function_to_link_partner_data.R

ecantrell

2024-06-01

```
# Emily Cantrell
# Exploration of fertility intentions questions from LISS
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# This is a draft of the code for linking partner's data with the primary respondent's data,
# in order to merge in the partner's fertility intention data.
# The actual merge happens in submission.R, but I am posting this code because it includes
# quality assurance checks and plots.

# Read in the data
train_full <- read.csv("/Users/ecantrell/Documents/PreFer\ 2024/prefer_data/training_data/PreFer_train_
outcome <- read.csv("/Users/ecantrell/Documents/PreFer\ 2024/prefer_data/training_data/PreFer_train_out
household_full <- read.csv("/Users/ecantrell/Documents/PreFer\ 2024/prefer_data/other_data/PreFer_train_
supplementary_full <- read.csv("/Users/ecantrell/Documents/PreFer\ 2024/prefer_data/other_data/PreFer_t

#### IF A PERSON HAS A PARTNER IN THE DATA, MERGE THE PARTNER'S DATA INTO THEIR ROW ####

# Select a few features of interest, plus features that will help us double-check that the merged in pe
train <- train_full %>%
  select(
    "nomem_encr",
    # Expected kids reported in 2020
    "cf20m128", "cf20m129", "cf20m130",
    # Expected kids reported in 2019
    "cf19l128", "cf19l129", "cf19l130",
    # Demographics
    "gender_bg", "birthyear_bg",
    # Do you live with partner?
    "cf08a025", "cf09b025", "cf10c025", "cf11d025", "cf12e025", "cf13f025", "cf14g025", "cf15h025", "cf
    # Partner's birth year
    "cf08a026", "cf09b026", "cf10c026", "cf11d026", "cf12e026", "cf13f026", "cf14g026", "cf15h026", "cf
    # Partner's gender
```

```
    "cf08a032", "cf09b032", "cf10c032", "cf11d032", "cf12e032", "cf13f032", "cf14g032", "cf15h032", "cf
    ) %>%
  # Collect the most recent response to whether they live with a partner in a single variable
  mutate(live_with_partner = coalesce(cf20m025, cf19l025, cf18k025, cf17j025, cf16i025, cf15h025, cf14g
  # Collect the most recently reported partner birth year in a single variable
  mutate(partner_birth_year = coalesce(cf20m026, cf19l026, cf18k026, cf17j026, cf16i026, cf15h026, cf14
  # Collect the most recent indicator of partner's gender in a single variable
  mutate(partner_gender = coalesce(cf20m032, cf19l032, cf18k032, cf17j032, cf16i032, cf15h032, cf14g032
  # Remove raw data that was used in the coalesced variables
  select(-c("cf08a025", "cf09b025", "cf10c025", "cf11d025", "cf12e025", "cf13f025", "cf14g025", "cf15h0
            "cf08a026", "cf09b026", "cf10c026", "cf11d026", "cf12e026", "cf13f026", "cf14g026", "cf15h0
            "cf08a032", "cf09b032", "cf10c032", "cf11d032", "cf12e032", "cf13f032", "cf14g032", "cf15h0

