```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "EDA  Healthcare Data"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "import all the necessary libraries.\n",
    "pandas and numpy are used for data manipulation and analysis.\n",
    "matplotlib.pyplot and seaborn are used for visualization.\n",
    "%matplotlib inline ensures that plots are displayed within the notebook.\n",
    "\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 1,
   "metadata": {},
   "outputs": [],
   "source": [
    "import pandas as pd\n",
    "import numpy as np\n",
    "import matplotlib.pyplot as plt\n",
    "import seaborn as sns\n",
    "\n",
    "# Set matplotlib to display inline\n",
    "%matplotlib inline\n"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "Load your dataset and handle any potential file loading errors.\n",
    "\n",
    "Replace 'path/to/healthcare_providers_data.csv' with the actual path to your CSV file.\n",
    "This step reads the dataset into a pandas DataFrame and checks if the file exists."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Dataset loaded successfully\n"
     ]
    }
   ],
   "source": [
    "# Load the dataset\n",
    "try:\n",
    "    df = pd.read_csv('Healthcare Providers.csv')\n",
    "    print(\"Dataset loaded successfully\")\n",
    "except FileNotFoundError:\n",
    "    print(\"Error: File not found. Please check the file path.\")\n",
    "# Step 2: Subset the Data (first 30 rows)\n",
    "df_subset = df.head(30)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "In the third cell, display basic information about the dataset.\n",
    "\n",
    "Display basic information about the dataset to understand its structure.\n",
    "df.info() provides information about the dataset, including column names, non-null counts, and data types.\n",
    "df.head() displays the first few rows of the dataset to give a preview of the data.\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Dataset Information:\n",
      "<class 'pandas.core.frame.DataFrame'>\n",
      "RangeIndex: 100000 entries, 0 to 99999\n",
      "Data columns (total 27 columns):\n",
      " #   Column                                                    Non-Null Count   Dtype  \n",
      "---  ------                                                    --------------   -----  \n",
      " 0   index                                                     100000 non-null  int64  \n",
      " 1   National Provider Identifier                              100000 non-null  int64  \n",
      " 2   Last Name/Organization Name of the Provider               100000 non-null  object \n",
      " 3   First Name of the Provider                                95745 non-null   object \n",
      " 4   Middle Initial of the Provider                            70669 non-null   object \n",
      " 5   Credentials of the Provider                               92791 non-null   object \n",
      " 6   Gender of the Provider                                    95746 non-null   object \n",
      " 7   Entity Type of the Provider                               100000 non-null  object \n",
      " 8   Street Address 1 of the Provider                          100000 non-null  object \n",
      " 9   Street Address 2 of the Provider                          40637 non-null   object \n",
      " 10  City of the Provider                                      100000 non-null  object \n",
      " 11  Zip Code of the Provider                                  100000 non-null  float64\n",
      " 12  State Code of the Provider                                100000 non-null  object \n",
      " 13  Country Code of the Provider                              100000 non-null  object \n",
      " 14  Provider Type                                             100000 non-null  object \n",
      " 15  Medicare Participation Indicator                          100000 non-null  object \n",
      " 16  Place of Service                                          100000 non-null  object \n",
      " 17  HCPCS Code                                                100000 non-null  object \n",
      " 18  HCPCS Description                                         100000 non-null  object \n",
      " 19  HCPCS Drug Indicator                                      100000 non-null  object \n",
      " 20  Number of Services                                        100000 non-null  object \n",
      " 21  Number of Medicare Beneficiaries                          100000 non-null  object \n",
      " 22  Number of Distinct Medicare Beneficiary/Per Day Services  100000 non-null  object \n",
      " 23  Average Medicare Allowed Amount                           100000 non-null  object \n",
      " 24  Average Submitted Charge Amount                           100000 non-null  object \n",
