Customer demographic analysis

It is going to contain the Exploratory data analysis and the reports are going to be presented in form of a PDF file. We are going to have a look at the customers and gather the necessary insights about their habits, demographics etc. We will make an attempt to delve deeper into their demographics and behavioral patterns.We will use charts and graphs to visualize the data and make it easier for the user to understand. These revelations will serve as a cornerstone for data-driven decision-making, guiding us to tailor marketing strategies, refine product offerings, and enhance customer engagement.

Step 1: Setting up the work environment

We are going to download the necessary packages for our work. We are going to view the dataset and examine the datatypes.

In [1]:
```python
#Getting the necessary packages
import pandas as pd         # For data manipulation and analysis
import numpy as np          # For numerical operations
import matplotlib.pyplot as plt    # For basic data visualization
import seaborn as sns       # For advanced data visualization
from scipy import stats # For statistics
```

In [2]:
```python
#getting the dataset

cd= pd.read_excel('C:\\Users\\sujoydutta\\Desktop\\Data analysis\\Projects\\Bank customer analysis\\CustomerDem
cd.head()
```

Out[2]:

	ID	account_type	gender	age	Income	Emp_Tenure_Years	Tenure_with_Bank	region_code	NetBanking_Flag	Avg_days_between_transact
0	19427	current	M	63	MEDIUM	30.1	10	628.0	1	
1	16150	current	M	36	MEDIUM	14.4	10	656.0	0	
2	11749	current	F	28	MEDIUM	4.8	10	314.0	1	
3	11635	current	M	32	MEDIUM	9.6	2	614.0	1	
4	8908	current	M	32	HIGH	12.0	7	750.0	1	

In [3]:
```python
#examining the dataset
cd.shape
```

Out[3]:
```
(20000, 10)
```

In [4]:
```python
#seeing datatypes
cd.dtypes
```

Out[4]:
```
ID                               int64
account_type                    object
gender                          object
age                              int64
Income                          object
Emp_Tenure_Years               float64
Tenure_with_Bank                 int64
region_code                    float64
NetBanking_Flag                  int64
Avg_days_between_transaction   float64
dtype: object
```

Step 2: Data cleaning/formatting

In this step,the data is going to be formatted and cleaned of its irregularities so as the anaalysis could be done better.

In [5]:
```python
#seeing missing values
cd.isnull().sum()
```

Out[5]:
```
ID                              0
account_type                    1
gender                          1
age                             0
Income                          1
Emp_Tenure_Years                0
Tenure_with_Bank                0
region_code                     1
NetBanking_Flag                 0
Avg_days_between_transaction    3
dtype: int64
```

In [6]:
```python
#dropping null values
cd=cd.dropna()
cd
```

Out[6]:

	ID	account_type	gender	age	Income	Emp_Tenure_Years	Tenure_with_Bank	region_code	NetBanking_Flag	Avg_days_between_tra
0	19427	current	M	63	MEDIUM	30.1	10	628.0	1	
1	16150	current	M	36	MEDIUM	14.4	10	656.0	0	
2	11749	current	F	28	MEDIUM	4.8	10	314.0	1	
3	11635	current	M	32	MEDIUM	9.6	2	614.0	1	
4	8908	current	M	32	HIGH	12.0	7	750.0	1	
...	...	...	...	...	...	...	...	...	...	
19995	1270	current	F	66	MEDIUM	32.2	6	354.0	1	
19996	15992	current	M	53	MEDIUM	19.8	5	809.0	0	
19997	7081	current	F	66	MEDIUM	18.4	1	466.0	0	
19998	6821	current	M	32	LOW	6.0	8	619.0	0	
19999	13730	current	F	118	LOW	78.4	7	882.0	1	

19993 rows × 10 columns

In [7]:
```python
#removing the trailing zero
cd['region_code'] = cd['region_code'].astype(int)
cd['region_code'] = cd['region_code'].astype(str)

#  cleaned 'region_code' column
print(cd['region_code'])
```

```
0        628
1        656
2        314
3        614
4        750
        ...
19995    354
19996    809
19997    466
19998    619
19999    882
Name: region_code, Length: 19993, dtype: object
```

In [8]:
```python
#making the data type appropriate
cd['ID'] = cd['ID'].astype(str)
cd['Avg_days_between_transaction'] = cd['Avg_days_between_transaction'].astype(int)
```

In [9]:
```python
#seeing new datatypes
cd.dtypes
```

Out[9]:
```
ID                              object
account_type                    object
gender                          object
age                              int64
Income                          object
Emp_Tenure_Years               float64
Tenure_with_Bank                 int64
region_code                     object
NetBanking_Flag                  int64
Avg_days_between_transaction     int32
dtype: object
```

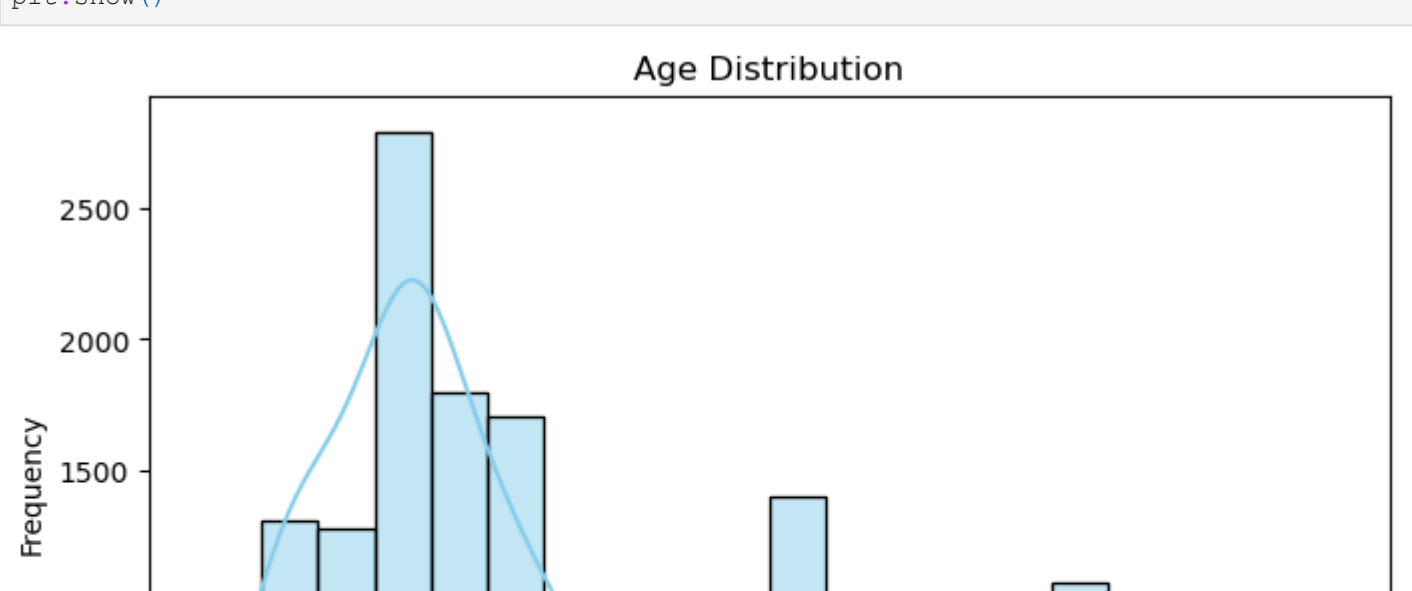
In [10]:
```python
#function to handle outliers



def handle_extreme_values_iqr(data, column_name, multiplier=1.5):
    # Calculate the IQR (Interquartile Range)
    Q1 = data[column_name].quantile(0.25)
    Q3 = data[column_name].quantile(0.75)
    IQR = Q3 - Q1

    lower_bound = Q1 - multiplier * IQR
    upper_bound = Q3 + multiplier * IQR


    extreme_values = (data[column_name] < lower_bound) | (data[column_name] > upper_bound)
    data.loc[extreme_values, column_name] = np.nan
```