# Select the same features from the supplementary data
# Note: the supplementary data is people outside the age range 18-45 during the outcome time window
supplementary <- supplementary_full %>%
  select(
    "nomem_encr",
    # Expected kids reported in 2020
    "cf20m128", "cf20m129", "cf20m130",
    # Expected kids reported in 2019
    "cf19l128", "cf19l129", "cf19l130",
    # Demographics
    "gender_bg", "birthyear_bg",
    # Do you live with partner?
    "cf08a025", "cf09b025", "cf10c025", "cf11d025", "cf12e025", "cf13f025", "cf14g025", "cf15h025", "cf
    # Partner's birth year
    "cf08a026", "cf09b026", "cf10c026", "cf11d026", "cf12e026", "cf13f026", "cf14g026", "cf15h026", "cf
    # Partner's gender
    "cf08a032", "cf09b032", "cf10c032", "cf11d032", "cf12e032", "cf13f032", "cf14g032", "cf15h032", "cf
  ) %>%
  # Collect the most recent response to whether they live with a partner in a single variable
  mutate(live_with_partner = coalesce(cf20m025, cf19l025, cf18k025, cf17j025, cf16i025, cf15h025, cf14g
  # Collect the most recently reported partner birth year in a single variable
  mutate(partner_birth_year = coalesce(cf20m026, cf19l026, cf18k026, cf17j026, cf16i026, cf15h026, cf14
  # Collect the most recent indicator of partner's gender in a single variable
  mutate(partner_gender = coalesce(cf20m032, cf19l032, cf18k032, cf17j032, cf16i032, cf15h032, cf14g032
  # Remove raw data that was used in the coalesced variables
  select(-c("cf08a025", "cf09b025", "cf10c025", "cf11d025", "cf12e025", "cf13f025", "cf14g025", "cf15h0
            "cf08a026", "cf09b026", "cf10c026", "cf11d026", "cf12e026", "cf13f026", "cf14g026", "cf15h0
            "cf08a032", "cf09b032", "cf10c032", "cf11d032", "cf12e032", "cf13f032", "cf14g032", "cf15h0

# For each person, filter to only the most recent wave in which they appeared
household <- household_full %>%
  group_by(nomem_encr) %>%
  arrange(desc(wave)) %>%
  slice_head() %>%
  ungroup() %>%
  select(nomem_encr, nohouse_encr, positie)

# Merge household ID and position data with training data
train <- left_join(train, household)
```

```
## Joining with `by = join_by(nomem_encr)`
```

```r
supplementary <- left_join(supplementary, household)
```

```
## Joining with `by = join_by(nomem_encr)`
```

```r
