IMT 573: Exploring the Fragile Families Dataset

Kiuk Paeng

Last Updated: October, 2022

Introduction

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# Load prepared data
# You can see the details in the RMarkdown document
load("data/ff_eda.Rdata")
```

Dataset Basics

When dealing with any new data, you first want to get a sense of the size of the data and its format. Let's investigate by calling the `dim` function in R:

```
# Report dimension of data
dim(df_eda)
```

```
## [1] 4898    8
```

This output of this function is the dimension of the data frame containing the data, in other words the number of number of observations (rows) and variables (columns) – assuming the data is in a tidy format. The rows represent the families, with a total of 4898. Here we have selected 8 variables to explore. Fun fact: there are over 17000 columns, meaning over 17000 variables collected/recorded in some of the wave of the Fragile Families data!

The `summary()` function is also a great place to start when exploring datasets in R. We see some variable names, along with some information about the data types that R believes best suited to the variables R does its best, but is not always right in determining data types – we will discuss this when we talk about data tidying and cleaning. As you can see R prints a reasonable summary for each variable, depending on the variable data type – some are stored as numbers and others as character strings/factors.

```
# Look at a summary of the data
summary(df_eda)
```

```
##      idnum          father_age      mother_age    father_bio_children
##  0001   :   1   Min.   :15.00   Min.   :15.00   Min.   : 1.000
##  0002   :   1   1st Qu.:22.00   1st Qu.:20.00   1st Qu.: 1.000
```

```
##  0003   :   1   Median :27.00   Median :24.00   Median : 1.000
##  0004   :   1   Mean   :27.92   Mean   :25.28   Mean   : 1.913
##  0005   :   1   3rd Qu.:32.00   3rd Qu.:29.00   3rd Qu.: 2.000
##  0006   :   1   Max.   :53.00   Max.   :43.00   Max.   :17.000
##  (Other):4892   NA's   :1068    NA's   :4       NA's   :2703
##  father_married_mother father_money_disagree mother_money_disagree
##  NO  :2754             NEVER    : 153        NEVER    : 335
##  YES :1076             OFTEN    :  43        OFTEN    : 132
##  NA's:1068             SOMETIMES:  65        SOMETIMES: 158
##                        NA's     :4637        NA's     :4273
##
##
##
##  mother_own_rent
##  Owned :1664
##  Rented:3196
##  NA's  :  38
##
##
##
##
```

I have renamed these variables to better represent the concepts they measure, but to do this I had to read the data description provided by the study investigators. Public data is often accompanied by a *codebook*, or a dictionary that describes the variables and lists their names. The Fragile Family Data codebooks give the column names and what they corresponds to in terms of the concepts measured.

You might go through the codebook prior to working on your own exploratory analysis if you choose to complete the extra credit assignment. Engaging in this process allows us start to get to know the data scheme and the kinds of data we have. It may also get you to start thinking about interesting data science questions. This data contains multiple waves of observation, but in this example we will consider only data from the baseline survey, i.e. Wave 1.

Originally, each of the variables was treated as type `factor`, but I converted them to numeric to better capture their concepts, e.g. age.

You can see that some data is missing or not observed for each of the families. We recoded missing data as `NA` in R. We will discuss these types of decisions in Module 4.

Parents' Age

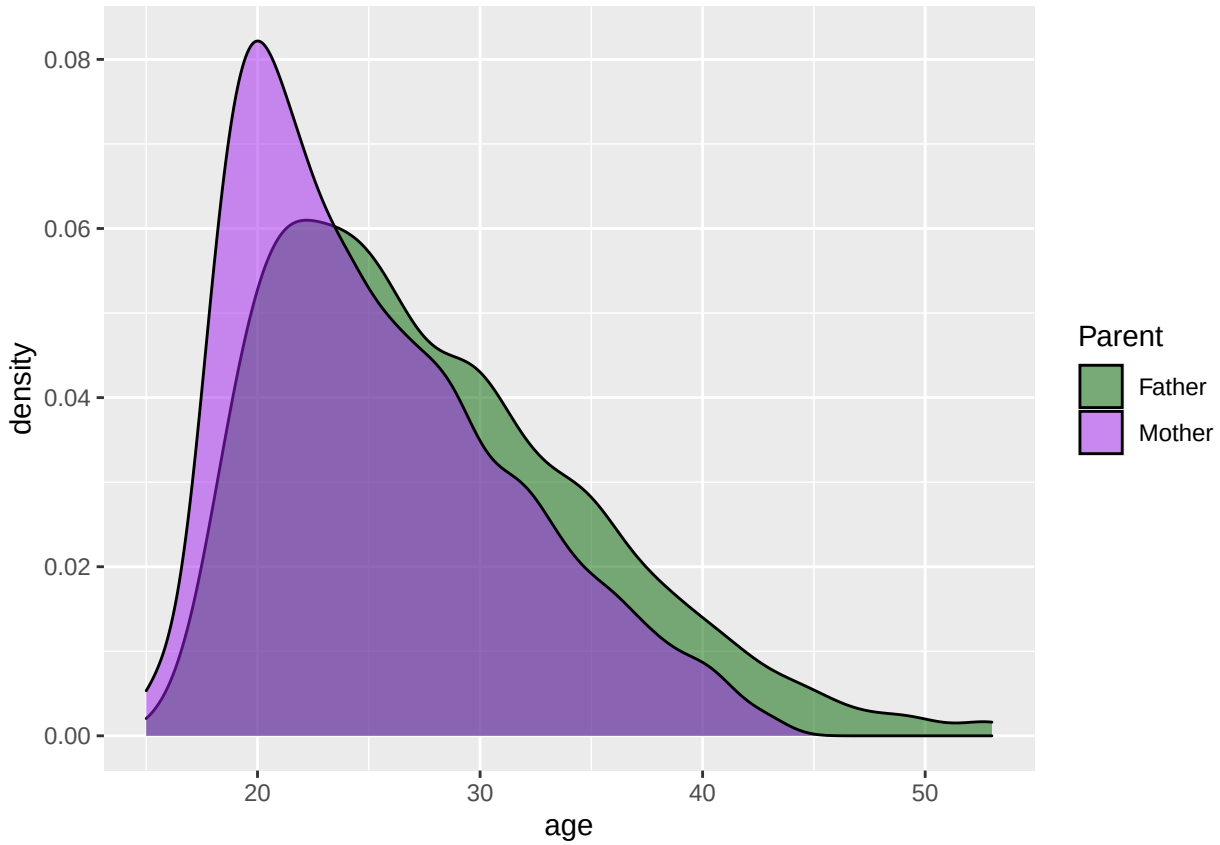
Let's illustrate a typical exploratory approach to this data by starting with questions about parents' age. We want to understand variability, missingness, and correlations.

Okay, so let's take a look at these variables and produce some visualizations that communicate variability.

