

Step 1: Setup & File Upload

```
# Install dependencies
!pip install pandas matplotlib seaborn openpyxl

Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
Requirement already satisfied: seaborn in /usr/local/lib/python3.12/dist-packages (0.13.2)
Requirement already satisfied: openpyxl in /usr/local/lib/python3.12/dist-packages (3.1.5)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cyclers>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.59.2)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (25.0)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.2.3)
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.12/dist-packages (from openpyxl) (2.0.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)

# Import libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
import warnings

# Upload Excel files
from google.colab import files
uploaded = files.upload() # Select CampaignData.xlsx, OutreachData.xlsx, ApplicantData.xlsx

Choose Files 3 files
• CampaignData.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 6032 bytes, last modified: 9/4/2025 - 100% done
• OutreachData.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 1342427 bytes, last modified: 9/4/2025 - 100% done
• ApplicantData.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 1005133 bytes, last modified: 9/4/2025 - 100% done
Saving CampaignData.xlsx to CampaignData (3).xlsx
Saving OutreachData.xlsx to OutreachData (3).xlsx
Saving ApplicantData.xlsx to ApplicantData (3).xlsx
```

Step 2: Load the Datasets

```
# Load datasets into DataFrames
campaign_df = pd.read_excel("CampaignData.xlsx")
outreach_df = pd.read_excel("OutreachData.xlsx")
applicant_df = pd.read_excel("ApplicantData.xlsx")

# Quick look at the data
print("Campaign Data:")
display(campaign_df.head())

Campaign Data:
      ID      Name      Category      Intake      University      Status      Start_Date
0  AANF23  GR GS FA24 Campaign- Admit, No Deposit  Post Admission  AY2024  Illinois Institute of Technology  Completed      3/20/2024 0:00
1   AND23  GR GS FA24 Campaign- Deposit No Action  Post Admission  AY2024  Illinois Institute of Technology  Completed  2024-11-09 00:00:00
2  BPNANF23    GR GS FA24 Campaign- Deposit, No I-20  Post Admission  AY2024  Illinois Institute of Technology  Completed  2024-11-07 00:00:00
3   BPNND23      GR GS FA24 Campaign- In Progress  Pre Admission  AY2024  Illinois Institute of Technology  Completed  2024-06-03 00:00:00
4  CTKANF23  GR GS FA24 Campaign- Submit, Incomplete  Pre Admission  AY2024  Illinois Institute of Technology  Completed  2024-08-03 00:00:00

# Quick look at the data
print("Outreach Data:")
display(outreach_df.head())

Outreach Data:
      Reference_ID      Recieved_At      University      Caller_Name      Outcome_1      Remark      Campaign_ID      Escalation_Required
0      12345      4/28/2023 12:15  Illinois Institute of Technology      Shailja      Connected      NaN      IANF23      No
1      12345      4/28/2023 13:04  Illinois Institute of Technology      Shailja      Reschedule      NaN      IANF23      No
2      12345      2023-01-05 11:14:00  Illinois Institute of Technology      Shailja      Connected      NaN      IANF23      No
3      347397      2023-01-05 11:16:00  Illinois Institute of Technology      Isha      Not connected      NaN      IANF23      No
4      347397      2023-01-05 11:18:00  Illinois Institute of Technology      Isha      Connected      NaN      IANF23      No

# Quick look at the data
print("Applicant Data:")
display(applicant_df.head())

Applicant Data:
      App_ID      Country      University      Phone_Number
0      12345      India  Illinois Institute of Technology      9823241234
1      12345      India  Illinois Institute of Technology      8805617501
2      12345      India  Illinois Institute of Technology      18019011222
3      347397      Nigeria  Illinois Institute of Technology      7738599513
4      347397      Nigeria  Illinois Institute of Technology      919182706838
```

Step 3: Check Schema & Data Types

```
# Schema (columns + data types)
print("\nCampaign Data Info")
campaign_df.info()

print("\nOutreach Data Info")
outreach_df.info()

print("\nApplicant Data Info")
applicant_df.info()

Campaign Data Info
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23 entries, 0 to 22
Data columns (total 7 columns):
```

```
#      Column      Non-Null Count  Dtype
---  -
0      ID           23 non-null    object
1      Name          23 non-null    object
2      Category       23 non-null    object
3      Intake          23 non-null    object
4      University      23 non-null    object
5      Status          23 non-null    object
6      Start_Date      23 non-null    object
dtypes: object(7)
memory usage: 1.4+ KB

Outreach Data Info
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37881 entries, 0 to 37880
Data columns (total 8 columns):
#      Column      Non-Null Count  Dtype
---  -
0      Reference_ID    37881 non-null  object
1      Recieved_At     37881 non-null  object
2      University      37881 non-null  object
3      Caller_Name     37881 non-null  object
4      Outcome_1       37881 non-null  object
5      Remark          4077 non-null   object
6      Campaign_ID     37881 non-null  object
7      Escalation_Required 37881 non-null  object
dtypes: object(8)
memory usage: 2.3+ MB

Applicant Data Info
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37882 entries, 0 to 37881
Data columns (total 4 columns):
#      Column      Non-Null Count  Dtype
---  -
0      App_ID        37881 non-null  object
1      Country       37882 non-null  object
2      University    37882 non-null  object
3      Phone_Number  37882 non-null  object
dtypes: object(4)
memory usage: 1.2+ MB
```

Step 4: Data Cleaning & Preprocessing

Steps are Included:

- Remove duplicates
- Handle missing values
- Standardize categorical fields (fix spelling/case issues)
- Convert date columns to datetime

```
# Handling Missing Values
print("Missing Values - Campaign:\n", campaign_df.isnull().sum())
print("Missing Values - Outreach:\n", outreach_df.isnull().sum())
print("Missing Values - Applicant:\n", applicant_df.isnull().sum())

for df in [campaign_df, outreach_df, applicant_df]:
    for col in df.columns:
        if df[col].dtype == "object": # categorical
            df[col] = df[col].fillna("Unknown")
        elif pd.api.types.is_numeric_dtype(df[col]): # numeric
            df[col] = df[col].fillna(df[col].median())
        elif pd.api.types.is_datetime64_any_dtype(df[col]): # datetime
            df[col] = df[col].fillna(pd.Timestamp("1900-01-01"))
```

Missing Values - Campaign:

ID	0
Name	0
Category	0
Intake	0
University	0
Status	0
Start_Date	0

dtype: int64

Missing Values - Outreach:

Reference_ID	0
Recieved_At	0
University	0
Caller_Name	0
Outcome_1	0
Remark	33804
Campaign_ID	0
Escalation_Required	0

dtype: int64

Missing Values - Applicant:

App_ID	1
Country	0
University	0
Phone_Number	0

dtype: int64

```
# Removing Duplicates
print("Duplicates - Campaign:", campaign_df.duplicated().sum())
print("Duplicates - Outreach:", outreach_df.duplicated().sum())
print("Duplicates - Applicant:", applicant_df.duplicated().sum())
```

```
campaign_df = campaign_df.drop_duplicates()
outreach_df = outreach_df.drop_duplicates()
applicant_df = applicant_df.drop_duplicates()
```

```
if "CampaignID" in campaign_df.columns:
    campaign_df = campaign_df.drop_duplicates(subset=["CampaignID"])
if "ApplicantID" in applicant_df.columns:
    applicant_df = applicant_df.drop_duplicates(subset=["ApplicantID"])
```

Duplicates - Campaign: 0

Duplicates - Outreach: 446

Duplicates - Applicant: 16489

```
# Correcting Inaccuracies
for df in [campaign_df, outreach_df, applicant_df]:
    for col in df.select_dtypes(include="object").columns:
        df[col] = df[col].str.strip().str.title()
```

```
if {"CampaignStartDate", "CampaignEndDate"}.issubset(campaign_df.columns):
    invalid = campaign_df[campaign_df["CampaignEndDate"] < campaign_df["CampaignStartDate"]]
    print("Invalid Campaign Dates:", invalid.shape[0])
    campaign_df.loc[campaign_df["CampaignEndDate"] < campaign_df["CampaignStartDate"], "CampaignEndDate"] = pd.NaT
```

```
# Standardizing Data
for df in [campaign_df, outreach_df, applicant_df]:
    for col in df.columns:
```

```
        if "date" in col.lower():
            df[col] = pd.to_datetime(df[col], errors="coerce")

for df in [campaign_df, outreach_df, applicant_df]:
    for col in df.select_dtypes(include="float"):
        df[col] = df[col].round(2)

# Validation
print("\nAfter Cleaning Validation:")
for name, df in [("Campaign", campaign_df), ("Outreach", outreach_df), ("Applicant", applicant_df)]:
    print(f"\n{name} Data → Shape: {df.shape}")
    print("Missing Values:\n", df.isnull().sum())
    print("Duplicates:", df.duplicated().sum())
```



After Cleaning Validation:

```
Campaign Data → Shape: (23, 7)
Missing Values:
ID          0
Name        0
Category    0
Intake       0
University   0
Status       0
Start_Date  13
dtype: int64
Duplicates: 0

Outreach Data → Shape: (37435, 8)
Missing Values:
Reference_ID      33219
Recieved_At      15771
University        0
Caller_Name       0
Outcome_1         0
Remark            1
Campaign_ID       0
Escalation_Required  0
dtype: int64
Duplicates: 14802



Applicant Data → Shape: (21393, 4)
Missing Values:
App_ID      17273
Country     0
University   0
Phone_Number 21327
dtype: int64
Duplicates: 19752
```

▼ Step 5: Exploratory Data Analysis (EDA)


```
# Data Overview
for name, df in [("Campaign", campaign_df), ("Outreach", outreach_df), ("Applicant", applicant_df)]:
    print(f"\n{name} Data → Shape: {df.shape}")
    print(df.info())
    display(df.head())
```



```
Campaign Data → Shape: (23, 7)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23 entries, 0 to 22
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0    ID          23 non-null    object
1    Name        23 non-null    object
2    Category    23 non-null    object
3    Intake       23 non-null    object
4    University  23 non-null    object
5    Status      23 non-null    object
6    Start_Date  10 non-null    datetime64[ns]
dtypes: datetime64[ns](1), object(6)
memory usage: 1.4+ KB
None
```

	ID	Name	Category	Intake	University	Status	Start_Date	
0	Aanf23	Gr Gs Fa24 Campaign- Admit, No Deposit	Post Admission	Ay2024	Illinois Institute Of Technology	Completed	2024-03-20	
1	And23	Gr Gs Fa24 Campaign- Deposit No Action	Post Admission	Ay2024	Illinois Institute Of Technology	Completed	NaT	
2	Bpnanf23	Gr Gs Fa24 Campaign- Deposit, No I-20	Post Admission	Ay2024	Illinois Institute Of Technology	Completed	NaT	
3	Bpnnd23	Gr Gs Fa24 Campaign- In Progress	Pre Admission	Ay2024	Illinois Institute Of Technology	Completed	NaT	
4	Ctkanf23	Gr Gs Fa24 Campaign- Submit, Incomplete	Pre Admission	Ay2024	Illinois Institute Of Technology	Completed	NaT	

```
Outreach Data → Shape: (37435, 8)
<class 'pandas.core.frame.DataFrame'>
Index: 37435 entries, 0 to 37880
Data columns (total 8 columns):
#   Column      Non-Null Count  Dtype
---  -
0    Reference_ID 4216 non-null  object
1    Recieved_At  21664 non-null object
2    University   37435 non-null object
3    Caller_Name  37435 non-null object
4    Outcome_1    37435 non-null object
5    Remark       37434 non-null object
6    Campaign_ID  37435 non-null object
7    Escalation_Required 37435 non-null object
dtypes: object(8)
memory usage: 2.6+ MB
None
```

	Reference_ID	Recieved_At	University	Caller_Name	Outcome_1	Remark	Campaign_ID	Escalation_Required	
0	NaN	4/28/2023 12:15	Illinois Institute Of Technology	Shailja	Connected	Unknown	Ianf23	No	
1	NaN	4/28/2023 13:04	Illinois Institute Of Technology	Shailja	Reschedule	Unknown	Ianf23	No	
2	NaN	NaN	Illinois Institute Of Technology	Shailja	Connected	Unknown	Ianf23	No	
3	NaN	NaN	Illinois Institute Of Technology	Isha	Not Connected	Unknown	Ianf23	No	
4	NaN	NaN	Illinois Institute Of Technology	Isha	Connected	Unknown	Ianf23	No	

```
Applicant Data → Shape: (21393, 4)
<class 'pandas.core.frame.DataFrame'>
Index: 21393 entries, 0 to 37881
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0    App_ID       4120 non-null  object
1    Country      21393 non-null object
2    University   21393 non-null object
3    Phone_Number 66 non-null    object
dtypes: object(4)
memory usage: 835.7+ KB
None
```

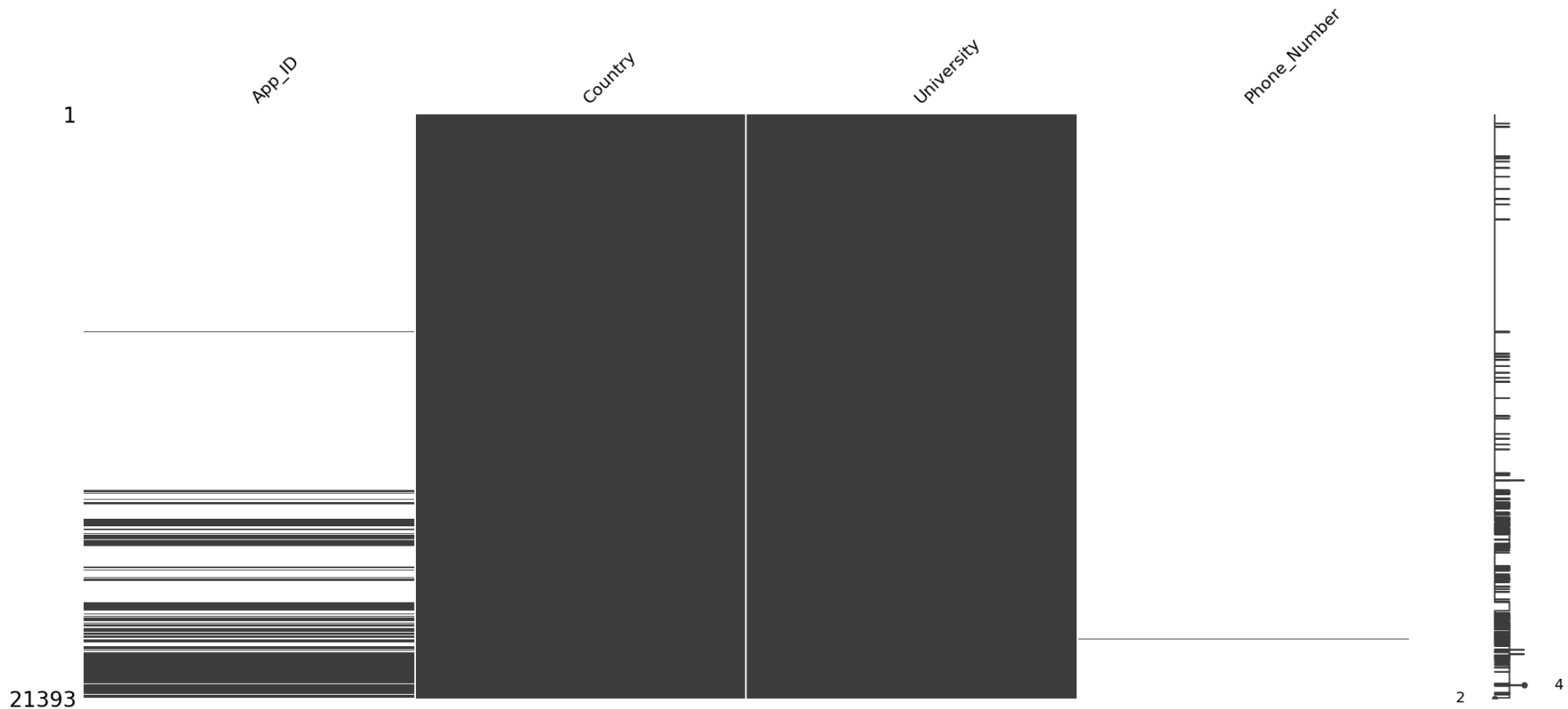
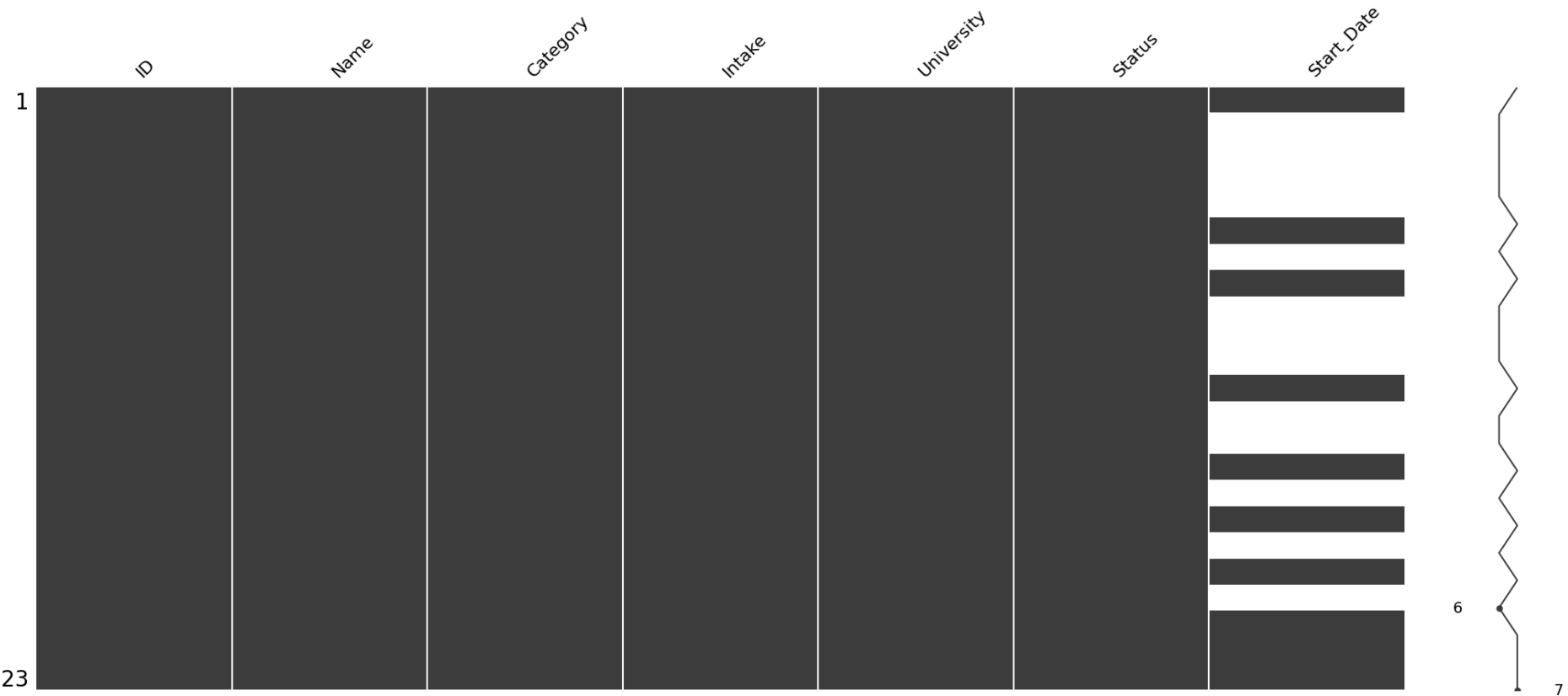
	App_ID	Country	University	Phone_Number	
0	NaN	India	Illinois Institute Of Technology	NaN	
1	NaN	India	Illinois Institute Of Technology	NaN	
2	NaN	India	Illinois Institute Of Technology	NaN	
3	NaN	Nigeria	Illinois Institute Of Technology	NaN	
4	NaN	Nigeria	Illinois Institute Of Technology	NaN	

```
# Missing Values Analysis
print("\nMissing Values")
print("Campaign:\n", campaign_df.isnull().sum())
print("Outreach:\n", outreach_df.isnull().sum())
print("Applicant:\n", applicant_df.isnull().sum())
```



```
Missing Values
Campaign:
ID          0
Name        0
Category    0
Intake       0
University  0
Status      0
Start_Date  13
dtype: int64
Outreach:
Reference_ID      33219
Recieved_At      15771
University        0
Caller_Name       0
Outcome_1         0
Remark           1
Campaign_ID       0
Escalation_Required 0
dtype: int64
Applicant:
App_ID          17273
Country         0
University      0
Phone_Number    21327
dtype: int64

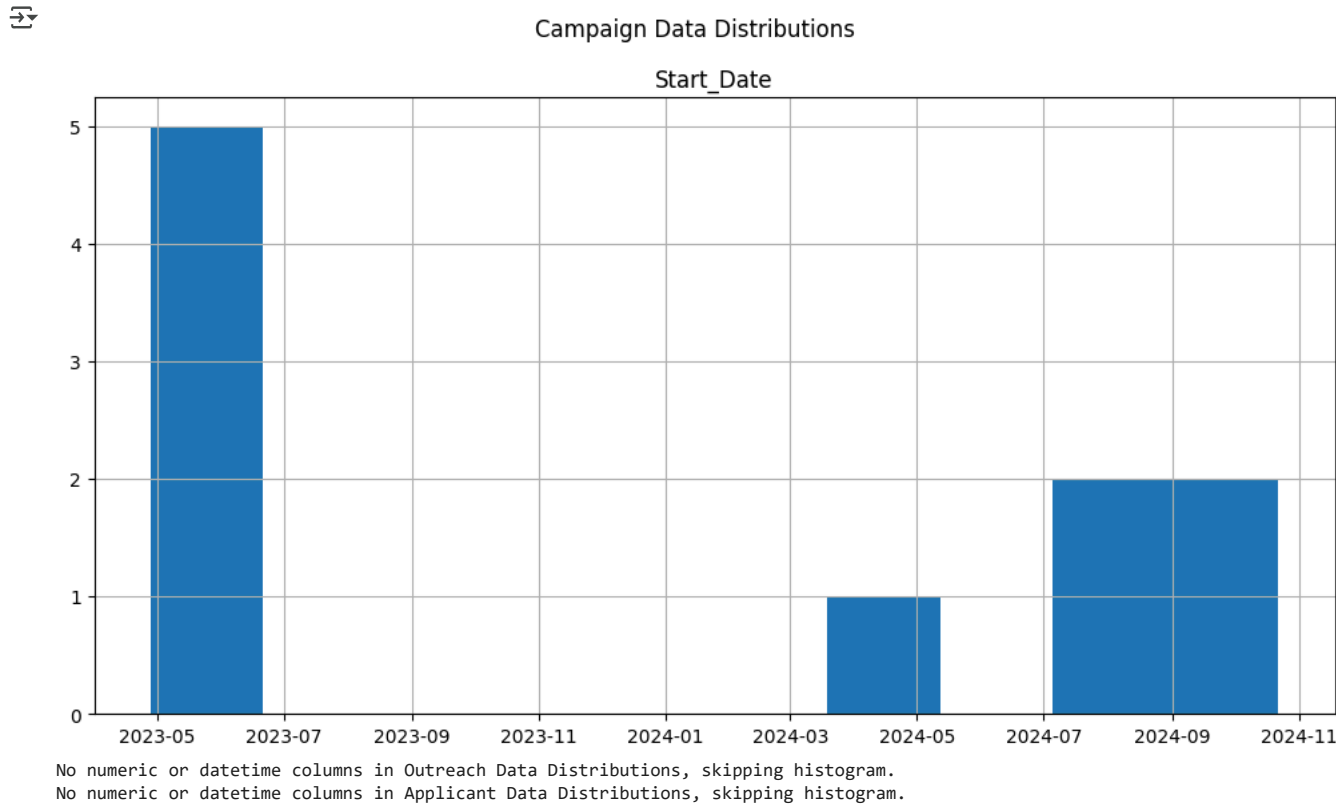
# Visualize missing values
msno.matrix(campaign_df); plt.show()
msno.matrix(outreach_df); plt.show()
msno.matrix(applicant_df); plt.show()
```



