Full dataset analysis

First read in all of the relevant data.

```
all_ws_1 <-
  readInWebCDI(all_data_ws1_path) %>%
  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
  )

all_ws_2 <-
  readInWebCDI(all_data_ws2_path) %>%
  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
  )

all_ws_raw <-
  all_ws_1 %>%
  bind_rows(all_ws_2) %>%
  mutate(completed = case_when(
    stringr::str_to_lower(completed) == "true" ~ TRUE,
    stringr::str_to_lower(completed) == "false" ~ FALSE
  )) %>%
  filter(completed == TRUE)

all_wg_raw <-
  readInWebCDI(all_data_wg_path) %>%
  filter(completed = TRUE)

save(
  all_ws_raw,
  file = path(
    project_root,
    "data",
    "full_dataset",
    "unfiltered",
    "ws_unfiltered.RData"
  )
```

```
)

save(
  all_wg_raw,
  file = path(
    project_root,
    "data",
    "full_dataset",
    "unfiltered",
    "wg_unfiltered.RData"
  )
)
```

Filter out: multilingual exposure, illnesses, vision and hearing problems.

```
#original sample size of 2868
#WG

#First we do all of the exclusions EXCEPT FOR TIMING.
wg_almost_filtered <-
  all_wg_raw %>%
  filter(completed == TRUE) %>%
  filter(repeat_num == "1") %>%
  filterBirthweight() %>%
  filterMultilingual() %>%
  filterIllnesses() %>%
  filterVision() %>%
  filterHearing() %>%
  getCompletionInterval() %>%
  getEthnicities() %>%
  getMaternalEd() %>%
  filter_age_wg() %>%
  filter_nwords_wg() #changes Words Understood to "understood" and likewise with Total Produced

##Now we look at Danielle's Facebook data to determine the cutoffs for WG

facebook_wg <-
  wg_almost_filtered %>%
  filter(study_name == "Facebook-2-WG2")

#use a quantile regression on Danielle's sample to figure out 5th percentile by age
facebook_wg_model <-
  facebook_wg %>%
  nest() %>%
  mutate(model = map(data, function(fb_wg_data) {
    rq(completion_time ~ age, data = fb_wg_data, tau = .05)
  }))
```

```
## Warning: `...` must not be empty for ungrouped data frames.
## Did you want `data = everything()`?
```

```
data_cells_wg <- facebook_wg %>% distinct(age)

facebook_wg_cutoffs <-
  facebook_wg_model %>%
  mutate(fits = map(model, function(m) {
    data_cells_wg %>%
      mutate(fitted = predict(m, newdata = data_cells_wg, type = "response"))
  })) %>%
  select(fits) %>%
  unnest(fits) %>%
  rename(minimum_time = fitted)

#get rid of people who took survey too fast
wg_filtered <-
  wg_almost_filtered %>%
  left_join(facebook_wg_cutoffs, by = "age") %>%
  filter(completion_time >= minimum_time)

wg_exclusion_n <- nrow(all_wg_raw) - nrow(wg_filtered)

#save filtered dataset to the project repo
save(
  wg_filtered,
  file = path(
    project_root,
    "data",
    "full_dataset",
    "filtered",
    "wg_filtered.RData"
  )
)

#save the quantile regression cutoffs to the project repo

save(
  facebook_wg_cutoffs,
  file = path(
    project_root,
    "data",
    "cutoffs",
    "wg_cutoffs.RData"
  )
)
```

```
ws_almost_filtered <-
  all_ws_raw %>%
  filter(completed == TRUE) %>%
  filter(repeat_num == "1") %>%
  filterBirthweight() %>%
  filterMultilingual() %>%
  filterIllnesses() %>%
  filterVision() %>%
  filterHearing() %>%
```

```
  getCompletionInterval() %>%
  getEthnicities() %>%
  getMaternalEd() %>%
  filter_age_ws() %>%
  filter_nwords_ws() #changes Words Understood to "understood" and likewise with Total Produced

#Get the prolific data that we are going to use for the WS cutoffs.
