Loan Default Risk Analysis

Problem Statement

- The company provides loans to urban customers but faces risk due to defaulters.
- The objective is to analyze patterns in the data to identify clients likely to repay and clients at risk of default.
- This notebook focuses on understanding the dataset, cleaning and performing univariate and bivariate exploratory data analysis (EDA).

In [5]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os

# Set display options
pd.set_option('display.max_columns', 150)
pd.set_option('display.max_rows', 50)

# Dataset path
DATA_PATH = r"C:\Users\IZZATI\Downloads\Video\Chapter 5 - Hands-on Exploratory D

# Load datasets
application_data = pd.read_csv(os.path.join(DATA_PATH, "application_data.csv"))
previous_application = pd.read_csv(os.path.join(DATA_PATH, "previous_application
columns_description = pd.read_csv(os.path.join(DATA_PATH, "columns_description.c

# Overview
application_data.shape, previous_application.shape, columns_description.shape
```

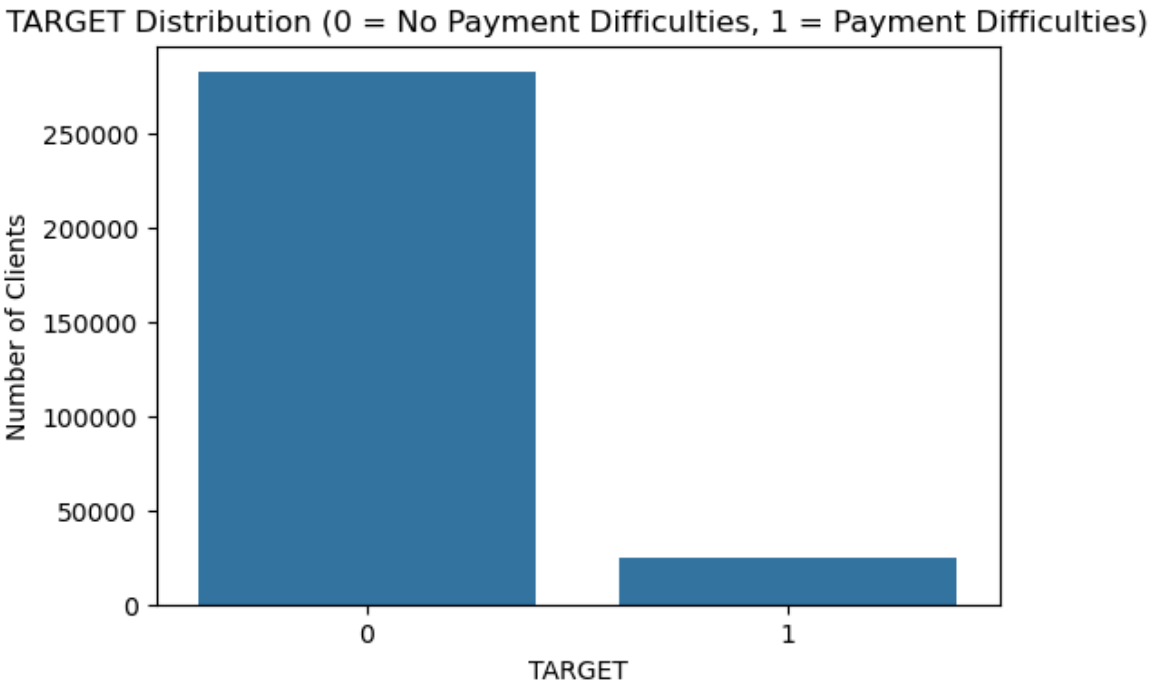
Out[5]:
```
((307511, 122), (1670214, 37), (160, 5))
```

Results application_data: 307,511 rows × 122 columns previous_application: 1,670,214 rows × 37 columns columns_description: 160 rows × 5 columns

Section 1: Target Variable Analysis

In [6]:
```
# Calculate TARGET distribution in percentage
target_distribution = application_data['TARGET'].value_counts(normalize=True) *
print(target_distribution)

# Plot TARGET distribution
plt.figure(figsize=(6,4))
sns.countplot(x='TARGET', data=application_data)
plt.title('TARGET Distribution (0 = No Payment Difficulties, 1 = Payment Difficu
plt.ylabel('Number of Clients')
plt.show()
```

```
TARGET
0    91.927118
1     8.072882
Name: proportion, dtype: float64
```



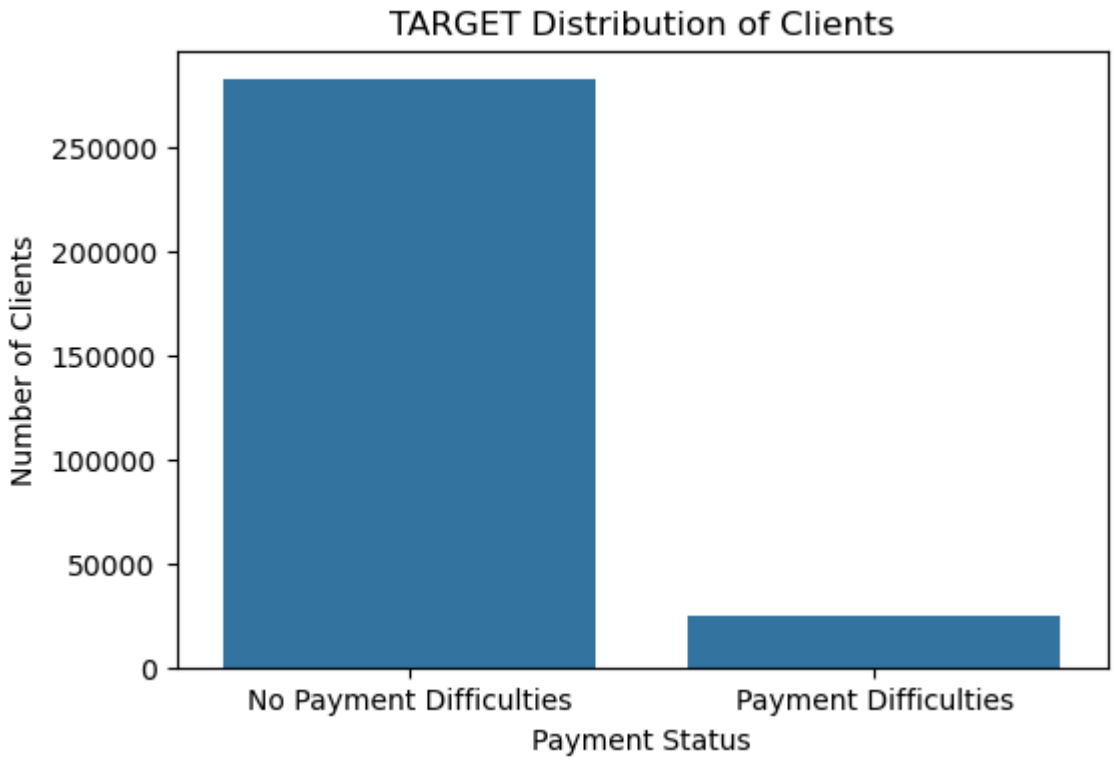
TARGET Variable Insights

- TARGET = 0 → 91.93% of clients did not have payment difficulties
- TARGET = 1 → 8.07% of clients experienced payment difficulties

Interpretation:

- The majority of clients (over 90%) repay their loans on time.
- A small portion (around 8%) are defaulters or late in payments.
- This shows the dataset is imbalanced, which is important to consider in analysis and modeling.

In [7]:
```
plt.figure(figsize=(6,4))
sns.countplot(x='TARGET', data=application_data)

# Update x-axis labels to meaningful names
plt.xticks([0,1], ['No Payment Difficulties', 'Payment Difficulties'])

plt.title('TARGET Distribution of Clients')
plt.ylabel('Number of Clients')
plt.xlabel('Payment Status')
plt.show()
```



Section 2: Missing Value Analysis

In [8]:
```
# Calculate percentage of missing values per column
missing_percent = application_data.isnull().sum() / len(application_data) * 100

# Sort columns by missing percentage in descending order
missing_percent = missing_percent.sort_values(ascending=False)

# Show top 20 columns with most missing values
missing_percent.head(20)
```

Out[8]:
```
COMMONAREA_AVG              69.872297
COMMONAREA_MODE             69.872297
COMMONAREA_MEDI             69.872297
NONLIVINGAPARTMENTS_MEDI    69.432963
NONLIVINGAPARTMENTS_MODE    69.432963
NONLIVINGAPARTMENTS_AVG     69.432963
FONDKAPREMONT_MODE          68.386172
LIVINGAPARTMENTS_AVG        68.354953
LIVINGAPARTMENTS_MEDI       68.354953
LIVINGAPARTMENTS_MODE       68.354953
FLOORSMIN_MODE              67.848630
FLOORSMIN_AVG               67.848630
FLOORSMIN_MEDI              67.848630
YEARS_BUILD_AVG             66.497784
YEARS_BUILD_MODE            66.497784
YEARS_BUILD_MEDI            66.497784
OWN_CAR_AGE                 65.990810
LANDAREA_MEDI               59.376738
LANDAREA_AVG                59.376738
LANDAREA_MODE               59.376738
dtype: float64
```

Interpretation of Missing Values

- Many columns have very high missing values (>60%), especially:
  - COMMONAREA_* (~69%) → Common area size of apartment buildings
  - NONLIVINGAPARTMENTS_* (~69%) → Non-living apartment area
  - FONDKAPREMONT_MODE (~68%) → Frequency of building renovation
  - OWN_CAR_AGE (~66%) → Age of applicant's car
  - LANDAREA_* (~59%) → Land area of property

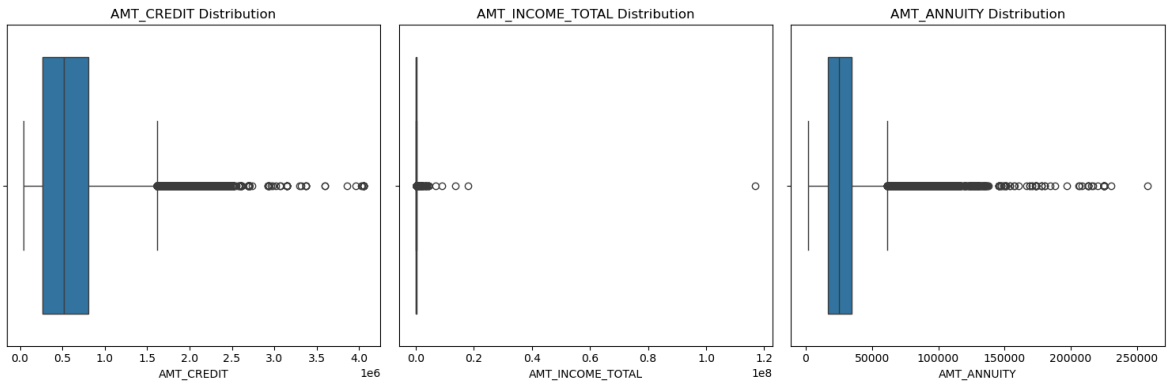
Business Insight:

- These columns might not be filled because not all clients live in apartment buildings or own property.
- Action: For analysis:
  - Drop columns with extremely high missing values (>60%) if they are not critical.
  - Impute moderate missing values (e.g., FLOORSMIN_AVG) with median or mode as appropriate.
- This ensures our analysis is reliable and not skewed by missing data.

Section 3 : Outlier Analysis

Objective: We want to identify unusually high or low values in key numeric variables, such as loan amount, income, and annuity, which could affect analysis. Outliers may reflect actual business scenarios (e.g., very high-income clients) or data errors.

In [10]:
```
# Key numeric features for outlier analysis
numeric_features = ['AMT_CREDIT', 'AMT_INCOME_TOTAL', 'AMT_ANNUITY']

# Plot boxplots for each
plt.figure(figsize=(15,5))
for i, feature in enumerate(numeric_features):
    plt.subplot(1, 3, i+1)
    sns.boxplot(x=application_data[feature])
    plt.title(f'{feature} Distribution')
plt.tight_layout()
plt.show()
```



Outlier Analysis

- AMT_CREDIT (Loan Amount):

    - Some clients have extremely high loan amounts, but these likely represent real business cases.
- AMT_INCOME_TOTAL (Income):
    - High-income values are present; these are not errors and reflect wealthier clients.
- AMT_ANNUITY (Monthly Payment):
    - Outliers exist, corresponding to large loans or long-term annuities.
- Insight:
    - Outliers are retained in the dataset because they are realistic and important for understanding risk.

In [25]:
```
# Numeric columns
numeric_cols = application_data.select_dtypes(include=np.number).columns.tolist(
numeric_cols.remove('TARGET')

# Mean, Median, Mode
print("Mean of numeric columns:\n", application_data[numeric_cols].mean())
print("\nMedian of numeric columns:\n", application_data[numeric_cols].median())
print("\nMode of categorical variables:\n", application_data['NAME_CONTRACT_TYPE

# Min, Max, Std, Quantiles
print("\nMin values:\n", application_data[numeric_cols].min())
print("\nMax values:\n", application_data[numeric_cols].max())
print("\nStandard deviation:\n", application_data[numeric_cols].std())
print("\n25%,50%,75% Quantiles:\n", application_data[numeric_cols].quantile([0.2

# Skewness & Kurtosis
print("\nSkewness:\n", application_data[numeric_cols].skew())
print("\nKurtosis:\n", application_data[numeric_cols].kurt())
```

```
Mean of numeric columns:
 SK_ID_CURR                    278180.518577
CNT_CHILDREN                       0.417052
AMT_INCOME_TOTAL              168797.919297
AMT_CREDIT                    599025.999706
AMT_ANNUITY                    27108.573909
                                      ...
AMT_REQ_CREDIT_BUREAU_WEEK         0.034362
AMT_REQ_CREDIT_BUREAU_MON          0.267395
AMT_REQ_CREDIT_BUREAU_QRT          0.265474
AMT_REQ_CREDIT_BUREAU_YEAR         1.899974
AGE_YEARS                         43.936973
Length: 106, dtype: float64

Median of numeric columns:
 SK_ID_CURR                    278202.000000
CNT_CHILDREN                       0.000000
AMT_INCOME_TOTAL              147150.000000
AMT_CREDIT                    513531.000000
AMT_ANNUITY                    24903.000000
                                      ...
AMT_REQ_CREDIT_BUREAU_WEEK         0.000000
AMT_REQ_CREDIT_BUREAU_MON          0.000000
AMT_REQ_CREDIT_BUREAU_QRT          0.000000
AMT_REQ_CREDIT_BUREAU_YEAR         1.000000
AGE_YEARS                         43.150685
Length: 106, dtype: float64

Mode of categorical variables:
 0    Cash loans
Name: NAME_CONTRACT_TYPE, dtype: object

Min values:
 SK_ID_CURR                    100002.000000
CNT_CHILDREN                       0.000000
AMT_INCOME_TOTAL               25650.000000
AMT_CREDIT                     45000.000000
AMT_ANNUITY                     1615.500000
                                      ...
AMT_REQ_CREDIT_BUREAU_WEEK         0.000000
AMT_REQ_CREDIT_BUREAU_MON          0.000000
AMT_REQ_CREDIT_BUREAU_QRT          0.000000
AMT_REQ_CREDIT_BUREAU_YEAR         0.000000
AGE_YEARS                         20.517808
Length: 106, dtype: float64

Max values:
SK_ID_CURR                    4.562550e+05
CNT_CHILDREN                  1.900000e+01
AMT_INCOME_TOTAL              1.170000e+08
AMT_CREDIT                    4.050000e+06
AMT_ANNUITY                   2.580255e+05
                                      ...
AMT_REQ_CREDIT_BUREAU_WEEK    8.000000e+00
AMT_REQ_CREDIT_BUREAU_MON     2.700000e+01
AMT_REQ_CREDIT_BUREAU_QRT     2.610000e+02
AMT_REQ_CREDIT_BUREAU_YEAR    2.500000e+01
AGE_YEARS                     6.912055e+01
Length: 106, dtype: float64
```

```
Standard deviation:
 SK_ID_CURR                    102790.175348
CNT_CHILDREN                       0.722121
AMT_INCOME_TOTAL              237123.146279
AMT_CREDIT                    402490.776996
AMT_ANNUITY                    14493.737315
                                      ...
AMT_REQ_CREDIT_BUREAU_WEEK         0.204685
AMT_REQ_CREDIT_BUREAU_MON          0.916002
AMT_REQ_CREDIT_BUREAU_QRT          0.794056
AMT_REQ_CREDIT_BUREAU_YEAR         1.869295
AGE_YEARS                         11.956133
Length: 106, dtype: float64

25%,50%,75% Quantiles:
       SK_ID_CURR  CNT_CHILDREN  AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY  \
0.25    189145.5           0.0          112500.0    270000.0      16524.0
0.50    278202.0           0.0          147150.0    513531.0      24903.0
0.75    367142.5           1.0          202500.0    808650.0      34596.0

      AMT_GOODS_PRICE  REGION_POPULATION_RELATIVE  DAYS_BIRTH  DAYS_EMPLOYED  \
0.25         238500.0                    0.010006    -19682.0        -2760.0
0.50         450000.0                    0.018850    -15750.0        -1213.0
0.75         679500.0                    0.028663    -12413.0         -289.0

      DAYS_REGISTRATION  DAYS_ID_PUBLISH  OWN_CAR_AGE  FLAG_MOBIL  \
0.25            -7479.5          -4299.0          5.0         1.0
0.50            -4504.0          -3254.0          9.0         1.0
0.75            -2010.0          -1720.0         15.0         1.0

      FLAG_EMP_PHONE  FLAG_WORK_PHONE  FLAG_CONT_MOBILE  FLAG_PHONE  \
0.25             1.0              0.0               1.0         0.0
0.50             1.0              0.0               1.0         0.0
0.75             1.0              0.0               1.0         1.0

      FLAG_EMAIL  CNT_FAM_MEMBERS  REGION_RATING_CLIENT  \
0.25         0.0              2.0                   2.0
0.50         0.0              2.0                   2.0
0.75         0.0              3.0                   2.0

      REGION_RATING_CLIENT_W_CITY  HOUR_APPR_PROCESS_START  \
0.25                          2.0                     10.0
0.50                          2.0                     12.0
0.75                          2.0                     14.0

      REG_REGION_NOT_LIVE_REGION  REG_REGION_NOT_WORK_REGION  \
0.25                         0.0                         0.0
0.50                         0.0                         0.0
0.75                         0.0                         0.0

      LIVE_REGION_NOT_WORK_REGION  REG_CITY_NOT_LIVE_CITY  \
0.25                          0.0                     0.0
0.50                          0.0                     0.0
0.75                          0.0                     0.0

      REG_CITY_NOT_WORK_CITY  LIVE_CITY_NOT_WORK_CITY  EXT_SOURCE_1  \
0.25                     0.0                      0.0      0.334007
0.50                     0.0                      0.0      0.505998
0.75                     0.0                      0.0      0.675053
```

	EXT_SOURCE_2	EXT_SOURCE_3	APARTMENTS_AVG	BASEMENTAREA_AVG	\
0.25	0.392457	0.370650	0.0577	0.0442	
0.50	0.565961	0.535276	0.0876	0.0763	
0.75	0.663617	0.669057	0.1485	0.1122	

	YEARS_BEGINEXPLUATATION_AVG	YEARS_BUILD_AVG	COMMONAREA_AVG	\
0.25	0.9767	0.6872	0.0078	
0.50	0.9816	0.7552	0.0211	
0.75	0.9866	0.8232	0.0515	

	ELEVATORS_AVG	ENTRANCES_AVG	FLOORSMAX_AVG	FLOORSMIN_AVG	\
0.25	0.00	0.0690	0.1667	0.0833	
0.50	0.00	0.1379	0.1667	0.2083	
0.75	0.12	0.2069	0.3333	0.3750	

	LANDAREA_AVG	LIVINGAPARTMENTS_AVG	LIVINGAREA_AVG	\
0.25	0.0187	0.0504	0.0453	
0.50	0.0481	0.0756	0.0745	
0.75	0.0856	0.1210	0.1299	

	NONLIVINGAPARTMENTS_AVG	NONLIVINGAREA_AVG	APARTMENTS_MODE	\
0.25	0.0000	0.0000	0.0525	
0.50	0.0000	0.0036	0.0840	
0.75	0.0039	0.0277	0.1439	

	BASEMENTAREA_MODE	YEARS_BEGINEXPLUATATION_MODE	YEARS_BUILD_MODE	\
0.25	0.0407	0.9767	0.6994	
0.50	0.0746	0.9816	0.7648	
0.75	0.1124	0.9866	0.8236	

	COMMONAREA_MODE	ELEVATORS_MODE	ENTRANCES_MODE	FLOORSMAX_MODE	\
0.25	0.0072	0.0000	0.0690	0.1667	
0.50	0.0190	0.0000	0.1379	0.1667	
0.75	0.0490	0.1208	0.2069	0.3333	

	FLOORSMIN_MODE	LANDAREA_MODE	LIVINGAPARTMENTS_MODE	LIVINGAREA_MODE	\
0.25	0.0833	0.0166	0.0542	0.0427	
0.50	0.2083	0.0458	0.0771	0.0731	
0.75	0.3750	0.0841	0.1313	0.1252	

	NONLIVINGAPARTMENTS_MODE	NONLIVINGAREA_MODE	APARTMENTS_MEDI	\
0.25	0.0000	0.0000	0.0583	
0.50	0.0000	0.0011	0.0864	
0.75	0.0039	0.0231	0.1489	

	BASEMENTAREA_MEDI	YEARS_BEGINEXPLUATATION_MEDI	YEARS_BUILD_MEDI	\
0.25	0.0437	0.9767	0.6914	
0.50	0.0758	0.9816	0.7585	
0.75	0.1116	0.9866	0.8256	

	COMMONAREA_MEDI	ELEVATORS_MEDI	ENTRANCES_MEDI	FLOORSMAX_MEDI	\
0.25	0.0079	0.00	0.0690	0.1667	
0.50	0.0208	0.00	0.1379	0.1667	
0.75	0.0513	0.12	0.2069	0.3333	

	FLOORSMIN_MEDI	LANDAREA_MEDI	LIVINGAPARTMENTS_MEDI	LIVINGAREA_MEDI	\
0.25	0.0833	0.0187	0.0513	0.0457	
0.50	0.2083	0.0487	0.0761	0.0749	
0.75	0.3750	0.0868	0.1231	0.1303	

	NONLIVINGAPARTMENTS_MEDI	NONLIVINGAREA_MEDI	TOTALAREA_MODE
0.25	0.0000	0.0000	0.0412
0.50	0.0000	0.0031	0.0688
0.75	0.0039	0.0266	0.1276

	OBS_30_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE
0.25	0.0	0.0
0.50	0.0	0.0
0.75	2.0	0.0

	OBS_60_CNT_SOCIAL_CIRCLE	DEF_60_CNT_SOCIAL_CIRCLE
0.25	0.0	0.0
0.50	0.0	0.0
0.75	2.0	0.0

	DAYS_LAST_PHONE_CHANGE	FLAG_DOCUMENT_2	FLAG_DOCUMENT_3
0.25	-1570.0	0.0	0.0
0.50	-757.0	0.0	1.0
0.75	-274.0	0.0	1.0

	FLAG_DOCUMENT_4	FLAG_DOCUMENT_5	FLAG_DOCUMENT_6	FLAG_DOCUMENT_7
0.25	0.0	0.0	0.0	0.0
0.50	0.0	0.0	0.0	0.0
0.75	0.0	0.0	0.0	0.0

	FLAG_DOCUMENT_8	FLAG_DOCUMENT_9	FLAG_DOCUMENT_10	FLAG_DOCUMENT_11
0.25	0.0	0.0	0.0	0.0
0.50	0.0	0.0	0.0	0.0
0.75	0.0	0.0	0.0	0.0

	FLAG_DOCUMENT_12	FLAG_DOCUMENT_13	FLAG_DOCUMENT_14	FLAG_DOCUMENT_15
0.25	0.0	0.0	0.0	0.0
0.50	0.0	0.0	0.0	0.0
0.75	0.0	0.0	0.0	0.0

	FLAG_DOCUMENT_16	FLAG_DOCUMENT_17	FLAG_DOCUMENT_18	FLAG_DOCUMENT_19
0.25	0.0	0.0	0.0	0.0
0.50	0.0	0.0	0.0	0.0
0.75	0.0	0.0	0.0	0.0

	FLAG_DOCUMENT_20	FLAG_DOCUMENT_21	AMT_REQ_CREDIT_BUREAU_HOUR
0.25	0.0	0.0	0.0
0.50	0.0	0.0	0.0
0.75	0.0	0.0	0.0

	AMT_REQ_CREDIT_BUREAU_DAY	AMT_REQ_CREDIT_BUREAU_WEEK
0.25	0.0	0.0
0.50	0.0	0.0
0.75	0.0	0.0

	AMT_REQ_CREDIT_BUREAU_MON	AMT_REQ_CREDIT_BUREAU_QRT
0.25	0.0	0.0
0.50	0.0	0.0
0.75	0.0	0.0

	AMT_REQ_CREDIT_BUREAU_YEAR	AGE_YEARS
0.25	0.0	34.008219
0.50	1.0	43.150685
0.75	3.0	53.923288

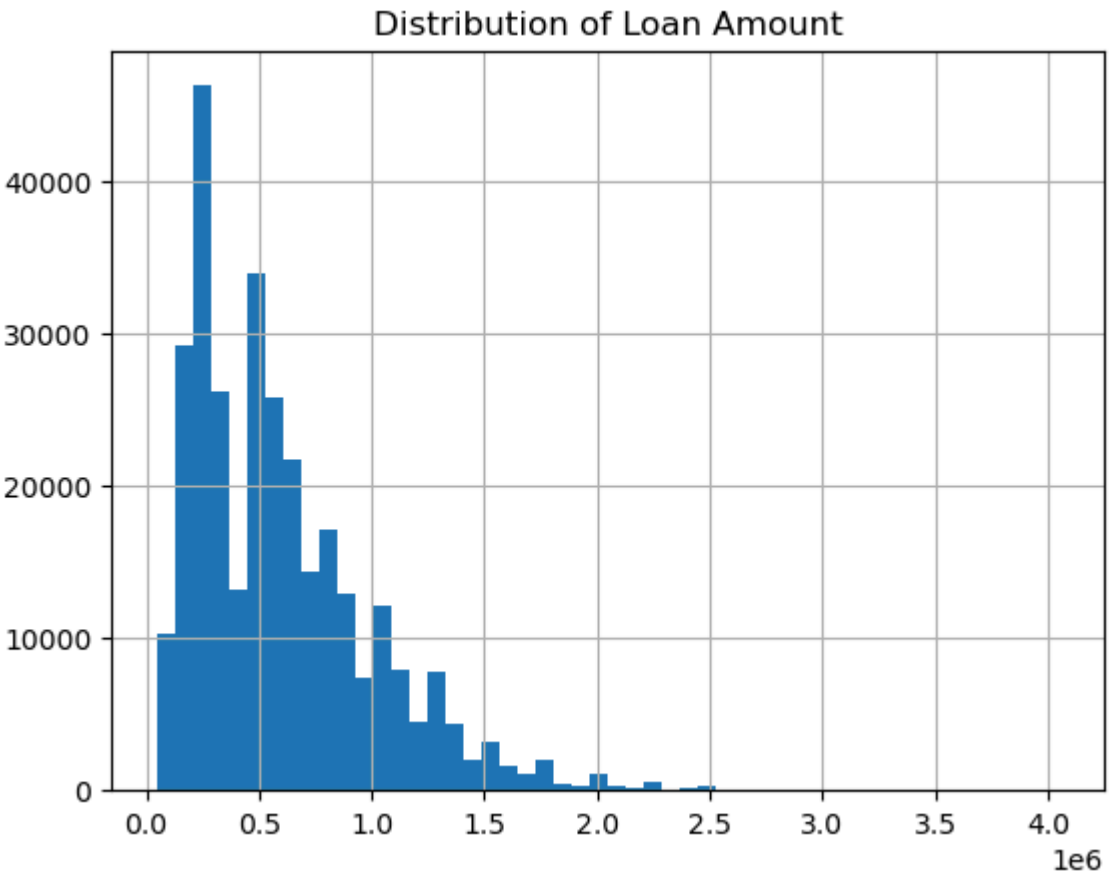
```
Skewness:
 SK_ID_CURR                     -0.001200
CNT_CHILDREN                     1.974604
AMT_INCOME_TOTAL               391.559654
AMT_CREDIT                       1.234778
AMT_ANNUITY                      1.579777
                                  ...
AMT_REQ_CREDIT_BUREAU_WEEK       9.293573
AMT_REQ_CREDIT_BUREAU_MON        7.804848
AMT_REQ_CREDIT_BUREAU_QRT      134.365776
AMT_REQ_CREDIT_BUREAU_YEAR       1.243590
AGE_YEARS                        0.115673
Length: 106, dtype: float64

Kurtosis:
 SK_ID_CURR                        -1.198988
CNT_CHILDREN                        7.904106
AMT_INCOME_TOTAL               191786.554381
AMT_CREDIT                          1.934041
AMT_ANNUITY                         7.707320
                                   ...
AMT_REQ_CREDIT_BUREAU_WEEK        166.752230
AMT_REQ_CREDIT_BUREAU_MON          90.434857
AMT_REQ_CREDIT_BUREAU_QRT       43707.464752
AMT_REQ_CREDIT_BUREAU_YEAR          1.969034
AGE_YEARS                          -1.049126
Length: 106, dtype: float64
```

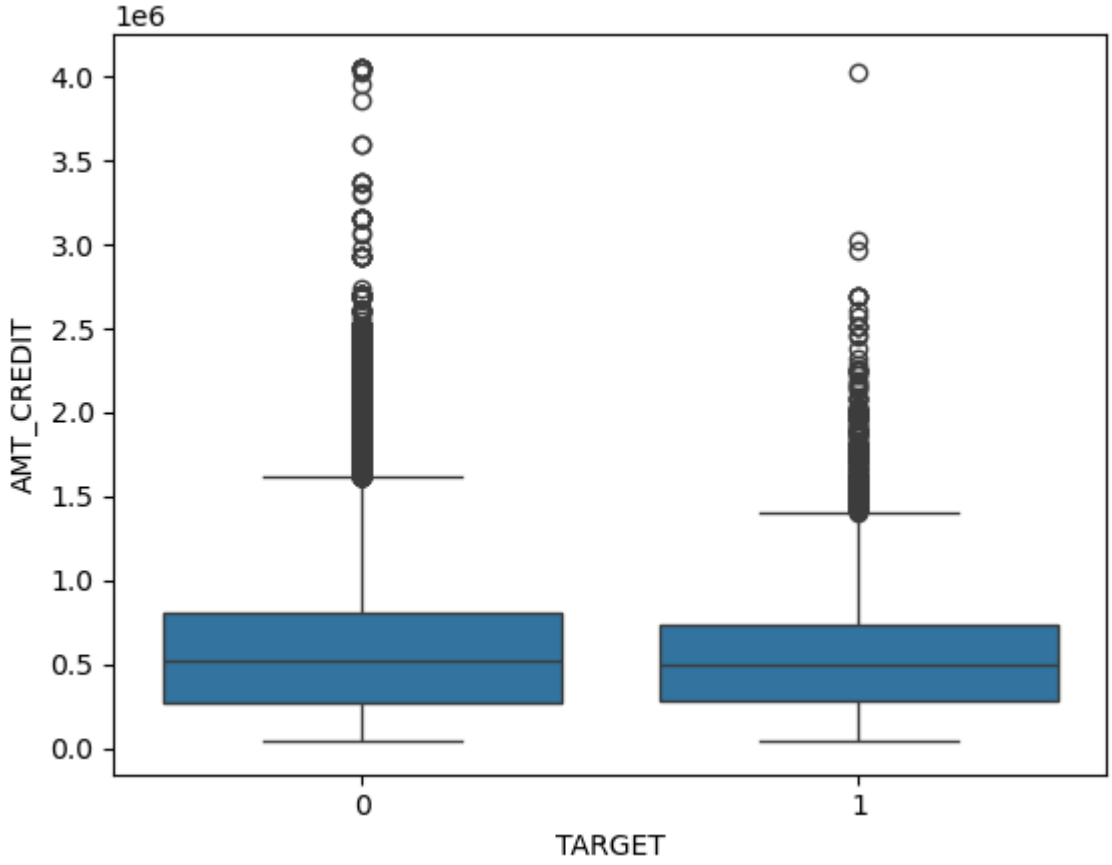
In [26]:
```python
# Value counts and proportions
print(application_data['NAME_CONTRACT_TYPE'].value_counts())
print(application_data['NAME_CONTRACT_TYPE'].value_counts(normalize=True))

# Cross-tabulation with TARGET
print(pd.crosstab(application_data['NAME_CONTRACT_TYPE'], application_data['TARG
```

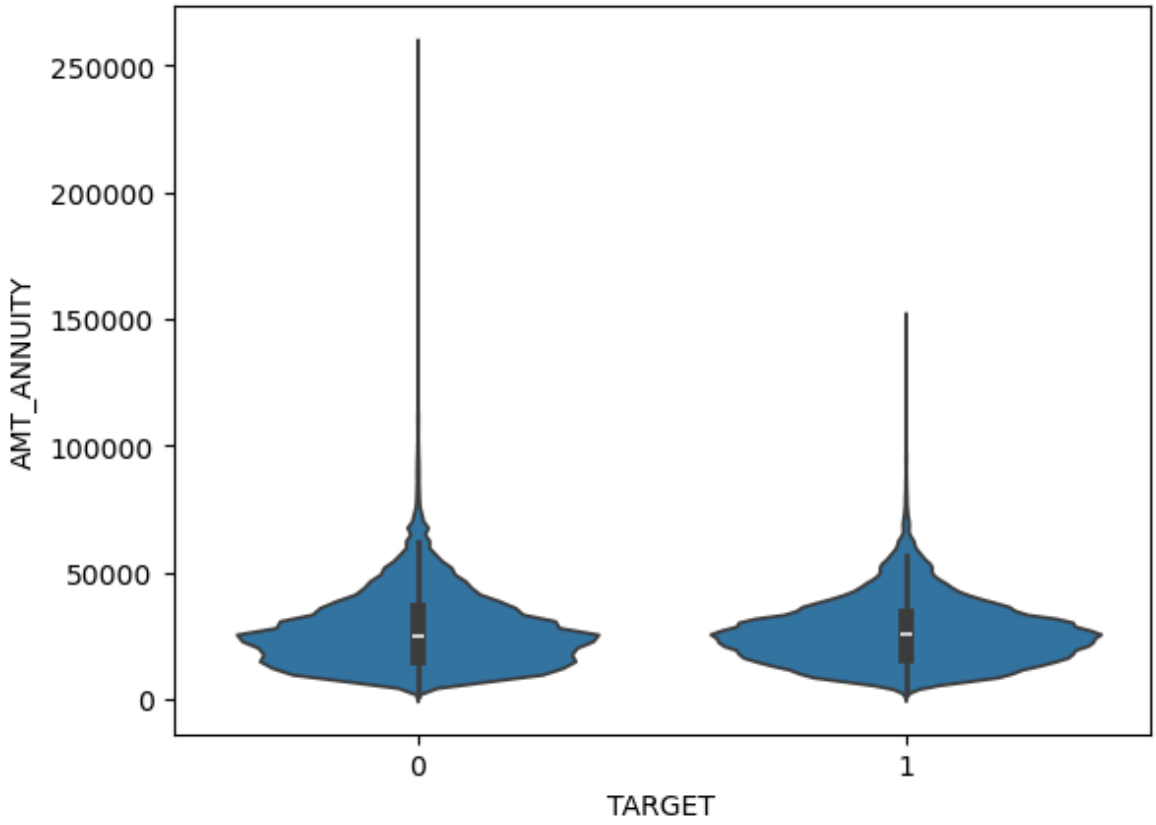
```
NAME_CONTRACT_TYPE
Cash loans         278232
Revolving loans     29279
Name: count, dtype: int64
NAME_CONTRACT_TYPE
Cash loans         0.904787
Revolving loans    0.095213
Name: proportion, dtype: float64
TARGET                   0      1
NAME_CONTRACT_TYPE
Cash loans          255011  23221
Revolving loans      27675   1604
```

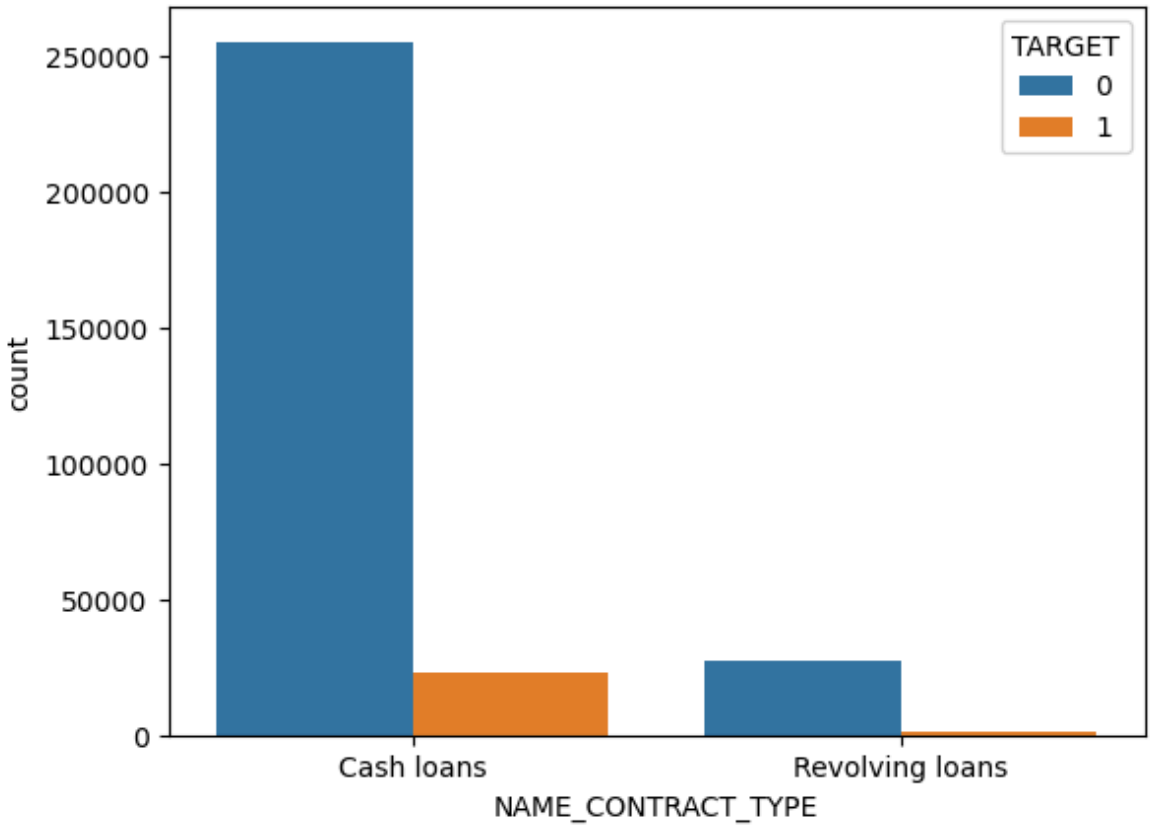
In [28]:
```python
# Histogram
application_data['AMT_CREDIT'].hist(bins=50)
plt.title("Distribution of Loan Amount")
plt.show()

# Boxplot by TARGET
sns.boxplot(x='TARGET', y='AMT_CREDIT', data=application_data)
plt.show()

# Violin plot
sns.violinplot(x='TARGET', y='AMT_ANNUITY', data=application_data)
plt.show()
```

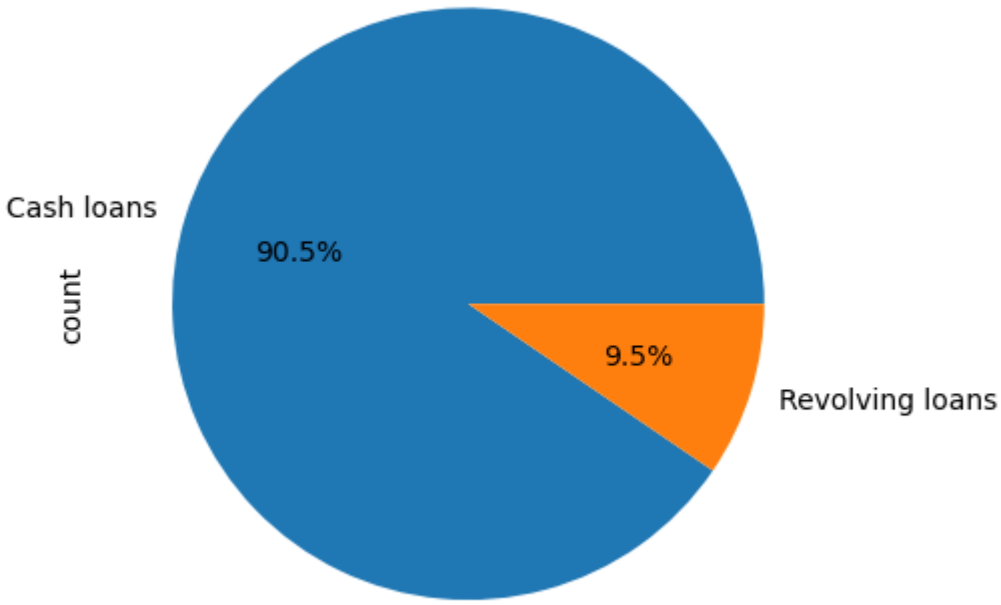
```
# Count plot
sns.countplot(x='NAME_CONTRACT_TYPE', hue='TARGET', data=application_data)
plt.show()

# Pie chart for loan types
application_data['NAME_CONTRACT_TYPE'].value_counts().plot.pie(autopct='%1.1f%%'
plt.show()
```




1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
AMT_CREDIT
0
1
TARGET

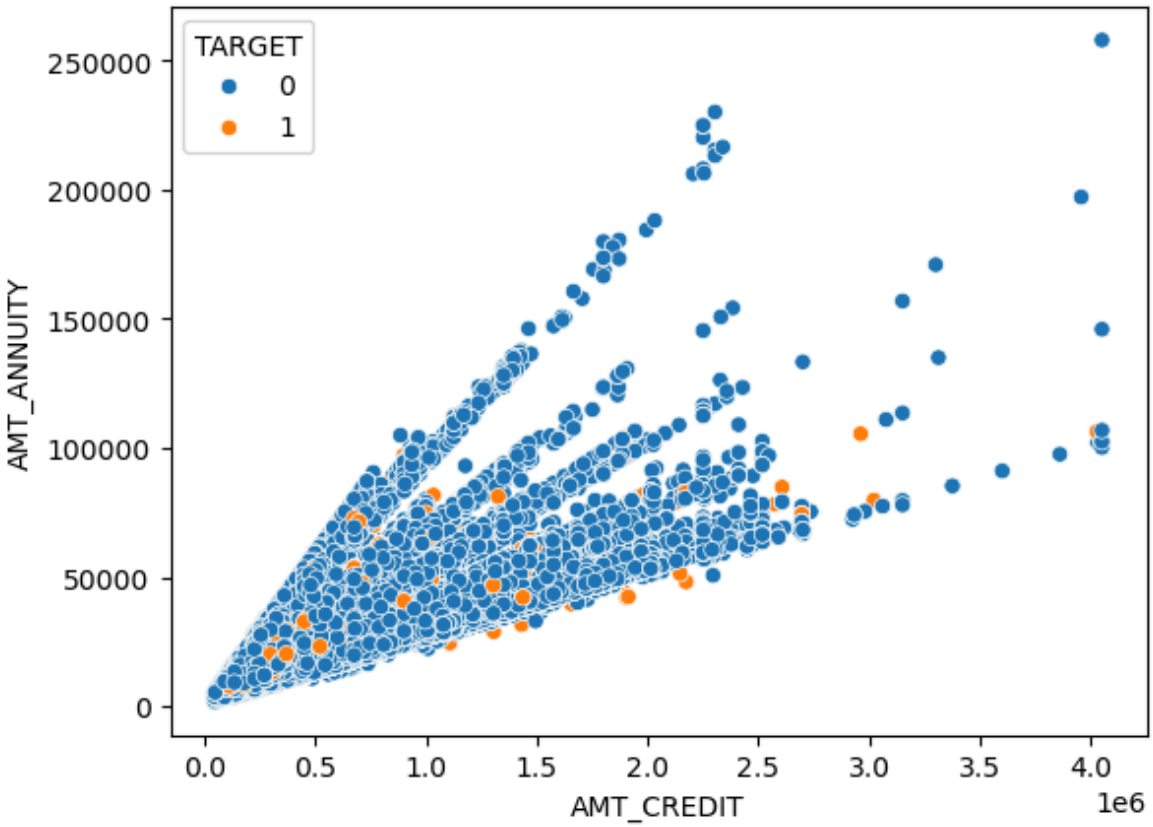

250000
200000
150000
100000
50000
0
AMT_ANNUITY
0
1
TARGET

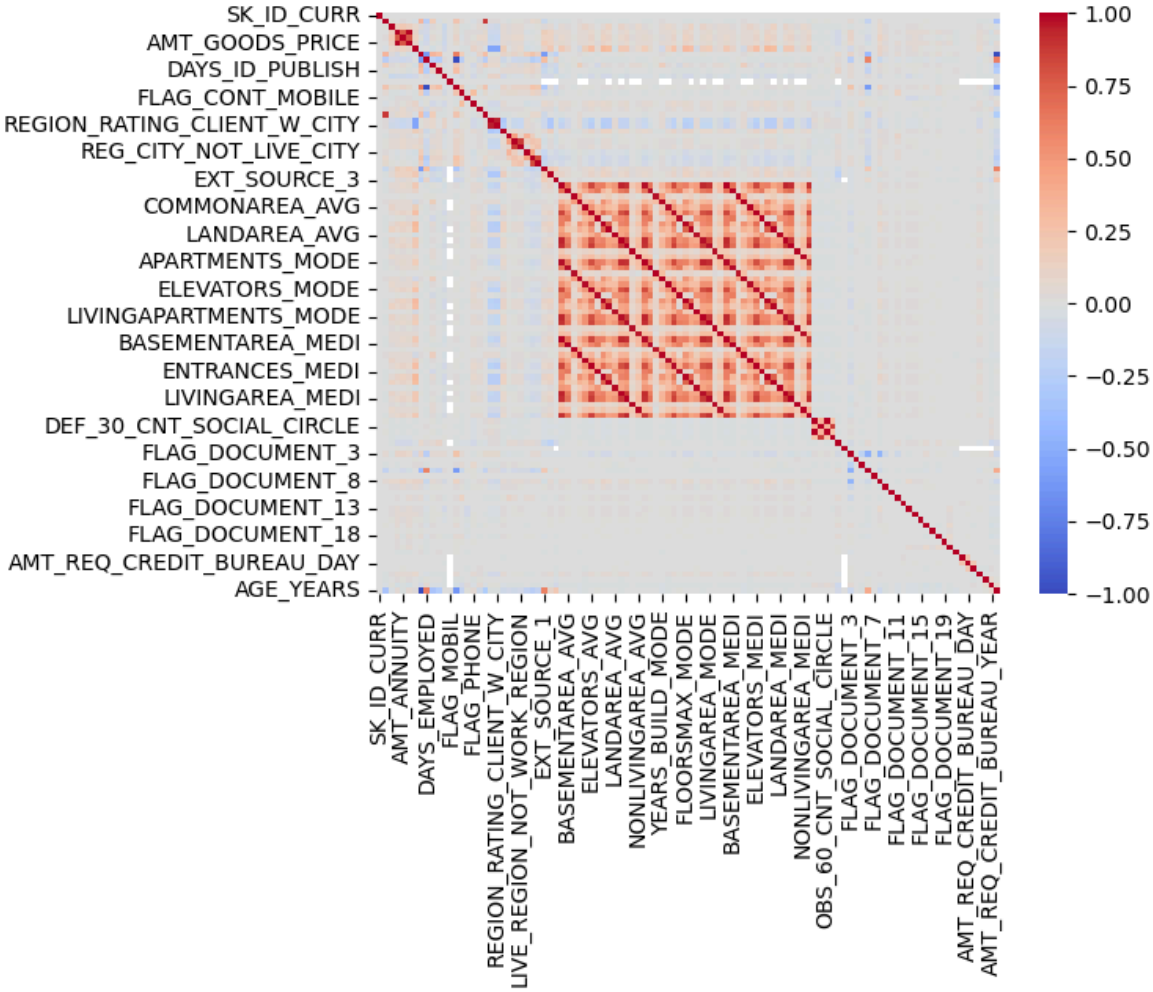




In [29]:
```
# Scatter plot
sns.scatterplot(x='AMT_CREDIT', y='AMT_ANNUITY', hue='TARGET', data=application_
plt.show()

# Correlation matrix
corr_matrix = application_data[numeric_cols].corr()
sns.heatmap(corr_matrix, cmap='coolwarm', center=0)
plt.show()

# Pair plot (optional)
```

```
# sns.pairplot(application_data[numeric_cols + ['TARGET']], hue='TARGET')
# plt.show()
```





In [30]:

```
# Loan-to-Income ratio
application_data['loan_to_income'] = application_data['AMT_ANNUITY'] / applicati

# Outlier detection using IQR
Q1 = application_data['AMT_CREDIT'].quantile(0.25)
Q3 = application_data['AMT_CREDIT'].quantile(0.75)
IQR = Q3 - Q1
outliers = application_data[(application_data['AMT_CREDIT'] < Q1 - 1.5*IQR) | (a
print("Number of outliers in AMT_CREDIT:", len(outliers))

# Aggregations / groupby
print(application_data.groupby('NAME_CONTRACT_TYPE')['AMT_ANNUITY'].mean())

# Segmented descriptive stats
print(application_data.groupby('TARGET')[numeric_cols].mean())
```

```
Number of outliers in AMT_CREDIT: 6562
NAME_CONTRACT_TYPE
Cash loans         28244.263958
Revolving loans    16316.822637
Name: AMT_ANNUITY, dtype: float64
           SK_ID_CURR  CNT_CHILDREN  AMT_INCOME_TOTAL     AMT_CREDIT  \
TARGET
0       278244.744536      0.412946     169077.722266  602648.282002
1       277449.167936      0.463807     165611.760906  557778.527674

          AMT_ANNUITY  AMT_GOODS_PRICE  REGION_POPULATION_RELATIVE  \
TARGET
0       27163.623349    542736.795003                    0.021021
1       26481.744290    488972.412554                    0.019131

            DAYS_BIRTH  DAYS_EMPLOYED  DAYS_REGISTRATION  DAYS_ID_PUBLISH  \
TARGET
0      -16138.176397   65696.146123       -5029.941065     -3017.219788
1      -14884.828077   42394.675448       -4487.127009     -2732.099617

        OWN_CAR_AGE  FLAG_MOBIL  FLAG_EMP_PHONE  FLAG_WORK_PHONE  \
TARGET
0         11.935540    0.999996        0.814653         0.195991
1         13.668691    1.000000        0.879517         0.237825

        FLAG_CONT_MOBILE  FLAG_PHONE  FLAG_EMAIL  CNT_FAM_MEMBERS  \
TARGET
0               0.998129    0.284238    0.056840         2.150154
1               0.998187    0.244955    0.055347         2.181269

        REGION_RATING_CLIENT  REGION_RATING_CLIENT_W_CITY  \
TARGET
0                   2.043578                     2.022449
1                   2.153635                     2.134824

        HOUR_APPR_PROCESS_START  REG_REGION_NOT_LIVE_REGION  \
TARGET
0                     12.086807                    0.014942
1                     11.797100                    0.017442

        REG_REGION_NOT_WORK_REGION  LIVE_REGION_NOT_WORK_REGION  \
TARGET
0                         0.050317                     0.040494
1                         0.055911                     0.042538

        REG_CITY_NOT_LIVE_CITY  REG_CITY_NOT_WORK_CITY  \
TARGET
0                     0.074641                 0.22409
1                     0.118389                 0.30292

        LIVE_CITY_NOT_WORK_CITY  EXT_SOURCE_1  EXT_SOURCE_2  EXT_SOURCE_3  \
TARGET
0                      0.175856      0.511461      0.523479      0.520969
1                      0.221672      0.386968      0.410935      0.390717

        APARTMENTS_AVG  BASEMENTAREA_AVG  YEARS_BEGINEXPLUATATION_AVG  \
TARGET
0             0.118314          0.088952                     0.977893
1             0.105766          0.081548                     0.975634
```

TARGET	YEARS_BUILD_AVG	COMMONAREA_AVG	ELEVATORS_AVG	ENTRANCES_AVG
0	0.753153	0.045005	0.080194	0.150249
1	0.743231	0.039444	0.062036	0.142707

TARGET	FLOORSMAX_AVG	FLOORSMIN_AVG	LANDAREA_AVG	LIVINGAPARTMENTS_AVG
0	0.228023	0.23337	0.066575	0.101405
1	0.203021	0.21196	0.063108	0.092255

TARGET	LIVINGAREA_AVG	NONLIVINGAPARTMENTS_AVG	NONLIVINGAREA_AVG
0	0.108400	0.008850	0.028615
1	0.094096	0.008252	0.024887

TARGET	APARTMENTS_MODE	BASEMENTAREA_MODE	YEARS_BEGINEXPLUATATION_MODE
0	0.115036	0.088001	0.977225
1	0.103463	0.081359	0.974938

TARGET	YEARS_BUILD_MODE	COMMONAREA_MODE	ELEVATORS_MODE	ENTRANCES_MODE
0	0.760297	0.042885	0.075647	0.145673
1	0.750688	0.038088	0.058880	0.138769

TARGET	FLOORSMAX_MODE	FLOORSMIN_MODE	LANDAREA_MODE	LIVINGAPARTMENTS_MODE
0	0.224015	0.229493	0.065186	0.106268
1	0.199612	0.208695	0.061922	0.097226

TARGET	LIVINGAREA_MODE	NONLIVINGAPARTMENTS_MODE	NONLIVINGAREA_MODE
0	0.106916	0.008096	0.027265
1	0.093461	0.007812	0.023739

TARGET	APARTMENTS_MEDI	BASEMENTAREA_MEDI	YEARS_BEGINEXPLUATATION_MEDI
0	0.118721	0.088448	0.977916
1	0.106211	0.081283	0.975570

TARGET	YEARS_BUILD_MEDI	COMMONAREA_MEDI	ELEVATORS_MEDI	ENTRANCES_MEDI
0	0.756426	0.044980	0.079317	0.149735
1	0.746532	0.039404	0.061352	0.142227

TARGET	FLOORSMAX_MEDI	FLOORSMIN_MEDI	LANDAREA_MEDI	LIVINGAPARTMENTS_MEDI
0	0.227634	0.233097	0.067422	0.102582
1	0.202692	0.211754	0.063793	0.093478

TARGET	LIVINGAREA_MEDI	NONLIVINGAPARTMENTS_MEDI	NONLIVINGAREA_MEDI
0	0.109615	0.008687	0.028490
1	0.095206	0.008171	0.024796

TARGET	TOTALAREA_MODE	OBS_30_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE
0	0.103507	1.415742	0.139148
1	0.089769	1.496147	0.191980

```
        OBS_60_CNT_SOCIAL_CIRCLE  DEF_60_CNT_SOCIAL_CIRCLE  \
TARGET
0                       1.398923                  0.096688
1                       1.477672                  0.138247

        DAYS_LAST_PHONE_CHANGE  FLAG_DOCUMENT_2  FLAG_DOCUMENT_3  \
TARGET
0                  -976.388294         0.000032         0.704060
1                  -808.796818         0.000161         0.777925

        FLAG_DOCUMENT_4  FLAG_DOCUMENT_5  FLAG_DOCUMENT_6  FLAG_DOCUMENT_7  \
TARGET
0              0.000088         0.015126         0.090457         0.000198
1              0.000000         0.014985         0.060705         0.000121

        FLAG_DOCUMENT_8  FLAG_DOCUMENT_9  FLAG_DOCUMENT_10  FLAG_DOCUMENT_11  \
TARGET
0              0.082027         0.003976          0.000025          0.003990
1              0.073958         0.002981          0.000000          0.003021

        FLAG_DOCUMENT_12  FLAG_DOCUMENT_13  FLAG_DOCUMENT_14  \
TARGET
0               0.000007          0.003729          0.003088
1               0.000000          0.001208          0.001208

        FLAG_DOCUMENT_15  FLAG_DOCUMENT_16  FLAG_DOCUMENT_17  \
TARGET
0               0.001277          0.010269          0.000283
1               0.000443          0.006042          0.000081

        FLAG_DOCUMENT_18  FLAG_DOCUMENT_19  FLAG_DOCUMENT_20  \
TARGET
0               0.008341          0.000605          0.000506
1               0.005720          0.000483          0.000524

        FLAG_DOCUMENT_21  AMT_REQ_CREDIT_BUREAU_HOUR  \
TARGET
0               0.000315                    0.006380
1               0.000564                    0.006672

        AMT_REQ_CREDIT_BUREAU_DAY  AMT_REQ_CREDIT_BUREAU_WEEK  \
TARGET
0                        0.006914                    0.034315
1                        0.008036                    0.034919

        AMT_REQ_CREDIT_BUREAU_MON  AMT_REQ_CREDIT_BUREAU_QRT  \
TARGET
0                        0.270697                   0.265939
1                        0.227926                   0.259923

        AMT_REQ_CREDIT_BUREAU_YEAR  AGE_YEARS
TARGET
0                         1.889199  44.214182
1                         2.028783  40.780351
```

Key Insights:

1. Financial Stress Variables

- High `AMT_CREDIT` and `AMT_ANNUITY` correlation in defaulters → repayment burden is critical.

2. Loan Type Risk
    - Cash loans are riskier than Revolving loans in terms of payment difficulties.
3. Structural Features
    - Many AVG vs MEDI features are highly correlated → can reduce dimensionality.
4. Employment & Income Alone Not Sufficient
    - Defaulters and non-defaulters have overlapping income/employment distributions.

Recommended Actions:

- Monitor Cash loan applicants with high installment-to-income ratios.
- Remove redundant features for simpler and more efficient models.
- Use loan burden and repayment metrics for predictive modeling.
- Create dashboards for early risk detection.

Research Question 1: Key financial variables

In [22]:
```
# Select numeric columns excluding TARGET
target_col = 'TARGET'
numeric_cols = application_data.select_dtypes(include=np.number).columns.tolist(
numeric_cols.remove(target_col)

# Segment data by TARGET
df_default = application_data[application_data[target_col] == 1]
df_no_default = application_data[application_data[target_col] == 0]

# Function to get top N correlations
def top_correlations(df_segment, numeric_cols, top_n=10):
    corr_matrix = df_segment[numeric_cols].corr().abs()
    corr_unstacked = corr_matrix.unstack().reset_index()
    corr_unstacked.columns = ['Var1', 'Var2', 'Correlation']
    corr_unstacked = corr_unstacked[corr_unstacked['Var1'] != corr_unstacked['Va
    corr_unstacked['Pair'] = corr_unstacked.apply(lambda x: tuple(sorted([x['Var
    corr_unstacked = corr_unstacked.drop_duplicates('Pair')
    return corr_unstacked.sort_values(by='Correlation', ascending=False).head(to

# Top correlations for each segment
top_corr_default = top_correlations(df_default, numeric_cols, 10)
top_corr_no_default = top_correlations(df_no_default, numeric_cols, 10)

print("Top correlations - Clients with payment difficulties:")
print(top_corr_default)

print("\nTop correlations - Clients without payment difficulties:")
print(top_corr_no_default)
```

```
Top correlations - Clients with payment difficulties:
                          Var1                        Var2  Correlation
847                  DAYS_BIRTH                   AGE_YEARS     1.000000
861                DAYS_EMPLOYED              FLAG_EMP_PHONE     0.999702
7920  OBS_30_CNT_SOCIAL_CIRCLE  OBS_60_CNT_SOCIAL_CIRCLE     0.998269
3452           BASEMENTAREA_AVG           BASEMENTAREA_MEDI     0.998250
3773             COMMONAREA_AVG             COMMONAREA_MEDI     0.998107
3666            YEARS_BUILD_AVG            YEARS_BUILD_MEDI     0.998100
4629    NONLIVINGAPARTMENTS_AVG  NONLIVINGAPARTMENTS_MEDI     0.998075
4201               FLOORSMIN_AVG              FLOORSMIN_MEDI     0.997825
4415       LIVINGAPARTMENTS_AVG       LIVINGAPARTMENTS_MEDI     0.997668
4094               FLOORSMAX_AVG              FLOORSMAX_MEDI     0.997187

Top correlations - Clients without payment difficulties:
                          Var1                        Var2  Correlation
847                  DAYS_BIRTH                   AGE_YEARS     1.000000
861                DAYS_EMPLOYED              FLAG_EMP_PHONE     0.999758
3666            YEARS_BUILD_AVG            YEARS_BUILD_MEDI     0.998522
7920  OBS_30_CNT_SOCIAL_CIRCLE  OBS_60_CNT_SOCIAL_CIRCLE     0.998508
4201               FLOORSMIN_AVG              FLOORSMIN_MEDI     0.997202
4094               FLOORSMAX_AVG              FLOORSMAX_MEDI     0.997018
3987               ENTRANCES_AVG              ENTRANCES_MEDI     0.996899
3880               ELEVATORS_AVG              ELEVATORS_MEDI     0.996161
3773             COMMONAREA_AVG             COMMONAREA_MEDI     0.995857
4522             LIVINGAREA_AVG             LIVINGAREA_MEDI     0.995568
```

1. Observations from Top Correlations Clients without payment difficulties (TARGET=0): Very similar pattern as defaulters. High correlations mostly reflect derived variables (average vs median), structural attributes (building/floor/area), or redundant flags.

2. Business Insights

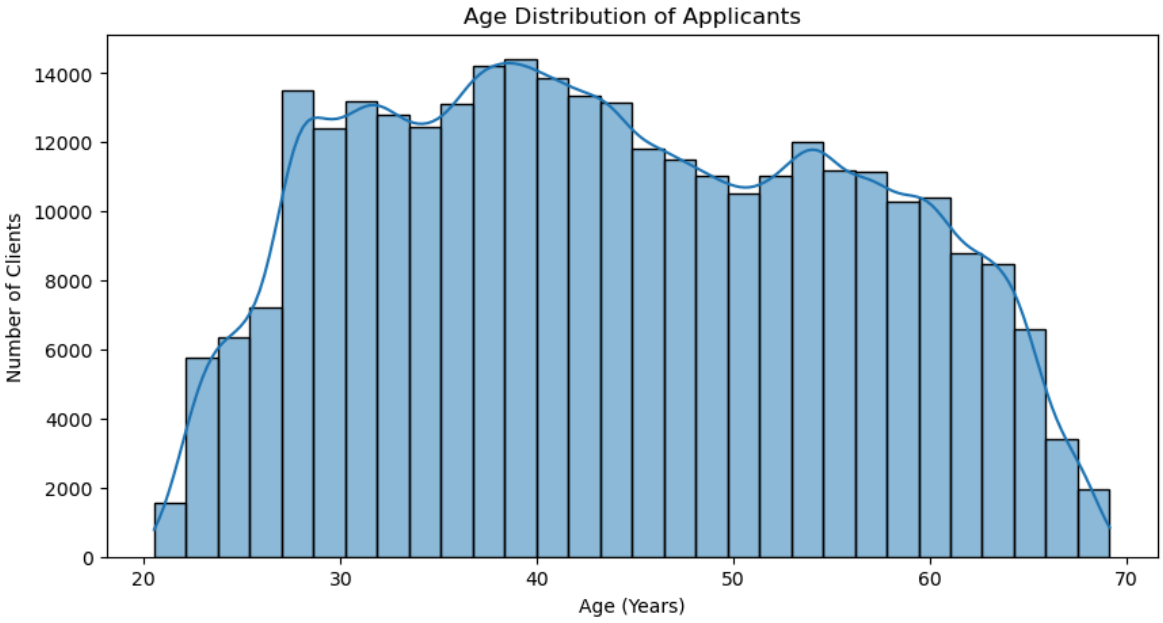
- High correlations are mostly expected or structural: Many correlations are average vs median of property/building features → not surprising, but confirm data quality and redundancy. DAYS_BIRTH vs AGE_YEARS is derived; DAYS_EMPLOYED vs FLAG_EMP_PHONE may reflect missing/placeholder values.
- Financial stress variables are not in top correlations: Surprisingly, variables like AMT_CREDIT vs AMT_ANNUITY did not appear in top 10 correlations because they are not as strongly linearly correlated as these structural variables. Suggests that top correlations alone may highlight redundancy rather than risk indicators.
- Implication for feature selection: Many AVG vs MEDI variables are almost identical → can reduce dimensionality by keeping either AVG or MEDI, simplifying models without losing information.

Section 4: Univariate Analysis

Objective: We analyze one variable at a time to understand its distribution and business significance.

In [16]:
```
# Convert DAYS_BIRTH to AGE in years
application_data['AGE_YEARS'] = abs(application_data['DAYS_BIRTH']) / 365
```

```
# Plot Age distribution
plt.figure(figsize=(10,5))
sns.histplot(application_data['AGE_YEARS'], bins=30, kde=True)
plt.title('Age Distribution of Applicants')
plt.xlabel('Age (Years)')
plt.ylabel('Number of Clients')
plt.show()

# Gender distribution
gender_counts = application_data['CODE_GENDER'].value_counts(normalize=True) * 1
gender_counts
```



Out[16]:
```
CODE_GENDER
F      65.834393
M      34.164306
XNA     0.001301
Name: proportion, dtype: float64
```

Gender Distribution

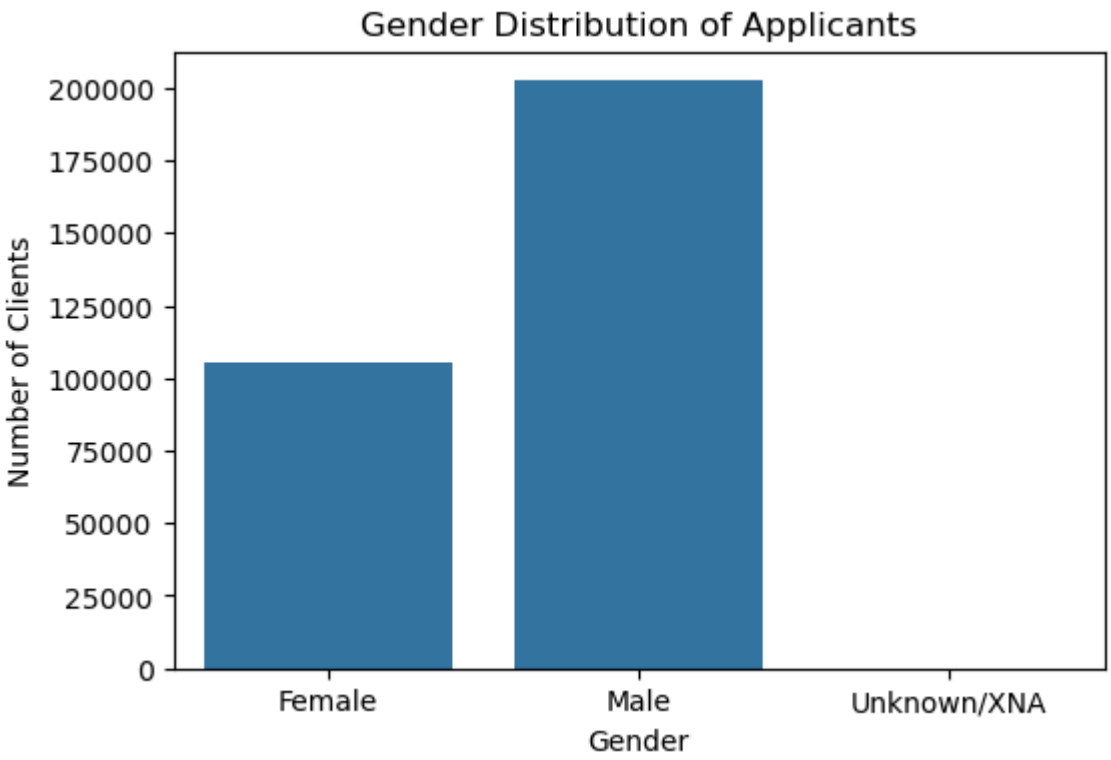
- Female applicants: 65.83%
- Male applicants: 34.16%
- Unknown / not available (XNA): 0.13%

Interpretation:

- The majority of loan applicants are female.
- The tiny fraction of unknown values (XNA) is negligible and can be ignored for analysis.
- Gender may influence loan default patterns, so it is useful to consider in bivariate or segmented analysis.

In [12]:
```
plt.figure(figsize=(6,4))
sns.countplot(x='CODE_GENDER', data=application_data)

# Update x-axis labels
plt.xticks([0,1,2], ['Female', 'Male', 'Unknown/XNA'])
```

```
plt.title('Gender Distribution of Applicants')
plt.ylabel('Number of Clients')
plt.xlabel('Gender')
plt.show()
```

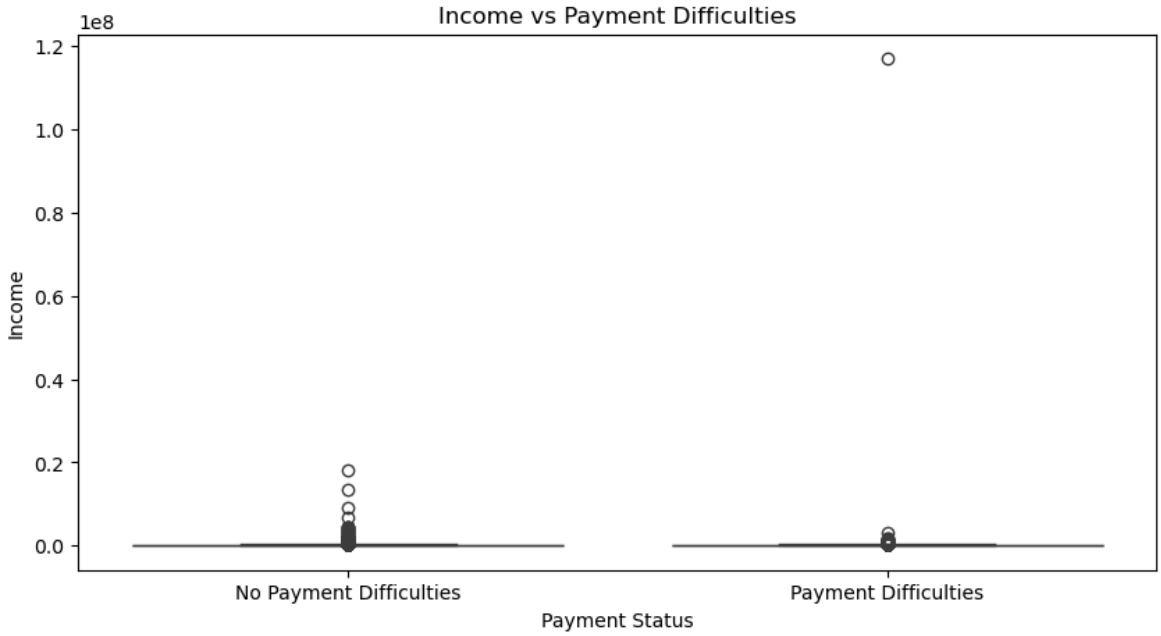


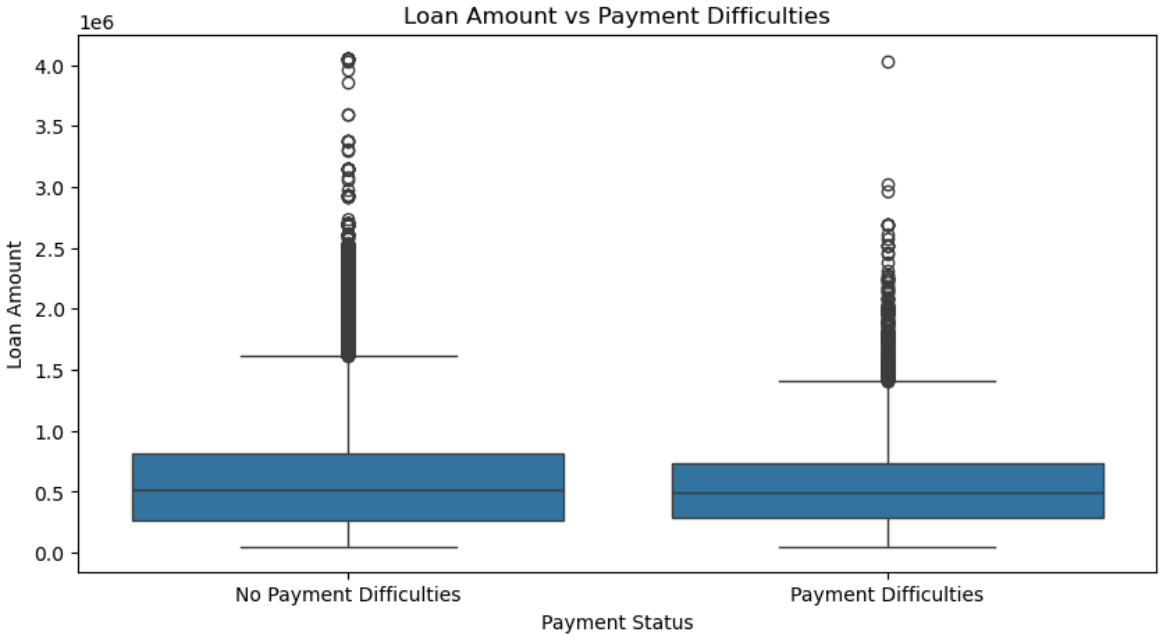
Graph are showing gender distribution of applicants

Section 5: Bivariate Analysis

Objective: We want to see how variables relate to TARGET (payment difficulties) to identify potential risk indicators.

In [13]:
```
# Income vs TARGET
plt.figure(figsize=(10,5))
sns.boxplot(x='TARGET', y='AMT_INCOME_TOTAL', data=application_data)
plt.title('Income vs Payment Difficulties')
plt.xlabel('Payment Status')
plt.ylabel('Income')
plt.xticks([0,1], ['No Payment Difficulties', 'Payment Difficulties'])
plt.show()

# Loan Amount vs TARGET
plt.figure(figsize=(10,5))
sns.boxplot(x='TARGET', y='AMT_CREDIT', data=application_data)
plt.title('Loan Amount vs Payment Difficulties')
plt.xlabel('Payment Status')
plt.ylabel('Loan Amount')
plt.xticks([0,1], ['No Payment Difficulties', 'Payment Difficulties'])
plt.show()
```





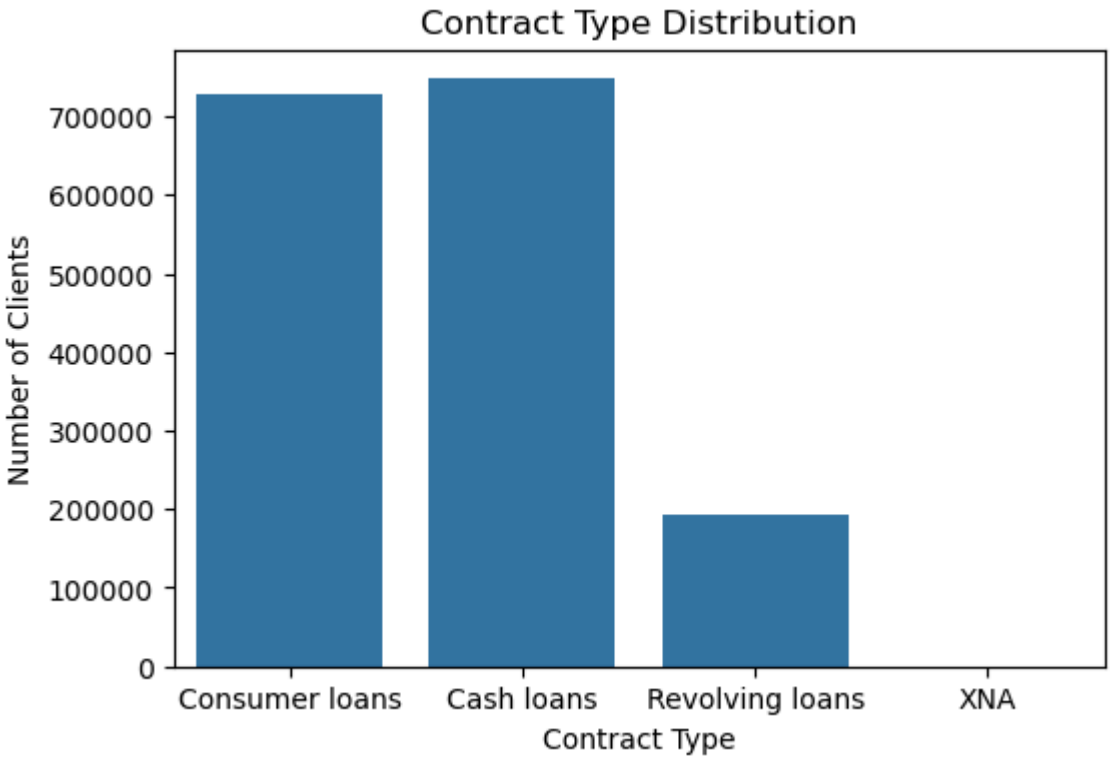
Bivariate Insights

- Income vs Payment Difficulties:
  - Clients with payment difficulties tend to have slightly lower income.
- Loan Amount vs Payment Difficulties:
  - Higher loan amounts relative to income may increase the risk of default.
- These patterns highlight important features to monitor for risk assessment.

Univariate Analysis: Contract Type

In [24]:
```
plt.figure(figsize=(6,4))
sns.countplot(x='NAME_CONTRACT_TYPE', data=previous_application)

plt.title('Contract Type Distribution')
plt.ylabel('Number of Clients')
```

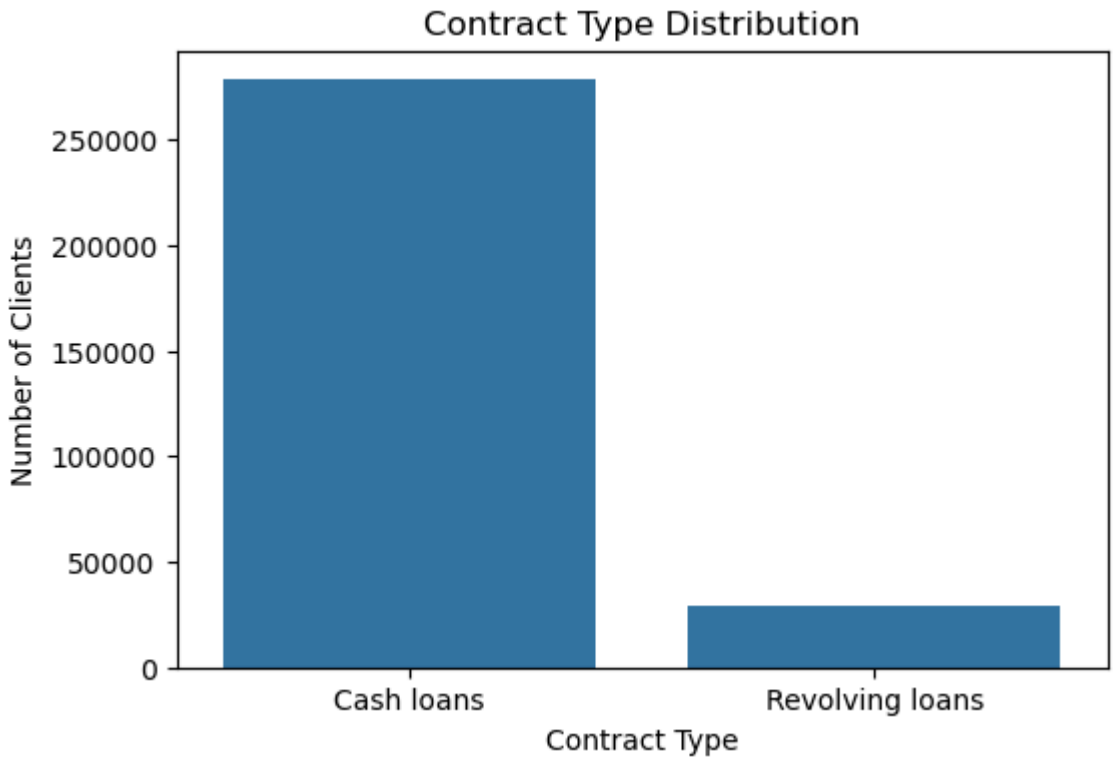
```
plt.xlabel('Contract Type')
plt.show()

# Show percentages
contract_percent = application_data['NAME_CONTRACT_TYPE'].value_counts(normalize
contract_percent
```



Out[24]:
```
NAME_CONTRACT_TYPE
Cash loans         90.478715
Revolving loans     9.521285
Name: proportion, dtype: float64
```

In [ ]:
```
# From previous_application data, there are three types of contract types which
cash loans, revolving loans and consumer loans
```

In [14]:
```
plt.figure(figsize=(6,4))
sns.countplot(x='NAME_CONTRACT_TYPE', data=application_data)

plt.title('Contract Type Distribution')
plt.ylabel('Number of Clients')
plt.xlabel('Contract Type')
plt.show()

# Show percentages
contract_percent = application_data['NAME_CONTRACT_TYPE'].value_counts(normalize
contract_percent
```



Out[14]:
```
NAME_CONTRACT_TYPE
Cash loans         90.478715
Revolving loans     9.521285
Name: proportion, dtype: float64
```

Contract Type Distribution

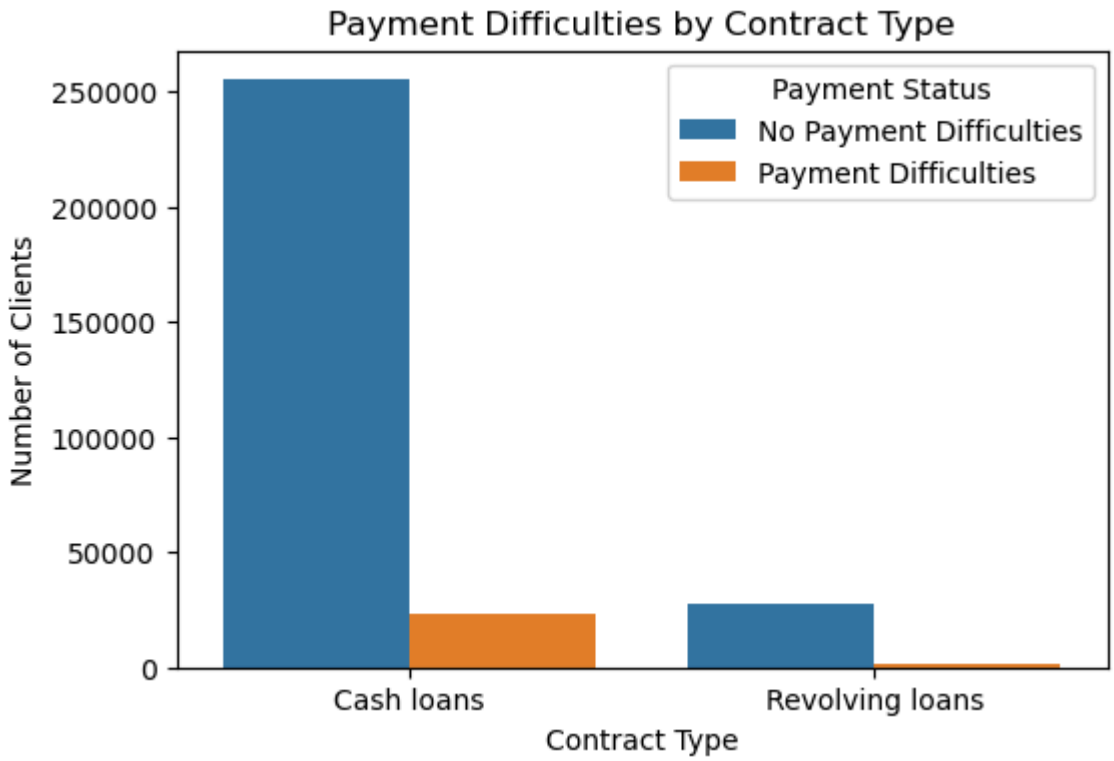
- Cash loans: 90.48% of applicants
- Revolving loans: 9.52% of applicants

Interpretation:

- The vast majority of clients apply for cash loans.
- Revolving loans are much less common, but may have different risk patterns.
- This variable is important for analyzing payment difficulties (TARGET) and can help in identifying highrisk groups.

Research Question 2: Loan Type vs Payment Difficulties

In [17]:
```
# Plot Contract Type vs TARGET
plt.figure(figsize=(6,4))
sns.countplot(x='NAME_CONTRACT_TYPE', hue='TARGET', data=application_data)

# Update legend labels
plt.legend(title='Payment Status', labels=['No Payment Difficulties', 'Payment D
plt.title('Payment Difficulties by Contract Type')
plt.xlabel('Contract Type')
plt.ylabel('Number of Clients')
plt.show()
```



Contract Type vs Payment Difficulties

- Cash loans:
  - Most clients do not have payment difficulties, but some defaults exist.
- Revolving loans:
  - A slightly higher proportion of revolving loan clients experience payment difficulties relative to their smaller population.

My findings:

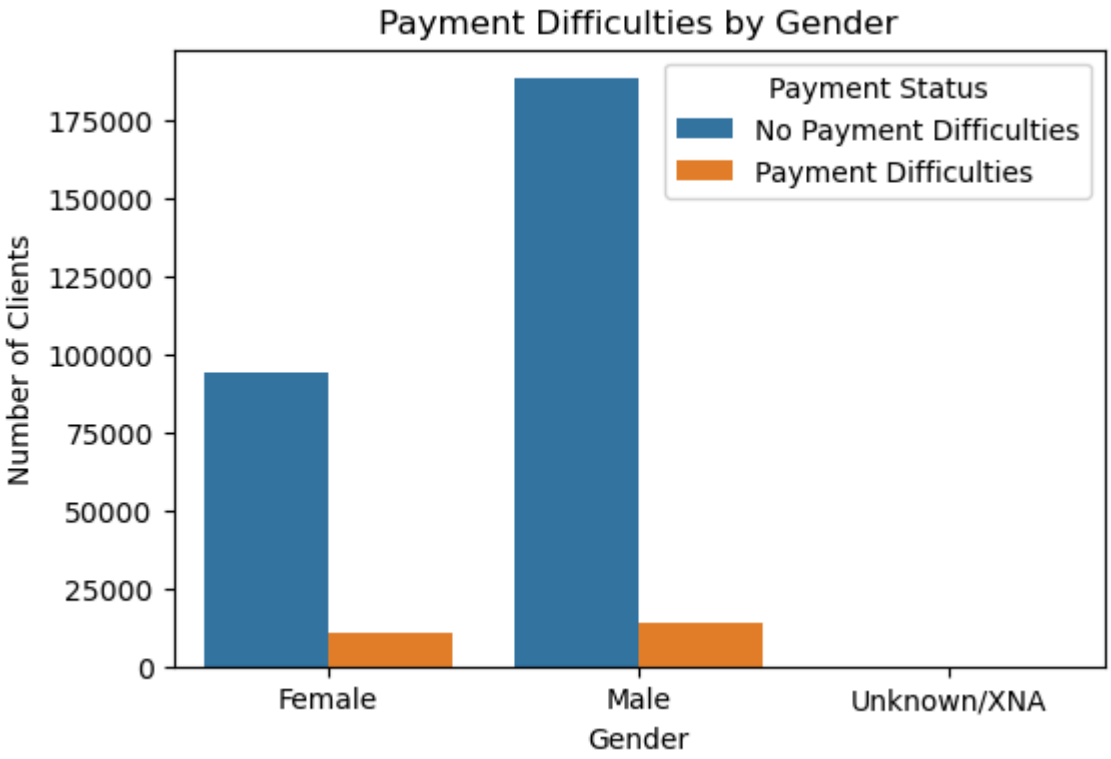
- Loan type can affect default risk.
- Revolving loans, though less common, may have slightly higher risk per client.
- This insight can help the company in risk assessment and loan approval strategies.

In [ ]:
```
# Plot Contract Type vs TARGET
plt.figure(figsize=(6,4))
sns.countplot(x='NAME_CONTRACT_TYPE', hue='TARGET', data=application_data)

# Update legend labels
plt.legend(title='Payment Status', labels=['No Payment Difficulties', 'Payment D
plt.title('Payment Difficulties by Contract Type')
plt.xlabel('Contract Type')
plt.ylabel('Number of Clients')
plt.show()
```

Bivariate Analysis: Gender vs TARGET

In [18]:
```
# Plot Gender vs TARGET
plt.figure(figsize=(6,4))
sns.countplot(x='CODE_GENDER', hue='TARGET', data=application_data)
```

```
# Update x-axis and legend labels
plt.xticks([0,1,2], ['Female', 'Male', 'Unknown/XNA'])
plt.legend(title='Payment Status', labels=['No Payment Difficulties', 'Payment D
plt.title('Payment Difficulties by Gender')
plt.xlabel('Gender')
plt.ylabel('Number of Clients')
plt.show()
```



Gender vs Payment Difficulties

- Female applicants:
  - Majority do not have payment difficulties.
- Male applicants:
  - Proportion of payment difficulties is slightly higher compared to females.
- Unknown/XNA:
  - Negligible, can be ignored for analysis.

My observation:

- Gender appears to influence payment behavior slightly.
- This insight can be used to segment clients for risk assessment or targeted monitoring.