In [3]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt
```

In [5]:

```
# Read the CSV file
df_train = pd.read_csv("train_new.csv")
```

In [9]:

```
df_train['log_gross_income'] = np.log(df_train['Gross_income'] + 1)
```

In [13]:

```
df_train['sqrt_gross_income'] = np.sqrt(df_train['Gross_income'])
```

In [16]:

```
# Display the updated DataFrame
print(df_train.head())
```

```
   fecha_dato  Customer_Code Employee_index Country_of_Residence    Sex  A
ge  \
0  2015-01-28        1375586   Not employed                   ES    Men
35
1  2015-01-28        1050611   Not employed                   ES  Women
23
2  2015-01-28        1050612   Not employed                   ES  Women
23
3  2015-01-28        1050613   Not employed                   ES    Men
22
4  2015-01-28        1050614   Not employed                   ES  Women
23

   fecha_alta  New_customer_Index  Seniority Customer_Type_1st_month  ...
\
0  2015-01-12                   0          6                       P  ...
1  2012-08-10                   0         35                       P  ...
2  2012-08-10                   0         35                       P  ...
3  2012-08-10                   0         35                       P  ...
4  2012-08-10                   0         35                       P  ...

  Loans Taxes Credit_card Securities Home_account Payroll  Pensions.1  \
0     0     0           0          0            0     0.0         0.0
1     0     0           0          0            0     0.0         0.0
2     0     0           0          0            0     0.0         0.0
3     0     0           0          0            0     0.0         0.0
4     0     0           0          0            0     0.0         0.0

  Direct_debit  log_gross_income  sqrt_gross_income
0            0         11.376179         295.327107
1            0         10.478688         188.543735
2            0         11.713252         349.541285
3            0         11.693383         346.086030
4            0          0.000000           0.000000

[5 rows x 47 columns]
```

In [21]:

```
from scipy.stats.mstats import winsorize
```

```
df_train['winsorized_gross_income'] = winsorize(df_train['Gross_income'],
```

In [37]:

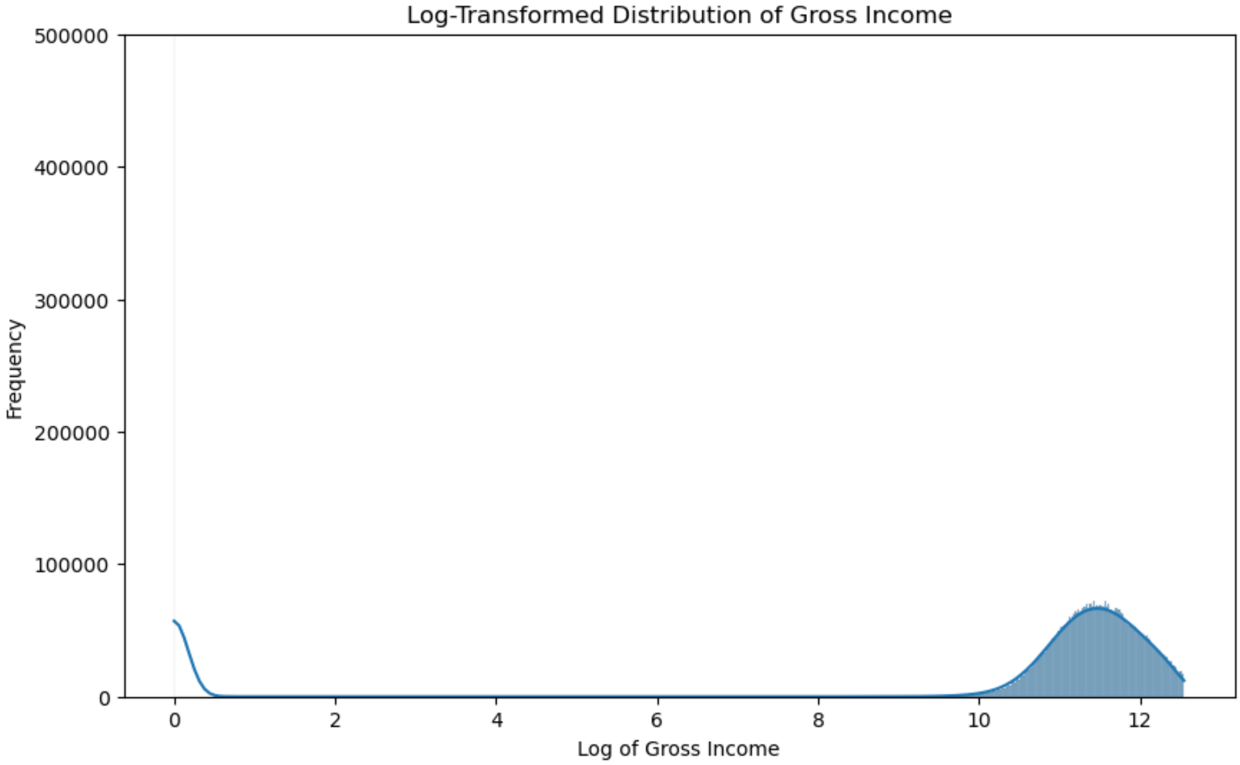
```
# Calculate the first quartile (Q1) and the third quartile (Q3)
Q1 = df_train['Gross_income'].quantile(0.25)
Q3 = df_train['Gross_income'].quantile(0.75)