```
age_data <- df_eda %>% pivot_longer(cols = c("mother_age","father_age"), names_to="parent",values_to="ag

ggplot(age_data, aes(x=age, fill = parent)) +
  geom_density(alpha=0.5) +
  scale_fill_manual(values = c("darkgreen","purple"), labels = c("Father","Mother"), name = "Parent")
```

```
## Warning: Removed 1072 rows containing non-finite values (stat_density).
```

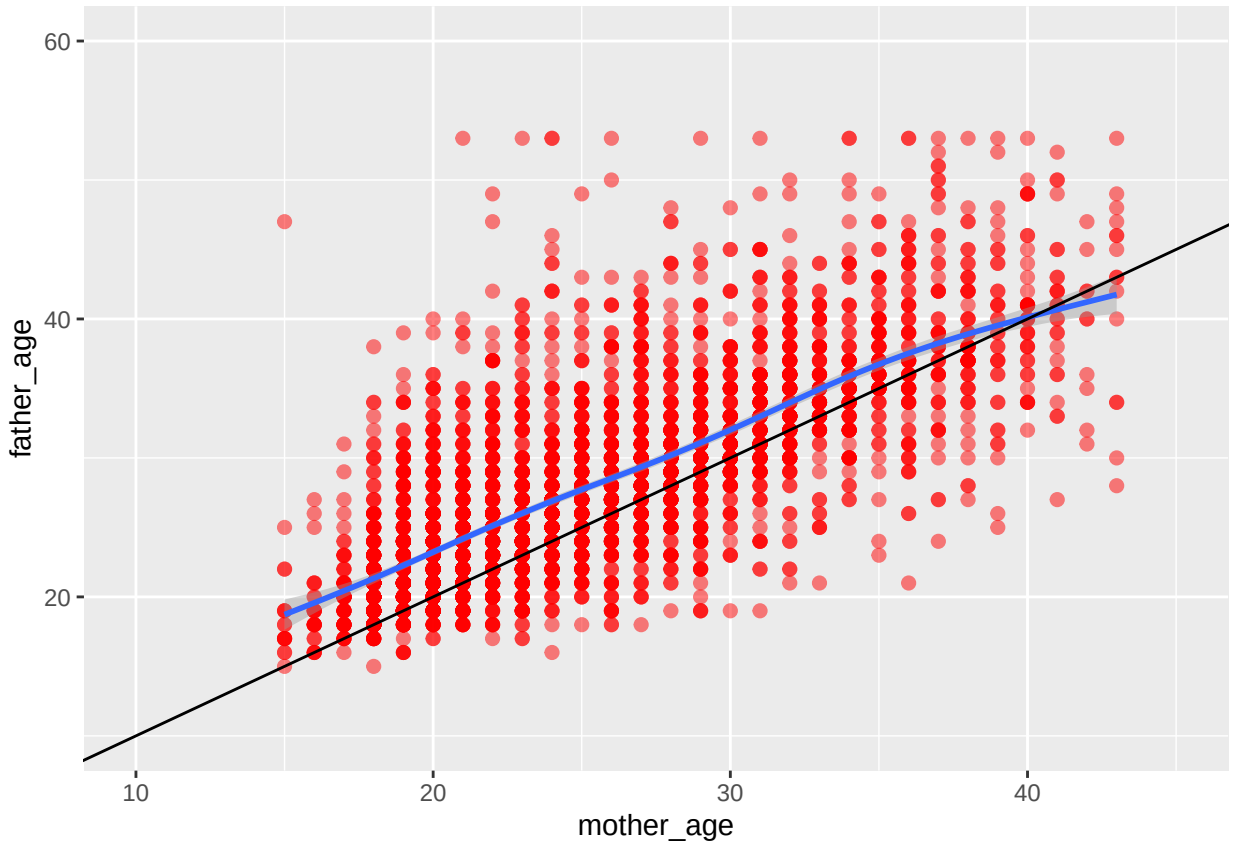


We might be interested in the relationship between these two variables. Together these figures suggest that as mother's age increases, father's age increases, but it also suggest that mothers are younger than fathers on average.

