```
import pandas as pd

# Load the uploaded CSV file to examine its contents
data_path = 'Final Data CA+FA.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  \
0  WOS:000174718100007     Janice                    England
GB
1  WOS:000207062600010   Kyoungho                South Korea
KR
2  WOS:000207451700010  Elizabeth  United States of America
US
3  WOS:000207695900002    Joachim  United States of America
US
4  WOS:000207784200003      Beate                    Germany
DE

   Gender Gender Probability  Publication Year    Author Type
0  female            100.00%              2002  Corresponding
1    male            100.00%              2007  Corresponding
2  female            100.00%              2007  Corresponding
3    male            100.00%              2008  Corresponding
4  female             99.00%              2009  Corresponding
```

```
# Preparing the data for analysis by creating 'Female CA', 'Female
FA', 'Male CA', 'Male FA' columns

# Initialize the columns to False
data['Female CA'] = False
data['Female FA'] = False
data['Male CA'] = False
data['Male FA'] = False

# Assign True based on conditions
data.loc[(data['Gender'] == 'female') & (data['Author Type'] ==
'Corresponding'), 'Female CA'] = True
data.loc[(data['Gender'] == 'female') & (data['Author Type'] ==
'First'), 'Female FA'] = True
data.loc[(data['Gender'] == 'male') & (data['Author Type'] ==
'Corresponding'), 'Male CA'] = True
data.loc[(data['Gender'] == 'male') & (data['Author Type'] ==
'First'), 'Male FA'] = True

# Summarize the data by 'WOS ID'
```

```
summary_data = data.groupby('WOS ID').agg({
    'Female CA': 'max',
    'Female FA': 'max',
    'Male CA': 'max',
    'Male FA': 'max'
})

# Calculate the correlations
correlation_female_CA_FA_unique = summary_data['Female
CA'].corr(summary_data['Female FA'], method='spearman')
correlation_male_CA_female_FA_unique = summary_data['Male
CA'].corr(summary_data['Female FA'], method='spearman')
correlation_male_CA_male_FA_unique = summary_data['Male
CA'].corr(summary_data['Male FA'], method='spearman')
correlation_female_CA_male_FA_unique = summary_data['Female
CA'].corr(summary_data['Male FA'], method='spearman')

# Calculate the individual numbers and total number of publications
total_publications = len(summary_data)
female_CA_count = summary_data['Female CA'].sum()
female_FA_count = summary_data['Female FA'].sum()
male_CA_count = summary_data['Male CA'].sum()
male_FA_count = summary_data['Male FA'].sum()

correlation_female_CA_FA_unique, correlation_male_CA_female_FA_unique,
correlation_male_CA_male_FA_unique,correlation_female_CA_male_FA_uniqu
e, total_publications, female_CA_count, female_FA_count,
male_CA_count, male_FA_count
```

```
(0.3334259173100722,
 -0.28686557112548444,
 0.3068203742066952,
 -0.2715145879241861,
 10576,
 3500,
 4400,
 6720,
 5208)
```

```
import pandas as pd

# Load the dataset
data_path = 'Final Data CA+FA.csv'
data = pd.read_csv(data_path)

# Aggregate the data for each WOS ID, counting male and female CAs and
FAs
ca_fa_gender_distribution = data.groupby(['WOS ID', 'Author Type',
'Gender']).size().unstack(fill_value=0).unstack(fill_value=0)
```

```
# Rearrange the columns for better readability
ca_fa_gender_distribution =
ca_fa_gender_distribution.reorder_levels([1, 0],
axis=1).sort_index(axis=1)

# Save the aggregated data to an Excel file
output_path = 'ca_fa_gender_distribution.xlsx'
ca_fa_gender_distribution.to_excel(output_path)

import pandas as pd

# Load the data
data_path = 'Final Data CA+FA.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  \
0  WOS:000174718100007     Janice                   England
GB
1  WOS:000207062600010   Kyoungho               South Korea
KR
2  WOS:000207451700010  Elizabeth  United States of America
US
3  WOS:000207695900002    Joachim  United States of America
US
4  WOS:000207784200003      Beate                   Germany
DE

   Gender Gender Probability  Publication Year    Author Type
0  female            100.00%              2002  Corresponding
1    male            100.00%              2007  Corresponding
2  female            100.00%              2007  Corresponding
3    male            100.00%              2008  Corresponding
4  female             99.00%              2009  Corresponding
```

```
# Load the Excel file
excel_path = 'ca_fa_gender_distribution.xlsx'
excel_data = pd.read_excel(excel_path)

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