# Calculate the Interquartile Range (IQR)
IQR = Q3 - Q1

# Define the lower and upper thresholds
lower_threshold = Q1 - 1.5 * IQR
upper_threshold = Q3 + 1.5 * IQR

# Filter the DataFrame to remove outliers
df_cleaned = df_train[(df_train['Gross_income'] >= lower_threshold) & (df

# Display the result (optional)
print(df_cleaned)
```

```
          fecha_dato  Customer_Code Employee_index Country_of_Residence  \
0         2015-01-28        1375586   Not employed                   ES   
1         2015-01-28        1050611   Not employed                   ES   
2         2015-01-28        1050612   Not employed                   ES   
3         2015-01-28        1050613   Not employed                   ES   
4         2015-01-28        1050614   Not employed                   ES   
...              ...            ...            ...                  ...   
13379651  2016-05-28        1166766   Not employed                   ES   
13379652  2016-05-28        1166765   Not employed                   ES   
13379653  2016-05-28        1166764   Not employed                   ES   
13379654  2016-05-28        1166763   Not employed                   ES   
13379655  2016-05-28        1166789   Not employed                   ES   

            Sex  Age  fecha_alta  New_customer_Index  Seniority  \
0           Men   35  2015-01-12                   0          6   
1         Women   23  2012-08-10                   0         35   
2         Women   23  2012-08-10                   0         35   
3           Men   22  2012-08-10                   0         35   
4         Women   23  2012-08-10                   0         35   
...         ...  ...         ...                 ...        ...   
13379651  Women   25  2013-08-14                   0         33   
13379652  Women   22  2013-08-14                   0         33   
13379653  Women   23  2013-08-14                   0         33   
13379654    Men   47  2013-08-14                   0         33   
13379655    Men   22  2013-08-14                   0         33   