# Create a copy of "train" merged with "supplementary" to represent the partner
# Some partners may be in the supplementary data because they are outside the 18-45 age range
train_partner <- rbind.data.frame(train, supplementary) %>%
  rename_with(~ paste0(., "_PartnerSurvey"), -nohouse_encr)

# Merge train with train_partner
train_linked_with_partner <- train %>%
  left_join(train_partner, by = "nohouse_encr", relationship = "many-to-many") %>%
  filter(nomem_encr != nomem_encr_PartnerSurvey,
         # Filter to only people who are head of household, wedded partner, or unwedded partner in most
         positie %in% c(1,2,3),
         positie_PartnerSurvey %in% c(1,2,3),
         # Filter to people from households where at least one supposed partner reported living togethe
         ((live_with_partner == 1) | (live_with_partner_PartnerSurvey ==1)),
         # Remove rows where reported birthyears are mismatched
         (partner_birth_year == birthyear_bg_PartnerSurvey | is.na(partner_birth_year) | is.na(birthyea
         (partner_birth_year_PartnerSurvey == birthyear_bg | is.na(partner_birth_year_PartnerSurvey) |
         # Remove rows where reported genders are mismatched
         (partner_gender == gender_bg_PartnerSurvey | is.na(partner_gender) | is.na(gender_bg_PartnerSu
         (partner_gender_PartnerSurvey == gender_bg | is.na(partner_gender_PartnerSurvey) | is.na(gende
         )

# Filter to only people with a non-missing outcome
train_linked_with_partner <- train_linked_with_partner %>%
  left_join(outcome) %>%
  filter(!is.na(new_child))
```

```
## Joining with `by = join_by(nomem_encr)`
```

```r
#### QUALITY ASSURANCE CHECKS ####

# Manually look at the responses
train_linked_with_partner %>%
  select(live_with_partner, live_with_partner_PartnerSurvey,
         partner_birth_year, birthyear_bg_PartnerSurvey,
         partner_birth_year_PartnerSurvey, birthyear_bg,
         partner_gender, gender_bg_PartnerSurvey,
         partner_gender_PartnerSurvey, gender_bg) %>%
  head()
```

```
