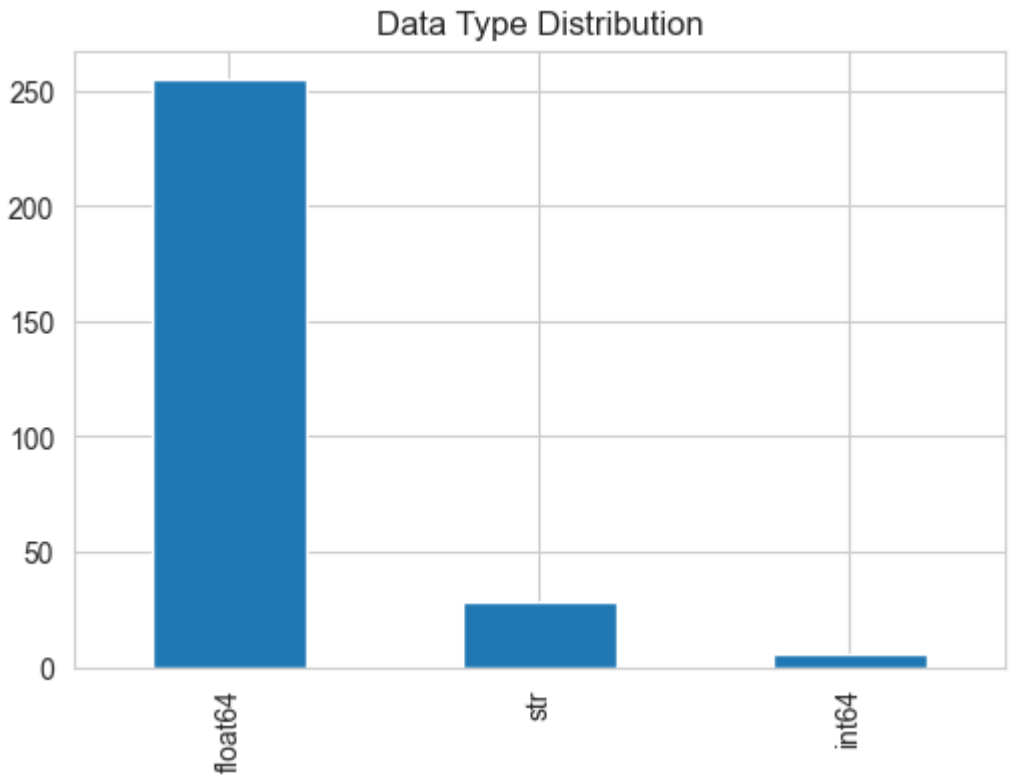
In [1]:

```
# ==============================
# DATA DESCRIPTIVE FULL REPORT
# ==============================

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

sns.set_style("whitegrid")

# ==============================
# Load Dataset
# ==============================

df = pd.read_csv("PS_2026.02.07_05.49.09.csv", comment="#", low_memory=False)

print("Dataset Shape:", df.shape)
print("\n")

# ==============================
# Basic Info
# ==============================

print("Dataset Info:\n")
df.info()

print("\nStatistical Summary:\n")
print(df.describe().T)

# ==============================
# Data Type Distribution
# ==============================

print("\nData Type Distribution:\n")
print(df.dtypes.value_counts())

plt.figure(figsize=(6,4))
df.dtypes.value_counts().plot(kind='bar')
plt.title("Data Type Distribution")
plt.show()

# ==============================
# Missing Value Analysis
# ==============================

missing = df.isnull().sum()
missing = missing[missing > 0].sort_values(ascending=False)

print("\nTop Missing Columns:\n")
print(missing.head(20))

plt.figure(figsize=(12,6))
missing.head(20).plot(kind='bar')
plt.title("Top 20 Columns with Missing Values")
plt.show()
```

```python
missing_percent = (df.isnull().sum() / len(df)) * 100
missing_percent = missing_percent[missing_percent > 0].sort_values(ascending=Fal

print("\nMissing Percentage:\n")
print(missing_percent.head(20))

plt.figure(figsize=(12,6))
sns.heatmap(df.isnull(), cbar=False)
plt.title("Missing Value Heatmap")
plt.show()

# ==============================
# Numerical Analysis
# ==============================

num_cols = df.select_dtypes(include=np.number).columns

print("\nSkewness & Kurtosis:\n")
skew_kurt = pd.DataFrame({
    "Skewness": df[num_cols].skew(),
    "Kurtosis": df[num_cols].kurt()
}).sort_values(by="Skewness", ascending=False)

print(skew_kurt.head(15))

# ==============================
# Distribution Plots
# ==============================

for col in num_cols[:3]:
    plt.figure(figsize=(6,4))
    sns.histplot(df[col], kde=True)
    plt.title(f"Distribution of {col}")
    plt.show()

# ==============================
# Correlation Heatmap
# ==============================

plt.figure(figsize=(12,8))
sns.heatmap(df[num_cols[:20]].corr(), cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()

# ==============================
# Strong Correlation Pairs
# ==============================

corr_matrix = df[num_cols].corr().abs()
upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool))
strong_pairs = upper.stack().sort_values(ascending=False)

print("\nTop Strong Correlation Pairs:\n")
print(strong_pairs.head(15))

# ==============================
# Outlier Detection
# ==============================
```

