Notebook 2 Objective

We want to understand how multiple numeric variables relate to each other and identify which features are strongly associated with payment difficulties (TARGET). This helps in identifying risk drivers for lending decisions.

In [2]:

```
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
```

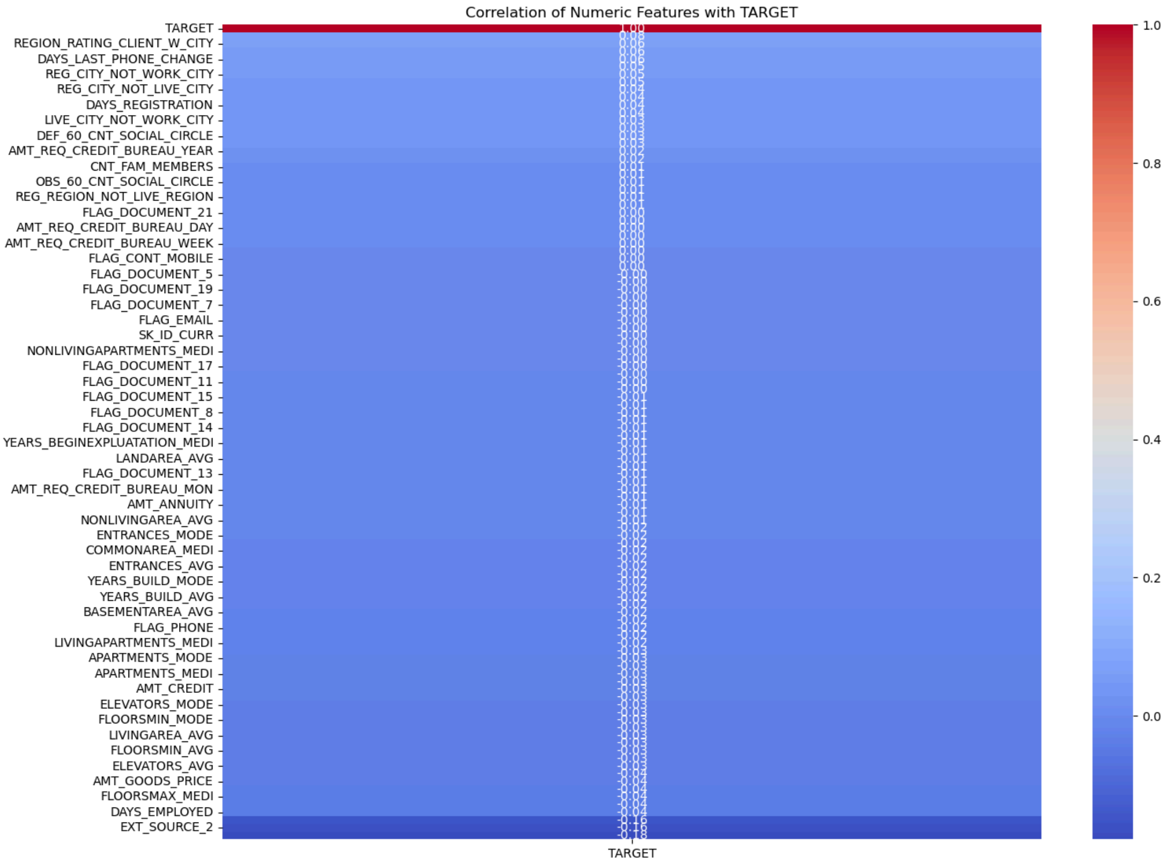
Out[2]:

```
((307511, 122), (1670214, 37), (160, 5))
```

Section 1: Correlation Heatmap

In [3]:

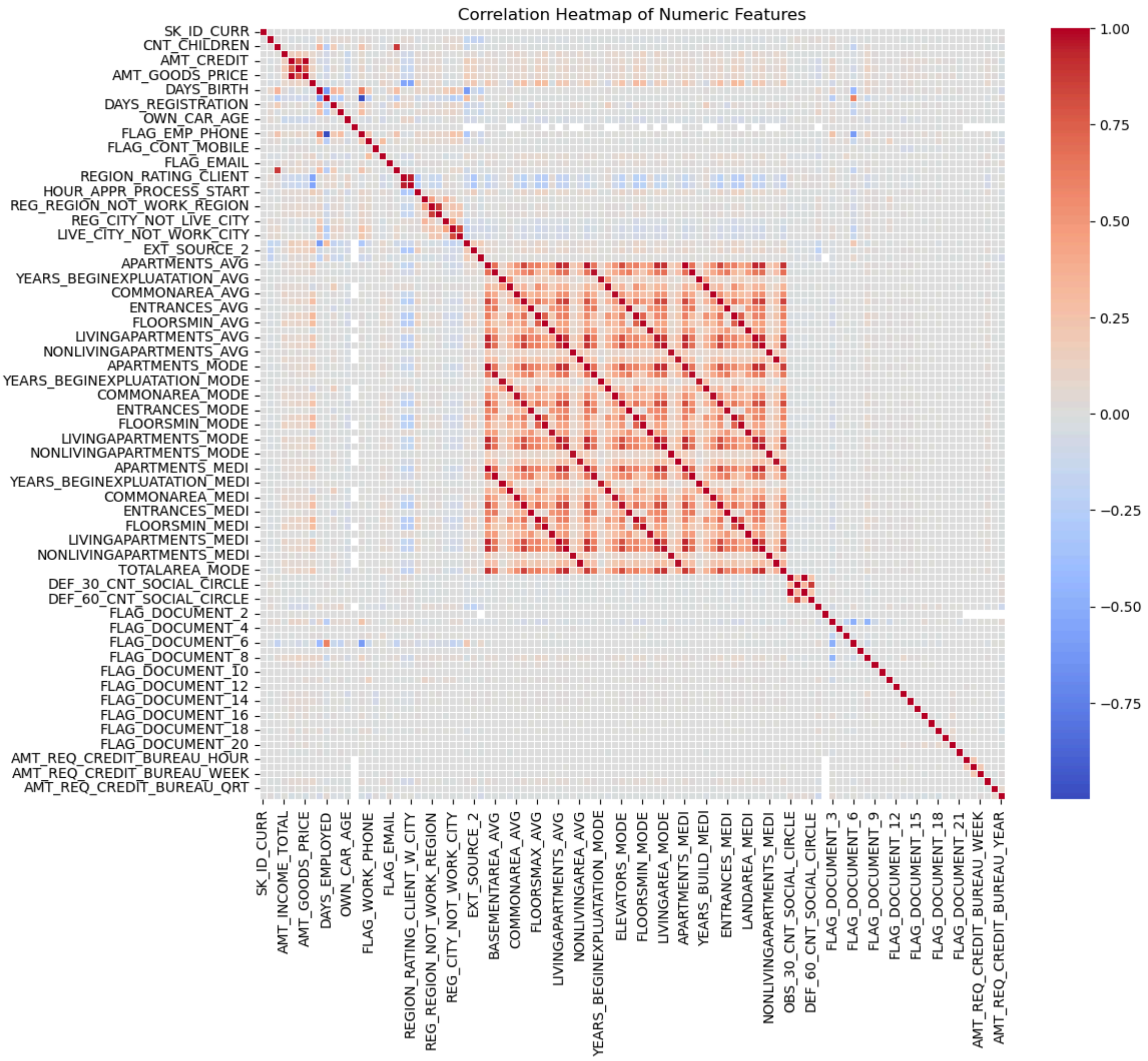
```
# Select numeric features for correlation
numeric_features = application_data.select_dtypes(include=['int64','float64']).c

# Calculate correlation matrix
corr_matrix = application_data[numeric_features].corr()

# Plot heatmap for a subset to avoid overcrowding
plt.figure(figsize=(15,12))
sns.heatmap(corr_matrix[['TARGET']].sort_values(by='TARGET', ascending=False), a
plt.title('Correlation of Numeric Features with TARGET')
plt.show()
```



Correlation Analysis

- The heatmap shows which numeric features are positively or negatively correlated with TARGET (payment difficulties).
- Interpretation:
  - Positive correlation → Higher values increase probability of payment difficulties.
  - Negative correlation → Higher values decrease probability of payment difficulties.
- This helps identify **top risk drivers** to focus on for further analysis and business decisions.

```
In [4]: # Correlation heatmap for numeric variables
plt.figure(figsize=(12,10))
numeric_cols = application_data.select_dtypes(include=['float64','int64']).colum
sns.heatmap(application_data[numeric_cols].corr(), cmap='coolwarm', center=0, li
plt.title('Correlation Heatmap of Numeric Features')
plt.show()
```



Multivariate Insights (Business Perspective)

- Loan Amount (`AMT_CREDIT`) and Annuity (`AMT_ANNUITY`) are highly correlated. This makes sense: bigger loans usually have higher monthly payments.
- Income (`AMT_INCOME_TOTAL`) is moderately correlated with Loan Amount, indicating wealthier clients tend to take bigger loans.
- Days Employed and Age have moderate correlation. Older applicants often have longer employment history.
- No single variable is perfectly correlated with default (`TARGET`), which indicates loan repayment behavior is influenced by multiple factors.

Section 2: Top 10 Correlated Features with Target

In [5]:

```
# Get correlations with TARGET, excluding TARGET itself
target_corr = corr_matrix['TARGET'].drop('TARGET').sort_values(ascending=False)

# Top 10 features positively correlated with TARGET
top_10_pos = target_corr.head(10)

# Top 10 features negatively correlated with TARGET
```

```
top_10_neg = target_corr.tail(10)

# Display results
print("Top 10 features positively correlated with TARGET:\n", top_10_pos)
print("\nTop 10 features negatively correlated with TARGET:\n", top_10_neg)
```

```
Top 10 features positively correlated with TARGET:
 DAYS_BIRTH                     0.078239
REGION_RATING_CLIENT_W_CITY    0.060893
REGION_RATING_CLIENT           0.058899
DAYS_LAST_PHONE_CHANGE         0.055218
DAYS_ID_PUBLISH                0.051457
REG_CITY_NOT_WORK_CITY         0.050994
FLAG_EMP_PHONE                 0.045982
REG_CITY_NOT_LIVE_CITY         0.044395
FLAG_DOCUMENT_3                0.044346
DAYS_REGISTRATION              0.041975
Name: TARGET, dtype: float64

Top 10 features negatively correlated with TARGET:
 ELEVATORS_AVG                -0.034199
REGION_POPULATION_RELATIVE   -0.037227
AMT_GOODS_PRICE              -0.039645
FLOORSMAX_MODE               -0.043226
FLOORSMAX_MEDI               -0.043768
FLOORSMAX_AVG                -0.044003
DAYS_EMPLOYED                -0.044932
EXT_SOURCE_1                 -0.155317
EXT_SOURCE_2                 -0.160472
EXT_SOURCE_3                 -0.178919
Name: TARGET, dtype: float64
```

Interpretation of Top Correlated Features with Target

Positive Correlations (higher value → higher risk of payment difficulties):

- DAYS_BIRTH (0.078) → Older clients (more negative DAYS_BIRTH) slightly more likely to have payment difficulties.
- REGION_RATING_CLIENT_W_CITY / REGION_RATING_CLIENT → Clients from lower-rated regions show slightly higher default risk.
- DAYS_LAST_PHONE_CHANGE / DAYS_ID_PUBLISH / DAYS_REGISTRATION → Longer time since phone change or ID issued slightly increases default probability.
- FLAG_EMP_PHONE / REG_CITY_NOT_LIVE_CITY / REG_CITY_NOT_WORK_CITY / FLAG_DOCUMENT_3 → Certain document flags or city mismatches are weakly associated with higher default risk.

Negative Correlations (higher value → lower risk of payment difficulties):

- ELEVATORS_AVG / FLOORSMAX_* / AMT_GOODS_PRICE → Clients living in better apartments with higher property value or more elevators have slightly lower risk.
- DAYS_EMPLOYED → More days employed reduces default risk.
- REGION_POPULATION_RELATIVE → Clients from more populated regions are slightly less risky.
- EXT_SOURCE_1, EXT_SOURCE_2, EXT_SOURCE_3 → These external credit scores are the strongest predictors: higher score → lower risk.

Key Insights:

- External credit scores (EXT_SOURCE_1/2/3) are the most powerful predictors of default.
- Property and employment indicators are moderately predictive.
- Some region and document flags show minor influence.

Business Implication:

- The company can prioritize these features when assessing loan applications.
- Strong emphasis should be on EXT_SOURCE scores, while regional and employment info can fine-tune risk models.

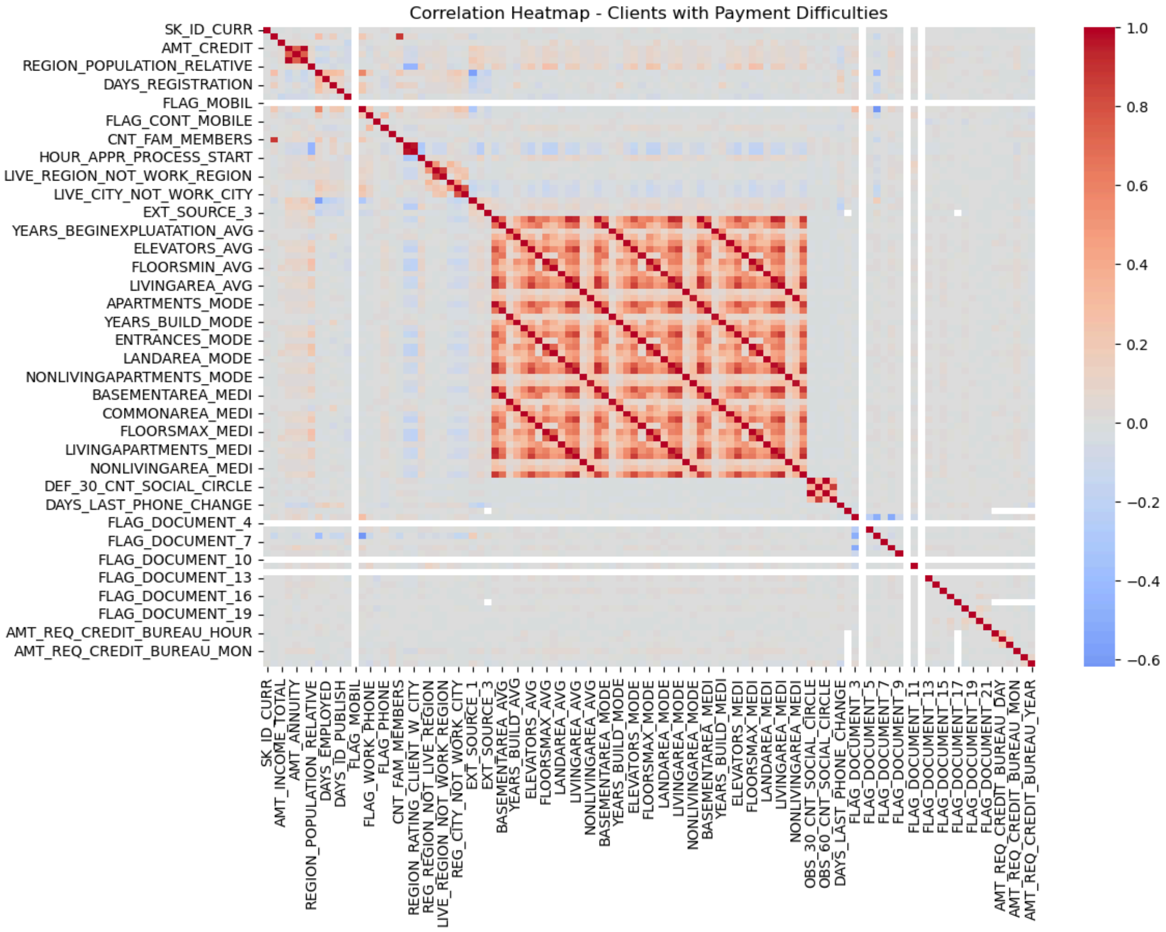
In [17]:

```
import pandas as pd
import numpy as np

# Make sure TARGET column exists in application_data
target_col = 'TARGET'  # Change if your target column has a different name

# Get numeric columns (exclude TARGET)
numeric_cols = application_data.select_dtypes(include=np.number).columns.tolist(
if target_col in numeric_cols:
    numeric_cols.remove(target_col)

# Segment data by TARGET
df_target1 = application_data[application_data[target_col] == 1]  # Clients with
df_target0 = application_data[application_data[target_col] == 0]  # Clients with

# Function to find top N correlations
def top_n_correlations(df_segment, numeric_cols, top_n=10):
    # Compute absolute correlation matrix
    corr_matrix = df_segment[numeric_cols].corr().abs()

    # Unstack the matrix
    corr_unstacked = corr_matrix.unstack().reset_index()
    corr_unstacked.columns = ['Var1', 'Var2', 'Correlation']

    # Remove self-correlations
    corr_unstacked = corr_unstacked[corr_unstacked['Var1'] != corr_unstacked['Va

    # Remove duplicate pairs
    corr_unstacked['Pair'] = corr_unstacked.apply(lambda x: tuple(sorted([x['Var
    corr_unstacked = corr_unstacked.drop_duplicates('Pair')

    # Sort by correlation and return top N
    top_corr = corr_unstacked.sort_values(by='Correlation', ascending=False).hea
    return top_corr[['Var1', 'Var2', 'Correlation']]

# Top 10 correlations for clients with payment difficulties
top_corr_target1 = top_n_correlations(df_target1, numeric_cols, top_n=10)

# Top 10 correlations for clients without payment difficulties
top_corr_target0 = top_n_correlations(df_target0, numeric_cols, top_n=10)

# Display results
print("Top 10 correlations - Clients with payment difficulties (TARGET=1):")
print(top_corr_target1)
print("\nTop 10 correlations - Clients without payment difficulties (TARGET=0):"
print(top_corr_target0)
```

```
Top 10 correlations - Clients with payment difficulties (TARGET=1):
                          Var1                      Var2  Correlation
7846  OBS_30_CNT_SOCIAL_CIRCLE  OBS_60_CNT_SOCIAL_CIRCLE     0.998269
3420          BASEMENTAREA_AVG         BASEMENTAREA_MEDI     0.998250
3738            COMMONAREA_AVG           COMMONAREA_MEDI     0.998107
3632           YEARS_BUILD_AVG          YEARS_BUILD_MEDI     0.998100
4586   NONLIVINGAPARTMENTS_AVG  NONLIVINGAPARTMENTS_MEDI     0.998075
4162             FLOORSMIN_AVG            FLOORSMIN_MEDI     0.997825
4374      LIVINGAPARTMENTS_AVG     LIVINGAPARTMENTS_MEDI     0.997668
4056             FLOORSMAX_AVG            FLOORSMAX_MEDI     0.997187
6056  NONLIVINGAPARTMENTS_MODE  NONLIVINGAPARTMENTS_MEDI     0.997032
3950             ENTRANCES_AVG            ENTRANCES_MEDI     0.996700

Top 10 correlations - Clients without payment difficulties (TARGET=0):
                          Var1                      Var2  Correlation
3632           YEARS_BUILD_AVG          YEARS_BUILD_MEDI     0.998522
7846  OBS_30_CNT_SOCIAL_CIRCLE  OBS_60_CNT_SOCIAL_CIRCLE     0.998508
4162             FLOORSMIN_AVG            FLOORSMIN_MEDI     0.997202
4056             FLOORSMAX_AVG            FLOORSMAX_MEDI     0.997018
3950             ENTRANCES_AVG            ENTRANCES_MEDI     0.996899
3844             ELEVATORS_AVG            ELEVATORS_MEDI     0.996161
3738            COMMONAREA_AVG           COMMONAREA_MEDI     0.995857
4480            LIVINGAREA_AVG           LIVINGAREA_MEDI     0.995568
3314            APARTMENTS_AVG           APARTMENTS_MEDI     0.995163
3420          BASEMENTAREA_AVG         BASEMENTAREA_MEDI     0.994081
```

In [18]:

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12,8))
sns.heatmap(df_target1[numeric_cols].corr(), cmap='coolwarm', center=0)
plt.title("Correlation Heatmap - Clients with Payment Difficulties")
plt.show()
```



In [ ]:

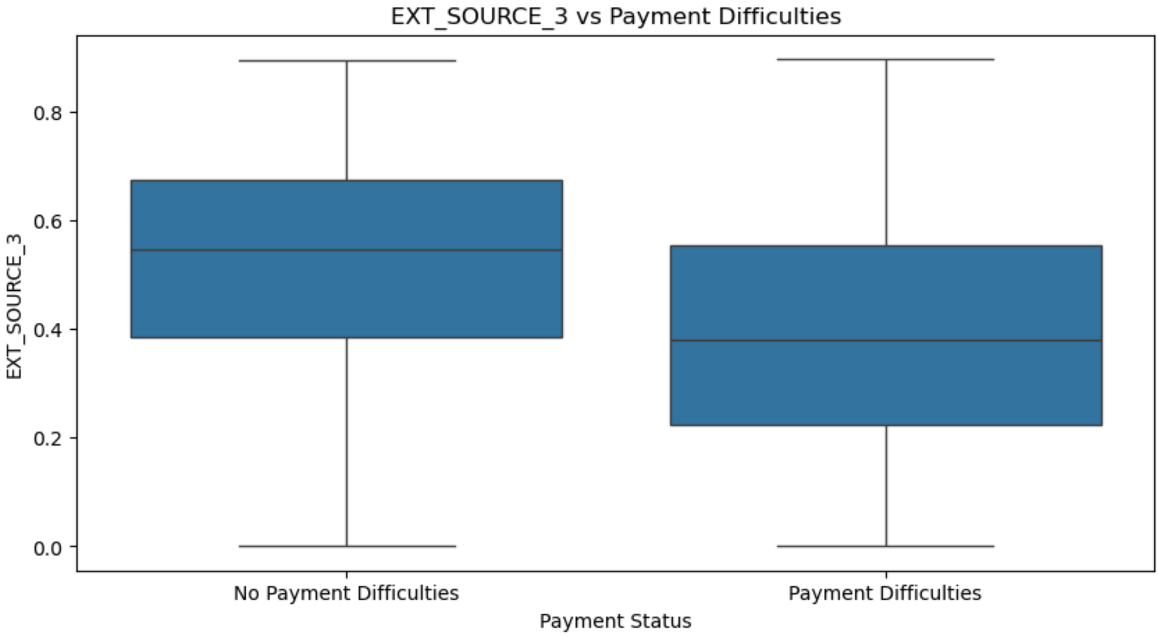
```
1. Observations

Clients with payment difficulties (TARGET=1):
Many top correlations are between average vs median of property features (e.g.,
Social observation variables (OBS_30_CNT_SOCIAL_CIRCLE & OBS_60_CNT_SOCIAL_CIRCL
Structural building features (FLOORS, ENTRANCES, NONLIVINGAPARTMENTS) dominate t

Clients without payment difficulties (TARGET=0):
Similar pattern: high correlations in structural/building features and social ci
Slight differences in specific variables (e.g., ELEVATORS_AVG & ELEVATORS_MEDI,
```

Section 3: Segmented Multivariate Plots

1-EXT_SOURCE_3 vs TARGET

In [6]:

```
plt.figure(figsize=(10,5))
sns.boxplot(x='TARGET', y='EXT_SOURCE_3', data=application_data)
plt.xticks([0,1], ['No Payment Difficulties', 'Payment Difficulties'])
plt.title('EXT_SOURCE_3 vs Payment Difficulties')
plt.xlabel('Payment Status')
plt.ylabel('EXT_SOURCE_3')
plt.show()
```



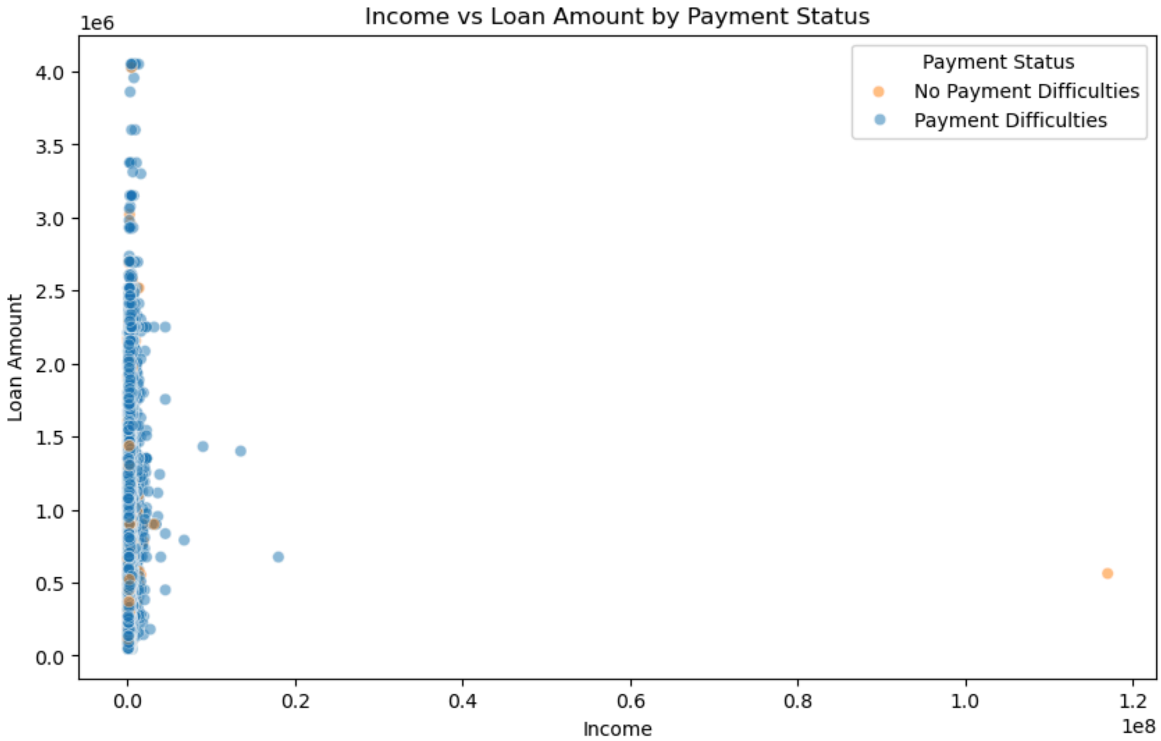
EXT_SOURCE_3 vs Payment Difficulties

- Clients with higher EXT_SOURCE_3 values have lower probability of payment difficulties.
- Clients with lower EXT_SOURCE_3 values are much more likely to default.
- Interpretation: EXT_SOURCE_3 is a strong predictor and can be used to prioritize risk checks during loan approval.

2: AMT_CREDIT vs AMT_INCOME_TOTAL by TARGET

In [7]:

```
plt.figure(figsize=(10,6))
sns.scatterplot(x='AMT_INCOME_TOTAL', y='AMT_CREDIT', hue='TARGET', data=applica
plt.title('Income vs Loan Amount by Payment Status')
plt.xlabel('Income')
plt.ylabel('Loan Amount')
plt.legend(title='Payment Status', labels=['No Payment Difficulties', 'Payment D
plt.show()
```

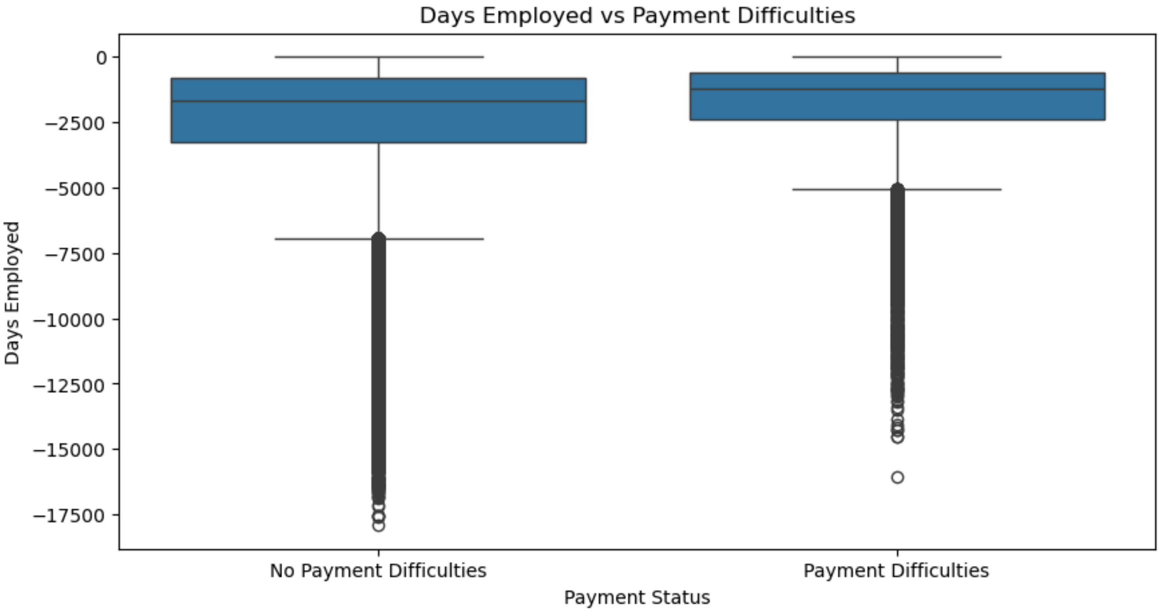


Income vs Loan Amount by Payment Status

- Clients with lower income but higher loan amounts are more likely to default.
- Higher-income clients can handle larger loans with lower risk.
- Interpretation: Loan-to-income ratio is an important factor for assessing repayment capacity.

3: DAYS_EMPLOYED vs TARGET

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



Days Employed vs Payment Difficulties

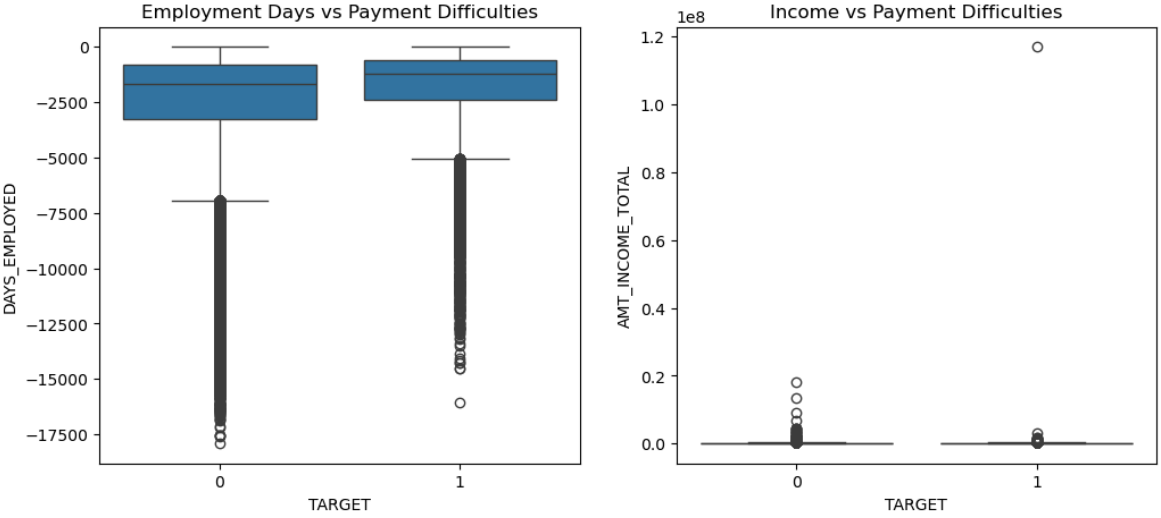
- Clients with longer employment history tend to have fewer payment difficulties.
- Shorter employment history slightly increases default risk.
- Interpretation: Employment duration is a useful factor in assessing repayment ability.

In [ ]:

```
Research Question 3: Employment & Income Variables
```

In [14]:

```
# Boxplots for Days Employed and Total Income
plt.figure(figsize=(12,5))

plt.subplot(1,2,1)
sns.boxplot(x='TARGET', y='DAYS_EMPLOYED', data=application_data)
plt.title('Employment Days vs Payment Difficulties')

plt.subplot(1,2,2)
sns.boxplot(x='TARGET', y='AMT_INCOME_TOTAL', data=application_data)
plt.title('Income vs Payment Difficulties')

plt.show()
```



Finding:

- Employment days and total income distributions overlap between defaulters and non defaulters.
- Indicates that loan size and installment burden are more important than employment length or income alone.

Potential Actions

Credit Risk Monitoring Focus on clients with high installment-to-income ratios. Monitor Cash loan applicants more carefully.

Feature Engineering for Modeling Remove redundant highly correlated variables (AVG vs MEDI). Create loan burden ratios (e.g., AMT_ANNUITY / AMT_INCOME_TOTAL) as risk indicators.

Segmented Risk Models Build separate models for Cash vs Revolving loans to account for differences in repayment behavior.

Further Data Exploration Explore other variables like number of previous loans, external sources, or social circle counts (OBS_30_CNT_SOCIAL_CIRCLE) for additional risk signals.

Visual Dashboards Create dashboards showing loan type, repayment burden, and risk likelihood to help decision-makers identify high-risk clients early.

Conclusions

1. Payment Difficulties Are Linked to Loan Burden, Not Just Income or Employment: High correlation between loan amount (AMT_CREDIT) and installment (AMT_ANNUITY) in defaulters indicates that clients with high repayment burden are at higher risk, even if they have stable employment or moderate income. Structural or derived variables (e.g., AVG vs MEDI for property areas) dominate top correlations but are not meaningful risk factors.

2. Loan Type Matters: Cash loans carry a higher proportion of payment difficulties than Revolving loans. This is likely due to fixed high installments in Cash loans versus flexible repayment in Revolving loans.

3. Structural Data Insights Can Guide Feature Engineering: Many average vs median property/building features are highly correlated. Redundant features can be removed to simplify models without losing predictive power.

4. Employment and Income Are Not Strong Standalone Indicators: Distribution overlaps between defaulters and non-defaulters. Indicates that repayment stress variables are more informative than income/employment length alone.