         Customer_Type_1st_month  ... Taxes Credit_card Securities  \
0                              P  ...     0           0          0   
1                              P  ...     0           0          0   
2                              P  ...     0           0          0   
3                              P  ...     0           0          0   
4                              P  ...     0           0          0   
...                          ...  ...   ...         ...        ...   
13379651                       1  ...     0           0          0   
13379652                       1  ...     0           0          0   
13379653                       1  ...     0           0          0   
13379654                       1  ...     0           0          0   
13379655                       P  ...     0           0          0   
```

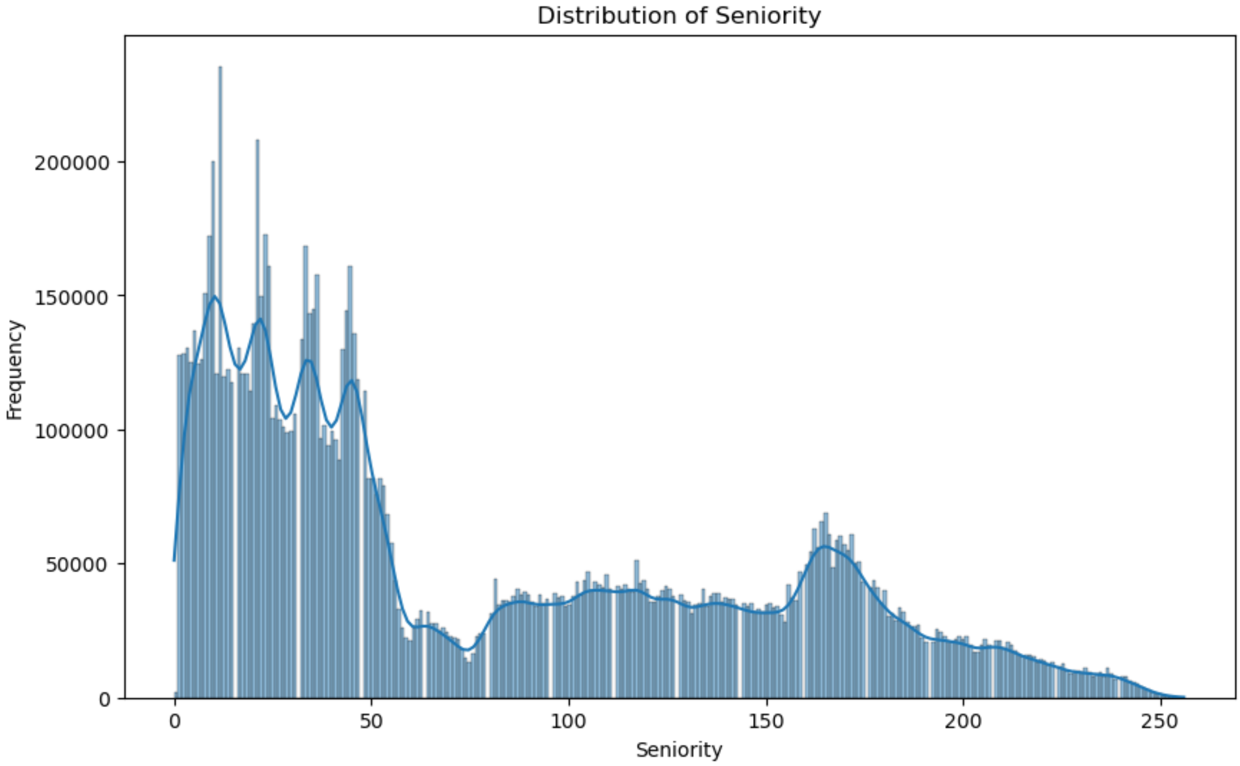
```
          Home_account  Payroll  Pensions.1  Direct_debit  log_gross_income  \
0                    0      0.0         0.0             0         11.376179   
1                    0      0.0         0.0             0         10.478688   
2                    0      0.0         0.0             0         11.713252   
3                    0      0.0         0.0             0         11.693383   
4                    0      0.0         0.0             0          0.000000   
...                ...      ...         ...           ...               ...   
13379651             0      0.0         0.0             0         10.838526   
13379652             0      0.0         0.0             0         10.689970   
13379653             0      0.0         0.0             0         10.057752   
13379654             0      0.0         0.0             0          0.000000   
13379655             0      0.0         0.0             0         12.204040   

          sqrt_gross_income  winsorized_gross_income  
0                295.327107                 87218.10  
1                188.543735                 35548.74  
2                349.541285                122179.11  
3                346.086030                119775.54  
4                  0.000000                     0.00  
...                     ...                      ...  
13379651         225.710545                 50945.25  
13379652         209.552309                 43912.17  
13379653         152.757946                 23334.99  
13379654           0.000000                     0.00  
13379655         446.758122                199592.82  

[12686208 rows x 48 columns]
```

In [46]:

```python
# Check if 'Gross_income' column exists
if 'Gross_income' in df_cleaned.columns:
    # Apply logarithmic transformation to the 'Gross_income' column
    log_data = np.log1p(df_cleaned['Gross_income'])

