Norming WebCDI

Benny deMayo

6/15/2020

```
#first we read in the WG data that is just from the SES pilot projects
all_wg_raw_ses <- readInWebCDI(fb_wg_directory)

save(
  all_wg_raw_ses,
  file = fs::path(
    project_root,
    "data",
    "ses_norming",
    "unfiltered",
    "wg_unfiltered_ses.RData"
  )
)

facebook_ws_raw_ses <-
  readInWebCDI(fb_ws_directory) %>%
  select( #drop a bunch of columns that were screwing up the merge with prolific data
    -opt_out,
    -country,
    -sibling_boolean,
    -sibling_data,
    -sibling_count,
    -caregiver_other
  )

prolific_raw_ses <-
  readInWebCDI(prolific_data_directory) %>%
  select(
    colnames(facebook_ws_raw_ses), #drop columns that are prolific specific
    -opt_out, #drop a bunch of columns that were screwing up the merge with facebook data
    -country,
    -sibling_boolean,
    -sibling_data,
    -sibling_count,
    -caregiver_other
  )

#all of the ws data collected
all_ws_raw_ses <-
  facebook_ws_raw_ses %>%
  bind_rows(prolific_raw_ses) %>%
  mutate(completed = case_when(
```

```
    stringr::str_to_lower(completed) == "true" ~ TRUE,
    stringr::str_to_lower(completed) == "false" ~ FALSE
  ))

save(
  all_ws_raw_ses,
  file = fs::path(
    project_root,
    "data",
    "ses_norming",
    "unfiltered",
    "ws_unfiltered_ses.RData"
  )
)
```

```
#we have to read in the full dataset and select out the manually-coded columns
wg_exclusion_info <-
  read_csv(all_data_wg_path) %>%
  select(
    study_name,
    subject_id,
    hearing_exclude,
    vision_exclude,
    illnesses_exclude
  ) %>%
  mutate(subject_id = as.character(subject_id))

wg_ses <-
  all_wg_raw_ses %>%
  filter(completed == TRUE) %>% #only take completed administrations
  #join it with the exclusion data we have from the screened files
  left_join(
    wg_exclusion_info,
    by = c("study_name", "subject_id")
  ) %>%
  filter(repeat_num == "1") %>% #done
  filterBirthweight() %>% #done
  filterMultilingual() %>% #done
  filterIllnesses() %>%
  filterVision() %>%
  filterHearing() %>%
  getCompletionInterval() %>%
  getEthnicities() %>%
  getMaternalEd() %>%
  filter_age_wg() %>%
  filter_nwords_wg() %>%
  left_join(facebook_wg_cutoffs, by = "age") %>%
  filter(completion_time >= minimum_time)

# wg_ses %>%
#   filter(completion_time < 15) %>%
#   mutate(mom_age = 2020 - mother_yob) %>%
#   filter(mom_age < 100) %>%
```

```
#   ggplot(aes(mom_age, completion_time, color = produced)) +
#   geom_point() +
#   scale_color_viridis_c()

save(
  wg_ses,
  file = fs::path(
    project_root,
    "data",
    "ses_norming",
    "filtered",
    "wg_filtered_ses.RData"
  )
)

ws1_exclusion_info <-
  read_csv(all_data_ws1_path) %>%
   select(
    study_name,
    subject_id,
    hearing_exclude,
    vision_exclude,
    illnesses_exclude
  ) %>%
  mutate(subject_id = as.character(subject_id))

ws2_exclusion_info <-
  read_csv(all_data_ws2_path) %>%
   select(
    study_name,
    subject_id,
    hearing_exclude,
    vision_exclude,
    illnesses_exclude
  ) %>%
  mutate(subject_id = as.character(subject_id))

ws_all_exclusion_info <-
  bind_rows(ws1_exclusion_info, ws2_exclusion_info)

ws_ses <-
  all_ws_raw_ses %>%
  filter(completed == TRUE) %>%
  #join it with the exclusion data we have from the screened files
  left_join(
    ws_all_exclusion_info,
    by = c("study_name", "subject_id")
  ) %>%
  filter(repeat_num == "1") %>%
  filterBirthweight() %>%
  filterMultilingual() %>%
  filterIllnesses() %>%
  filterVision() %>%
```

```
  filterHearing() %>%
  getCompletionInterval() %>%
  getEthnicities() %>%
  getMaternalEd() %>%
  filter_age_ws() %>%
  filter_nwords_ws() %>%
