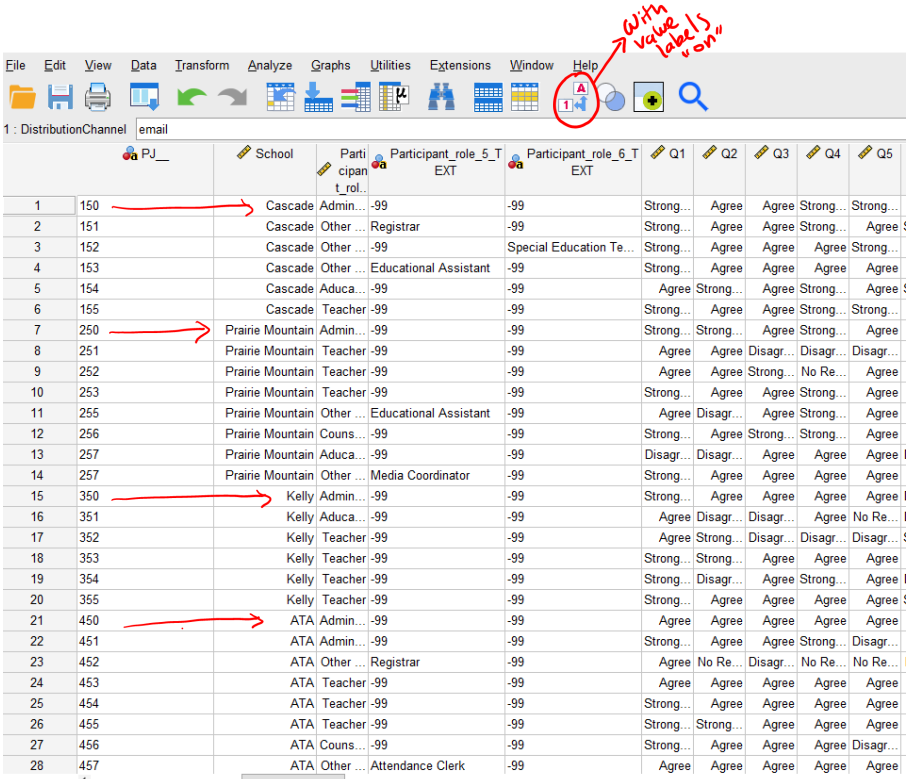
By visual inspection I dentified that the first digit of the indvidual ids in the `id` variable did not correspond to the ids in the school id variable `school` for schools 3, 4, 5, and 6. In the table below, I selected four variables and only the first row of data of each of the six schools to ilustrate this point.

id	school	q1	q2	q3
150	1	4	3	3
250	2	4	4	3
350	4	4	3	3
450	3	3	3	3
550	6	3	3	3
650	5	4	3	3

As can be seen in the table above, ids in the 300's are coded to belong to `school` 4 and ids in the 400's are coded to belong to `school` 3. I am calling this flip-flopped school ids. Schools 5 and 6 were also flip-flopped.

At first, I thought that this could be due to an error in the data exporting process and it seemed like an easy enough fix to make. I thought I just needed to recode the names of the levels of the `school` variable. Later I found that this fix did not solve the issue. It took me a couple of months to identify that the error was coded in the Qualtrics survey.