In [12]:
```python
#handling outliers for numerical variables

handle_extreme_values_iqr(cd, 'age', multiplier=1.5)
handle_extreme_values_iqr(cd, 'Emp_Tenure_Years',  multiplier=1.5)
handle_extreme_values_iqr(cd, 'Tenure_with_Bank', multiplier=1.5)
handle_extreme_values_iqr(cd, 'Avg_days_between_transaction',  multiplier=1.5)
```

Step 3: Exploratory Data Analysis

The dataset has been cleaned and formatted. Now it is time to ask our questions to the dataset to reveal some interesting insights.

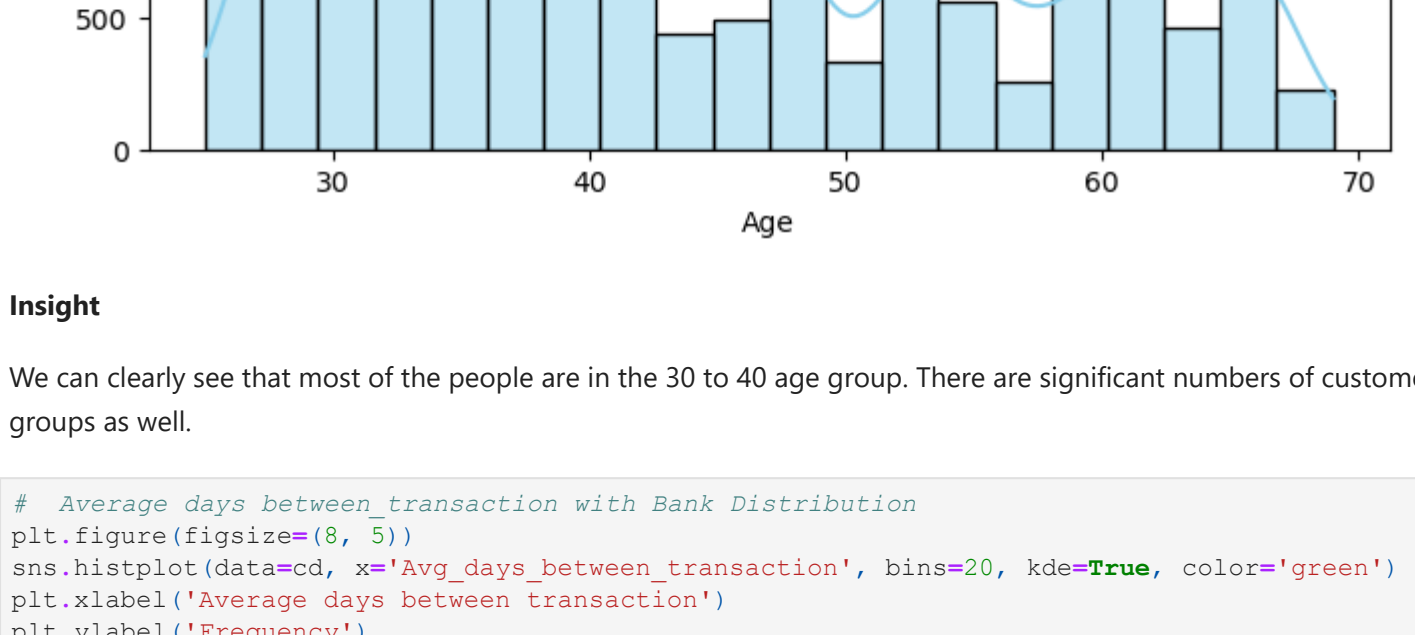
In [13]:
```python
# Age Distribution
plt.figure(figsize=(8, 5))
sns.histplot(data=cd, x='age', bins=20, kde=True, color='skyblue')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Age Distribution')
plt.show()
```



Insight

We can clearly see that most of the people are in the 30 to 40 age group. There are significant numbers of customers in the 50 to 70 age groups as well.