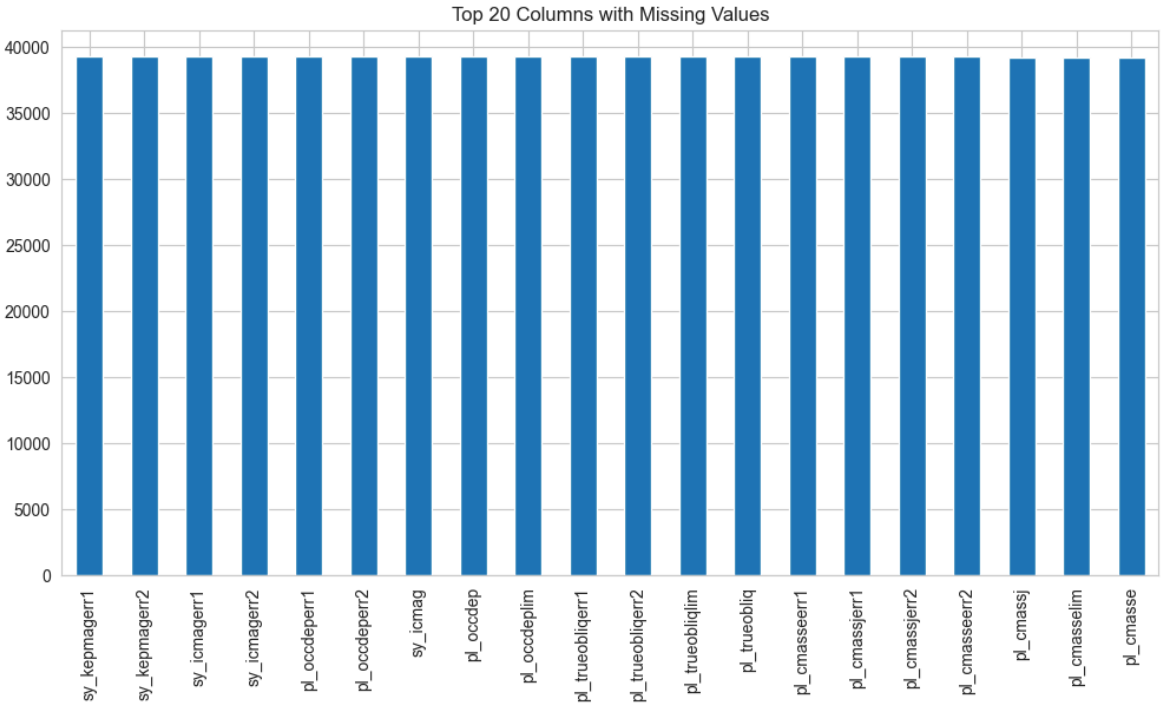
```python
plt.figure(figsize=(12,6))
sns.boxplot(data=df[num_cols[:5]])
plt.xticks(rotation=45)
plt.title("Boxplot for Outlier Detection")
plt.show()

# ==============================
# Variance Analysis
# ==============================

variance = df[num_cols].var().sort_values(ascending=False)

print("\nTop Variance Features:\n")
print(variance.head(15))

# ==============================
# Data Quality Score
# ==============================

total_cells = df.shape[0] * df.shape[1]
missing_cells = df.isnull().sum().sum()

data_quality = 100 - ((missing_cells / total_cells) * 100)

print(f"\nOverall Data Quality Score: {data_quality:.2f}%")

# ==============================
# PCA Visualization
# ==============================

num_data = df[num_cols].fillna(0)

scaler = StandardScaler()
scaled_data = scaler.fit_transform(num_data)

pca = PCA(n_components=2)
pca_result = pca.fit_transform(scaled_data)

plt.figure(figsize=(8,6))
plt.scatter(pca_result[:,0], pca_result[:,1], alpha=0.5)
plt.title("PCA Projection (2D)")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.show()

print("\n===== DATA DESCRIPTIVE REPORT COMPLETED =====")
```

```
Dataset Shape: (39315, 289)


Dataset Info:

<class 'pandas.DataFrame'>
RangeIndex: 39315 entries, 0 to 39314
Columns: 289 entries, rowid to pl_ndispec
dtypes: float64(255), int64(6), str(28)
memory usage: 86.7 MB

Statistical Summary:

                count          mean           std  min     25%      50%  \
rowid         39315.0  19658.000000  11349.407253  1.0  9829.5  19658.0
default_flag  39315.0      0.155157      0.362059  0.0     0.0      0.0
sy_snum       39315.0      1.085133      0.308189  1.0     1.0      1.0
sy_pnum       39315.0      1.918199      1.238170  1.0     1.0      1.0
sy_mnum       39315.0      0.000000      0.000000  0.0     0.0      0.0
...               ...           ...           ...  ...     ...      ...
st_nrvc       39265.0      0.196027      0.887007  0.0     0.0      0.0
st_nspec      39265.0      0.110480      0.778309  0.0     0.0      0.0
pl_nespec     39265.0      0.258780      1.919063  0.0     0.0      0.0
pl_ntranspec  39265.0      0.218643      1.619760  0.0     0.0      0.0
pl_ndispec    39265.0      0.005577      0.155038  0.0     0.0      0.0

                  75%      max
rowid         29486.5  39315.0
default_flag      0.0      1.0
sy_snum           1.0      4.0
sy_pnum           2.0      8.0
sy_mnum           0.0      0.0
...               ...      ...
st_nrvc           0.0     12.0
st_nspec          0.0     13.0
pl_nespec         0.0     35.0
pl_ntranspec      0.0     32.0
pl_ndispec        0.0      6.0

[261 rows x 8 columns]

Data Type Distribution:

float64    255
str         28
int64        6
Name: count, dtype: int64
```