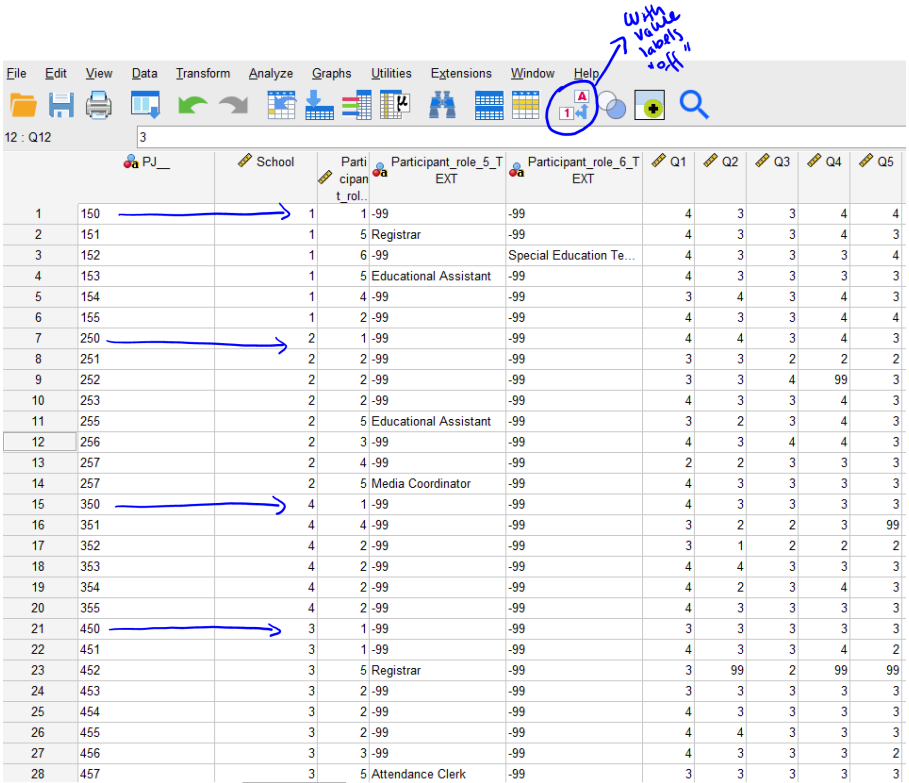
The images below are screenshots of the same raw data SPSS file downloaded directly from Qualtrics. In figure 1, it can be seen that when the *value labels* button is "on" (i.e. showing value labels and not values), it appears as if there was no flip-flop because the names of the schools coincided with the numbers that were assigned to them. Indeed, "Kelly" was school 3 and its participants were identified with ids in the 300's and "ATA" was school 4 and its participants were identified with ids in the 400's, and so on.



	PJ__	School	Participant_role_5_TEXT	Participant_role_6_TEXT	Q1	Q2	Q3	Q4	Q5
1	150	Cascade	Admin...	-99	-99	Strong...	Agree	Agree	Strong...	Strong...
2	151	Cascade	Other ...	Registrar	-99	Strong...	Agree	Agree	Strong...	Agree
3	152	Cascade	Other ...	-99	Special Education Te...	Strong...	Agree	Agree	Agree	Strong...
4	153	Cascade	Other ...	Educational Assistant	-99	Strong...	Agree	Agree	Agree	Agree
5	154	Cascade	Aduca...	-99	-99	Agree	Strong...	Agree	Strong...	Agree
6	155	Cascade	Teacher	-99	-99	Strong...	Agree	Agree	Strong...	Strong...
7	250	Prairie Mountain	Admin...	-99	-99	Strong...	Strong...	Agree	Strong...	Agree
8	251	Prairie Mountain	Teacher	-99	-99	Agree	Agree	Disagr...	Disagr...	Disagr...
9	252	Prairie Mountain	Teacher	-99	-99	Agree	Agree	Strong...	No Re...	Agree
10	253	Prairie Mountain	Teacher	-99	-99	Strong...	Agree	Agree	Strong...	Agree
11	255	Prairie Mountain	Other ...	Educational Assistant	-99	Agree	Disagr...	Agree	Strong...	Agree
12	256	Prairie Mountain	Couns...	-99	-99	Strong...	Agree	Strong...	Strong...	Agree
13	257	Prairie Mountain	Aduca...	-99	-99	Disagr...	Disagr...	Agree	Agree	Agree
14	257	Prairie Mountain	Other ...	Media Coordinator	-99	Strong...	Agree	Agree	Agree	Agree
15	350	Kelly	Admin...	-99	-99	Strong...	Agree	Agree	Agree	Agree
16	351	Kelly	Aduca...	-99	-99	Agree	Disagr...	Disagr...	Agree	No Re...
17	352	Kelly	Teacher	-99	-99	Agree	Strong...	Disagr...	Disagr...	Disagr...
18	353	Kelly	Teacher	-99	-99	Strong...	Strong...	Agree	Agree	Agree
19	354	Kelly	Teacher	-99	-99	Strong...	Disagr...	Agree	Strong...	Agree
20	355	Kelly	Teacher	-99	-99	Strong...	Agree	Agree	Agree	Agree
21	450	ATA	Admin...	-99	-99	Agree	Agree	Agree	Agree	Agree
22	451	ATA	Admin...	-99	-99	Strong...	Agree	Agree	Strong...	Disagr...
23	452	ATA	Other ...	Registrar	-99	Agree	No Re...	Disagr...	No Re...	No Re...
24	453	ATA	Teacher	-99	-99	Agree	Agree	Agree	Agree	Agree
25	454	ATA	Teacher	-99	-99	Strong...	Agree	Agree	Agree	Agree
26	455	ATA	Teacher	-99	-99	Strong...	Strong...	Agree	Agree	Agree
27	456	ATA	Couns...	-99	-99	Strong...	Agree	Agree	Agree	Disagr...
28	457	ATA	Other ...	Attendance Clerk	-99	Agree	Agree	Agree	Agree	Agree