  left_join(prolific_ws_cutoffs, by = "age") %>%
  filter(completion_time >= minimum_time)


save(
  ws_ses,
  file = fs::path(
    project_root,
    "data",
    "ses_norming",
    "filtered",
    "ws_filtered_ses.RData"
  )
)

# prolific_filtered_n <-
#   ws_ses %>%
#   filter(str_detect(study_name, "prolific")) %>%
#   nrow()
#
# ws_ses %>%
#   filter(completion_time < 10) %>%
#   ggplot(aes(completion_time, produced, color = `Total Produced Percentile-sex`)) +
#   geom_point(position = position_jitter(width = 0.2)) +
#   geom_smooth(method = "lm") +
#   scale_color_viridis_c()
#
# prolific_ws <-
#   ws_ses %>%
#   filter(str_detect(study_name, "prolific")) %>%
#   mutate(factor_age = case_when(
#     age %in% 16:24 ~ "16 - 24 months",
#     age %in% 25:30 ~ "25 - 30 months"
#   ))
#
# younger_ws_quantiles <-
#   prolific_ws %>%
#   filter(factor_age == "16 - 24 months") %>%
#   pull(completion_time) %>%
#   quantile(seq(0, 1, .01))
#
# older_ws_quantiles <-
#   prolific_ws %>%
#   filter(factor_age == "25 - 30 months") %>%
#   pull(completion_time) %>%
#   quantile(seq(0, 1, .01))
#
```

```
# prolific_ws %>%
#   ggplot(aes(factor_age, completion_time)) +
#   geom_boxplot() +
#   geom_point(
#     aes(color = produced),
#     alpha = 0.5,
#     position = position_jitter(width = 0.2)
#   ) +
#   scale_color_viridis_c()
#
# wg_ses %>%
#   mutate(factor_age = case_when(
#     age %in% 8:12 ~ "8 - 12 months",
#     age %in% 13:18 ~ "13 - 18 months"
#   )) %>%
#   mutate(factor_age = fct_relevel(factor_age, "8 - 12 months")) %>%
#   filter(completion_time < 100) %>%
#   ggplot(aes(factor_age, completion_time)) +
#   geom_boxplot() +
#   geom_point(
#     aes(color = age),
#     alpha = 0.5,
#     position = position_jitter(width = 0.2)
#   ) +
#   scale_color_viridis_c()
```

```
#exploring mom age

demographics_df_ses <-
  bind_rows(
    wg_ses %>%
      select(
        study_name,
        subject_id,
        sex,
        age,
        ethnicity,
        maternal_ed,
        produced,
        completion_time,
        mother_yob
      ) %>%
      mutate(mom_age = 2020 - mother_yob) %>%
      filter(mom_age < 100)
    ,
    ws_ses %>%
      select(
        study_name,
        subject_id,
        sex,
        age,
        ethnicity,
        maternal_ed,
```

```
        produced,
        completion_time,
        mother_yob
      ) %>%
      mutate(mom_age = 2020 - mother_yob) %>%
      filter(mom_age < 100)
  )

demographics_df_ses %>%
  arrange(desc(mom_age)) %>% View()

momed_ses_df <-
  demographics_df_ses %>%
  mutate(highschool = case_when(
    maternal_ed == "Some high school or less" ~ "High school",
    maternal_ed == "High school diploma" ~ "High school",
    maternal_ed == "Some college education" ~ "College",
    maternal_ed == "College diploma or more" ~ "College"
  )) %>%
  filter(completion_time > 8.5)

momed_ses_df %>%
  filter(
    age <= 30
  ) %>%
  ggplot(aes(age, produced, shape = highschool, color = mom_age)) +
  geom_jitter(alpha = 0.5, width = 0.225) +
  geom_smooth(aes(linetype = highschool), method = "lm") +
  labs(
    x = "Age in months",
    y = "Words produced"
  ) +
  ggthemes::theme_few() +
  theme(legend.title = element_blank()) +
  scale_color_viridis_c()
```

```
#Calculating exclusions

completed_wg_ses <-
  all_wg_raw_ses %>%
  filter(completed == TRUE) %>%
  left_join(
    wg_exclusion_info,
    by = c("study_name", "subject_id")
  ) %>%
  getCompletionInterval()


completed_ws_ses <-
  all_ws_raw_ses %>%
  filter(completed == TRUE) %>%
  left_join(
    ws_all_exclusion_info,
```

```
    by = c("study_name", "subject_id")
  ) %>%
  getCompletionInterval()
```

More fine-grained exclusion information.