Duplicates after cleaning:
Campaign: 0
Outreach: 14802
Applicant: 19752

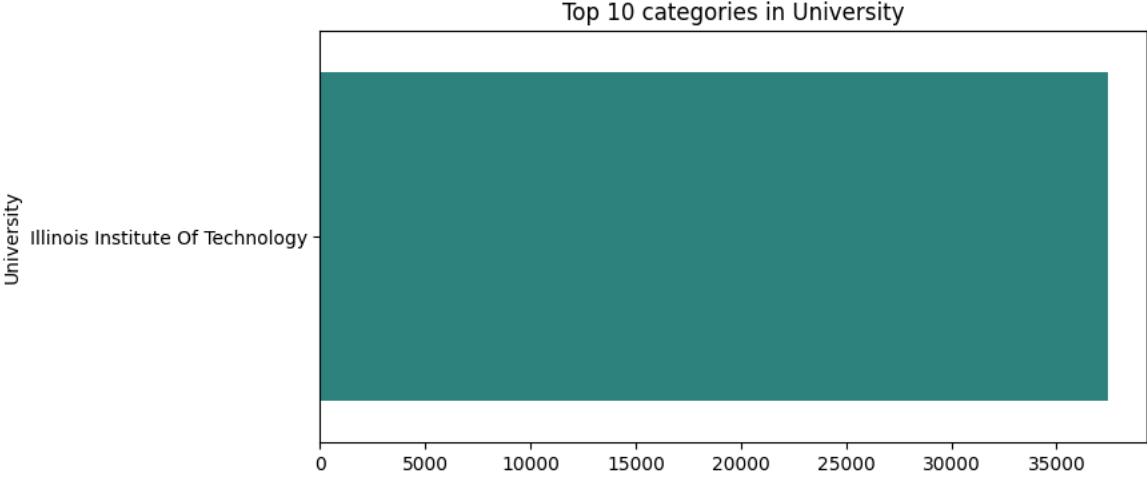
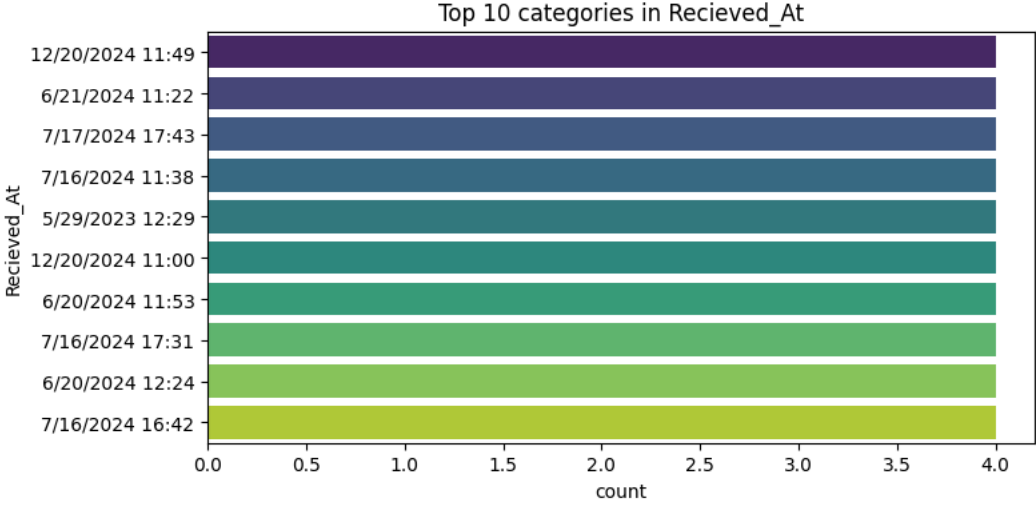
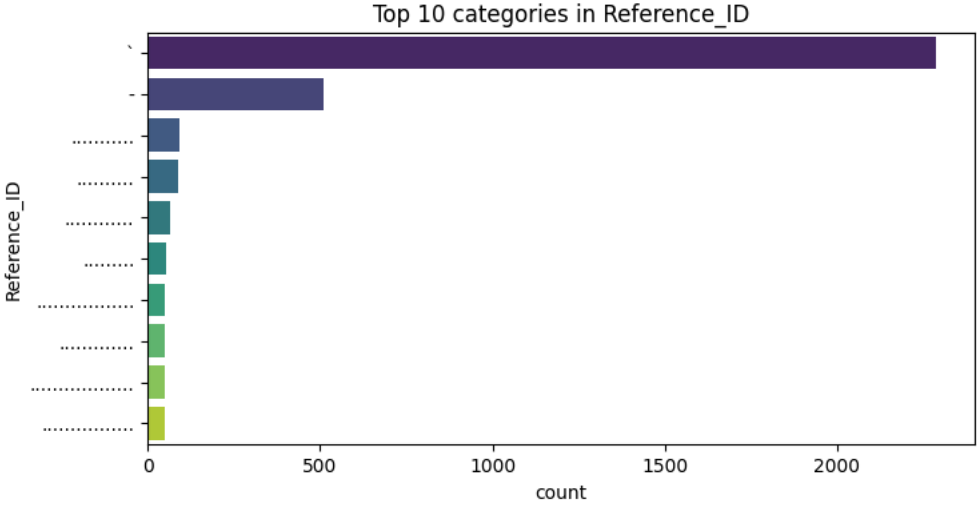
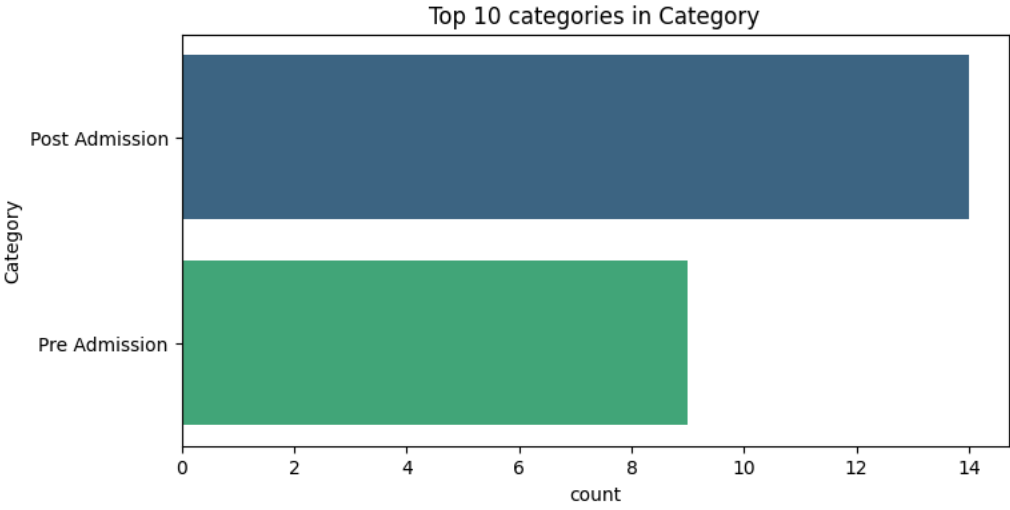
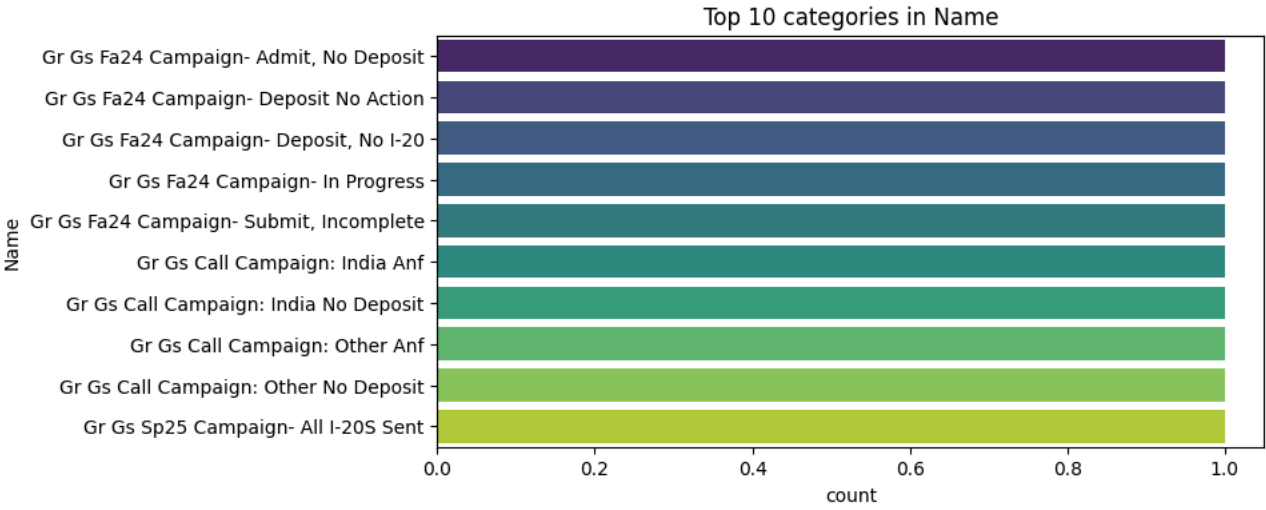
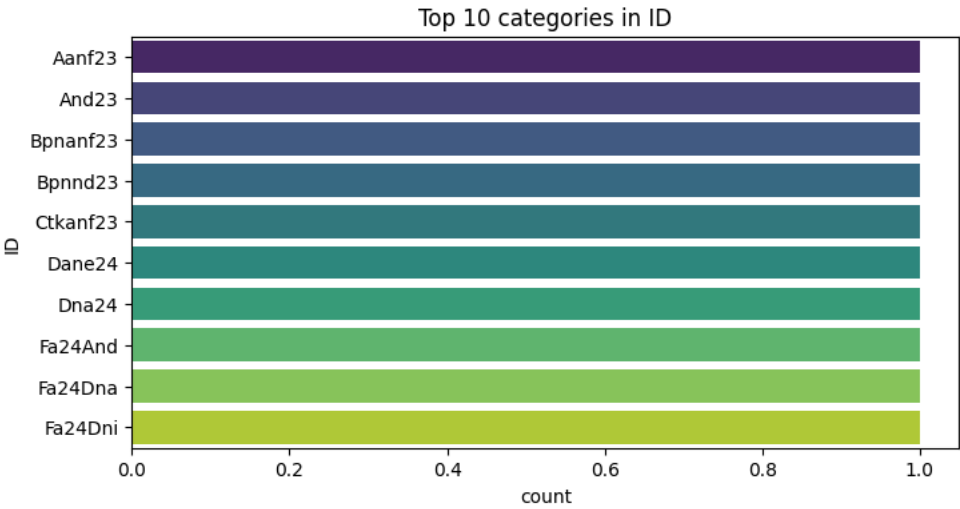
```
# Univariate Analysis
def safe_hist(df, title="Data Distributions"):
    numeric_cols = df.select_dtypes(include=["number", "datetime64"]).columns
    if len(numeric_cols) == 0:
        print(f"No numeric or datetime columns in {title}, skipping histogram.")
        return
    df[numeric_cols].hist(figsize=(12, 6))
    plt.suptitle(title)
    plt.show()
```