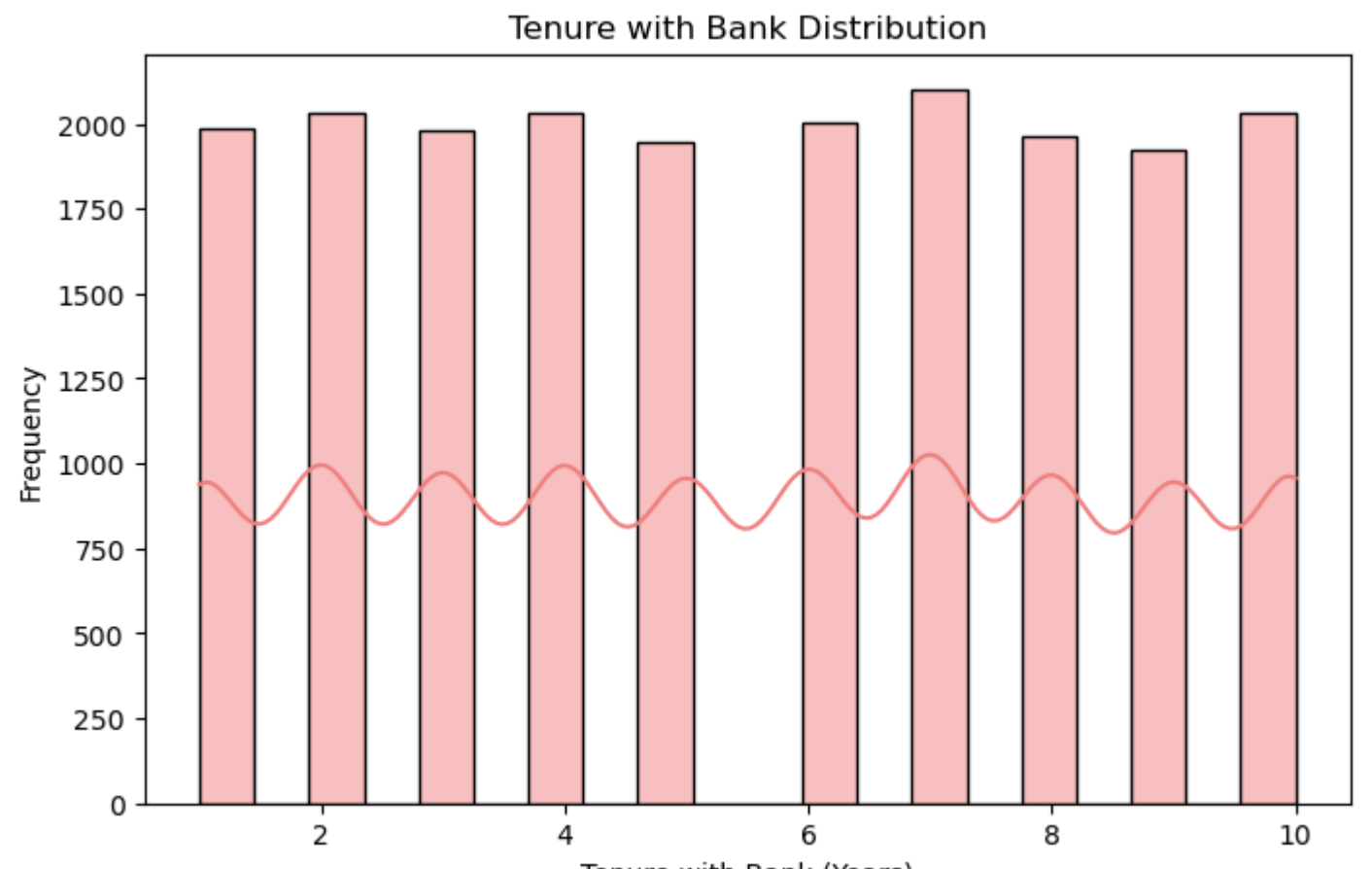
In [50]:
```python
#  Average days between_transaction with Bank Distribution
plt.figure(figsize=(8, 5))
sns.histplot(data=cd, x='Avg_days_between_transaction', bins=20, kde=True, color='green')
plt.xlabel('Average days between transaction')
plt.ylabel('Frequency')
plt.title('Average number of days between transaction Distribution')
plt.show()
```



Insight

The distribution is almost uniform for all types of days between transaction.

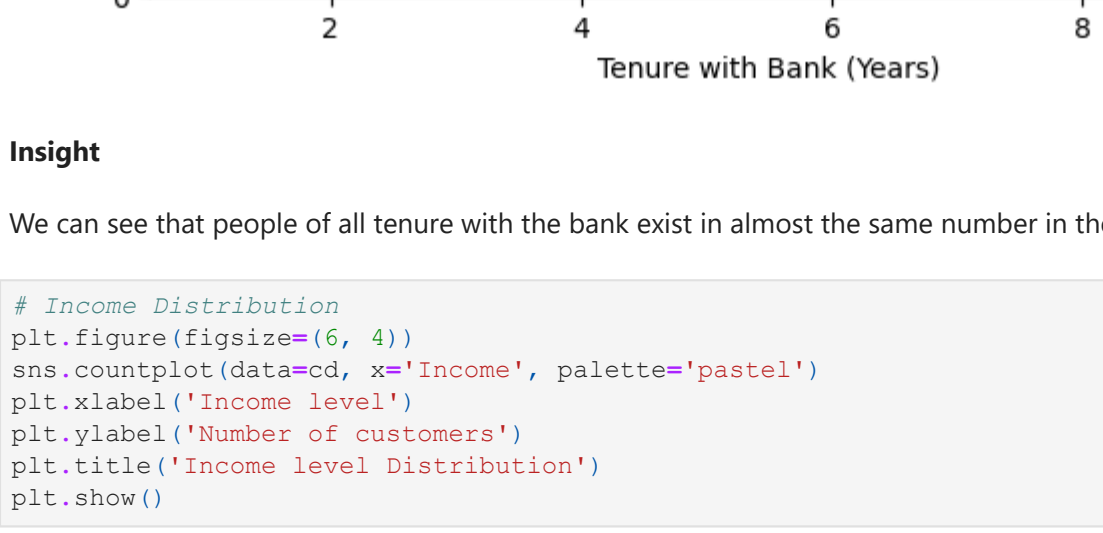
In [21]:
```python
#  Tenure with Bank Distribution
plt.figure(figsize=(8, 5))
sns.histplot(data=cd, x='Tenure_with_Bank', bins=20, kde=True, color='lightcoral')
plt.xlabel('Tenure with Bank (Years)')
plt.ylabel('Frequency')
plt.title('Tenure with Bank Distribution')
plt.show()
```



Insight

We can see that people of all tenure with the bank exist in almost the same number in the bank.