```
#Disclaimer that this is some of the worst code I've ever written. Sorry everyone.

n_total_wg_ses <- nrow(completed_wg_ses)
n_total_ws_ses <- nrow(completed_ws_ses)

excl_col_names <-
  c(
    "Exclusion",
    "WG exclusions",
    "% of full WG sample excluded",
    "WS exclusions",
    "% of full WS sample excluded"
  )


#First take away kids who have done the survey more than once.
wg_minus_repeats <-
  completed_wg_ses %>%
  filter(repeat_num == "1")

wg_repeats_n <- n_total_wg_ses - nrow(wg_minus_repeats)

ws_minus_repeats <-
  completed_ws_ses %>%
  filter(repeat_num == "1")

ws_repeats_n <- n_total_ws_ses - nrow(ws_minus_repeats)

repeat_admins <-
  c(
    "Not first administration",
    wg_repeats_n,
    percent(wg_repeats_n / n_total_wg_ses, accuracy = 0.01),
    ws_repeats_n,
    percent(ws_repeats_n / n_total_ws_ses, accuracy = 0.01)
  )

names(repeat_admins) <- excl_col_names

#Next take away kids born pre-term or with low birthweight.

wg_minus_premie <-
  wg_minus_repeats %>%
  filterBirthweight()

wg_premie_n <- nrow(wg_minus_repeats) - nrow(wg_minus_premie)
```

```
ws_minus_premie <-
  ws_minus_repeats %>%
  filterBirthweight()

ws_premie_n <- nrow(ws_minus_repeats) - nrow(ws_minus_premie)

premies <-
  c(
    "Premature or low birthweight",
    wg_premie_n,
    percent(wg_premie_n / n_total_wg_ses, accuracy = 0.01),
    ws_premie_n,
    percent(ws_premie_n / n_total_ws_ses, accuracy = 0.01)
  )

names(premies) <- excl_col_names

#Next take away kids with multilingual exposure

wg_minus_multiling <-
  wg_minus_premie %>%
  filterMultilingual()

wg_multiling_n <- nrow(wg_minus_premie) - nrow(wg_minus_multiling)

ws_minus_multiling <-
  ws_minus_premie %>%
  filterMultilingual()

ws_multiling_n <- nrow(ws_minus_premie) - nrow(ws_minus_multiling)

multiling <-
  c(
    "Multilingual exposure",
    wg_multiling_n,
    percent(wg_multiling_n / n_total_wg_ses, accuracy = 0.01),
    ws_multiling_n,
    percent(ws_multiling_n / n_total_ws_ses, accuracy = 0.01)
  )

names(multiling) <- excl_col_names

#Next exclude kids with problems of illness, vision, or hearing
wg_minus_health <-
  wg_minus_multiling %>%
  filterIllnesses() %>%
  filterVision() %>%
  filterHearing()

wg_health_n <- nrow(wg_minus_multiling) - nrow(wg_minus_health)

ws_minus_health <-
  ws_minus_multiling %>%
```

```
  filterIllnesses() %>%
  filterVision() %>%
  filterHearing()

ws_health_n <- nrow(ws_minus_multiling) - nrow(ws_minus_health)

health <-
  c(
    "Illnesses/Vision/Hearing",
    wg_health_n,
    percent(wg_health_n / n_total_wg_ses, accuracy = 0.01),
    ws_health_n,
    percent(ws_health_n / n_total_ws_ses, accuracy = 0.01)
  )

names(health) <- excl_col_names