    # Create a histogram for the log-transformed 'Gross_income' column
    plt.figure(figsize=(10, 6))
    sns.histplot(log_data, kde=True)
    plt.title('Log-Transformed Distribution of Gross Income')
    plt.xlabel('Log of Gross Income')
    plt.ylabel('Frequency')
    plt.ylim(0, 500000)  # Set the y-axis limit
    plt.show()
else:
    print("The 'Gross_income' column does not exist in the dataset.")
```

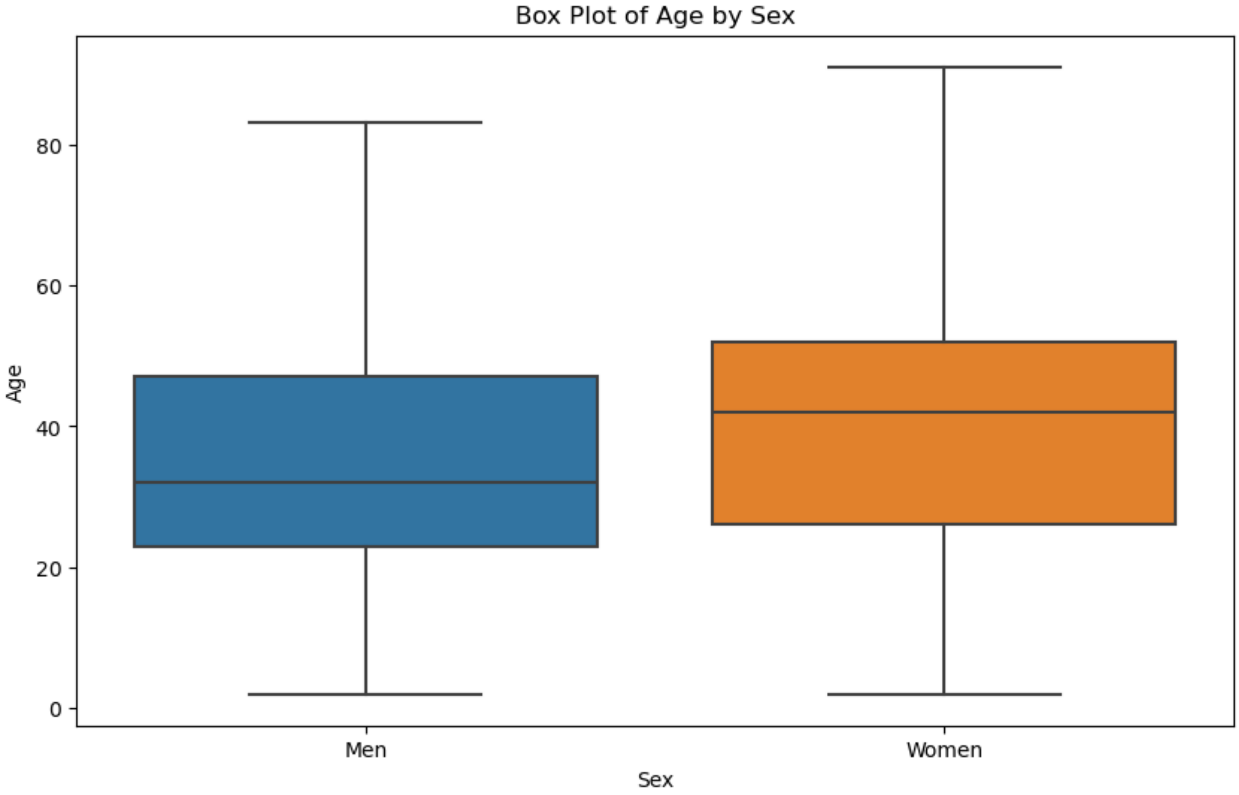


```
In [52]: # Check if Seniority column exists
         if 'Seniority' in df_cleaned.columns:
             # Create a histogram for the Seniority column
             plt.figure(figsize=(10, 6))
             sns.histplot(df_cleaned['Seniority'], kde=True)
             plt.title('Distribution of Seniority')
             plt.xlabel('Seniority')
             plt.ylabel('Frequency')
             plt.show()
         else:
             print("The Seniority column does not exist in the dataset.")
```



In [77]:

```
# Calculate the first quartile (Q1) and the third quartile (Q3) for each
Q1 = df_train.groupby('Sex')['Age'].quantile(0.25)
Q3 = df_train.groupby('Sex')['Age'].quantile(0.75)

# Calculate the Interquartile Range (IQR) for each group
IQR = Q3 - Q1

# Define the lower and upper thresholds for each group
lower_threshold = Q1 - 1.5 * IQR
upper_threshold = Q3 + 1.5 * IQR

# Filter the DataFrame to remove outliers
def filter_outliers(group):
    lower = lower_threshold[group.name]
    upper = upper_threshold[group.name]
    return group[(group['Age'] >= lower) & (group['Age'] <= upper)]