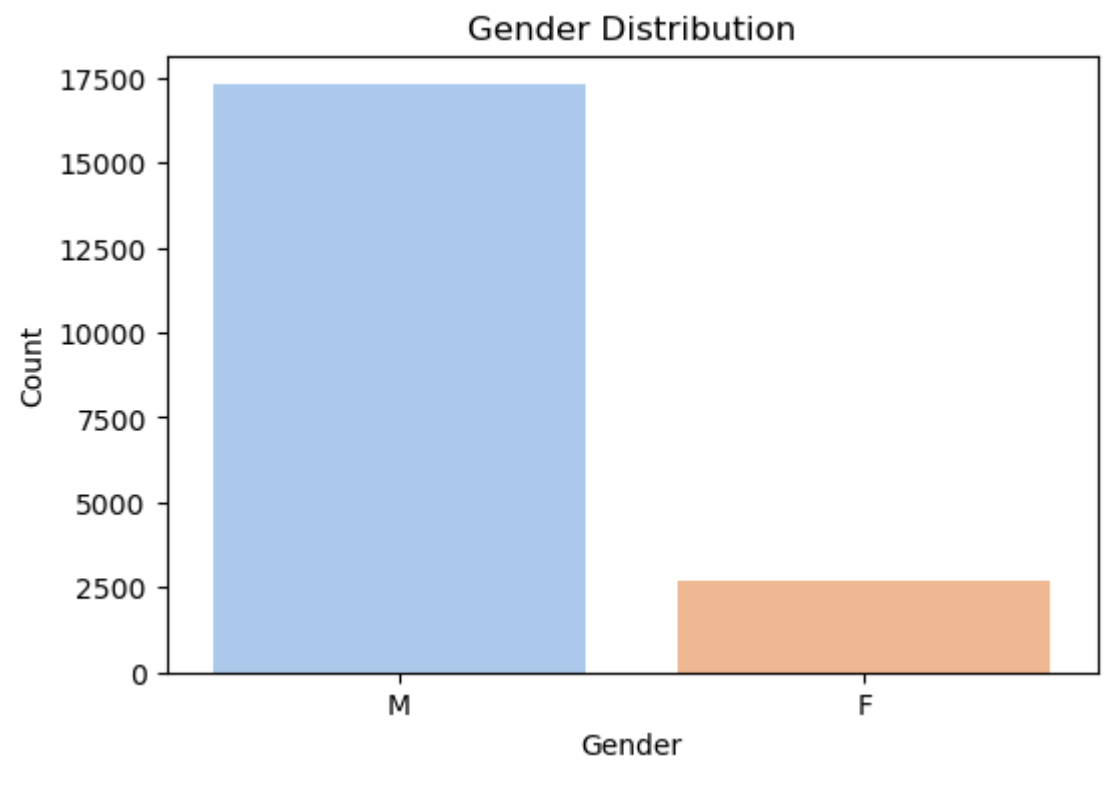
In [15]:
```python
# Income Distribution
plt.figure(figsize=(6, 4))
sns.countplot(data=cd, x='Income', palette='pastel')
plt.xlabel('Income level')
plt.ylabel('Number of customers')
plt.title('Income level Distribution')
plt.show()
```



Insight

Only 10% of the customers are in the high income group. People in the middle income comprise the majority and are 60% of the clientele rest 30% are low income customers.

In [40]:
```python
# Gender Distribution
plt.figure(figsize=(6, 4))
sns.countplot(data=cd, x='gender', palette='pastel')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.title('Gender Distribution')
plt.show()
```



Insights

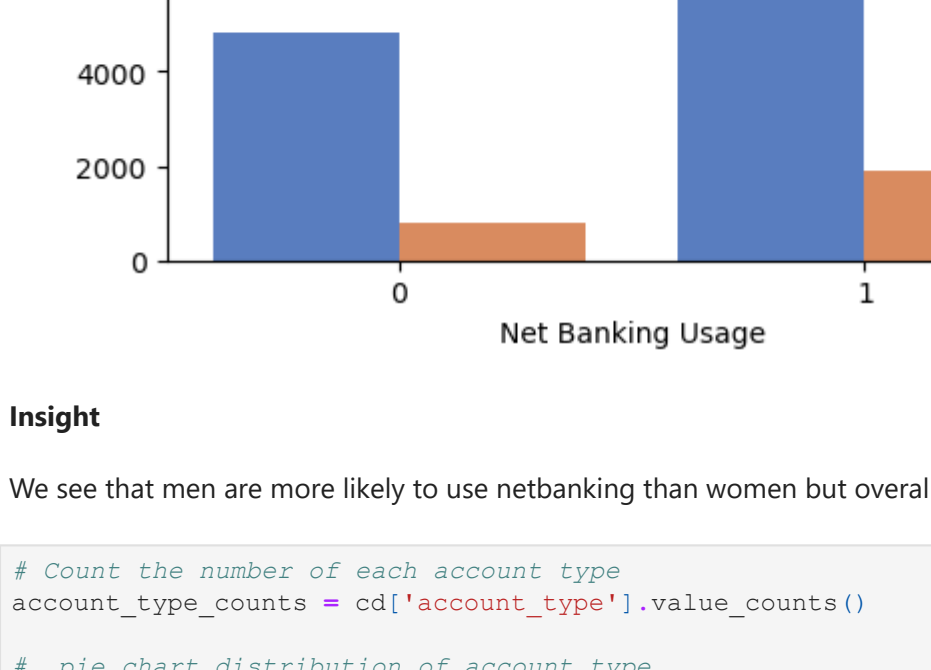
Pretty much self explatory that around 88% of the clients are male and the rest 12% are women. The bank needs to bring more women friendly schemes to encourage more female clients to invest.

In [42]:
```python
# Net Banking Usage by Gender
plt.figure(figsize=(6, 4))
sns.countplot(data=cd, x='NetBanking_Flag', hue='gender', palette='muted')
plt.xlabel('Net Banking Usage')
plt.ylabel('Count')
plt.title('Net Banking Usage by Gender')
plt.show()
```

Insight

We see that men are more likely to use netbanking than women but overall both genders favour netbanking more.