prolific_ws <-
  ws_almost_filtered %>%
  filter(str_detect(study_name, "prolific"))

prolific_ws_model <-
  prolific_ws %>%
  nest() %>%
  mutate(model = map(data, function(prolific_wg_data) {
    rq(completion_time ~ age, data = prolific_wg_data, tau = .05)
  }))
```

```
## Warning: `...` must not be empty for ungrouped data frames.
## Did you want `data = everything()`?
```

```
data_cells_ws <- prolific_ws %>% distinct(age)

prolific_ws_cutoffs <-
  prolific_ws_model %>%
  mutate(fits = map(model, function(m) {
    data_cells_ws %>%
      mutate(fitted = predict(m, newdata = data_cells_ws, type = "response"))
  })) %>%
  select(fits) %>%
  unnest(fits) %>%
  rename(minimum_time = fitted)

ws_filtered <-
  ws_almost_filtered %>%
  left_join(prolific_ws_cutoffs, by = "age") %>%
  filter(completion_time >= minimum_time)


ws_exclusion_n <- nrow(all_ws_raw) - nrow(ws_filtered)

save(
  ws_filtered,
  file = path(
    project_root,
    "data",
    "full_dataset",
    "filtered",
    "ws_filtered.RData"
  )
)

save(
```

```
  prolific_ws_cutoffs,
  file = path(
    project_root,
    "data",
    "cutoffs",
    "ws_cutoffs.RData"
  )
)

total_n <- nrow(all_ws_raw) + nrow(all_wg_raw)

filtered_n <- nrow(ws_filtered) + nrow(wg_filtered)

nrow(all_wg_raw)
```

```
## [1] 2867
```

```
nrow(all_ws_raw)
```

```
## [1] 3564
```

```
#Comprehension and production measures
medians <-
  ws_filtered %>%
  mutate(
    maternal_ed = fct_recode(
      maternal_ed,
      "High school diploma or less" = "High school diploma",
      "High school diploma or less" = "Some high school or less"
    )
  ) %>%
  filter(produced < ws_total_words & maternal_ed != "Not reported") %>%
  group_by(maternal_ed, age) %>%
  summarize(median = median(produced))

ws_filtered %>%
  mutate(
    maternal_ed = fct_recode(
      maternal_ed,
      "High school diploma or less" = "High school diploma",
      "High school diploma or less" = "Some high school or less"
    )
  ) %>%
  filter(produced < ws_total_words & maternal_ed != "Not reported") %>%
  ggplot(aes(age, produced, color = maternal_ed)) +
  ggthemes::theme_few() +
  geom_jitter(alpha = 0.3, width = 0.225) +
  coord_cartesian(ylim = c(0, 686)) +
  geom_quantile(quantiles = .5, method = "rqss", lambda = 5, size = 1) +
  labs(
    x = "age in months",
    y = "total words produced",
```

```
    color = "Maternal education"
  )

wg_filtered %>%
  filter(!is.na(maternal_ed)) %>%
  ggplot(aes(age, understood, color = maternal_ed)) +
  ggthemes::theme_few() +
  geom_jitter(alpha = 0.3, width = 0.225) +
  geom_smooth(method = "lm") +
  coord_cartesian(ylim = c(0, 390)) +
  labs(
    x = "Age in months",
    y = "Total words understood",
    color = "Maternal education"
  )

wg_filtered %>%
  filter(!is.na(maternal_ed)) %>%
  ggplot(aes(age, produced, color = maternal_ed)) +
  ggthemes::theme_few() +
  geom_jitter(alpha = 0.3, width = 0.225) +
  geom_smooth(method = "lm") +
  coord_cartesian(ylim = c(0, 390)) +
  labs(
    x = "Age in months",
    y = "Total words produced",
    color = "Maternal education"
  )

wg_filtered %>%
   mutate(
    maternal_ed = fct_recode(
      maternal_ed,
      "High school diploma or less" = "High school diploma",
      "High school diploma or less" = "Some high school or less"
    )
  ) %>%
  filter(!is.na(maternal_ed)) %>%
  select(
    age,
    `Words Understood` = understood,
    `Words Produced` = produced,
    maternal_ed
  ) %>%