```
ggplot(df_eda, aes(x = mother_age, y = father_age)) +
  geom_point(size = 2, alpha = 0.5, colour = "red") + geom_smooth() +
  geom_abline() + xlim(10,45) + ylim(10,60)
```

```
## Warning: Removed 1069 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1069 rows containing missing values (geom_point).
```



Money Disagreements

One of the variables reported here is about disagreements regarding money. First, let's take a look at these variables, and then again ask about the association between them.

First we consider the varibles. We see each is reported as a categorical variables with three responses: `Never`, `Sometimes`, and `Often`. There is a lot of missing data in these variables as well.

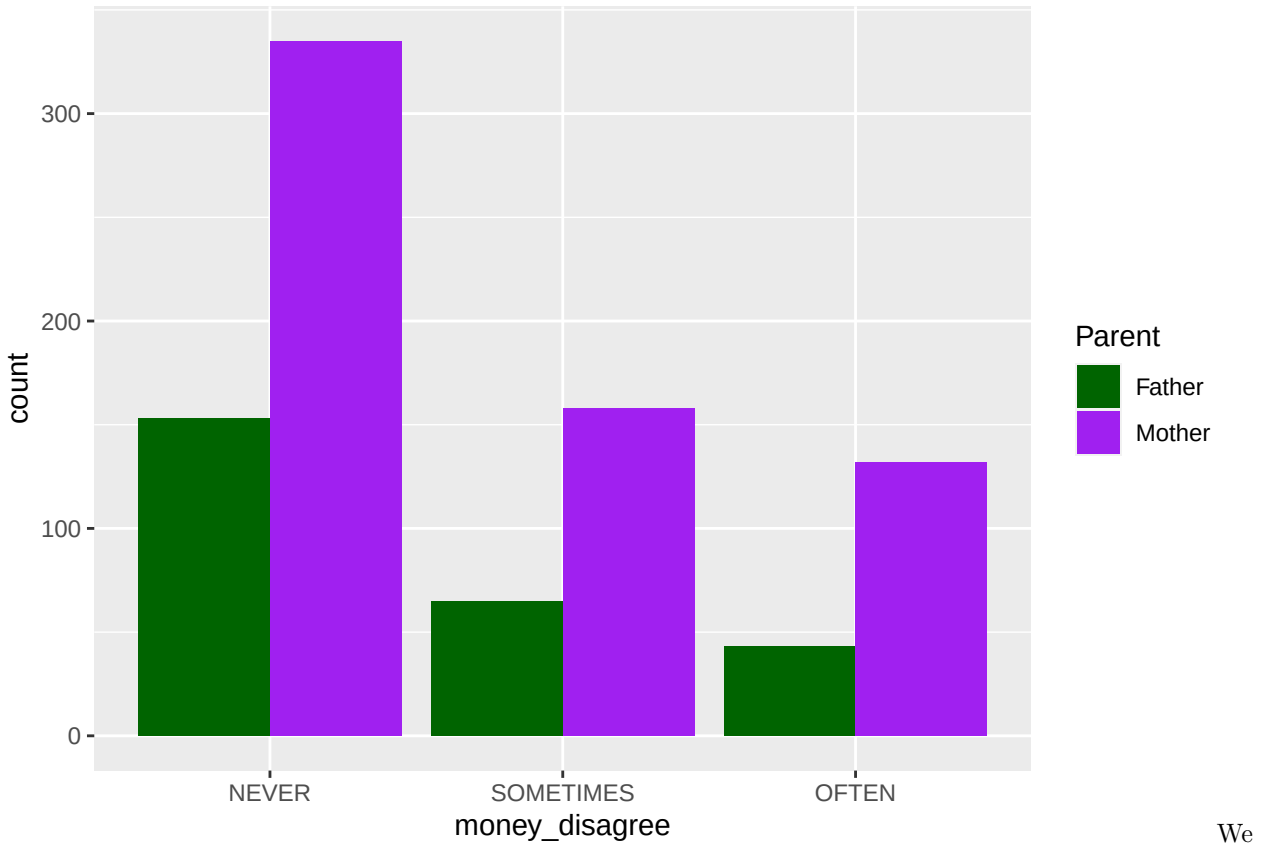
Let's look at the distributions across parents.