```
           Author Type Corresponding Unnamed: 2 Unnamed: 3   First
Unnamed: 5  \
0              Gender        female       male    unknown  female
```

```
male
1                WOS ID            NaN         NaN         NaN      NaN
NaN
2  WOS:000072120200009              0           1           0        0
0
3  WOS:000072564900002              1           0           0        0
0
4  WOS:000073629800014              1           0           0        0
1

  Unnamed: 6
0    unknown
1        NaN
2          0
3          0
4          0
```

```
# Drop the first row which is essentially a sub-header
excel_data_cleaned = excel_data.drop(index=0).reset_index(drop=True)
# Rename columns properly based on the understanding of the structure
excel_data_cleaned.columns = ['WOS ID', 'CA Female', 'CA Male', 'CA
Unknown', 'FA Female', 'FA Male', 'FA Unknown']

# Merge the two datasets on WOS ID
merged_data = pd.merge(data[['WOS ID', 'Gender']], excel_data_cleaned,
on='WOS ID', how='inner')

# Since we need to filter based on the gender of the CA and then look
at the gender of the FA,
# we first need to ensure that the 'Gender' from the CSV matches the
'CA Female' and 'CA Male' columns.
# Transform the Gender column to match the binary representation in
the excel sheet
merged_data['CA Gender'] = merged_data['Gender'].apply(lambda x:
'female' if x == 'female' else ('male' if x == 'male' else 'unknown'))

# Now, filter the data for Female and Male CAs and prepare it for
calculating Pearson's correlation
female_ca_data = merged_data[merged_data['CA Gender'] == 'female']
[['FA Female', 'FA Male', 'FA Unknown']]
male_ca_data = merged_data[merged_data['CA Gender'] == 'male'][['FA
Female', 'FA Male', 'FA Unknown']]

# Display the first few rows of each to ensure correctness
female_ca_data.head(), male_ca_data.head()
```

```
(  FA Female FA Male FA Unknown
 0         0       0          0
 3         1       0          0
```

```
 4          1       0          0
 7          1       0          0
 8          1       0          0,
   FA Female FA Male FA Unknown
 1          0       1          0
 2          0       1          0
 5          0       1          0
 6          0       1          0
 9          0       1          0)
```

```
# Calculate probabilities for the various scenarios
# The probability of FA being male given CA is male
prob_fa_male_given_ca_male = male_ca_data['FA Male'].sum() /
male_ca_data[['FA Male', 'FA Female']].sum().sum()

# The probability of FA being female given CA is male
prob_fa_female_given_ca_male = male_ca_data['FA Female'].sum() /
male_ca_data[['FA Male', 'FA Female']].sum().sum()

# The probability of FA being male given CA is female
prob_fa_male_given_ca_female = female_ca_data['FA Male'].sum() /
female_ca_data[['FA Male', 'FA Female']].sum().sum()

# The probability of FA being female given CA is female
prob_fa_female_given_ca_female = female_ca_data['FA Female'].sum() /
female_ca_data[['FA Male', 'FA Female']].sum().sum()

(prob_fa_male_given_ca_male, prob_fa_female_given_ca_male,
prob_fa_male_given_ca_female, prob_fa_female_given_ca_female)
```

```
(0.8104057922209126,
 0.1895942077790874,
 0.1384850803366488,
 0.8615149196633511)
```

```
# Calculate the counts that were used to compute the probabilities
fa_male_given_ca_male_count = male_ca_data['FA Male'].sum()
fa_female_given_ca_male_count = male_ca_data['FA Female'].sum()
total_ca_male = male_ca_data[['FA Male', 'FA Female']].sum().sum()

fa_male_given_ca_female_count = female_ca_data['FA Male'].sum()
fa_female_given_ca_female_count = female_ca_data['FA Female'].sum()
total_ca_female = female_ca_data[['FA Male', 'FA Female']].sum().sum()

{
    "FA Male given CA Male": fa_male_given_ca_male_count,
    "FA Female given CA Male": fa_female_given_ca_male_count,
    "Total CA Male": total_ca_male,
    "FA Male given CA Female": fa_male_given_ca_female_count,
    "FA Female given CA Female": fa_female_given_ca_female_count,
```

```
    "Total CA Female": total_ca_female
}
```

```
{'FA Male given CA Male': 9626,
 'FA Female given CA Male': 2252,
 'Total CA Male': 11878,
 'FA Male given CA Female': 1086,
 'FA Female given CA Female': 6756,
 'Total CA Female': 7842}
```