  pivot_longer(
    cols = c("Words Understood", "Words Produced"),
    names_to = "measure",
    values_to = "words"
  ) %>%
  group_by(age, maternal_ed, measure) %>%
  summarize(
    median = median(words, na.rm = TRUE),
    first_quartile = quantile(words, probs = 0.25, na.rm = TRUE),
```

```
    third_quartile = quantile(words, probs = 0.75, na.rm = TRUE)
  ) %>%
  ggplot(aes(age, median, color = maternal_ed)) +
  facet_grid(~measure) +
  geom_point(position = position_dodge(width = 0.5)) +
  geom_linerange(
    aes(ymin = first_quartile, ymax = third_quartile),
    position = position_dodge(width = 0.5)
  ) +
  scale_x_continuous(breaks = seq(from = 8, to = 18, by = 2)) +
  coord_cartesian(ylim = c(0, 390)) +
  ggthemes::theme_few() +
  labs(
    color = "Maternal education level",
    x = "age in months",
    y = "number of words"
  )

ggsave(
  "median_plot.png",
  plot = last_plot(),
  path = fig_directory,
  width = 9,
  height = 4.5
)

wg_filtered %>%
  mutate(
    maternal_ed = fct_recode(
      maternal_ed,
      "High school diploma or less" = "High school diploma",
      "High school diploma or less" = "Some high school or less"
    )
  ) %>%
  filter(!is.na(maternal_ed)) %>%
  select(age, produced, understood, maternal_ed) %>%
  pivot_longer(
    cols = c("produced", "understood"),
    names_to = "measure",
    values_to = "words"
  ) %>%
  ggplot(aes(age, words, color = maternal_ed)) +
  facet_grid(~measure) +
  geom_jitter(alpha = 0.2, width = 0.225) +
  geom_quantile(quantiles = .5, method = "rqss", lambda = 3, size = 1) +
  coord_cartesian(ylim = c(0, 390)) +
  ggthemes::theme_few() +
  labs(
    color = "Maternal education level",
    x = "age in months",
    y = "number of words"
  ) +
  scale_x_continuous(breaks = seq(from = 8, to = 18, by = 2))
```

Inferential statistics on SES and gender

```
wg_lm_df <-
  wg_filtered %>%
  mutate(
    age_c = age - mean(age, na.rm = TRUE),
    maternal_ed_c = mother_education - mean(mother_education, na.rm = TRUE),
    maternal_ed = fct_recode(
      maternal_ed,
      "High school diploma or less" = "High school diploma",
      "High school diploma or less" = "Some high school or less"
    )
  )

ses_wg_lm_prod <-
  lm_robust(formula = produced ~ age_c * maternal_ed, data = wg_lm_df)

ses_wg_lm_comp <-
  lm_robust(formula = understood ~ age_c * maternal_ed, data = wg_lm_df)

summary(ses_wg_lm_prod)
summary(ses_wg_lm_comp)

ws_lm_df <-
  ws_filtered %>%
  filter(maternal_ed != "Not reported") %>%
  mutate(
    age_c = age - mean(age, na.rm = TRUE),
    maternal_ed_c = mother_education - mean(mother_education, na.rm = TRUE),
    maternal_ed = fct_recode(
      maternal_ed,
      "High school diploma or less" = "High school diploma",
      "High school diploma or less" = "Some high school or less"
    )
  ) %>%
  filter(maternal_ed_c > -10)

ses_ws_lm <-
  lm(formula = produced ~ age_c * maternal_ed, data = ws_lm_df)

summary(ses_ws_lm)

gender_ws_lm <-
  lm(formula = produced ~ age_c * sex, data = ws_lm_df)

summary(gender_ws_lm)
```

Gender analyses

```
wg_filtered_gender <-
  wg_filtered %>%
  select(age, produced, sex)

ws_filtered_gender <-
```

```
  ws_filtered %>%
  select(age, produced, sex)

#filter out kids without a binary gender listed
all_d_gender <-
  bind_rows(ws_filtered_gender, wg_filtered_gender) %>%
  filter(sex != "Other")

all_gender_n <- nrow(all_d_gender)

all_d_gender %>%
  ggplot(aes(age, produced, color = sex)) +