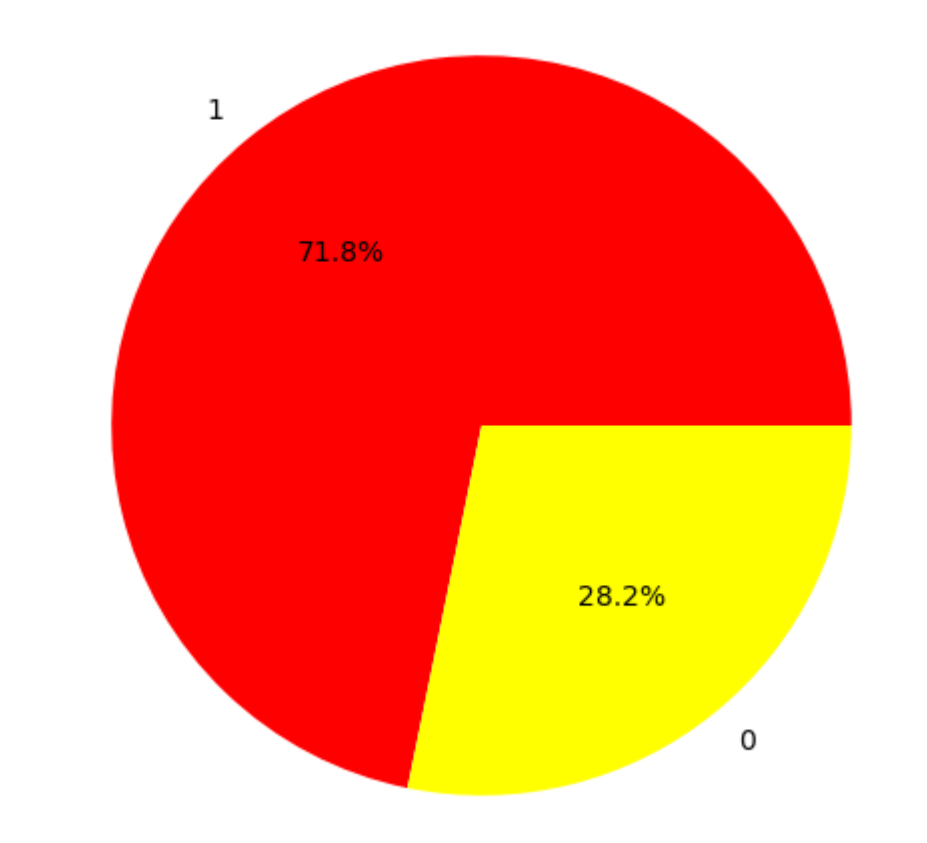
In [17]:
```python
# Count the number of each account type
account_type_counts = cd['account_type'].value_counts()

#  pie chart distribution of account type
plt.figure(figsize=(6, 6))
plt.pie(account_type_counts, labels=account_type_counts.index, autopct='%1.1f%%', colors=['lightblue', 'lightgr
plt.title('Account Type Distribution')
plt.show()
```



Insight

Majority of customers open current account and only a few open a saving account.

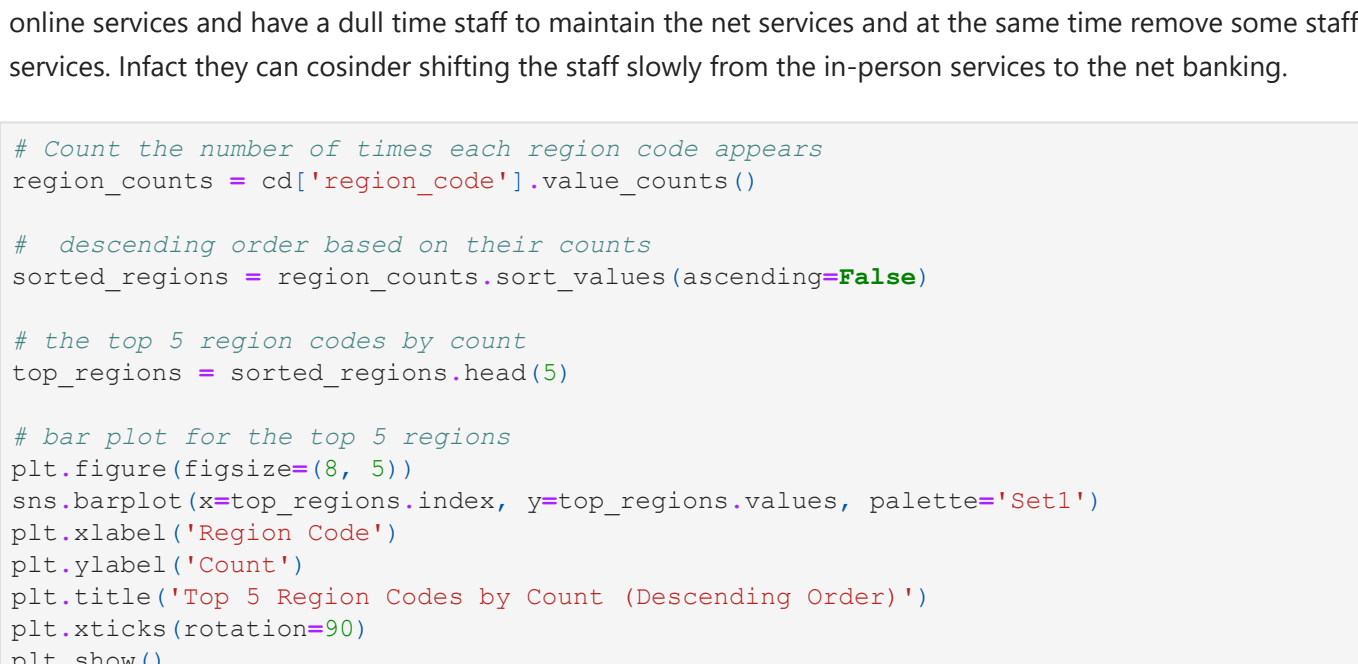
In [22]:
```python
# Count the number of each account type
NetBanking_Flag_counts = cd['NetBanking_Flag'].value_counts()

#  pie chart distribution of account type
plt.figure(figsize=(6, 6))
plt.pie(NetBanking_Flag_counts, labels=NetBanking_Flag_counts.index, autopct='%1.1f%%', colors=['red', 'yellow'
plt.title('NetBanking Flag Distribution')
plt.show()
```



Insights

This leads us to conclude that every 7 customers out of 10 prefers net banking so it is suggested that the bank pays full attention to the online services and have a dull time staff to maintain the net services and at the same time remove some staff from the in-person services. Infact they can cosinder shifting the staff slowly from the in-person services to the net banking.