#Now filter out kids who are the wrong age
wg_minus_age <-
  wg_minus_health %>%
  filter_age_wg()

wg_age_n <- nrow(wg_minus_health) - nrow(wg_minus_age)

ws_minus_age <-
  ws_minus_health %>%
  filter_age_ws()

ws_age_n <- nrow(ws_minus_health) - nrow(ws_minus_age)

age <-
  c(
    "Out of age range",
    wg_age_n,
    percent(wg_age_n / n_total_wg_ses, accuracy = 0.01),
    ws_age_n,
    percent(ws_age_n / n_total_ws_ses, accuracy = 0.01)
  )

names(age) <- excl_col_names

#Now we need to get rid of people who did the survey too fast

wg_minus_fakes <-
  wg_minus_age %>%
  left_join(facebook_wg_cutoffs, by = "age") %>%
  filter(completion_time >= minimum_time)

wg_fake_n <- nrow(wg_minus_age) - nrow(wg_minus_fakes)

ws_minus_fakes <-
  ws_minus_age %>%
  left_join(prolific_ws_cutoffs, by = "age") %>%
```

```
  filter(completion_time >= minimum_time)

ws_fake_n <- nrow(ws_minus_age) - nrow(ws_minus_fakes)

fakes <-
  c(
    "Completed survey too quickly",
    wg_fake_n,
    percent(wg_fake_n / n_total_wg_ses, accuracy = 0.01),
    ws_fake_n,
    percent(ws_fake_n / n_total_ws_ses, accuracy = 0.01)
  )

names(fakes) <- excl_col_names

#lastly filter out kids who have buggy word totals (more than possible)

wg_minus_wordbugs <-
  wg_minus_fakes %>%
  filter_nwords_wg()

wg_wordbugs_n <- nrow(wg_minus_fakes) - nrow(wg_minus_wordbugs)

ws_minus_wordbugs <-
  ws_minus_fakes %>%
  filter_nwords_ws()

ws_wordbugs_n <- nrow(ws_minus_fakes) - nrow(ws_minus_wordbugs)

wordbugs <-
  c(
    "System error in word tabulation",
    wg_wordbugs_n,
    percent(wg_wordbugs_n / n_total_wg_ses, accuracy = .01),
    ws_wordbugs_n,
    percent(ws_wordbugs_n / n_total_ws_ses, accuracy = .01)
  )

names(wordbugs) <- excl_col_names


#calculate total amount of WG exclusions
total_wg_exclusions_ses <-
  wg_repeats_n +
  wg_premie_n +
  wg_multiling_n +
  wg_health_n +
  wg_age_n +
  wg_fake_n +
  wg_wordbugs_n

#calculate total amount of WS exclusions
total_ws_exclusions_ses <-
```

```
  ws_repeats_n +
  ws_premie_n +
  ws_multiling_n +
  ws_health_n +
  ws_age_n +
  ws_fake_n +
  ws_wordbugs_n

#make a row in the table for this
totals <-
  c(
    "Total exclusions",
    total_wg_exclusions_ses,
    percent(total_wg_exclusions_ses / n_total_wg_ses),
    total_ws_exclusions_ses,
    percent(total_ws_exclusions_ses / n_total_ws_ses)
  )

names(totals) <- excl_col_names

#now make the table
exclusion_tbl_ses <-
  bind_rows(
    repeat_admins,
    premies,
    multiling,
    health,
    age,
    fakes,
    wordbugs,
    totals
  )

knitr::kable(exclusion_tbl_ses)
```

Exclusion	WG exclusions	% of full WG sample excluded	WS exclusions	% of full WS sample excluded
Not first administration	0	0.00%	0	0.00%