```
money_data <- df_eda %>% pivot_longer(cols = c("mother_money_disagree","father_money_disagree"), names_
money_data$money_disagree <- factor(money_data$money_disagree,
                                    levels = c("NEVER", "SOMETIMES", "OFTEN"))

table(money_data$money_disagree)
```

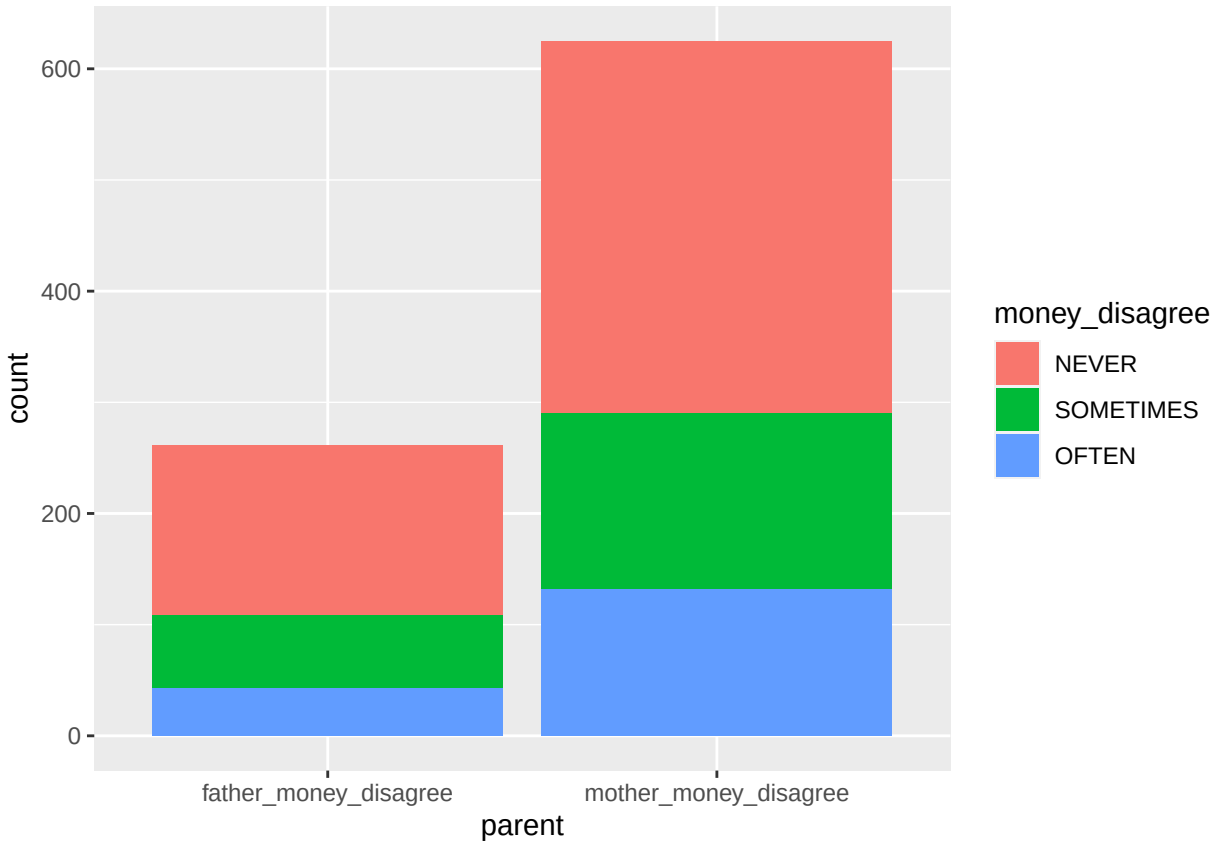
```
##
##     NEVER SOMETIMES     OFTEN
##       488       223       175
```

```
ggplot(subset(money_data, !is.na(money_disagree)), aes(x=money_disagree, fill = parent)) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = c("darkgreen","purple"), labels = c("Father","Mother"), name = "Parent")
```



We see that mothers seen higher, but this might be misleading based on the total observed values. We might instead want to look at this as a proportion.