  ggthemes::theme_few() +
  geom_jitter(alpha = 0.3, width = 0.225) +
  geom_smooth(method = "loess") +
  coord_cartesian(ylim = c(0, 686)) +
  labs(
    x = "Age in months",
    y = "Total words produced",
    color = "Child gender"
  )
```

```
#Counting hispanic and latino heritage
wg_filtered %>%
  mutate(hispanic = !is.na(child_hispanic_latino) & child_hispanic_latino == "TRUE") %>%
  count(hispanic) %>%
  mutate(prop = n / sum(n))

ws_filtered %>%
  mutate(hispanic = !is.na(child_hispanic_latino) & child_hispanic_latino == "TRUE") %>%
  count(hispanic)%>%
  mutate(prop = n / sum(n))
```

```
#Demographic analyses on the entire sample


demographics_df <-
  bind_rows(
    wg_filtered %>%
      mutate(
        hispanic = !is.na(child_hispanic_latino) &
          child_hispanic_latino == "TRUE",
        year_of_test = year(created_date),
        mom_age = year_of_test - mother_yob
      ) %>%
      select(
        study_name,
        subject_id,
        age,
        ethnicity,
        maternal_ed,
        produced,
        sex,
```

```
        hispanic,
        mom_age,
        completion_time
      ),
    ws_filtered %>%
      mutate(
        hispanic = !is.na(child_hispanic_latino) &
          child_hispanic_latino == "TRUE",
        year_of_test = year(created_date),
        mom_age = year_of_test - mother_yob
      ) %>%
      select(
        study_name,
        subject_id,
        age,
        ethnicity,
        maternal_ed,
        produced,
        sex,
        hispanic,
        completion_time
      )
  ) %>%
  filter(
    !(study_name %in% ses_studies),
    completion_time != 0
  )


ethnicity_plot_df <-
  demographics_df %>%
  count(ethnicity) %>%
  filter(!is.na(ethnicity)) %>%
  filter(ethnicity != "No ethnicity reported") %>%
  mutate(`Web-CDI sample` = prop.table(n)) %>%
  left_join(old_ethnicity_numbers, by = "ethnicity") %>%
  select(-n) %>%
  pivot_longer(
    cols = c(`Web-CDI sample`, `2007 manual`),
    names_to = "study",
    values_to = "proportion"
  )

whites_vs_census <-
  demographics_df %>%
  mutate(ethnicity = as.character(ethnicity)) %>%
  mutate(ethnicity = factor(case_when(
    ethnicity == "White" & !hispanic ~ "White, non-Hispanic",
    ethnicity == "White" & hispanic ~ "White, Hispanic",
    TRUE ~ ethnicity
  ))) %>%
  count(ethnicity) %>%
  filter(ethnicity != "No ethnicity reported") %>%
```

```
  mutate(`Web-CDI sample` = prop.table(n)) %>%
  left_join(census_ethnicity_hispanic_white, by = "ethnicity") %>%
  select(-n) %>%
  pivot_longer(
    cols = c(`Web-CDI sample`, `U.S. Census Estimates`),
    names_to = "study",
    values_to = "proportion"
  ) %>%
  mutate(study = fct_relevel(
    study,
    levels = c("Web-CDI sample", "U.S. Census Estimates")
  ))

white_hispanic_plot <-
  whites_vs_census %>%
  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 = "Race") +
  scale_fill_viridis_d() +
  coord_cartesian(ylim = c(0, .81))

white_hispanic_plot

ethnicity_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")
```

```
# +
#   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)
#   )

ethnicity_plot
```

```
#Maternal ed analysis on the full sample

maternal_ed_plot_df <-
  demographics_df %>%
  count(maternal_ed) %>%
  mutate(`Full Web-CDI sample to date` = prop.table(n)) %>%
  left_join(old_momed_numbers, by = "maternal_ed") %>%
  select(-n) %>%
  pivot_longer(
    cols = c(`Full Web-CDI sample to date`, `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"