Premature or low birthweight	7	2.53%	1	0.33%
Multilingual exposure	18	6.50%	23	7.62%
Illnesses/Vision/Hearing	4	1.44%	4	1.32%
Out of age range	1	0.36%	26	8.61%
Completed survey too quickly	119	42.96%	133	44.04%
System error in word tabulation	0	0.00%	0	0.00%
Total exclusions	149	54%	187	62%

```
total_admin <- nrow(ws_ses) + nrow(wg_ses)
```

```
save(
  exclusion_tbl_ses,
  file = path(
    project_root,
    "data",
    "exclusion_tables",
    "ses_norming_exclusions",
    ext = "RData"
  )
)
```

Looking at people who did and didn't take enough time to finish

```
#All data, incomplete and complete
all_d <-
  bind_rows(
    all_wg_raw %>%
      getCompletionInterval() %>%
      getMaternalEd() %>%
      getEthnicities() %>%
      select(
        study_name,
        subject_id,
        link,
        completed,
        completedBackgroundInfo,
        sex,
        age,
        ethnicity,
        maternal_ed,
        words_produced = `Words Produced`,
        completion_time,
        maternal_ed,
        ethnicity
      ) %>%
      mutate(completed = as.character(completed)),
    all_ws_raw %>%
      getCompletionInterval() %>%
      getMaternalEd() %>%
      getEthnicities() %>%
      select(
        study_name,
        subject_id,
        link,
        completed,
        completedBackgroundInfo,
        sex,
        age,
        ethnicity,
        maternal_ed,
        words_produced = `Total Produced`,
        completion_time,
        maternal_ed,
```

```
        ethnicity
      ) %>%
      mutate(completed = as.character(completed))
  )

incomplete <- all_d %>% filter(completed == "FALSE")

complete_bg <- incomplete %>% filter(completedBackgroundInfo == "TRUE")

incomplete_ethnicity <-
  incomplete %>%
  count(ethnicity)

complete_bg_ethnicity <-
  complete_bg %>%
  count(ethnicity)

incomplete_momed <-
  incomplete %>%
  count(maternal_ed)

complete_bg_momed <-
  complete_bg %>%
  count(maternal_ed)

incomplete_vocab <-
  incomplete %>%
  filter(!is.na(words_produced))

complete_bg_vocab <-
  complete_bg %>%
  filter(!is.na(words_produced))

incomplete_vocab %>%
  ggplot(aes(age, words_produced)) +
  geom_point() +
  geom_smooth(method = "lm")

complete_bg %>%
  filter(completion_time > min_completion_time) %>%
  ggplot(aes(age, words_produced)) +
  geom_point() +
  geom_smooth(method = "lm")

complete_bg %>%
  filter(completion_time > min_completion_time) %>%
  count(ethnicity)
```

```
#combine relevant demographic information from both WS and WG
demographics_df_ses <-
  bind_rows(
    wg_ses %>%
      select(
```

```
        study_name,
        subject_id,
        sex,
        age,
        ethnicity,
        maternal_ed,
        produced
      ),
    ws_ses %>%
      select(
        study_name,