Figure 1: Value labels button on.

This changed when the *value labels* button was "off". In the image below, the flip-flopped school ids is evident again:



	PJ__	School	Participant_role_5_TEXT	Participant_role_6_TEXT	Q1	Q2	Q3	Q4	Q5
1	150	1	1	-99	-99	4	3	3	4	4
2	151	1	5	Registrar	-99	4	3	3	4	3
3	152	1	6	-99	Special Education Te...	4	3	3	3	4
4	153	1	5	Educational Assistant	-99	4	3	3	3	3
5	154	1	4	-99	-99	3	4	3	4	3
6	155	1	2	-99	-99	4	3	3	4	4
7	250	2	1	-99	-99	4	4	3	4	3
8	251	2	2	-99	-99	3	3	2	2	2
9	252	2	2	-99	-99	3	3	4	99	3
10	253	2	2	-99	-99	4	3	3	4	3
11	255	2	5	Educational Assistant	-99	3	2	3	4	3
12	256	2	3	-99	-99	4	3	4	4	3
13	257	2	4	-99	-99	2	2	3	3	3
14	257	2	5	Media Coordinator	-99	4	3	3	3	3
15	350	4	1	-99	-99	4	3	3	3	3
16	351	4	4	-99	-99	3	2	2	3	99
17	352	4	2	-99	-99	3	1	2	2	2
18	353	4	2	-99	-99	4	4	3	3	3
19	354	4	2	-99	-99	4	2	3	4	3
20	355	4	2	-99	-99	4	3	3	3	3
21	450	3	1	-99	-99	3	3	3	3	3
22	451	3	1	-99	-99	4	3	3	4	2
23	452	3	5	Registrar	-99	3	99	2	99	99
24	453	3	2	-99	-99	3	3	3	3	3
25	454	3	2	-99	-99	4	3	3	3	3
26	455	3	2	-99	-99	4	4	3	3	3
27	456	3	3	-99	-99	4	3	3	3	2
28	457	3	5	Attendance Clerk	-99	3	3	3	3	3

Figure 2: Value labels button off

This survey coding error meant that the `school` variable's value labels properly corresponded to the participants' ids, but the variable's values did not. Instead of recoding the values, I decided to create a new variable called `school_id` and delete the flawed original variable `school`.

In the code below, I used the first digit of the individual participant id variable `id` as the reference for the new `school_id` variable, following CEQP'S id protocol. I also created a new variable called `condition` to indicate which schools were randomly assigned to the control group (coded as 1) or to the intervention group (coded as 2).