    )
  ) %>%
  filter(!is.na(maternal_ed))

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 %>%
  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)

maternal_ed_plot

#this table can be printed out
maternal_ed_table <-
  maternal_ed_plot_df %>%
  filter(maternal_ed != "Not reported") %>%
  pivot_wider(names_from = "study", values_from = "proportion") %>%
  mutate(
    `Current study proportions` = round(`Full Web-CDI sample to date`, digits = 4)
  ) %>%
  select(-`Full Web-CDI sample to date`)

prow <- cowplot::plot_grid(
  ethnicity_plot +
    theme(
      legend.position = "none",
      plot.margin = (margin(r = 2, l = 0)),
      axis.text = element_text(size = 12)
    ),
  maternal_ed_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_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))
```

‘

```
#More fine grained analyses of exclusions. Copied and pasted from all_norming_analysis.Rmd.

n_total_wg <- nrow(all_wg_raw %>% filter(completed == TRUE))
n_total_ws <- nrow(all_ws_raw %>% filter(completed == TRUE))
```

```
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 <-
  all_wg_raw %>%
  getCompletionInterval() %>%
  filter(repeat_num == "1")

wg_repeats_n <- n_total_wg - nrow(wg_minus_repeats)

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

ws_repeats_n <- n_total_ws - nrow(ws_minus_repeats)

repeat_admins <-
  c(
    "Not first administration",
    wg_repeats_n,
    percent(wg_repeats_n / n_total_wg, accuracy = 0.01),
    ws_repeats_n,
    percent(ws_repeats_n / n_total_ws, 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, accuracy = 0.01),
    ws_premie_n,
    percent(ws_premie_n / n_total_ws, 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, accuracy = 0.01),
    ws_multiling_n,
    percent(ws_multiling_n / n_total_ws, 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, accuracy = 0.01),
    ws_health_n,
    percent(ws_health_n / n_total_ws, 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, accuracy = 0.01),
    ws_age_n,
    percent(ws_age_n / n_total_ws, 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, accuracy = 0.01),
```

```
    ws_fake_n,
    percent(ws_fake_n / n_total_ws, 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, accuracy = .01),
    ws_wordbugs_n,
    percent(ws_wordbugs_n / n_total_ws, accuracy = .01)
  )

names(wordbugs) <- excl_col_names

#calculate total amount of WG exclusions
total_wg_exclusions <-
  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 <-
  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,
    percent(total_wg_exclusions / n_total_wg),
    total_ws_exclusions,
    percent(total_ws_exclusions / n_total_ws)
  )

names(totals) <- excl_col_names

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

knitr::kable(exclusion_tbl_full)
```

Exclusion	WG exclusions	% of full WG sample excluded	WS exclusions	% of full WS sample excluded
Not first administration	163	5.69%	444	12.46%
Premature or low birthweight	37	1.29%	67	1.88%
Multilingual exposure	449	15.66%	492	13.80%
Illnesses/Vision/Hearing	191	6.66%	203	5.70%
Out of age range	88	3.07%	199	5.58%
Completed survey too quickly	319	11.13%	256	7.18%
System error in word tabulation	1	0.03%	4	0.11%
Total exclusions	1248	44%	1665	47%

```
save(
  exclusion_tbl_full,
  file = path(
    project_root,
    "data",
    "exclusion_tables",
    "full_sample_norming_exclusions",
    ext = "RData"
  )
)
```