##   live_with_partner live_with_partner_PartnerSurvey partner_birth_year
## 1                 1                               1              1990
## 2                 1                               1              1982
## 3                 1                               1              1982
## 4                 1                               1              1973
## 5                 1                               1              1968
## 6                 1                               1              1987
##   birthyear_bg_PartnerSurvey partner_birth_year_PartnerSurvey birthyear_bg
## 1                       1990                             1990         1990
## 2                       1982                             1989         1989
## 3                       1982                             1984         1984
```

```
## 4                       1973                             1979         1979
## 5                       1968                             1976         1976
## 6                       1987                             1989         1989
##   partner_gender gender_bg_PartnerSurvey partner_gender_PartnerSurvey gender_bg
## 1              2                       2                            1         1
## 2              1                       1                            2         2
## 3              2                       2                            1         1
## 4              1                       1                            2         2
## 5              1                       1                            2         2
## 6              1                       1                            2         2
```

```
# Some households should appear once, and some should appear twice
train_linked_with_partner %>%
  count(nohouse_encr) %>%
  group_by(n) %>% # Count the number of times a household appears
  count() # counts the number of households that appear a given number of times
```

```
## Storing counts in `nn`, as `n` already present in input
## i Use `name = "new_name"` to pick a new name.
```

```
## # A tibble: 2 x 2
## # Groups:   n [2]
##       n    nn
##   <int> <int>
## 1     1   124
## 2     2   133
```

```
# Check that the proportion of same-sex and different-sex couples is roughly aligned
# with the expected proportion based on population rates for same-sex households
train_linked_with_partner %>%
  group_by(nohouse_encr) %>%
  slice_head() %>%
  ungroup() %>%
  mutate(same_sex = gender_bg == gender_bg_PartnerSurvey) %>%
  count(same_sex) %>%
  mutate(proportion = n / sum(n))
```