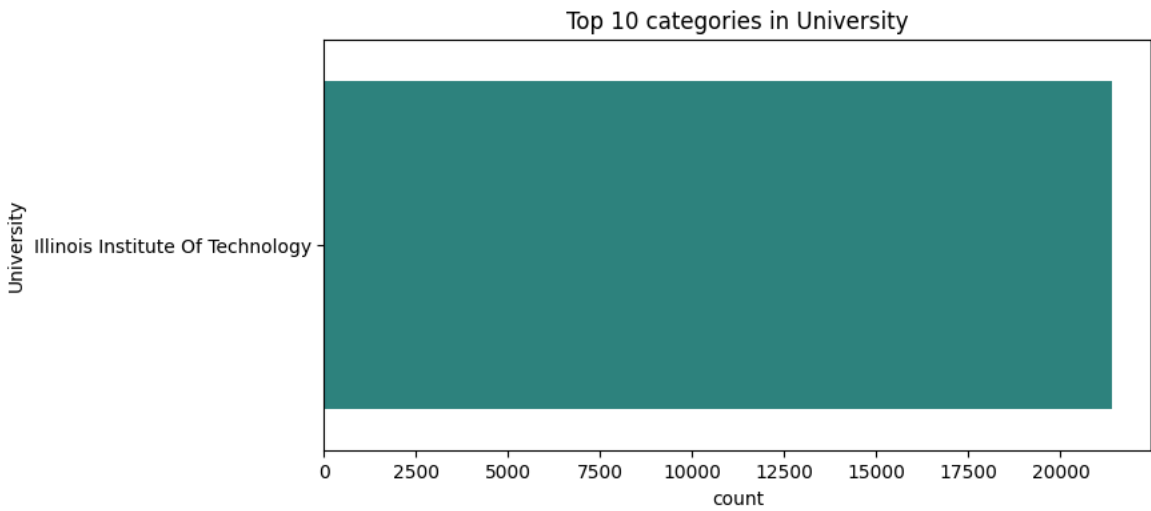
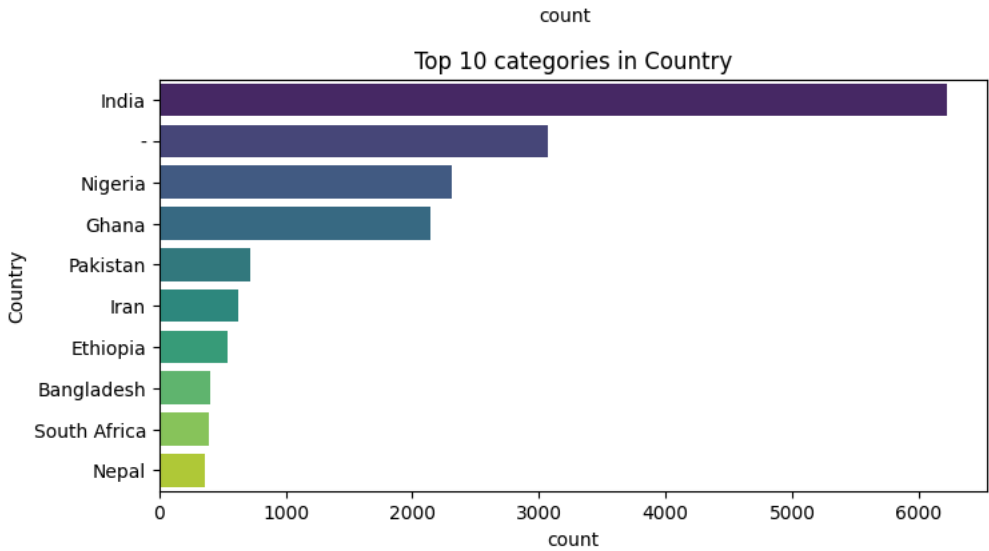
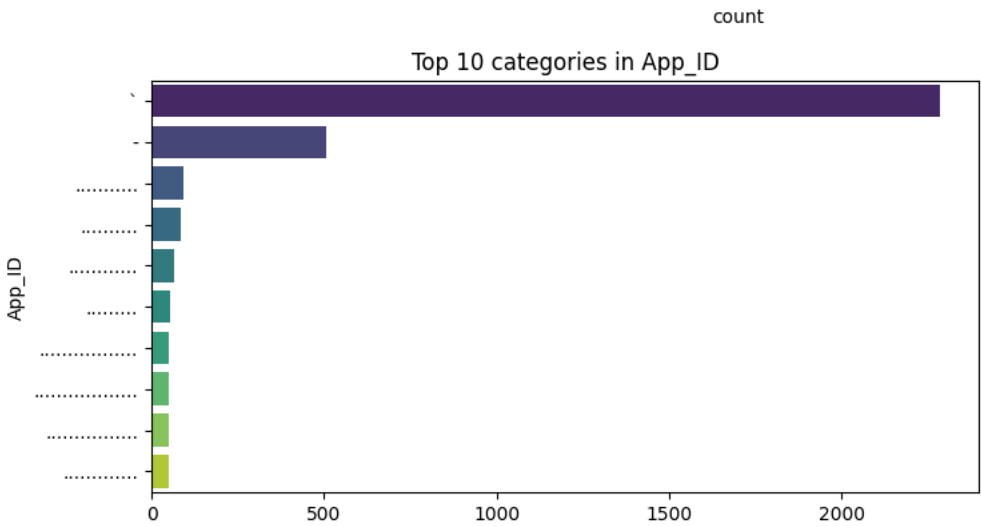
```
safe_hist(campaign_df, "Campaign Data Distributions")
safe_hist(outreach_df, "Outreach Data Distributions")
safe_hist(applicant_df, "Applicant Data Distributions")
```



```
# Bar plots for categorical fields
def plot_top_categories(df, col, top_n=10):
    plt.figure(figsize=(8,4))
    sns.countplot(y=df[col], order=df[col].value_counts().head(top_n).index, palette="viridis")
    plt.title(f"Top {top_n} categories in {col}")
    plt.show()

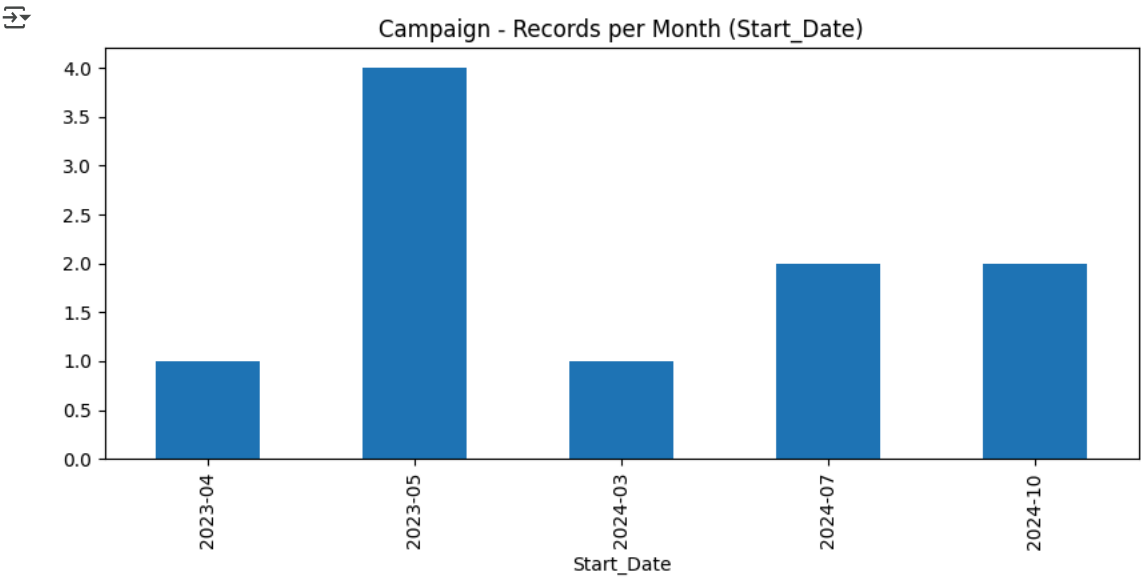
for df, name in [(campaign_df, "Campaign"), (outreach_df, "Outreach"), (applicant_df, "Applicant")]:
    for col in df.select_dtypes(include="object").columns[:3]:
        plot_top_categories(df, col)
```





```
# Bivariate Analysis
# Correlation heatmap for numeric fields
for df, name in [(campaign_df, "Campaign"), (outreach_df, "Outreach"), (applicant_df, "Applicant")]:
    numeric_df = df.select_dtypes(include="number")
    if not numeric_df.empty:
        plt.figure(figsize=(8,6))
        sns.heatmap(numeric_df.corr(), annot=True, cmap="coolwarm")
        plt.title(f"{name} - Correlation Heatmap")
        plt.show()

# Time-Based Trends (if date exists)
for df, name in [(campaign_df, "Campaign"), (outreach_df, "Outreach"), (applicant_df, "Applicant")]:
    date_cols = df.select_dtypes(include="datetime64").columns
    if len(date_cols) > 0:
        for col in date_cols:
            plt.figure(figsize=(10,4))
            df[col].groupby(df[col].dt.to_period("M")).count().plot(kind="bar")
            plt.title(f"{name} - Records per Month ({col})")
            plt.show()
```



```
# Outlier Detection (Boxplots)
```