```
ggplot(subset(money_data, !is.na(money_disagree)), aes(x=parent, fill = money_disagree)) +
  geom_bar()
```



This show us more clearly that we have differences in missingness across parents. But really we might be more interested in how parents agree/disagree in these reports.

```
df_eda$mother_money_disagree <- factor(df_eda$mother_money_disagree,
                                       levels = c("NEVER", "SOMETIMES", "OFTEN"))
df_eda$father_money_disagree <- factor(df_eda$father_money_disagree,
                                       levels = c("NEVER", "SOMETIMES", "OFTEN"))
table(df_eda$mother_money_disagree, df_eda$father_money_disagree)
```

```
##
##              NEVER SOMETIMES OFTEN
##   NEVER         48        21     8
##   SOMETIMES     24        11     6
##   OFTEN         12         7     8
```

Here we see that disagreements are frequent, with both mothers reporting higher frequencies on average.

There is so much more we could do with this data. For one examples, we that the team at Princeton did below.

The Fragile Families Challenge

The Fragile Families Data was used in a unique, data-science focused challenge to understand just how *predictable* life trajectories are in practice. In this organized challenge, hundreds of researchers attempted to predict six life outcomes:

(1) child grade point average (GPA)
(2) child grit
(3) household eviction
(4) household material hardship

(5) caregiver layoff
(6) caregiver participation in job training

These researchers used machine-learning methods optimized for prediction, and they drew on the vast Fragile Families dataset. You can learn more about this challenge, as well as the work produced in the Introduction to the Special Collection on the Fragile Families Challenge, as well as the Special Collection itself.

A nice summary article is also available: Measuring the predictability of life outcomes with a scientific mass collaboration

Despite hundreds of researchers taking on this challenge, and very sophisticated data science methods being put to the test, **no one made very accurate predictions.**

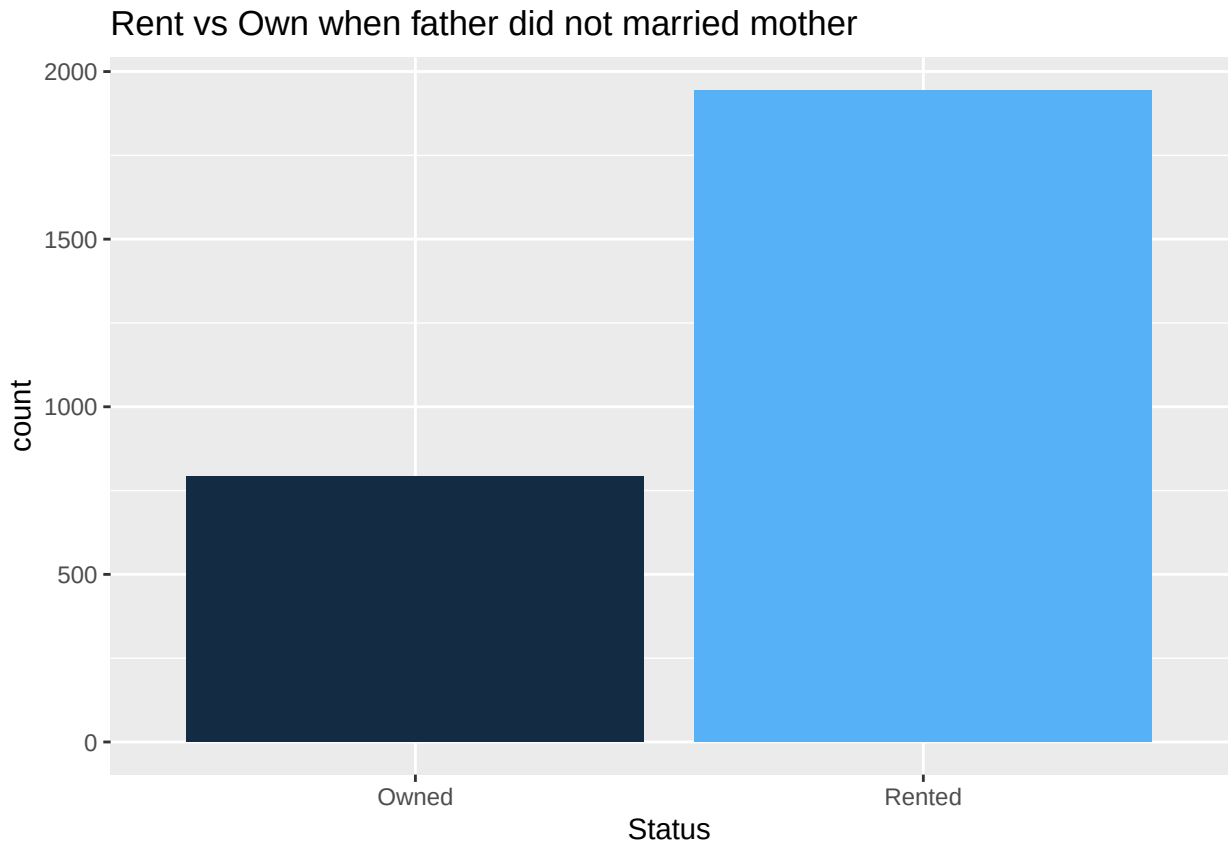
As the authors say, these results have significant implications: "For policymakers considering using predictive models in settings such as criminal justice and child-protective services, these results raise a number of concerns. Additionally, researchers must reconcile the idea that they understand life trajectories with the fact that none of the predictions were very accurate."

Rented vs Owned when father did not marry mother

Whether mother lived in a rented property or a owned property is a variable on the dataset. Question is if there is any differences between the number of rented and owned status based on the marriage (when father married mother or did not married mother). First, column chart below shows that more than rented status is twice more than owned when father did not married mother.