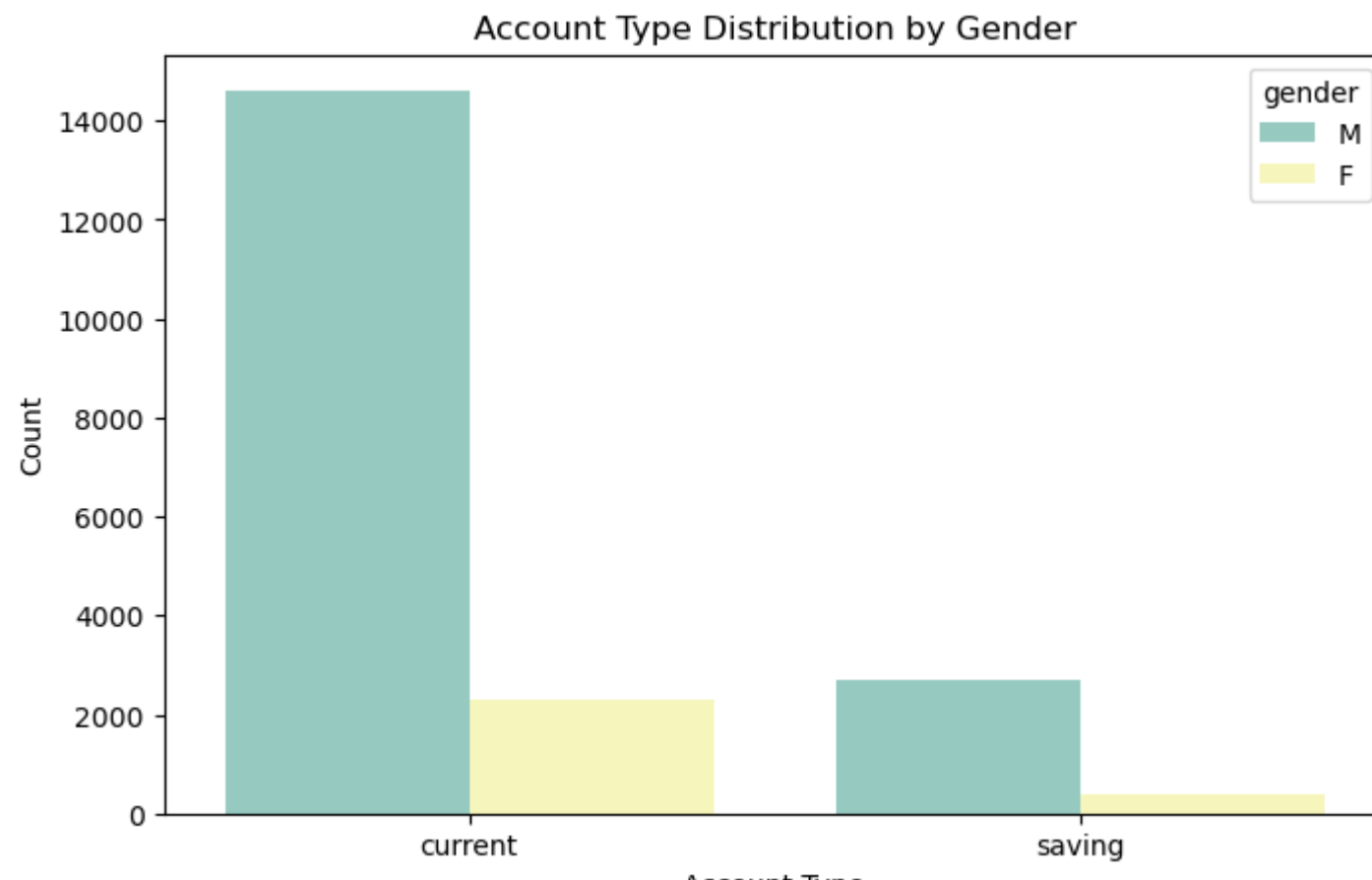
In [31]:
```python
# Count the number of times each region code appears
region_counts = cd['region_code'].value_counts()

#  descending order based on their counts
sorted_regions = region_counts.sort_values(ascending=False)

# the top 5 region codes by count
top_regions = sorted_regions.head(5)

# bar plot for the top 5 regions
plt.figure(figsize=(8, 5))
sns.barplot(x=top_regions.index, y=top_regions.values, palette='Set1')
plt.xlabel('Region Code')
plt.ylabel('Count')
plt.title('Top 5 Region Codes by Count (Descending Order)')
plt.xticks(rotation=90)
plt.show()
```



Insights

These are the following regions with the most customers. Region 249 is the most precious as it supplies the most customers.

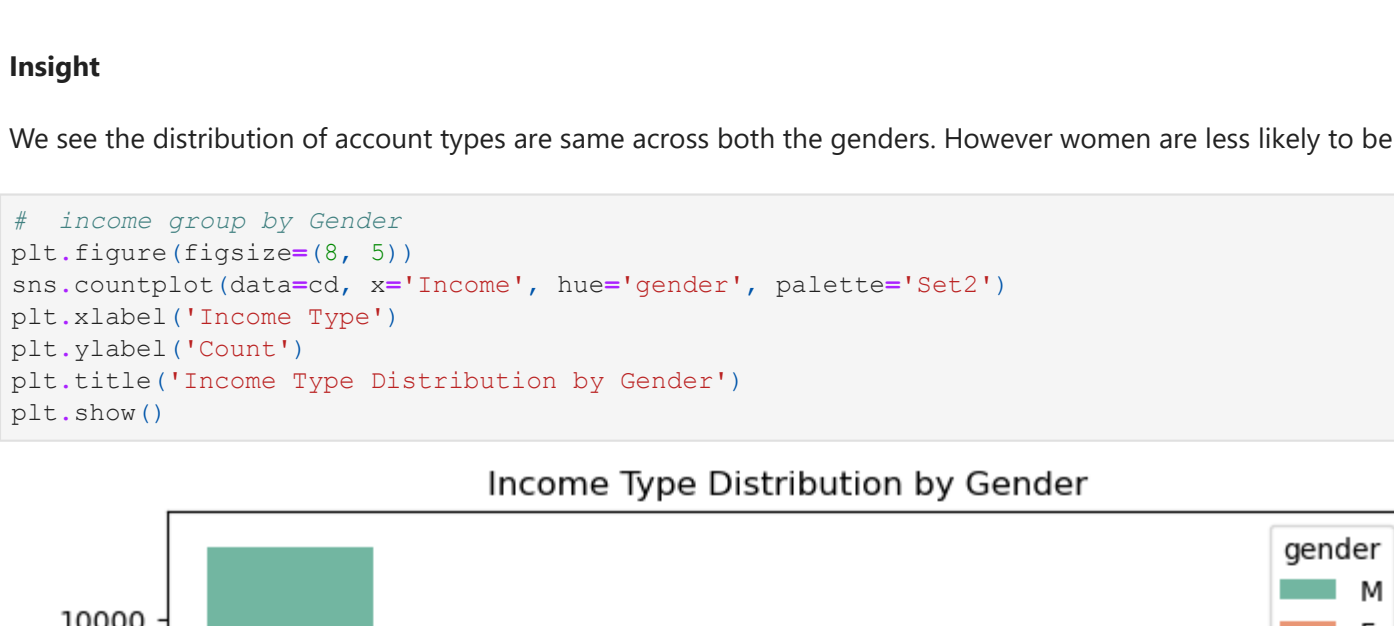
In [32]:
```python
#  Account Type by Gender
plt.figure(figsize=(8, 5))
sns.countplot(data=cd, x='account_type', hue='gender', palette='Set3')
plt.xlabel('Account Type')
plt.ylabel('Count')
plt.title('Account Type Distribution by Gender')
plt.show()
```