```
## # A tibble: 2 x 3
##   same_sex     n proportion
##   <lgl>    <int>      <dbl>
## 1 FALSE      252     0.981
## 2 TRUE         5     0.0195
```

```
# Check that partners are usually of similar ages
train_linked_with_partner %>%
  group_by(nohouse_encr) %>%
  slice_head() %>%
  ungroup() %>%
  mutate(age_gap = birthyear_bg - birthyear_bg_PartnerSurvey) %>%
  count(age_gap) %>%
  mutate(proportion = n / sum(n)) %>%
  print(n = "Inf")
```

```
## # A tibble: 26 x 3
##    age_gap     n proportion
##      <int> <int>      <dbl>
##  1     -11     1    0.00389
```

```
##  2     -10     1    0.00389
##  3      -7     3    0.0117
##  4      -6     2    0.00778
##  5      -5     8    0.0311
##  6      -4     5    0.0195
##  7      -3    18    0.0700
##  8      -2    18    0.0700
##  9      -1    32    0.125
## 10       0    37    0.144
## 11       1    23    0.0895
## 12       2    18    0.0700
## 13       3    21    0.0817
## 14       4    17    0.0661
## 15       5    12    0.0467
## 16       6    11    0.0428
## 17       7     6    0.0233
## 18       8     6    0.0233
## 19       9     4    0.0156
## 20      10     4    0.0156
## 21      11     3    0.0117
## 22      12     2    0.00778
## 23      13     2    0.00778
## 24      15     1    0.00389
## 25      21     1    0.00389
## 26      25     1    0.00389
```

```
# Check that all partners are at least 18
train_linked_with_partner %>%
  filter(birthyear_bg_PartnerSurvey > 2002)
```

```
##  [1] nomem_encr                      cf20m128
##  [3] cf20m129                        cf20m130
##  [5] cf19l128                        cf19l129
##  [7] cf19l130                        gender_bg
##  [9] birthyear_bg                    live_with_partner
## [11] partner_birth_year              partner_gender
## [13] nohouse_encr                    positie
## [15] nomem_encr_PartnerSurvey        cf20m128_PartnerSurvey
## [17] cf20m129_PartnerSurvey          cf20m130_PartnerSurvey
## [19] cf19l128_PartnerSurvey          cf19l129_PartnerSurvey
## [21] cf19l130_PartnerSurvey          gender_bg_PartnerSurvey
## [23] birthyear_bg_PartnerSurvey      live_with_partner_PartnerSurvey
## [25] partner_birth_year_PartnerSurvey partner_gender_PartnerSurvey
## [27] positie_PartnerSurvey           new_child
## <0 rows> (or 0-length row.names)
```

```
#### MERGE PARTNERED PEOPLE AND NON-PARTNERED PEOPLE BACK INTO SAME DATAFRAME ####
full_train_linked_with_partner <- left_join(train, train_linked_with_partner)
```

```
## Joining with `by = join_by(nomem_encr, cf20m128, cf20m129, cf20m130, cf19l128,
## cf19l129, cf19l130, gender_bg, birthyear_bg, live_with_partner,
## partner_birth_year, partner_gender, nohouse_encr, positie)`
```

```
#### EXAMINE PATTERNS ####

# HOW MANY PEOPLE HAVE FERTILITY QUESTION RESPONSES FROM THE PARTNER?
```

```r
train_linked_with_partner %>%
  group_by(is.na(cf20m128_PartnerSurvey)) %>%
  count()
```

```
## # A tibble: 2 x 2
## # Groups:   is.na(cf20m128_PartnerSurvey) [2]
##   `is.na(cf20m128_PartnerSurvey)`     n
##   <lgl>                           <int>
## 1 FALSE                             298
## 2 TRUE                               92
```

```r
# We get partner data on 2020 fertility intentions for 298 people (among those with non-missing outcome
# In total, 987 people have non-missing outcome data.
# That means we have data on partner's 2020 fertility intentions for 30% of people.

# HOW MANY PEOPLE HAVING MISSING DATA FOR FERTILITY QUESTIONS, BUT THEIR PARTNER ANSWERED IT?
train_linked_with_partner %>%
  group_by(is.na(cf20m128), is.na(cf20m128_PartnerSurvey)) %>%
  count()
```

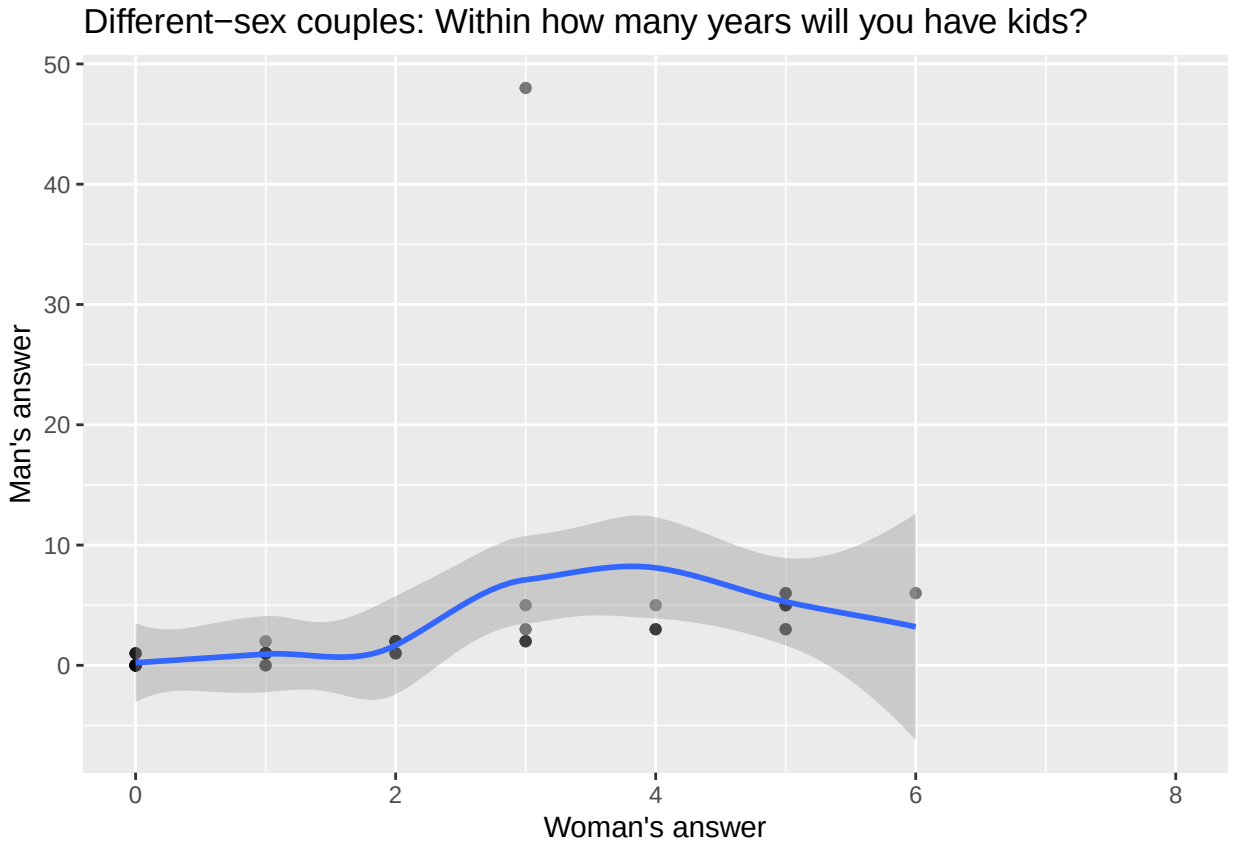
```
## # A tibble: 4 x 3
## # Groups:   is.na(cf20m128), is.na(cf20m128_PartnerSurvey) [4]
##   `is.na(cf20m128)` `is.na(cf20m128_PartnerSurvey)`     n
##   <lgl>             <lgl>                           <int>
## 1 FALSE             FALSE                             276
## 2 FALSE             TRUE                               76
## 3 TRUE              FALSE                              22
## 4 TRUE              TRUE                               16
```

```r
# There are 22 people for whom the ego didn't answer 2020 fertility questions, but partner did answer 2

# HOW WELL DO PARTNERS' ANSWERS ALIGN?

# Correction to data: Change the response "2025" to "5" for "within how many years will you have kids?"
train_linked_with_partner <- train_linked_with_partner %>%
  mutate(cf20m130 = ifelse(cf20m130 == 2025, 5, cf20m130),
         cf20m130_PartnerSurvey = ifelse(cf20m130_PartnerSurvey == 2025, 5, cf20m130_PartnerSurvey))

# Plot for different-sex couples
train_linked_with_partner %>%
  filter(gender_bg == 2, # Filter to different-sex couples, with woman as the primary person
         gender_bg_PartnerSurvey == 1) %>%
  ggplot(aes(x = cf20m130, y = cf20m130_PartnerSurvey)) +
  geom_point(alpha = 0.5) +
  geom_smooth() +
  xlab("Woman's answer") +
  ylab("Man's answer") +
  ggtitle("Different-sex couples: Within how many years will you have kids?")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## Warning: Removed 155 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 155 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

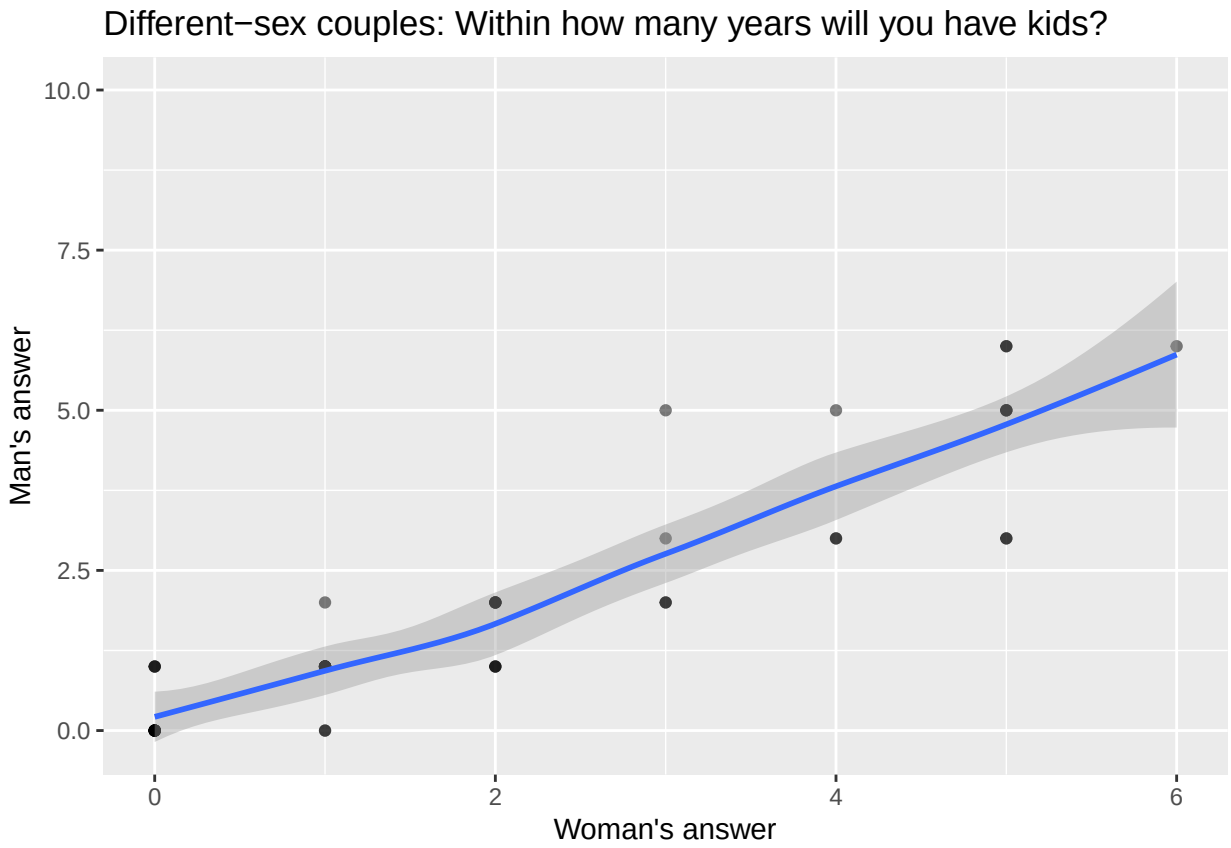



```
# Same plot as above, but remove the outlier so we can see the other points better
train_linked_with_partner %>%
  filter(gender_bg == 2, # Filter to different-sex couples, with woman as the primary person
         gender_bg_PartnerSurvey == 1) %>%
  filter(cf20m130_PartnerSurvey < 40) %>%
  ggplot(aes(x = cf20m130, y = cf20m130_PartnerSurvey)) +
  geom_point(alpha = 0.5) +
  geom_smooth() +
  xlab("Woman's answer") +
  ylab("Man's answer") +
  ggtitle("Different-sex couples: Within how many years will you have kids?")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## Warning: Removed 5 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

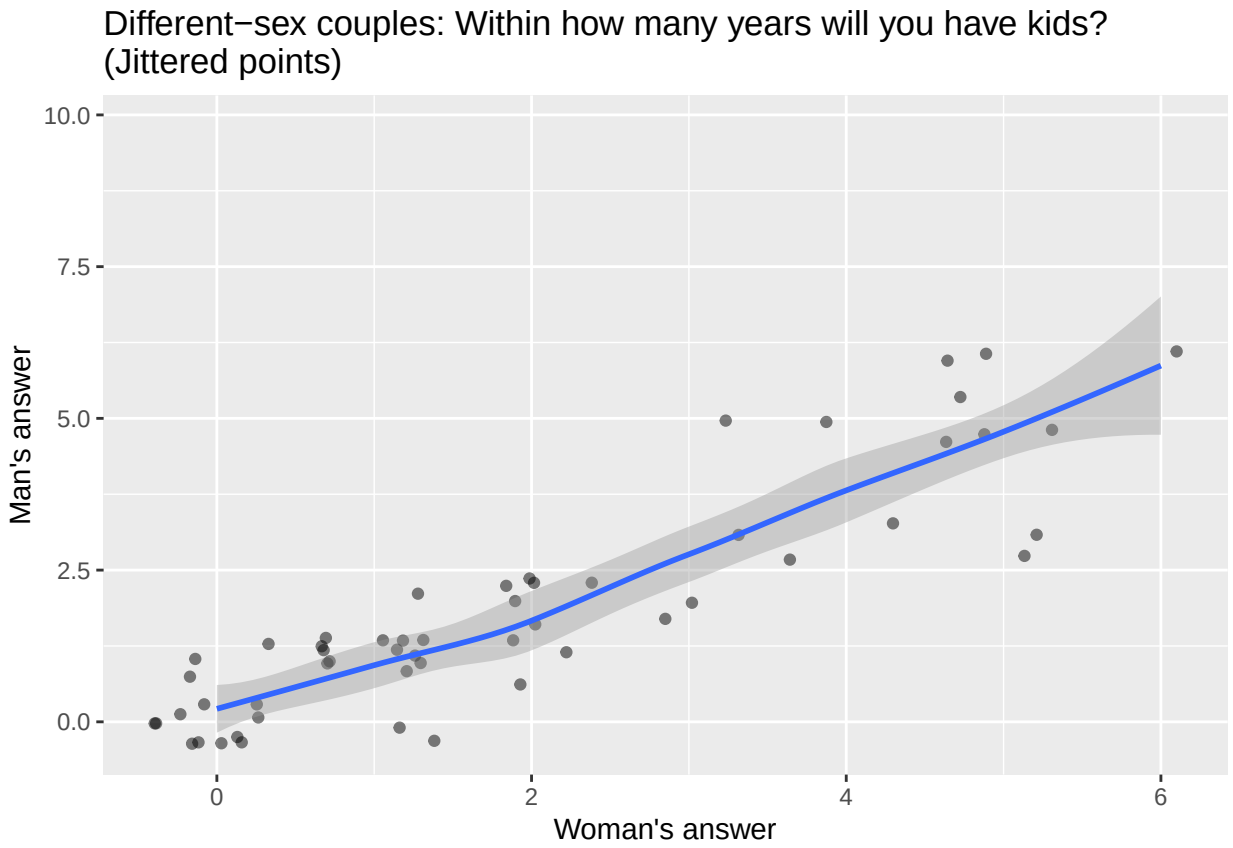
```
## Warning: Removed 5 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
# Same plot as above, with outlier removed, jittered
train_linked_with_partner %>%
  filter(gender_bg == 2, # Filter to different-sex couples, with woman as the primary person
         gender_bg_PartnerSurvey == 1) %>%
  filter(cf20m130_PartnerSurvey < 40) %>%
  ggplot(aes(x = cf20m130, y = cf20m130_PartnerSurvey)) +
  geom_jitter(alpha = 0.5) +
  geom_smooth() +
  xlab("Woman's answer") +
  ylab("Man's answer") +
  ggtitle("Different-sex couples: Within how many years will you have kids?
(Jittered points)")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## Warning: Removed 5 rows containing non-finite outside the scale range (`stat_smooth()`).
## Removed 5 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
# Almost all data on fertility intentions is missing among the few same-sex couples,
# so I didn't make a plot for them.
```