```
import pandas as pd

# Load the data
data_path = 'High Confidence Data - Sheet2.csv'
data = pd.read_csv(data_path)

# Display the first few rows of the dataframe to understand its
structure
data.head()
```

```
                WOS ID   First Name      Country Country Code  Gender
\
0  WOS:000207784200003        Beate      Germany           DE  female

1  WOS:000208863500007       Rustam     Scotland           GB    male

2  WOS:000209649600001  Subramanian        India           IN    male

3  WOS:000209649600042  Subramanian        India           IN    male

4  WOS:000230225300007     Margriet  Netherlands           NL  female


  Gender Probability  Publication Year    Author Type
0             99.00%              2009  Corresponding
1             99.00%              2011  Corresponding
2             98.00%              2014  Corresponding
3             98.00%              2014  Corresponding
4             98.00%              2005  Corresponding
```

```
# Group data by "Publication Year" and "Gender" and count the number
of occurrences
author_counts_by_year_gender = data.groupby(['Publication Year',
'Gender']).size().unstack(fill_value=0)

# Display the aggregated data
author_counts_by_year_gender
```

```
Gender            female  male
Publication Year
1996                   0     3
1997                   0     1
1998                   0     1
1999                   1     2
2000                   2     5
2001                   4     3
2002                   9     3
2003                   6     9
2004                  12    11
2005                  24    13
2006                  34    27
```

```
2007                  39    41
2008                  48    43
2009                  51    53
2010                  57    76
2011                  88    68
2012                  86    94
2013                  91    93
2014                 129   133
2015                 143   134
2016                 173   177
2017                 172   144
2018                 217   180
2019                 204   154
2020                 277   202
2021                 273   222
2022                 309   271
2023                 320   222
2024                  16    16
```

```
from numpy import polyfit, polyval
from sklearn.metrics import mean_squared_error

# Filter data for pre-pandemic years for modeling
pre_pandemic_data =
author_counts_by_year_gender[author_counts_by_year_gender.index <=
2019]

# Prepare data for modeling
years = pre_pandemic_data.index.values
male_authors = pre_pandemic_data['male'].values
female_authors = pre_pandemic_data['female'].values

# Polynomial degree (let's start with a quadratic model, degree=2)
degree = 2

# Fit the models
male_model = polyfit(years, male_authors, degree)
female_model = polyfit(years, female_authors, degree)

# Years for forecasting
forecast_years = [2020, 2021, 2022, 2023]

# Forecasting
male_forecasts = polyval(male_model, forecast_years)
female_forecasts = polyval(female_model, forecast_years)

# Actual data for comparison
actual_male = author_counts_by_year_gender.loc[forecast_years,
'male'].values
actual_female = author_counts_by_year_gender.loc[forecast_years,
```

```
'female'].values

# Prepare a DataFrame for manuscript-ready table
forecast_vs_actual = pd.DataFrame({
    'Year': forecast_years,
    'Forecasted Male Authors': male_forecasts.round(0),
    'Actual Male Authors': actual_male,
    'Forecasted Female Authors': female_forecasts.round(0),
    'Actual Female Authors': actual_female
})
print(male_authors)
forecast_vs_actual
```

```
[  3   1   1   2   5   3   3   9  11  13  27  41  43  53  76  68  94
93
 133 134 177 144 180 154]
```

```
   Year  Forecasted Male Authors  Actual Male Authors  \
0  2020                    206.0                  202
1  2021                    224.0                  222
2  2022                    243.0                  271
3  2023                    263.0                  222

   Forecasted Female Authors  Actual Female Authors
0                      241.0                    277
1                      265.0                    273
2                      289.0                    309
3                      314.0                    320
```

```
# Calculate Mean Squared Error (MSE) for each forecast against actual
data

# MSE for male authors
mse_male = mean_squared_error(actual_male, male_forecasts)

# MSE for female authors
mse_female = mean_squared_error(actual_female, female_forecasts)

mse_male, mse_female
```

```
(617.9520469551784, 446.6701248077866)
```

```
# Function to print the polynomial equation from coefficients
def print_polynomial_equation(coefficients, label="Model"):
    # Assuming a quadratic model, coefficients are in the order of [c,
b, a]
    a, b, c = coefficients
    print(f"{label} Authors Model: y = {a:.4f}x^2 + {b:.4f}x +
{c:.4f}")
```

```
# Print the equations for both models
print_polynomial_equation(male_model, "Male")
print_polynomial_equation(female_model, "Female")
```

```
Male Authors Model: y = 0.3803x^2 + -1518.4097x + 1515812.0989
Female Authors Model: y = 0.5340x^2 + -2134.6654x + 2133301.9768
```

```
from sklearn.metrics import mean_squared_error, r2_score

# Calculate RMSE and R-squared for males
rmse_male = np.sqrt(mean_squared_error(male_authors,
polyval(male_model, years)))
r_squared_male = r2_score(male_authors, polyval(male_model, years))

# Calculate RMSE and R-squared for females
rmse_female = np.sqrt(mean_squared_error(female_authors,
polyval(female_model, years)))
r_squared_female = r2_score(female_authors, polyval(female_model,
years))

print(f"Male - RMSE: {rmse_male:.2f}, R-squared:
{r_squared_male:.2f}")
print(f"Female - RMSE: {rmse_female:.2f}, R-squared:
{r_squared_female:.2f}")
```

```
Male - RMSE: 12.56, R-squared: 0.96
Female - RMSE: 7.97, R-squared: 0.99
```

```
import numpy as np

# Splitting the data into pre- and post-2020
pre_2020_data =
author_counts_by_year_gender[author_counts_by_year_gender.index <
2020]
post_2020_data =
author_counts_by_year_gender[author_counts_by_year_gender.index >=
2020]

# Fitting separate models for pre- and post-2020 data
# Pre-2020
pre_years = pre_2020_data.index.values
pre_male_authors = pre_2020_data['male'].values
pre_female_authors = pre_2020_data['female'].values
pre_male_model = polyfit(pre_years, pre_male_authors, degree)
pre_female_model = polyfit(pre_years, pre_female_authors, degree)

# Post-2020
post_years = post_2020_data.index.values
post_male_authors = post_2020_data['male'].values
post_female_authors = post_2020_data['female'].values
post_male_model = polyfit(post_years, post_male_authors, degree)
```

```
post_female_model = polyfit(post_years, post_female_authors, degree)

# Predicting values at 2020 to assess discontinuity
pre_2020_male_predict = polyval(pre_male_model, 2020)
post_2020_male_predict = polyval(post_male_model, 2020)
male_discontinuity = post_2020_male_predict - pre_2020_male_predict

pre_2020_female_predict = polyval(pre_female_model, 2020)
post_2020_female_predict = polyval(post_female_model, 2020)
female_discontinuity = post_2020_female_predict -
pre_2020_female_predict

(pre_2020_male_predict, post_2020_male_predict, male_discontinuity,
pre_2020_female_predict, post_2020_female_predict,
female_discontinuity)
```

```
(206.1541501963511,
 182.42857152223587,
 -23.72557867411524,
 241.27470355760306,
 244.7142858505249,
 3.439582292921841)
```

```
# Preparing the full range of years in the dataset
full_years = author_counts_by_year_gender.index.values

# Splitting the full range into pre- and post-2020
pre_years_full = full_years[full_years < 2020]
post_years_full = full_years[full_years >= 2020]

# Predicting counts for all years using the respective models
# Male authors
pre_male_predictions = polyval(pre_male_model, pre_years_full)
post_male_predictions = polyval(post_male_model, post_years_full)
full_male_predictions = np.concatenate([pre_male_predictions,
post_male_predictions])

# Female authors
pre_female_predictions = polyval(pre_female_model, pre_years_full)
post_female_predictions = polyval(post_female_model, post_years_full)
full_female_predictions = np.concatenate([pre_female_predictions,
post_female_predictions])

# Compiling results
predictions_df = pd.DataFrame({
    'Year': full_years,
    'Predicted Male Authors': full_male_predictions.round(0),
    'Predicted Female Authors': full_female_predictions.round(0)
})
```

```
predictions_df
```

	Year	Predicted Male Authors	Predicted Female Authors
0	1996	-2.0	3.0
1	1997	-2.0	1.0
2	1998	-2.0	-0.0
3	1999	-0.0	-0.0
4	2000	2.0	0.0
5	2001	5.0	2.0
6	2002	9.0	5.0
7	2003	13.0	9.0
8	2004	19.0	14.0
9	2005	25.0	21.0
10	2006	31.0	28.0
11	2007	39.0	36.0
12	2008	47.0	45.0
13	2009	56.0	56.0
14	2010	66.0	67.0
15	2011	77.0	80.0
16	2012	88.0	94.0
17	2013	100.0	108.0
18	2014	113.0	124.0
19	2015	127.0	141.0
20	2016	141.0	159.0
21	2017	156.0	178.0
22	2018	172.0	198.0
23	2019	189.0	219.0
24	2020	182.0	245.0
25	2021	263.0	331.0
26	2022	265.0	328.0
27	2023	189.0	236.0
28	2024	34.0	55.0