I coded schools identified with a `school_id` odd number (1, 3, and 5) as the control schools and the schools identified with an even number (2, 4, and 6) as the intervention schools, as directed by CEQP's research assistant. Finally, I also created a `wave` variable to indicate the wave of the data.

```
elt_w1_clean_2 <- elt_w1_clean %>%
 mutate(school_id = str_sub(id, 1, 1), # new school id variable
      condition = case_when(
        school_id == "1" | school_id == "3" | school_id == "5" ~ "1",
        school_id == "2" | school_id == "4" | school_id == "6" ~ "2")) %>% # new
   condition variable
 select(school_id, condition, everything()) %>%
 add_column(wave = 1, .before = 10) %>% # new wave variable
 select(- school) # deleting school variable
```

The `condition` and `school_id` variables I created in the previous code were string variables. In the code below I made them numeric so they can be used in quantitative analyses. I also added value labels with the `set_vall` function so that SPSS users can use the *value labels* button.

In the code below I also fixed a response option coding error I identified in the variable `q68`. Throughout most of the survey, response options were coded as "Strongly Disagree" = 1, "Disagree" = 2, "Agree" = 3, "Strongly Agree" = 4, "No response" = 99; however, in variable `q68` the response option "No response" was coded as 5.

I fixed this using the ifelse function, specifying that if this variable had a response of 5, it should be changed to 99. Finally, I set the variable and value labels with the set_varl and set_vall functions, respectively, because sometimes procedures performed with R strips out these labels.

```
elt_w1_clean_3 <- elt_w1_clean_2 %>%
  mutate(condition = as.numeric(condition),
         condition = set_vall(condition, c("control" = 1, "intervention" = 2)),
         school_id = as.numeric(school_id),
         school_id = set_vall(school_id, c("cascade" = 1, "prairie_mountain" = 2, "kelly"
    = 3, "ata" = 4, "briggs" = 5, "agnes_stewart" = 6)),
         q68 = ifelse(q68 == 5, 99, q68),
         q68 = set_varl(q68, "When I communicate with Latino families, I keep in mind
    that many Latino parents may not understand how to navigate the educational
    system in this
country."),
         q68 = set_vall(q68, c("Strongly Disagree" = 1, "Disagree" = 2, "Agree" = 3,
    "Strongly Agree" = 4, "No response" = 99)))
```

0.0.3 Dealing with split out responses from multiple choice, unique answer variables

In this dataset, several multiple choice variables that were originally meant to have a single answer, were spread out as if they had multiple answers. I believe this was because in the Qualtrics survey development process, the option for *Multiple answer* was selected, instead of *Single answer*.

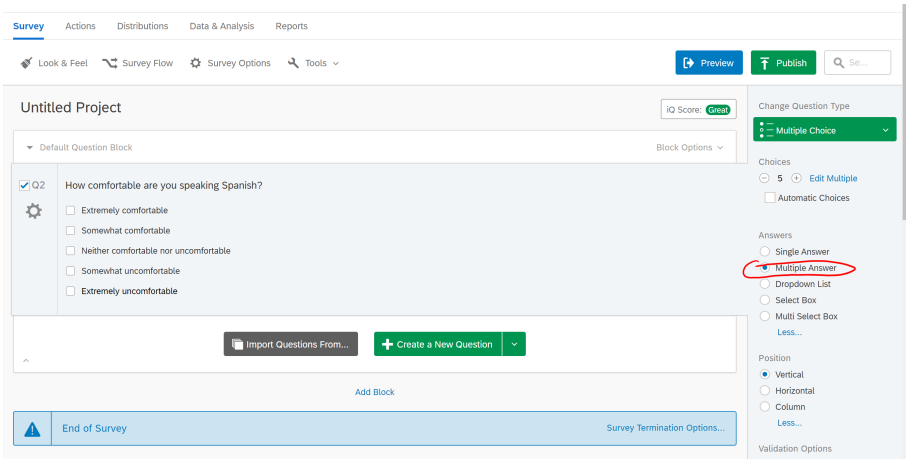


Figure 3: Qualtrics survey development

When this happens, participants could select mutually exclusive options, like this:

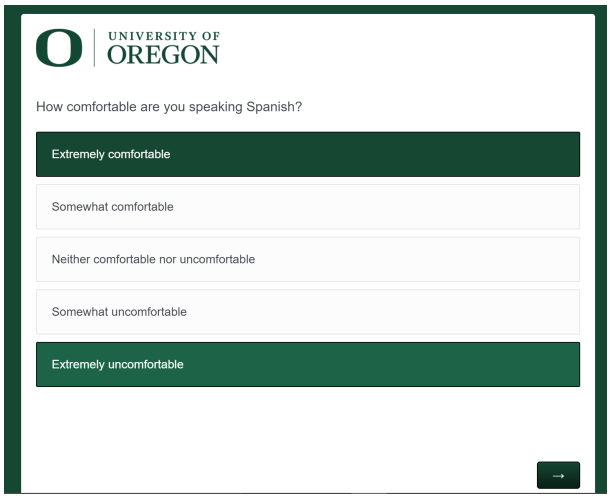


Figure 4: Qualtrics survey preview

When *Multiple answer* is selected, Qualtrics splits these multi-value fields into columns, assigning a value of 1 if a response option is chosen and a value of 0 if a response option is not chosen. In the following code, I collapsed the Spanish variable that was split out so it could be easily used in analyses. To avoid overwhelming the reader, I am omitting the code I used to collapse other language variables. I used the same procedure in all of these variables.

In the code below, the function pivot_longer makes the dataset "long" as it increases the number of rows and decreases the number of columns. This function gathers varables' names under the new variable `item_2` and gathers the values of these variables under the new variable `spanish_comfort`. Then, I chose only the options that had a value of 1, indicating when a participant chose that response option.

Finally, I recoded the response options to follow this scheme: "Not at all comfortable" = 1, "Somewhat comfortable" = 2, "Comfortable" = 3, "Very comfortable" = 4, "No response" = 99.

```
# collapsing spanish variables
spa <- elt_w1_clean_3 %>%
 select(id, starts_with("q132_2")) %>% # creating a dataset with only the id and
   Spanish variables
  pivot_longer(
    cols = starts_with("q132_2"),
    names_to = "item_2",
    values_to = "spanish_comfort",
    values_drop_na = TRUE) %>%
```

```
  filter(spanish_comfort == 1) %>%
 mutate(spanish_comfort = case_when(item_2 == "q132_2_1" ~ "1",
                                    item_2 == "q132_2_2" ~ "2",
                                    item_2 == "q132_2_3" ~ "3",
                                    item_2 == "q132_2_4" ~ "4",
                                    item_2 == "q132_2_99" ~ "99",
                            TRUE ~ as.character(spanish_comfort))) %>%
 select(-item_2) # selecting out variable with repetitive information
```

When all the language variables were collapsed I tested if there were duplicated cases and I found that participant identified with `id` 454 chose response option 1 and response option 2.

id	spanish_comfort
451	3
452	3
453	2
454	1
454	2
455	2
456	1
457	2
458	2

Because I can only assume that this was an entry error because the choices are mutually exclusive, “Not at all comfortable” = 1, vs. “Somewhat comfortable” = 2, I used the distinct function to retain only unique values.

For this case, option 1 = “Not at all comfortable” was retained as the function “assumes” the second option is the duplicative.

```
spa_2 <- spa %>%
 distinct(id, .keep_all = TRUE)
```

id	spanish_comfort
451	3
452	3
453	2
454	1
455	2
456	1
457	2
458	2

The last step in this process was making the language variables numeric so they could be used in quantitative analyses and adding the value labels so that the SPSS users can use the *value labels* button. I used the code below to do this.

```
elt_w1_clean_4 <- elt_w1_clean_4 %>%
 mutate(english_comfort = as.numeric(english_comfort),
        english_comfort = set_vall(english_comfort, c("not at all comfortable" = 1,
  "somewhat comfortable" = 2, "comfortable" = 3, "very comfortable" = 4, "no
  response" = 99)),
        spanish_comfort = as.numeric(spanish_comfort),
        spanish_comfort = set_vall(spanish_comfort, c("not at all comfortable" = 1,
  "somewhat comfortable" = 2, "comfortable" = 3, "very comfortable" = 4, "no
  response" = 99)),
        other1_lang_comfort = as.numeric(other1_lang_comfort),
        other1_lang_comfort = set_vall(other1_lang_comfort, c("not at all
  comfortable" = 1, "somewhat comfortable" = 2, "comfortable" = 3, "very
  comfortable" = 4, "no response" = 99)),
```

```
    other2_lang_comfort = as.numeric(other2_lang_comfort),
    other2_lang_comfort = set_vall(other2_lang_comfort, c("not at all
  comfortable" = 1, "somewhat comfortable" = 2, "comfortable" = 3, "very
  comfortable" = 4, "no response" = 99)),
    )
```

As shown, the following variables were the result of the collapsing process described above: `english_comfort`, `spanish_comfort`, `other1_lang_comfort`, and `other2_lang_comfort`.

0.0.4 Renaming demographic variables

In the code below, I used the rename function to rename the demographic variables that I will use to describe participants's characteristics in the next section of this report. This function uses a "new name" = "old name" pattern. Very straightforward!

At the end I selected out a few variables that did not have meaningful information. For instance, variable `q127` was a response/no response question that only indicated if participants chose to answer it. The meaningul information was contained in variable `q127_1_text` that was renamed as `age`.

```
elt_w1_clean_5 <- elt_w1_clean_4 %>%
 rename(c("age" = "q127_1_text"),
      c("birth_country" = "q128"),
      c("another_birth_country_text" = "q128_2_text"),
      c("age_first_moved_us" = "q129_1_text"),
      c("white" = "q130_1"),
      c("hispanic_latino_spanish" = "q130_2"),
      c("black_african_american" = "q130_3"),
      c("asian" = "q130_4"),
      c("american_indian_alaska_native" = "q130_5"),
      c("indigenous_americas" = "q130_6"),
      c("middle_eastern_north_african" = "q130_7"),
      c("native_hawaiian_pacific_islander" = "q130_8"),
      c("race_ethnicity_other" = "q130_9"),
      c("race_ethnicity_no_response" = "q130_99"),
      c("indigenous_americas_text" = "q130_6_text"),
      c("race_ethnicity_other_text" = "q130_9_text"),
      c("gender_id" = "q131"),
      c("years_in_position" = "q133"),
      c("years_in_school" = "q134"),
      c("equity_leadership" = "q135_1"),
      c("cultural_responsiveness" = "q135_2"),
      c("restorative_practices" = "q135_3"),
      c("diversity" = "q135_4"),
```

```
        c("ell" = "q135_5"),
        c("cont_ed_other" = "q135_6"),
        c("cont_ed_na" = "q135_88"),
        c("cont_ed_no_response" = "q135_99"),
        c("cont_ed_other_text" = "q135_6_text")) %>%
  select(-q127, -q129, -q131_3_text) # selecting out because they did not have
    meaningful info
```

Parent dataset

...

Youth dataset

...

Participant descriptives

In the following section, I will use descriptive statistics to summarize participants' characteristics.

to do: use function to create summariers from several vars. create nice table.

Educator's characteristics

...

Parent characteristics

...

Youth characteristics

...

Scale creation and Testing

Say something about scales

Educator's scales

...

Parent scales

...

Youth scales

...

Average Scores of Major Study Constructs

Say something about the average scores...

Educator's average scores

...

Parent average scores

...

Youth average scores

...

note: include plots with average scores

Recommendations

I recommend...

- id protocol
  - When developing the id protocol for schools, make sure that both values and values labels coincide.
  - Assign an unique identifier for each participant and an unique identifier per family.


Center for
Equity Promotion
Centro para la Promoción de la Equidad