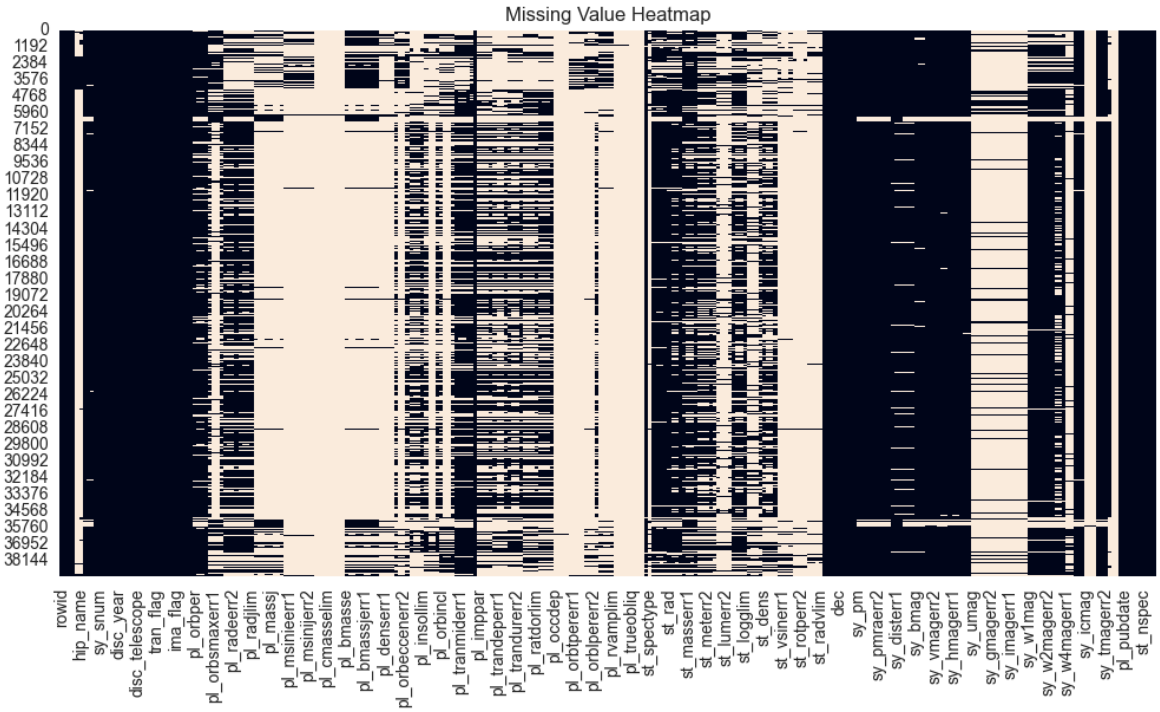
Top Missing Columns:

```
sy_kepmagerr1       39315
sy_kepmagerr2       39315
sy_icmagerr1        39275
sy_icmagerr2        39275
pl_occdeperr1       39270
pl_occdeperr2       39270
sy_icmag            39270
pl_occdep           39267
pl_occdeplim        39267
pl_trueobliqerr1    39252
pl_trueobliqerr2    39252
pl_trueobliqlim     39247
pl_trueobliq        39247
pl_cmasseerr1       39231
pl_cmassjerr1       39231
pl_cmassjerr2       39231
pl_cmasseerr2       39231
pl_cmassj           39227
pl_cmasselim        39227
pl_cmasse           39227
dtype: int64
```