df_cleaned = df_train.groupby('Sex').apply(filter_outliers).reset_index(d

# Plot the cleaned data
plt.figure(figsize=(10, 6))
sns.boxplot(x='Sex', y='Age', data=df_cleaned)
plt.title('Box Plot of Age by Sex')
plt.xlabel('Sex')
plt.ylabel('Age')
plt.show()
```



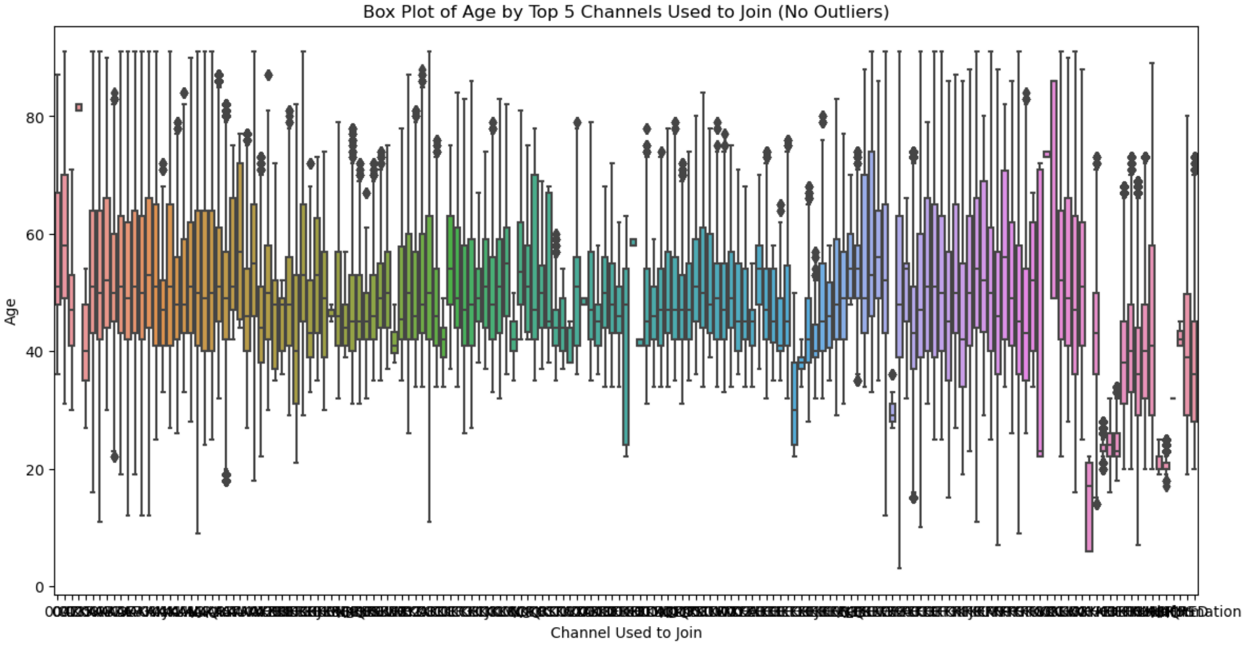
In [87]:

```
# Calculate the first quartile (Q1) and the third quartile (Q3) for each
Q1 = df_cleaned.groupby('Channel_used_to_join')['Age'].quantile(0.25)
Q3 = df_cleaned.groupby('Channel_used_to_join')['Age'].quantile(0.75)

# Calculate the Interquartile Range (IQR) for each channel
IQR = Q3 - Q1
```

```python
# Define the lower and upper thresholds for each channel
lower_threshold = Q1 - 1.5 * IQR
upper_threshold = Q3 + 1.5 * IQR

# Function to filter out outliers based on IQR
def filter_outliers(group):
    lower = lower_threshold[group.name]
    upper = upper_threshold[group.name]
    return group[(group['Age'] >= lower) & (group['Age'] <= upper)]

# Apply the filter function to each group
df_no_outliers = df_cleaned.groupby('Channel_used_to_join').apply(filter_
```

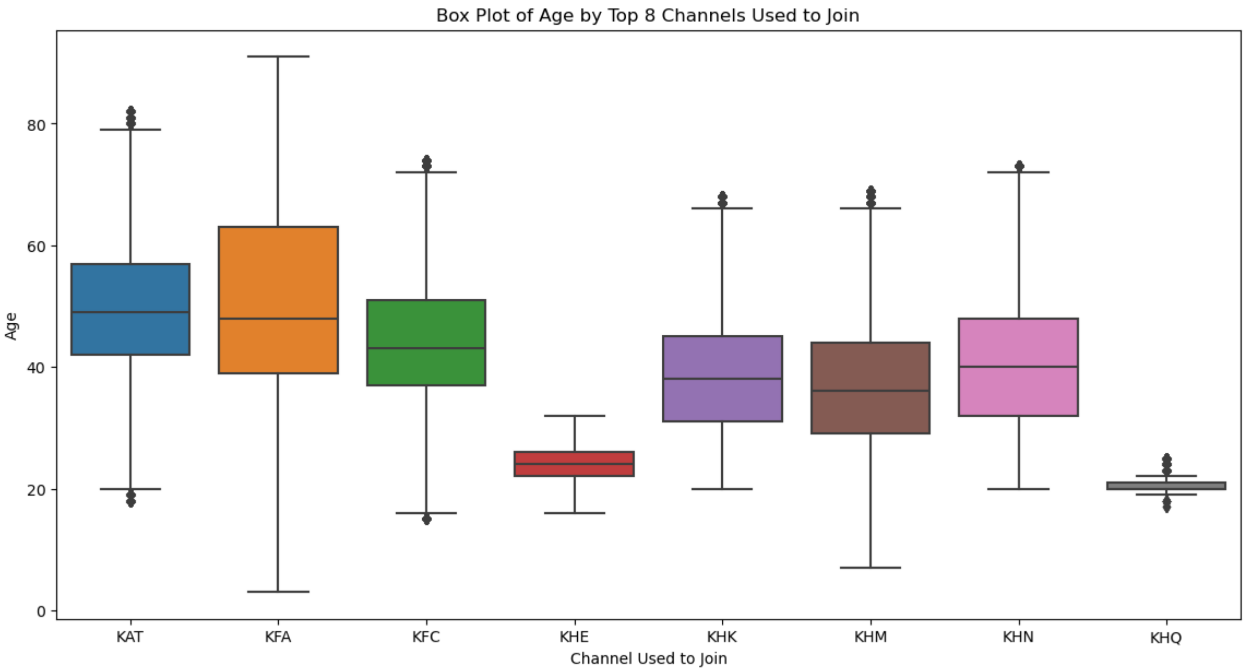
```
          fecha_dato  Customer_Code Employee_index Country_of_Residence  \
0         2015-01-28         851959   Not employed                   ES
1         2015-02-28         851959   Not employed                   ES
2         2015-03-28         851959   Not employed                   ES
3         2015-04-28         851959   Not employed                   ES
4         2015-05-28         851959   Not employed                   ES
...              ...            ...            ...                  ...
12617096  2016-05-28        1173589   Not employed                   ES
12617097  2016-05-28        1172121   Not employed                   ES
12617098  2016-05-28        1171673   Not employed                   ES
12617099  2016-05-28        1165203   Not employed                   ES
12617100  2016-05-28        1168589   Not employed                   ES

            Sex  Age  fecha_alta  New_customer_Index  Seniority  \
0           Men   36  2009-09-15                   0         69
1           Men   36  2009-09-15                   0         69
2           Men   36  2009-09-15                   0         69
3           Men   36  2009-09-15                   0         69
4           Men   36  2009-09-15                   0         69
...         ...  ...         ...                 ...        ...
12617096  Women   38  2014-05-13                   0         26
12617097  Women   32  2013-09-03                   0         28
12617098  Women   21  2013-09-02                   0         12
12617099  Women   24  2013-08-13                   0         23
12617100  Women   59  2013-09-17                   0         23

         Customer_Type_1st_month  ... Taxes Credit_card Securities  \
0                              P  ...     0           0          0
1                              P  ...     0           0          0
2                              P  ...     0           0          0
3                              P  ...     0           0          0
4                              P  ...     0           0          0
...                          ...  ...   ...         ...        ...
12617096                       1  ...     0           0          0
12617097                       1  ...     0           0          0
12617098                       P  ...     0           0          0
12617099                       1  ...     0           0          0
12617100                       P  ...     0           0          0

         Home_account Payroll Pensions.1  Direct_debit log_gross_income  \
0                   0     0.0        0.0             0        12.010900
1                   0     0.0        0.0             0        12.010900
2                   0     0.0        0.0             0        12.010900
3                   0     0.0        0.0             0        12.010900
```

```
4                     0     0.0        0.0              0        12.010900
...                 ...     ...        ...            ...              ...
12617096              0     0.0        0.0              0        11.286214
12617097              0     0.0        0.0              0        10.746474
12617098              0     0.0        0.0              0        11.816476
12617099              0     0.0        0.0              0        11.253567
12617100              0     0.0        0.0              1         0.000000

          sqrt_gross_income  winsorized_gross_income
0                405.632321                164537.58
1                405.632321                164537.58
2                405.632321                164537.58
3                405.632321                164537.58
4                405.632321                164537.58
...                     ...                      ...
12617096         282.336873                 79714.11
12617097         215.557185                 46464.90
12617098         368.055784                135465.06
12617099         277.765387                 77153.61
12617100           0.000000                     0.00

[12617101 rows x 48 columns]
```



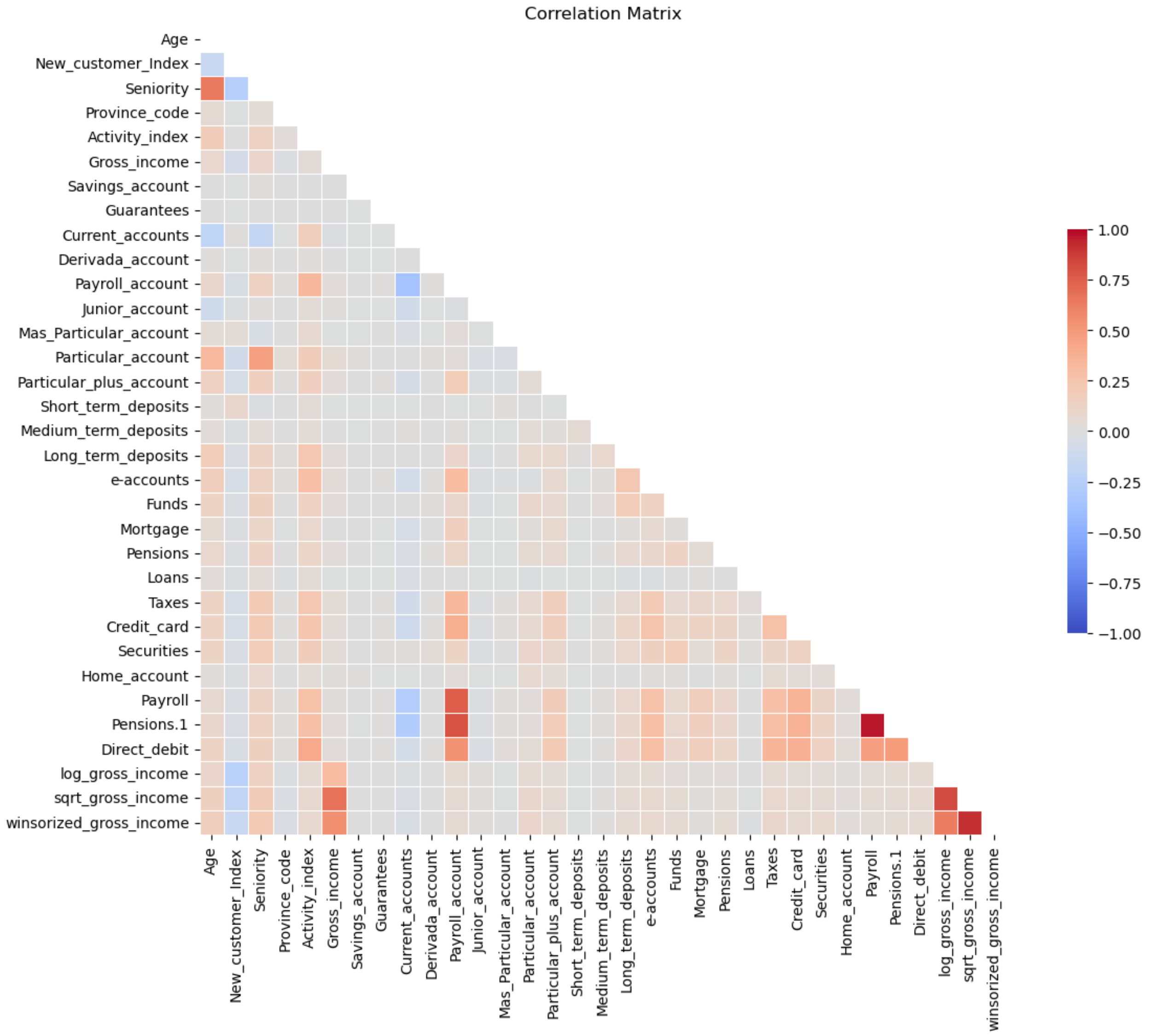
In [93]:

```python
# Calculate the count of each channel
channel_counts = df_no_outliers['Channel_used_to_join'].value_counts()

# Get the top 8 channels
top_8_channels = channel_counts.head(8).index

# Filter the DataFrame to include only the top 5 channels
df_top_8_channels = df_no_outliers[df_no_outliers['Channel_used_to_join']

# Create the box plot
plt.figure(figsize=(14, 7))
sns.boxplot(x='Channel_used_to_join', y='Age', data=df_top_8_channels)
plt.title('Box Plot of Age by Top 8 Channels Used to Join')
plt.xlabel('Channel Used to Join')
plt.ylabel('Age')
plt.show()
```



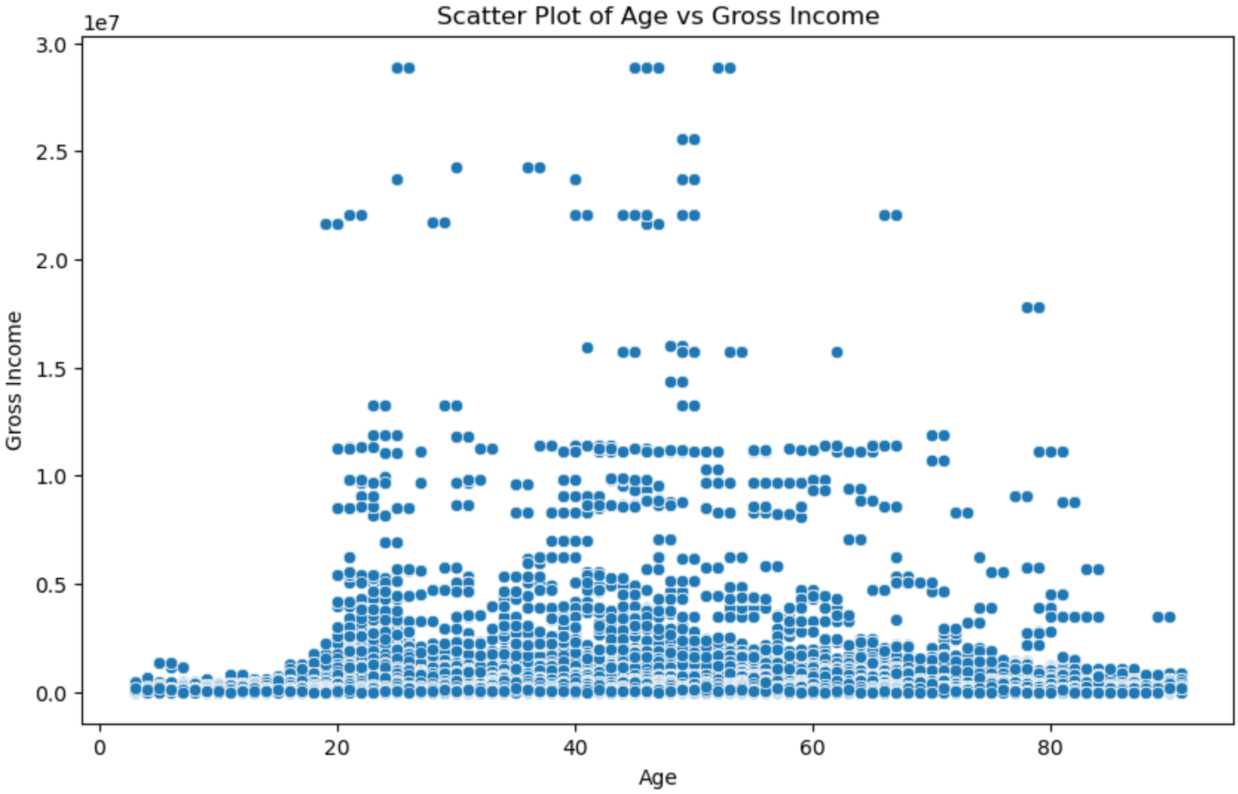
In [99]:

```
# List of columns to exclude from the correlation matrix
columns_to_exclude = ['Customer_Code']  # Add other columns to exclude if

# Drop the specified columns and select numerical columns
relevant_numerical_df = df_no_outliers.drop(columns=columns_to_exclude).s

# Compute the correlation matrix
corr = relevant_numerical_df.corr()

# Generate a mask for the upper triangle
mask = np.triu(np.ones_like(corr, dtype=bool))

# Set up the matplotlib figure
plt.figure(figsize=(15, 10))

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, mask=mask, annot=True, fmt='.2f', cmap='coolwarm', vmin
            square=True, linewidths=.5, cbar_kws={"shrink": .5})

plt.title('Correlation Matrix')
plt.show()
```



In [100…

```python
# Scatter plot between 'Age' and 'Gross_income'
if 'Age' in df_no_outliers.columns and 'Gross_income' in df_no_outliers.c
    plt.figure(figsize=(10, 6))
    sns.scatterplot(x='Age', y='Gross_income', data=df_no_outliers)
    plt.title('Scatter Plot of Age vs Gross Income')
    plt.xlabel('Age')
    plt.ylabel('Gross Income')
    plt.show()
else:
    print("One or more columns do not exist in the dataset.")
```


Scatter Plot of Age vs Gross Income
1e7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Gross Income
0
20
40
60
80
Age