```
df_rent_own <- df_eda %>%
  select(mother_own_rent, mother_age, father_married_mother) %>%
  filter(father_married_mother == "NO") %>%
  group_by(mother_own_rent) %>%
  summarise(count = n()) %>%
  drop_na()


ggplot(data = df_rent_own, mapping = aes(x = mother_own_rent, y = count, fill = count)) +
 geom_col() +
  labs(title = "Rent vs Own when father did not married mother", x = "Status") +
  theme(legend.position = "none")
```

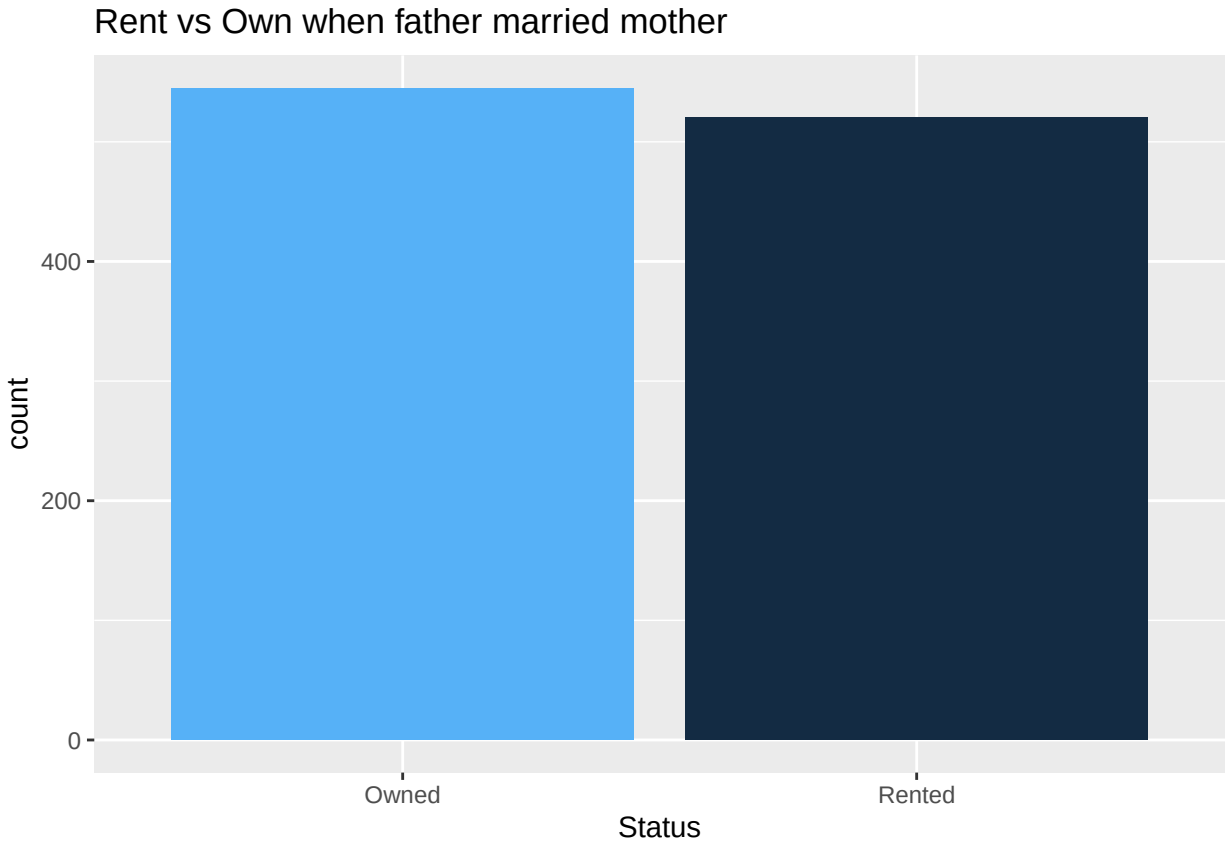


Rented vs Owned when father married mother

When father married mother, there is no signification difference between owned and rented status as below chart. From this, it might need further research to reach the conclusion that a family is likely to live in rented property when father did not married mother.

```
df_rent_own_yes <- df_eda %>%
  select(mother_own_rent, mother_age, father_married_mother) %>%
  filter(father_married_mother == "YES") %>%
  group_by(mother_own_rent) %>%
  summarise(count = n()) %>%
  drop_na()


ggplot(data = df_rent_own_yes, mapping = aes(x = mother_own_rent, y = count, fill = count)) +
 geom_col() +
  labs(title = "Rent vs Own when father married mother", x = "Status") +
  theme(legend.position = "none")
```

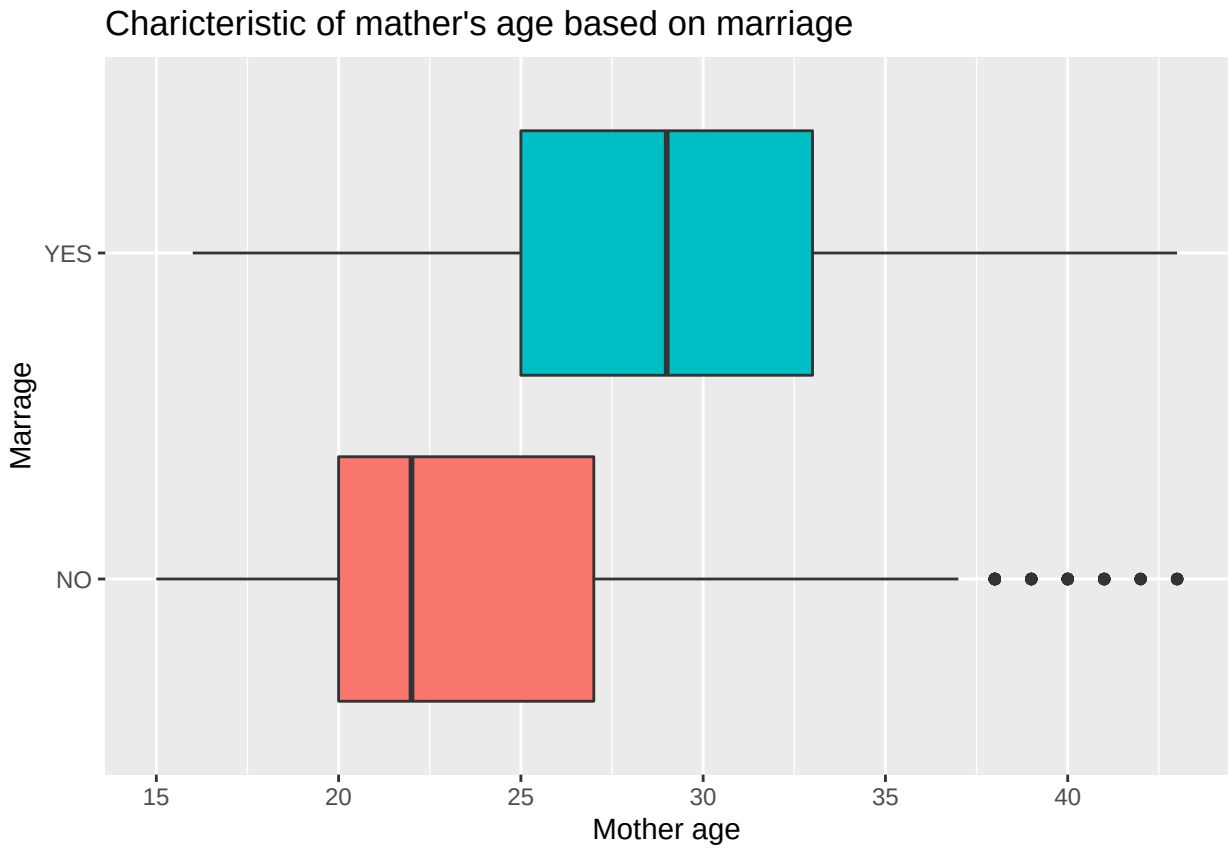


Marriage status and mother's age

Another analysis is on how mother's age is different depending on marriage status (whether or not father married mother). As shown on the box plot below, mothers with "no marriage" status are younger than the mothers with "yes marriage status". IQR is quite different based on the marriage status. When father did not marry mother, IRQ of mother's age is around 20 ~ 27. When father married mother, IRQ of mother's age is around 25 ~ 33.

```
age_marriage_df <- df_eda %>%
  select(mother_age, father_married_mother) %>%
  drop_na()

ggplot(data = age_marriage_df, mapping = aes(x = mother_age, y = father_married_mother, fill = father_ma
```

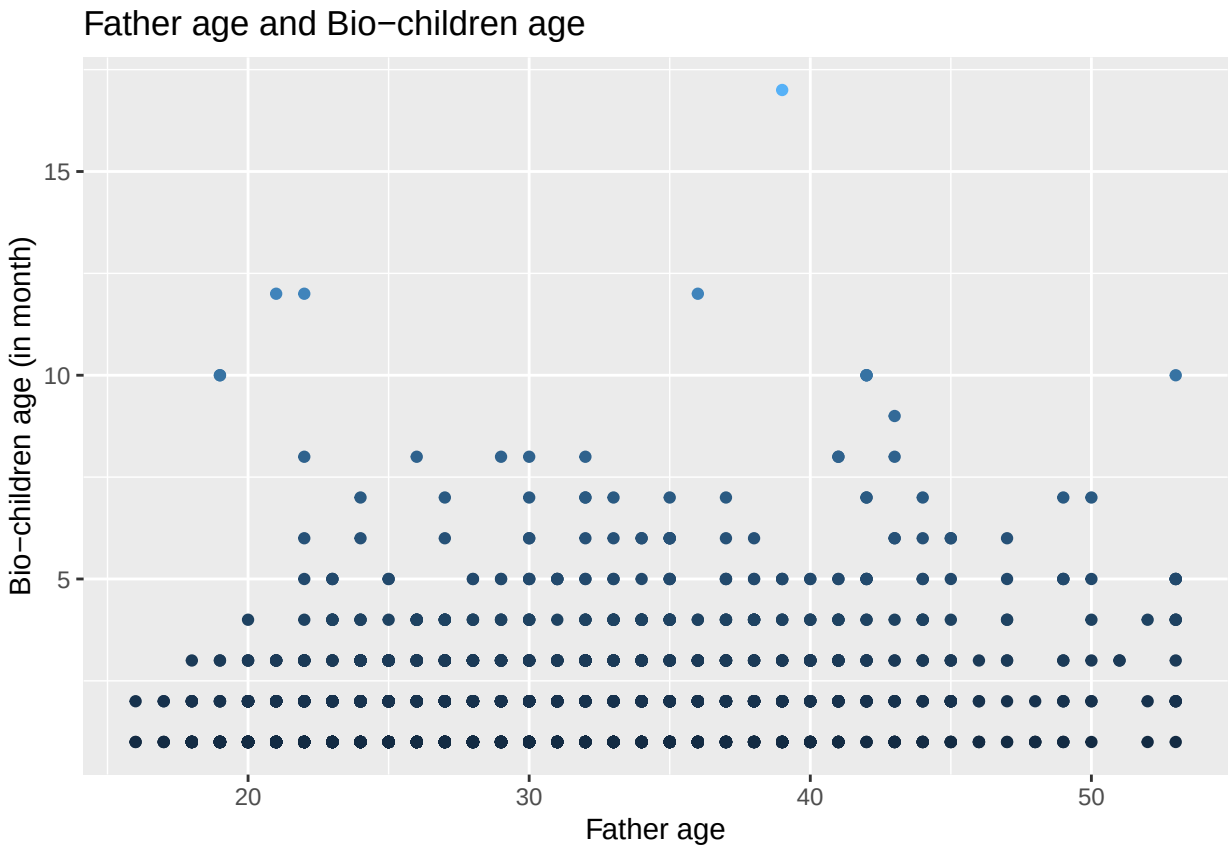


Father's age and bio-children's age

The scatter plot below shows the distribution of bio-child age per father's age. The distribution shows that the most age of bio child is around less than 10 months regardless of father's age. Whether the father is above 50 or under 20, children age is mostly under 10 month, excluding some outlines.

```
father_age_bio_child <- df_eda %>%
  select(father_age, father_bio_children) %>%
  drop_na()

ggplot(data = father_age_bio_child, mapping = aes(x = father_age, y = father_bio_children, col = father
  geom_point() +
  labs(title = "Father age and Bio-children age", x = "Father age", y = "Bio-children age (in month)") +
  theme(legend.position = "none")
```


Father age and Bio−children age
Bio−children age (in month)
15
10
5
20
30
40
50
Father age