Insight

We see the distribution of account types are same across both the genders. However women are less likely to be in the saving accounts.

In [34]:
```python
#  income group by Gender
plt.figure(figsize=(8, 5))
sns.countplot(data=cd, x='Income', hue='gender', palette='Set2')
plt.xlabel('Income Type')
plt.ylabel('Count')
plt.title('Income Type Distribution by Gender')
plt.show()
```



Insight

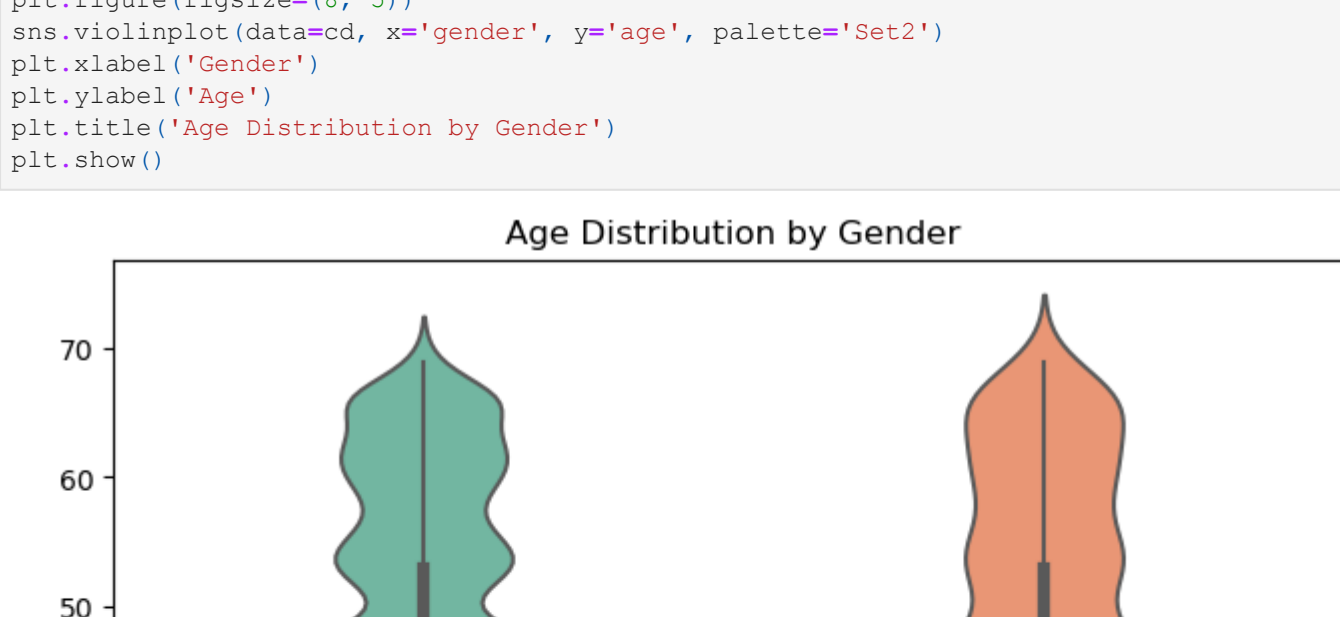
We see the distribution of Income types are same across both the genders.However women are less likely to be in the high income group.

In [36]:
```python
# Employee tenure by Account Type
plt.figure(figsize=(8, 5))
sns.boxplot(data=cd, x='account_type', y='Emp_Tenure_Years', palette='Set2')
plt.xlabel('Account Type')
plt.ylabel('Tenure in career (Years)')
plt.title('Tenure in career by Account Type')
plt.xticks(rotation=45)  # Rotate x-axis labels for readability
plt.show()
```

Insight

We can conclude that there is no significant difference between the average career tenure of both account types.

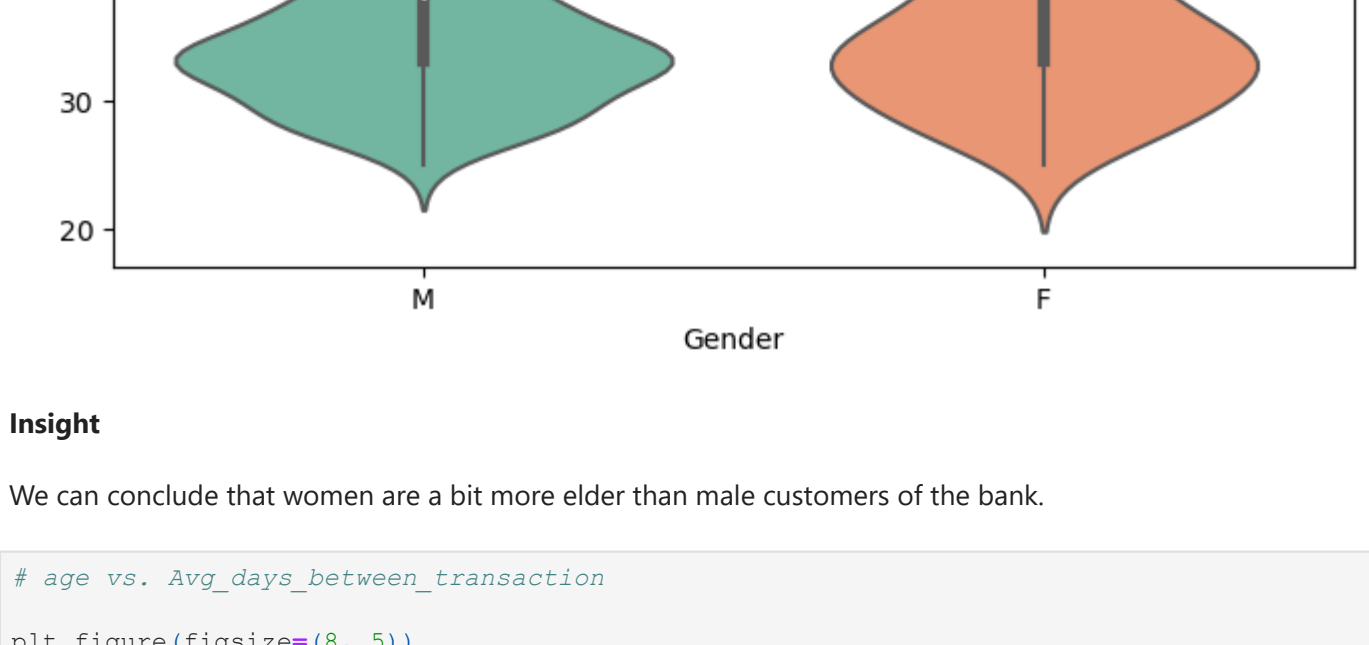
In [38]:
```python
# Age Distribution by Gender
plt.figure(figsize=(8, 5))
sns.violinplot(data=cd, x='gender', y='age', palette='Set2')
plt.xlabel('Gender')
plt.ylabel('Age')
plt.title('Age Distribution by Gender')
plt.show()
```