        subject_id,
        sex,
        age,
        ethnicity,
        maternal_ed,
        produced
      )
  )

total_n_ses <- nrow(demographics_df_ses)
ethnicity_na_n_ses <-
  nrow(demographics_df_ses %>% filter(ethnicity == "No ethnicity reported"))
ethnicity_total_n_ses <- total_n_ses - ethnicity_na_n_ses
maternal_ed_na_n_ses <- nrow(demographics_df_ses %>% filter(maternal_ed == "Not reported"))
maternal_ed_total_n_ses <- total_n_ses - maternal_ed_na_n_ses
```

```
#Ethnicity plot creation
ethnicity_plot_df_ses <-
  demographics_df_ses %>%
  getEthnicitySummary() %>%
  filter(
    !is.na(ethnicity),
    ethnicity != "No ethnicity reported"
  ) %>%
  mutate(`Current study` = prop.table(n)) %>%
  left_join(old_ethnicity_numbers, by = "ethnicity") %>%
  select(-n) %>%
  pivot_longer(
    cols = c(`Current study`, `2007 manual`),
    names_to = "study",
    values_to = "proportion"
  )

ethnicity_plot_ses <-
  ethnicity_plot_df_ses %>%
  ggplot(aes(ethnicity, proportion, fill = study)) +
  geom_col(position = "dodge") +
  labs(
    y = "Proportion of\nrespondents"
    # caption = str_c(
    #   "A total of ",
    #   total_admin,
```

```
    #   " CDI's were collected. ",
    #   total_exclusions,
    #   " did not meet inclusion criteria\n(multilingual status, vision/hearing impairments, premature
    #   min_completion_time,
    #   " minutes); ",
    #   ethnicity_na_n,
    #   " did not report ethnicity, leaving a final N = ",
    #   ethnicity_total_n,
    #   "."
    # )
  ) +
  theme_few() +
  theme(
    legend.title = element_blank(),
    axis.text = element_text(size = 14),
    axis.title = element_text(size = 13),
    legend.text = element_text(size = 13),
    axis.title.x = element_blank(),
    plot.title = element_text(size = 15),
    plot.caption = element_text(hjust = 0)
  )
```

```
#Maternal education plots

maternal_ed_plot_df <-
  demographics_df %>%
  count(maternal_ed) %>%
  mutate(`Current study` = prop.table(n)) %>%
  left_join(old_momed_numbers, by = "maternal_ed") %>%
  select(-n) %>%
  pivot_longer(
    cols = c(`Current study`, `2007 manual`),
    names_to = "study",
    values_to = "proportion"
  ) %>%
  mutate(
    maternal_ed = fct_relevel(
      maternal_ed,
      "Some high school or less",
      "High school diploma",
      "Some college education",
      "College diploma or more"
    )
  )

x_axis_labs <- c(
  "Some high school\n or less",
  "High school\ndiploma",
  "Some college\neducation",
  "College diploma\nor more"
)

maternal_ed_plot <-
```

```
  maternal_ed_plot_df %>%
  ggplot(aes(maternal_ed, proportion, fill = study)) +
  geom_col(position = "dodge") +
  labs(
    x = "Education Level",
    y = "Proportion of\nrespondents"