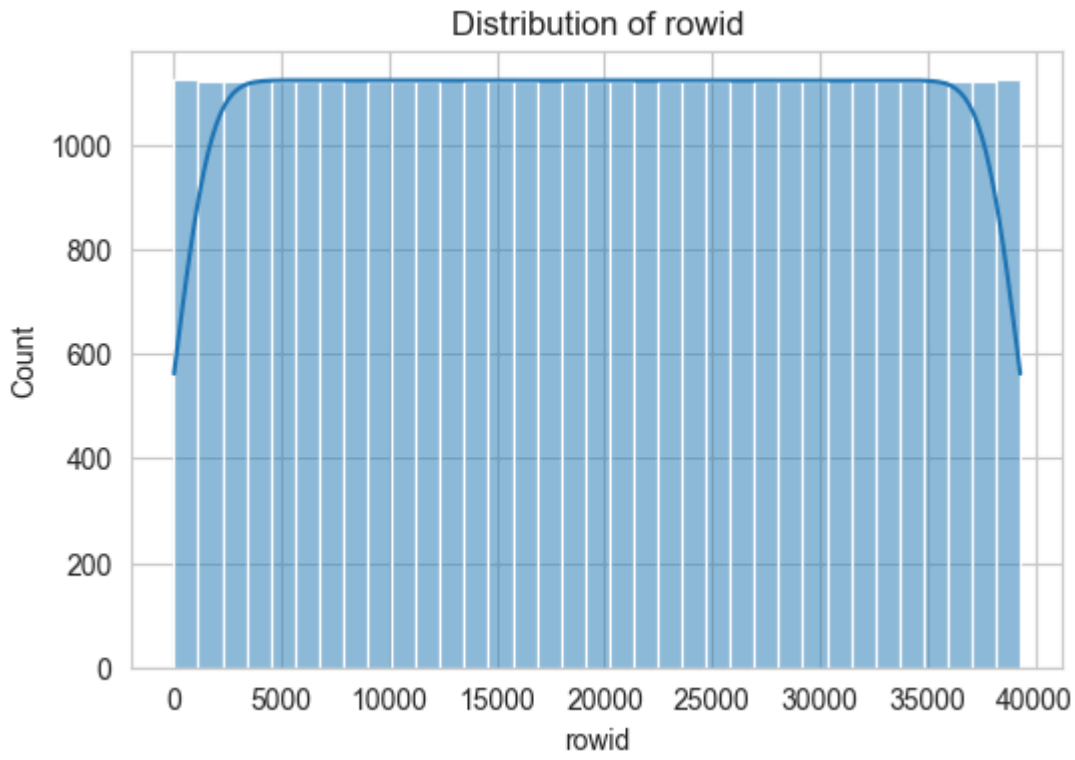
Missing Percentage:

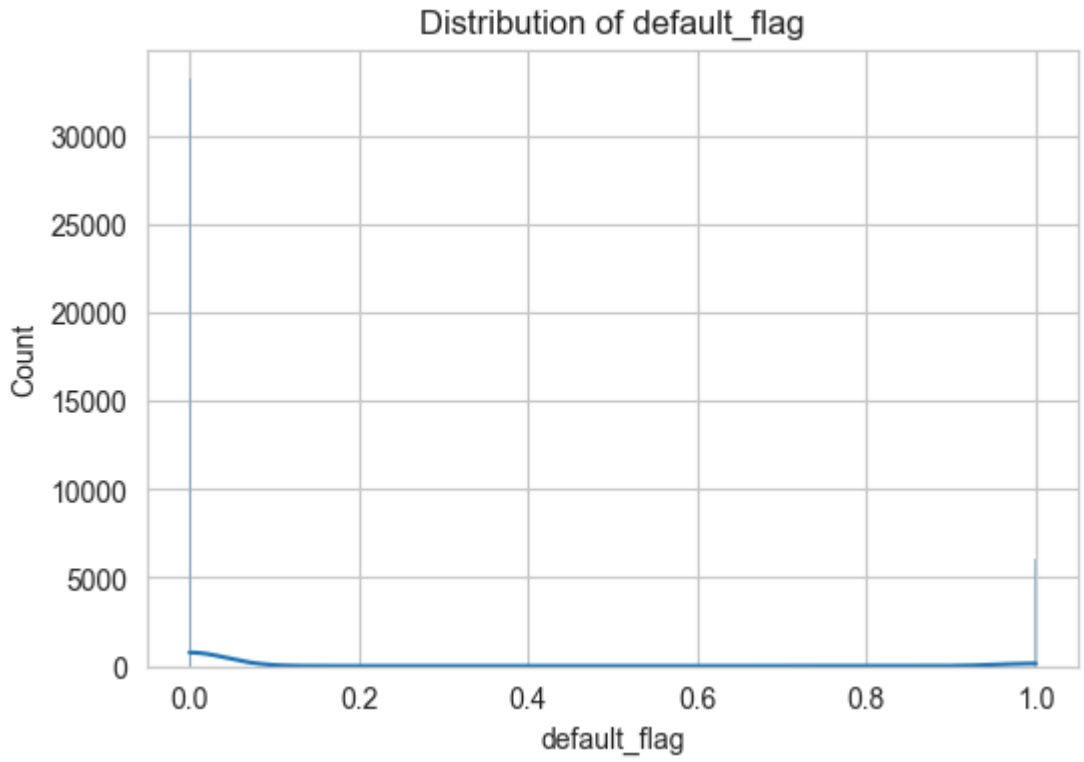
```
sy_kepmagerr1       100.000000
sy_kepmagerr2       100.000000
sy_icmagerr1         99.898258
sy_icmagerr2         99.898258
pl_occdeperr1        99.885540
pl_occdeperr2        99.885540
sy_icmag             99.885540
pl_occdep            99.877909
pl_occdeplim         99.877909
pl_trueobliqerr1     99.839756
pl_trueobliqerr2     99.839756
pl_trueobliqlim      99.827038
pl_trueobliq         99.827038
pl_cmasseerr1        99.786341
pl_cmassjerr1        99.786341
pl_cmassjerr2        99.786341
pl_cmasseerr2        99.786341
pl_cmassj            99.776167
pl_cmasselim         99.776167
pl_cmasse            99.776167
dtype: float64
```