Insight

We can conclude that women are a bit more elder than male customers of the bank.

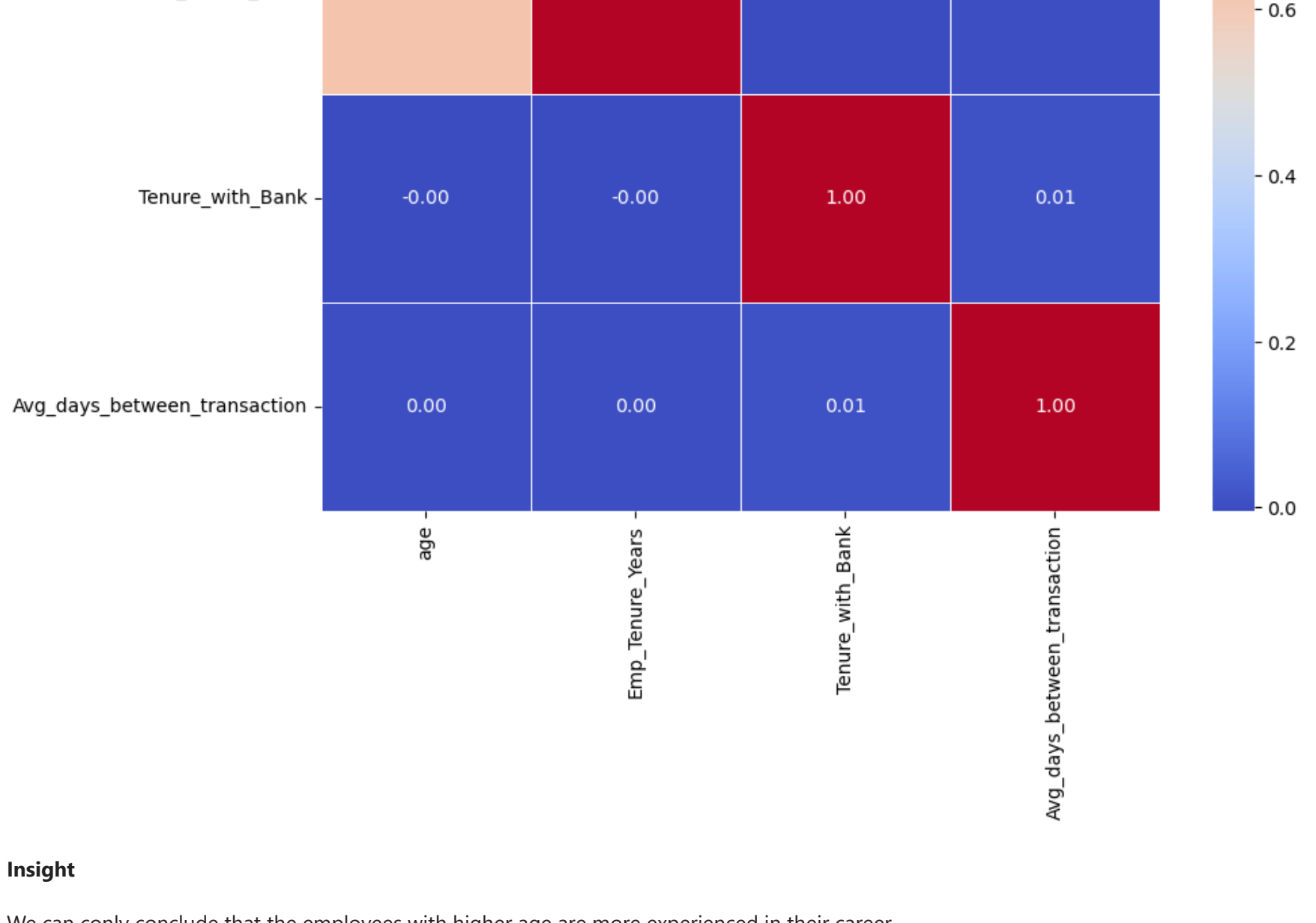
In [45]:
```python
# age vs. Avg_days_between_transaction

plt.figure(figsize=(8, 5))
sns.scatterplot(data=cd, x='age', y='Avg_days_between_transaction', color='lightseagreen')
plt.xlabel('Age')
plt.ylabel('Avg Days Between Transactions')
plt.title('Scatterplot of Age vs. Avg Days Between Transactions')
plt.show()
```



Insight

No clear conclusion can be made.

In [46]:
```python
# list of numerical variables
selected_numerical_variables = ['age', 'Emp_Tenure_Years', 'Tenure_with_Bank', 'Avg_days_between_transaction']

#  correlation matrix for the selected numerical variables
correlation_matrix = cd[selected_numerical_variables].corr()

# Create a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('Correlation Heatmap of Selected Numerical Variables')
plt.show()
```



Insight

We can conly conclude that the employees with higher age are more experienced in their career.

In [ ]: