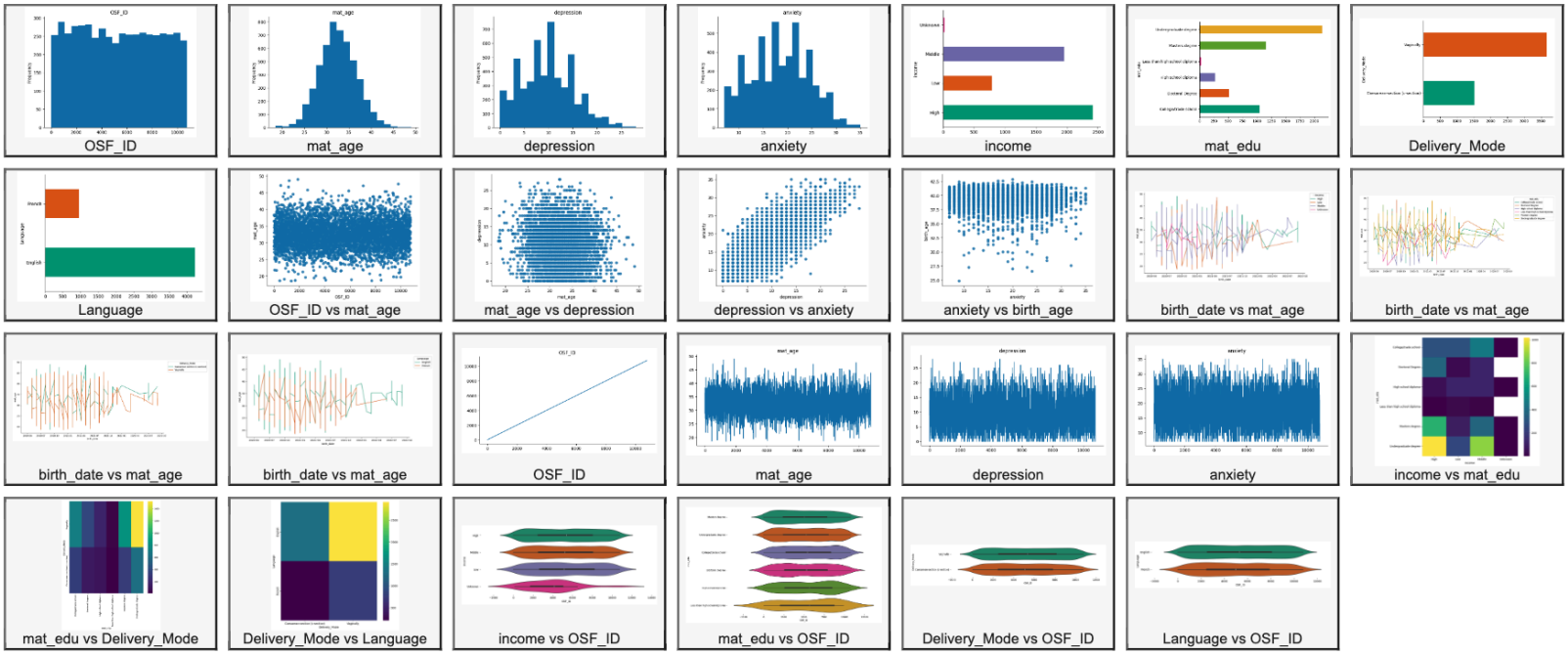
JUSCheckpoint #2: EDA

Names:
Sophia Ashraf
Dylan Oquendo
Karun Mokha
Jake Kondo
Ekrem Ersoz



Given the multitude of variables in our dataset related to each mothers background, mental health status, and births, we have explored our dataset for possible outliers, skews, and trends that should align with common birth knowledge. For maternal age, we have a normal distribution centered around are thirty, suggesting the women of this dataset are mostly in their 30's. Interestingly enough, the Edinburgh Postnatal Depression Scale Score is actually slightly right-skewed, meaning more respondents had lower depression scores, with fewer individuals reporting high levels of depression in their model's assessment. We saw a similar trend in the PROMIS Anxiety Score as well. The Gestational Age at Birth distribution also had peaks at 38-40 weeks, indicating that the majority of our births in the dataset occurred on term, according to the WHO, who we compared our birth related values to. The majority of babies had a weight of 2500 to 4000 grams, which is within the normal range also according to the WHO.

Furthermore, some other trends in our dataset amongst our variables were majority of our mothers were high or middle class, and majority spoke english with some french speakers (as our dataset comes from canadian population), and the absolute majority of our mothers received a college degree.The histograms for depression and anxiety scores hint at potential outliers in the higher score ranges, but these could just be cases of severe anxiety or depression, potentially people with overlapping conditions. In terms of our birth data weight and gestational age distributions seemed relatively clean, with no extreme outliers present. It will be important to keep in mind when we start running our models how we clean our data if we choose to reevaluate how we choose our null values or distributions, as we have some ranges to define relative to publicly known depression and birth knowledge (i.e how we compared values to the WHO).

Revised Question:

"How did the interplay between mental health and specific Covid-19-related events influence pregnancy outcomes, and what novel patterns emerge when comparing pre-pandemic, pandemic, and post-vaccine introduction phases?"

Data

Data overview

- Dataset #1
    - Mental health in the pregnancy during the COVID-19
    - https://www.kaggle.com/datasets/yeganehbavafa/mental-health-in-the-pregnancy-during-the-covid-19/data
    - Number of observations
    - Number of variables

Mental Health in the pregnancy during the COVID-19

Code:

```
import pandas as pd
import numpy as np


file_path = 'Pregnancy During the COVID-19 Pandemic.csv'
df = pd.read_csv(file_path)

# Display the first few rows to understand the structure of the dataset
df.head()
```

```
#drop OSF ID



df = df.dropna(subset=['PROMIS_Anxiety', 'Birth_Length', 'Birth_Weight',
'NICU_Stay', 'Edinburgh_Postnatal_Depression_Scale'])
#Drop any row with null value for the values we are most interested in
```

```
#drop language column

df.drop('Language', axis=1)
```

```
# Checking for missing values
missing_values = df.isnull().sum()

missing_values
```

```
df.rename(columns={'Maternal_Age': 'mat_age', 'Household_Income':
'income', 'Maternal_Education': 'mat_edu',
                   'Edinburgh_Postnatal_Depression_Scale': 'depression',
                   'PROMIS_Anxiety': 'anxiety', 'Gestational_Age_At_Birth':
'birth_age',
                   'Delivery_Date(converted to month and year)':
'birth_date'}, inplace=True)
df.head()
```

```
# Summary statistics for numerical columns
summary_statistics = df.describe()


summary_statistics, missing_values
```

```
standardize_income = {'$100,000 -$124,999': '$100,000-$124,999',
                      '$70,000-$99,999': '$70,000-$99,999',
                      '$125,000- $149,999': '$125,000-$149,999',
                      '$150,000 - $174,999': '$150,000-$174,999',
                      '$175,000- $199,999': '$175,000-$199,999',
                      '$20,000- $39,999': '$20,000-$39,999',
                      'Less than $20, 000': '<$20,000'}
```

```
#this function converts the income into numerical values and fixes the
def standardize_income(income):
   if pd.isna(income):
       # Return NaN as is, you can also choose to fill it with a specific
value if required
       return np.nan
   elif isinstance(income, str):
       # Check for non-standard strings and convert them
       if 'Less than' in income:
           return 20000  # Example value, adjust based on your dataset
       # Check if income is a range
       elif '-' in income:
           parts = income.replace('$', '').replace(',', '').split('-')
           # Calculate midpoint for ranges
           if len(parts) == 2 and parts[1]:
               low, high = map(int, parts)
               return (low + high) / 2
           else:  # Handle cases like '$150,000 -'
               low = int(parts[0])
               return low * 1.25
       elif '+' in income:
           # Handle open-ended values like '$200,000+'
           low = int(income.replace('$', '').replace(',', '').replace('+',
''))
           return low * 1.25
```

```
        else:
            # Handle single values without range
            return int(income.replace('$', '').replace(',', '').replace('
', ''))
    else:
        # If income is already a number, just return it
        return income

# Assuming 'df' is your dataframe
df['income'] = df['income'].apply(standardize_income)

df.head()
```

```
# Convert 'Yes'/'No' to 1/0 in NICU_Stay
df['NICU_Stay'] = df['NICU_Stay'].map({'Yes': 1, 'No': 0})
df.head()
```

```
df['birth_date'] = pd.to_datetime(df['birth_date'], errors='coerce')
df.head()
```