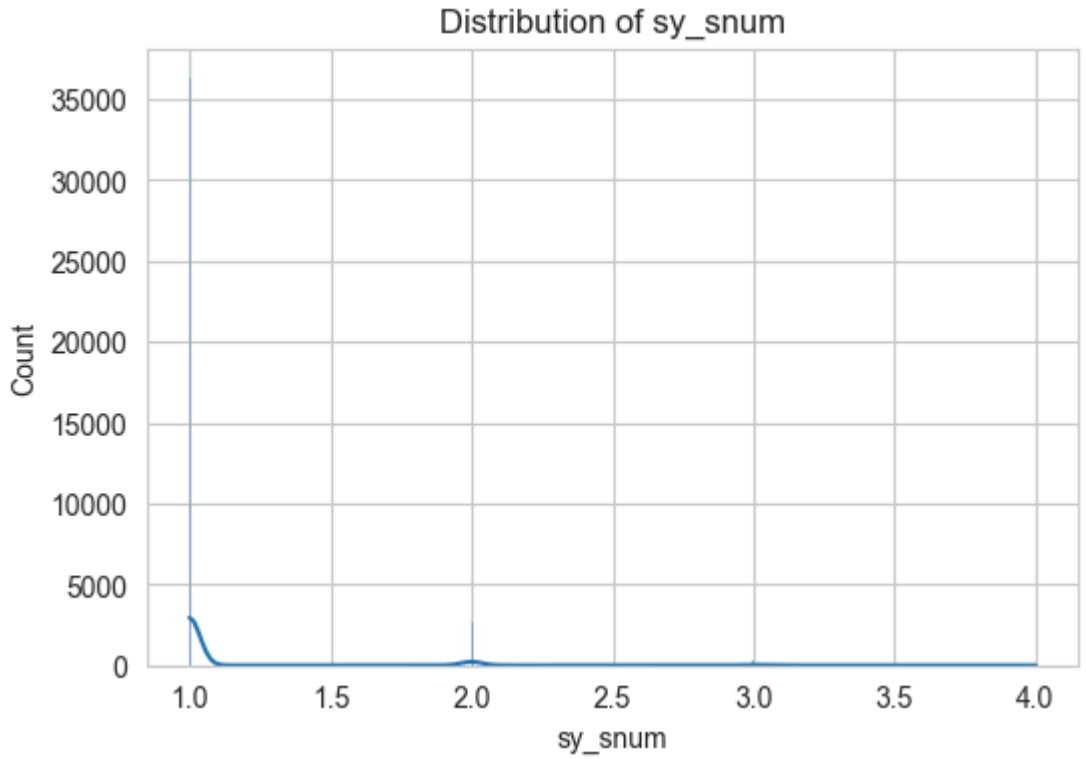


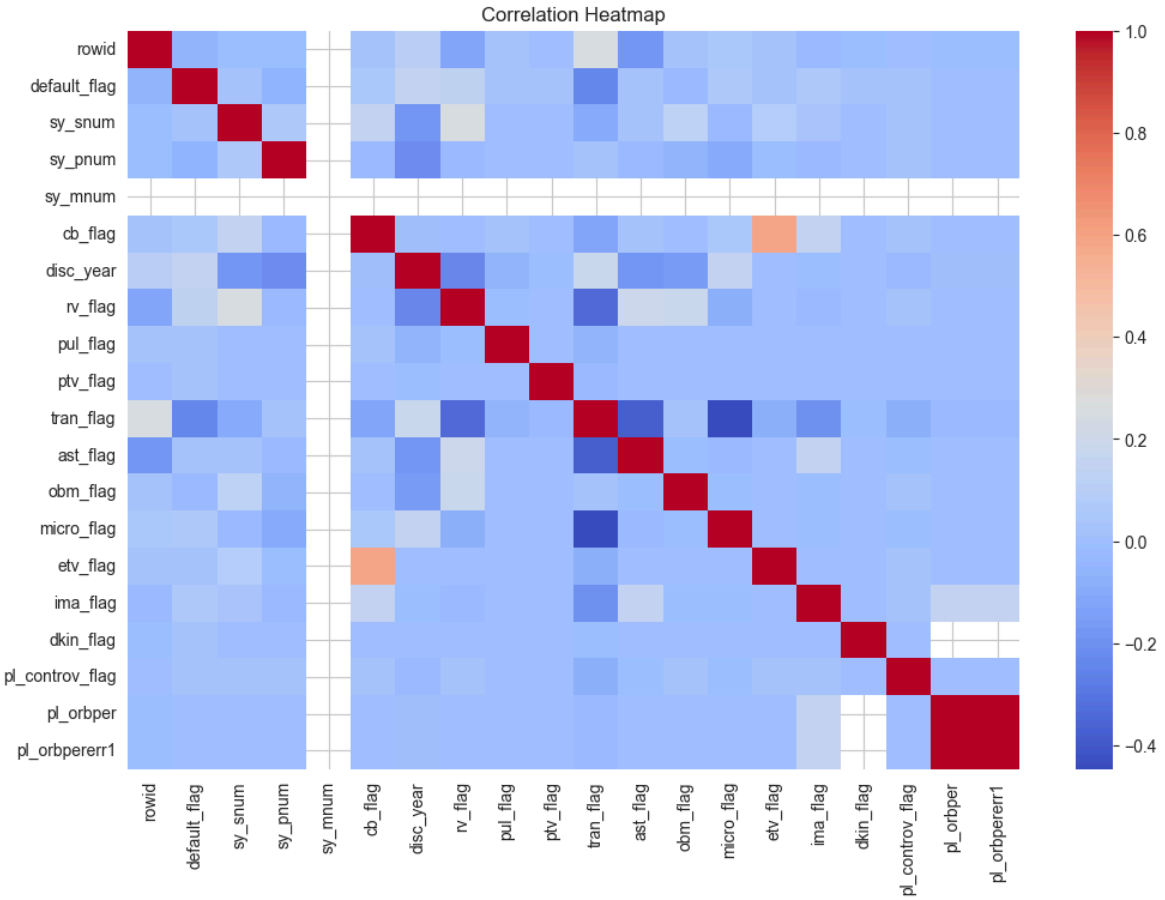
Skewness & Kurtosis:

	Skewness	Kurtosis
dkin_flag	198.229665	39295.000000
pl_orbper	189.306157	35877.216306
st_tefflim	189.148090	35777.000000
pl_orbpererr1	185.062291	34252.988129
pl_trandurlim	158.003165	24965.000000
st_masserr1	146.877592	23086.656200
ptv_flag	140.164189	19644.999796
pl_tranmiderr1	135.827121	19637.819835
st_radlim	134.195752	18007.499778
pl_imppar	103.380772	12175.453380
pl_tranmid	100.297199	10091.763091
pl_radeerr1	94.762842	10103.481691
pl_radjerr1	94.762767	10103.469277
pl_orbsmaxerr1	85.944655	7428.151885
st_raderr1	83.233480	8886.585795


Distribution of rowid
Count
1000
800
600
400
200
0
0
5000
10000
15000
20000
25000
30000
35000
40000
rowid

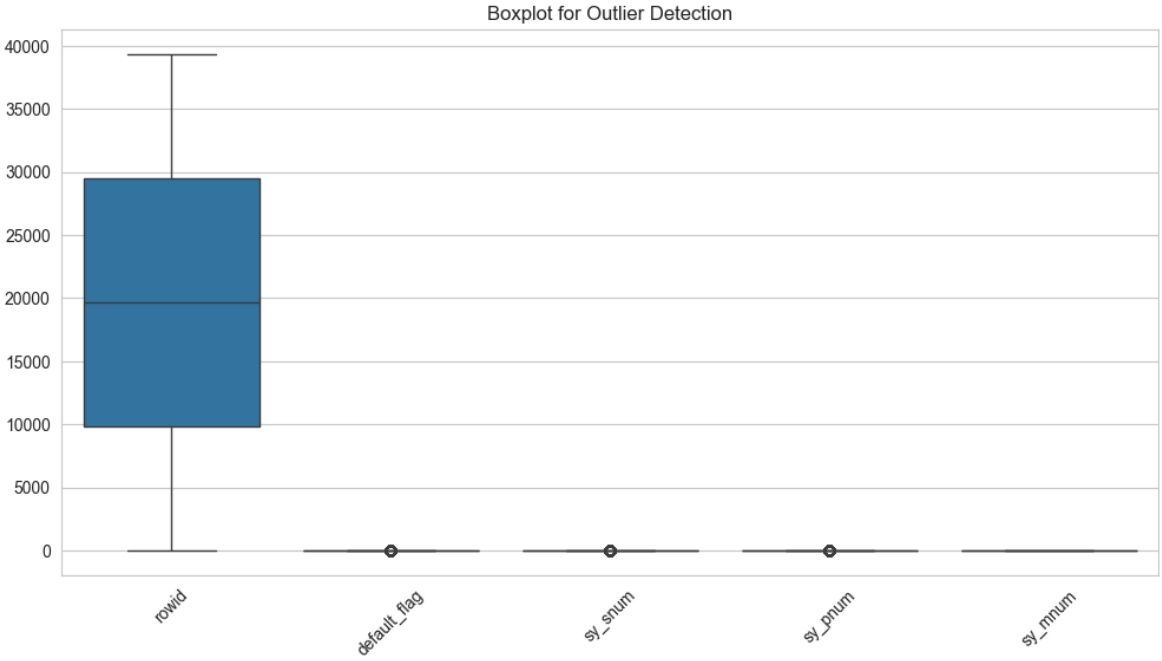

Distribution of default_flag
Count
30000
25000
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
default_flag


Distribution of sy_snum
Count
35000
30000
25000
20000
15000
10000
5000
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
sy_snum


Correlation Heatmap
rowid
default_flag
sy_snum
sy_pnum
sy_mnum
cb_flag
disc_year
rv_flag
pul_flag
ptv_flag
tran_flag
ast_flag
obm_flag
micro_flag
etv_flag
ima_flag
dkin_flag
pl_controv_flag
pl_orbper
pl_orbpererr1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

Top Strong Correlation Pairs:

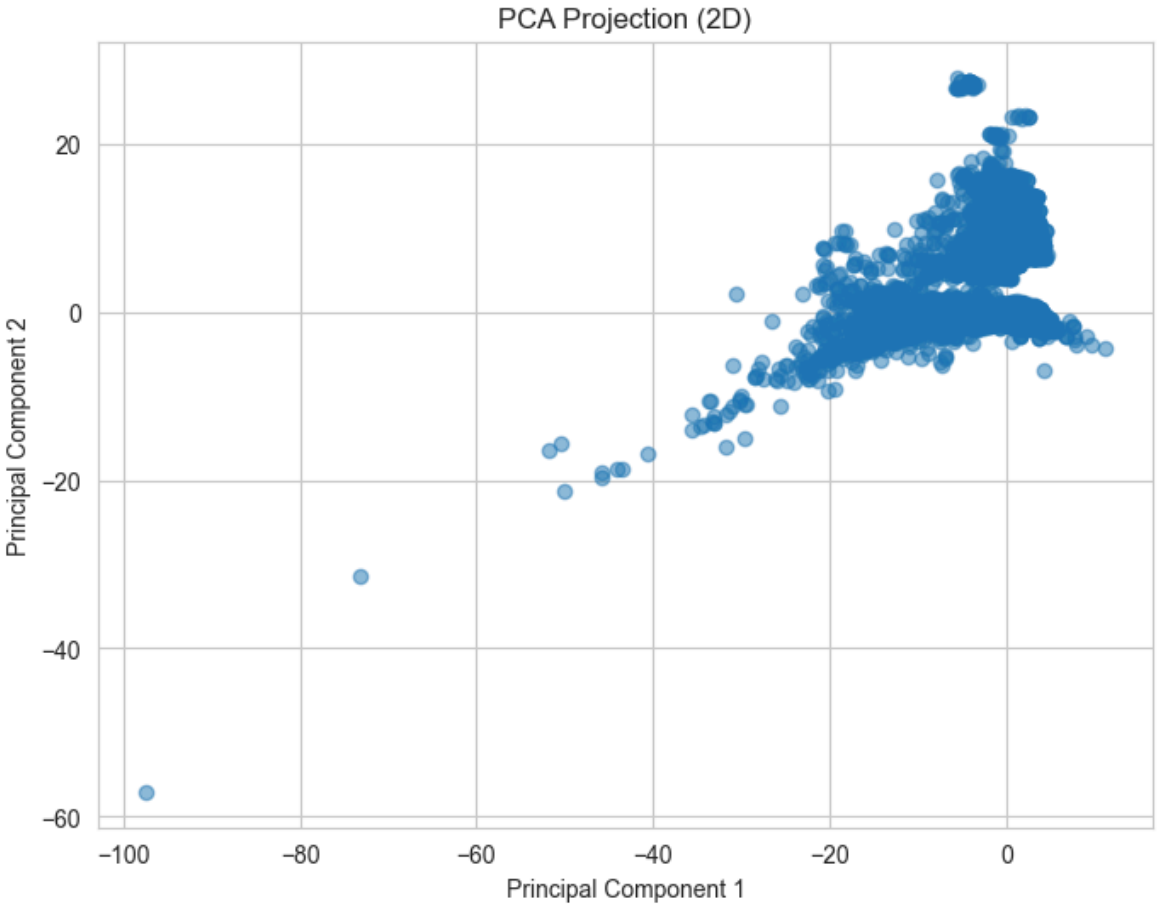
```
pl_cmasseerr1  pl_cmasseerr2       1.0
pl_masseerr1   pl_bmasseerr1       1.0
sy_zmagerr1    sy_zmagerr2         1.0
pl_insolerr1   pl_occdeperr1       1.0
               pl_occdeperr2       1.0
sy_rmagerr1    sy_rmagerr2         1.0
sy_plxerr1     sy_plxerr2          1.0
pul_flag       st_tefflim          1.0
pl_massj       pl_bmassj           1.0
pl_msinieerr2  pl_trueobliqerr1    1.0
pl_massjlim    pl_bmasselim        1.0
               pl_bmassjlim        1.0
pl_insolerr2   pl_occdeperr1       1.0
               pl_occdeperr2       1.0
sy_gmagerr1    sy_gmagerr2         1.0
dtype: float64
```



Top Variance Features:

```
pl_orbpererr1     6.448120e+12
pl_orbper         4.499091e+12
pl_orbpererr2     2.932157e+11
pl_tranmid        4.824202e+10
rowid             1.288090e+08
pl_orbtper        1.164760e+08
pl_orbtpererr2    7.724957e+06
pl_orbtpererr1    3.513886e+06
pl_masse          2.926226e+06
pl_bmasse         2.439002e+06
pl_insol          1.908406e+06
pl_msinie         1.503975e+06
pl_insolerr1      1.498624e+06
st_teff           9.894498e+05
pl_denserr1       9.702835e+05
dtype: float64
```

Overall Data Quality Score: 53.73%




```
===== DATA DESCRIPTIVE REPORT COMPLETED =====
```