    # title = "Maternal education",
    # caption = str_c(
    #   "A total of ",
    #   total_admin,
    #   " CDI's were collected. ",
    #   total_exclusions,
    #   " did not meet inclusion\ncriteria",
    #   " (multilingual status, vision/hearing impairments, premature\nbirth, completion time under ",
    #   min_completion_time,
    #   " minutes); leaving a final N = ",
    #   total_n,
    #   "."
    # )
  ) +
  theme_few() +
  theme(
    legend.title = element_blank(),
    axis.text.x = element_text(angle = 30, vjust = 0.9, hjust = 1, size = 13.5),
    axis.title.x = element_blank(),
    legend.text = element_text(size = 13),
    plot.caption = element_text(hjust = 0)
  ) +
  scale_x_discrete(labels = x_axis_labs)

maternal_ed_plot
```

Joint plot of maternal ed and ethnicity

```
#tweak the ethnicity plot
ethnicity_grid_plot <-
  ethnicity_plot_df %>%
  ggplot(aes(ethnicity, proportion, fill = study)) +
  geom_col(position = "dodge") +
  labs(
    y = "Proportion of\nrespondents"
  ) +
  theme_few() +
  theme(
    legend.title = element_blank(),
    axis.text.x = element_text(
       angle = 25,
       vjust = 0.9,
       hjust = 1
     )
  ) +
  labs(x = "Ethnicity")
```

```r
#tweak the maternal ed plot
maternal_ed_grid_plot <-
  maternal_ed_plot_df %>%
  filter(maternal_ed != "Not reported") %>%
  ggplot(aes(maternal_ed, proportion, fill = study)) +
  geom_col(position = "dodge") +
  labs(x = "Primary caregiver education") +
  theme_few() +
  theme(
    legend.title = element_blank(),
    axis.text.x = element_text(angle = 25, vjust = 0.9, hjust = 1)
  ) +
  scale_x_discrete(labels = x_axis_labs)

prow <- cowplot::plot_grid(
  ethnicity_grid_plot +
    theme(
      legend.position = "none",
      plot.margin = (margin(r = 2, l = 0)),
      axis.text = element_text(size = 12)
    ),
  maternal_ed_grid_plot +
    ylab(NULL) +
    theme(
      legend.position = "none",
      plot.margin = (margin(r = 2, l = 2)),
      axis.text = element_text(size = 12)
    ),
  align = "vh",
  labels = c("A", "B")
)

legend <-
  get_legend(
    ethnicity_grid_plot +
      # guides(color = guide_legend(nrow = 1)) +
      # theme(legend.position = "bottom")
      theme(
        legend.box.margin = margin(0, 0, 0, 15),
        legend.text = element_text(size = 12)
      )
  )

plot_grid(prow, legend, rel_widths = c(3, .5))
```

```r
momed_ses_df <-
  demographics_df_ses %>%
  mutate(highschool = case_when(
    maternal_ed == "Some high school or less" ~ "High school",
    maternal_ed == "High school diploma" ~ "High school",
    maternal_ed == "Some college education" ~ "College",
    maternal_ed == "College diploma or more" ~ "College"
  ))
```

```
momed_ses_df %>%
  filter(
    age <= 30
  ) %>%
  ggplot(aes(age, produced, color = highschool)) +
  geom_jitter(alpha = 0.3, width = 0.225) +
  geom_smooth(method = "lm") +
  labs(
    x = "age in months",
    y = "words produced"
  ) +
  ggthemes::theme_few() +
  theme(legend.title = element_blank())

demographics_df %>%
  filter(
    age <= 30
  ) %>%
  ggplot(aes(age, words_produced, color = maternal_ed)) +
  geom_jitter(alpha = 0.3, width = 0.225) +
  geom_smooth(method = "lm") +
  labs(
    x = "age in months",
    y = "words produced"
  ) +
  ggthemes::theme_few() +
  theme(legend.title = element_blank())

demographics_df %>%
  filter(
    age <= 30,
    sex != "Other"
  ) %>%
  ggplot(aes(age, words_produced, color = sex)) +
  geom_jitter(alpha = 0.3, width = 0.225) +
  geom_smooth(method = "lm") +
  labs(
    x = "age in months",
    y = "words produced"
  ) +
  ggthemes::theme_few() +
  theme(legend.title = element_blank())

demographics_df %>%
  filter(
    age <= 30,
    ethnicity != "No ethnicity reported"
  ) %>%
  ggplot(aes(age, words_produced, color = ethnicity)) +
  geom_jitter(alpha = 0.3, width = 0.225) +
  geom_smooth(method = "lm", alpha = 0.5) +
```

```
  labs(
    x = "age in months",
    y = "words produced"
  ) +
  ggthemes::theme_few() +
  theme(legend.title = element_blank())
```

```
#Ethnicities on the people who completed too fast (if they even are people).

eth_fake_wg <-
  completed_wg %>%
  getEthnicities() %>%
  filter(completion_time <= min_completion_time) %>%
  count(ethnicity) %>%
  mutate(prop = n / sum(n))

eth_fake_ws <-
  completed_ws %>%
  getEthnicities() %>%
  filter(completion_time <= min_completion_time) %>%
  count(ethnicity) %>%
  mutate(prop = n / sum(n))
```