     " 25  Average Medicare Payment Amount                           100000 non-null  object \n",
     " 26  Average Medicare Standardized Amount                      100000 non-null  object \n",
     "dtypes: float64(1), int64(2), object(24)\n",
     "memory usage: 20.6+ MB\n",
     "\n",
     "First few rows of the dataset:\n"
    ]
   },
   {
    "data": {
     "text/html": [
      "<div>\n",
      "<style scoped>\n",
      "    .dataframe tbody tr th:only-of-type {\n",
      "        vertical-align: middle;\n",
      "    }\n",
      "\n",
      "    .dataframe tbody tr th {\n",
      "        vertical-align: top;\n",
      "    }\n",
      "\n",
      "    .dataframe thead th {\n",
      "        text-align: right;\n",
      "    }\n",
      "</style>\n",
      "<table border=\"1\" class=\"dataframe\">\n",
      "  <thead>\n",
      "    <tr style=\"text-align: right;\">\n",
      "      <th></th>\n",
      "      <th>index</th>\n",
      "      <th>National Provider Identifier</th>\n",
      "      <th>Last Name/Organization Name of the Provider</th>\n",
      "      <th>First Name of the Provider</th>\n",
      "      <th>Middle Initial of the Provider</th>\n",
      "      <th>Credentials of the Provider</th>\n",
      "      <th>Gender of the Provider</th>\n",
      "      <th>Entity Type of the Provider</th>\n",
      "      <th>Street Address 1 of the Provider</th>\n",
      "      <th>Street Address 2 of the Provider</th>\n",
      "      <th>...</th>\n",
      "      <th>HCPCS Code</th>\n",
      "      <th>HCPCS Description</th>\n",
      "      <th>HCPCS Drug Indicator</th>\n",
      "      <th>Number of Services</th>\n",
      "      <th>Number of Medicare Beneficiaries</th>\n",
      "      <th>Number of Distinct Medicare Beneficiary/Per Day Services</th>\n",
      "      <th>Average Medicare Allowed Amount</th>\n",
      "      <th>Average Submitted Charge Amount</th>\n",
      "      <th>Average Medicare Payment Amount</th>\n",
      "      <th>Average Medicare Standardized Amount</th>\n",
      "    </tr>\n",
      "  </thead>\n",
      "  <tbody>\n",
      "    <tr>\n",
      "      <th>0</th>\n",
      "      <td>8774979</td>\n",
      "      <td>1891106191</td>\n",
      "      <td>UPADHYAYULA</td>\n",
      "      <td>SATYASREE</td>\n",
      "      <td>NaN</td>\n",
      "      <td>M.D.</td>\n",
      "      <td>F</td>\n",
      "      <td>I</td>\n",
      "      <td>1402 S GRAND BLVD</td>\n",
      "      <td>FDT 14TH FLOOR</td>\n",
      "      <td>...</td>\n",
      "      <td>99223</td>\n",
      "      <td>Initial hospital inpatient care, typically 70 ...</td>\n",
      "      <td>N</td>\n",
      "      <td>27</td>\n",
      "      <td>24</td>\n",
      "      <td>27</td>\n",
      "      <td>200.58777778</td>\n",
      "      <td>305.21111111</td>\n",
      "      <td>157.26222222</td>\n",
      "      <td>160.90888889</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>1</th>\n",
      "      <td>3354385</td>\n",
      "      <td>1346202256</td>\n",
      "      <td>JONES</td>\n",
      "      <td>WENDY</td>\n",
      "      <td>P</td>\n",
      "      <td>M.D.</td>\n",
      "      <td>F</td>\n",
      "      <td>I</td>\n",
      "      <td>2950 VILLAGE DR</td>\n",
      "      <td>NaN</td>\n",
      "      <td>...</td>\n",
      "      <td>G0202</td>\n",
      "      <td>Screening mammography, bilateral (2-view study...</td>\n",
      "      <td>N</td>\n",
      "      <td>175</td>\n",
      "      <td>175</td>\n",
      "      <td>175</td>\n",
      "      <td>123.73</td>\n",
      "      <td>548.8</td>\n",
      "      <td>118.83</td>\n",
      "      <td>135.31525714</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>2</th>\n",
      "      <td>3001884</td>\n",
      "      <td>1306820956</td>\n",
      "      <td>DUROCHER</td>\n",
      "      <td>RICHARD</td>\n",
      "      <td>W</td>\n",
      "      <td>DPM</td>\n",
      "      <td>M</td>\n",
      "      <td>I</td>\n",
      "      <td>20 WASHINGTON AVE</td>\n",
      "      <td>STE 212</td>\n",
      "      <td>...</td>\n",
      "      <td>99348</td>\n",
      "      <td>Established patient home visit, typically 25 m...</td>\n",
      "      <td>N</td>\n",
      "      <td>32</td>\n",
      "      <td>13</td>\n",
      "      <td>32</td>\n",
        "      <td>90.65</td>\n",
        "      <td>155</td>\n",
        "      <td>64.4396875</td>\n",
        "      <td>60.5959375</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>3</th>\n",
        "      <td>7594822</td>\n",
        "      <td>1770523540</td>\n",
        "      <td>FULLARD</td>\n",
        "      <td>JASPER</td>\n",
        "      <td>NaN</td>\n",
        "      <td>MD</td>\n",
        "      <td>M</td>\n",
        "      <td>I</td>\n",
        "      <td>5746 N BROADWAY ST</td>\n",
        "      <td>NaN</td>\n",
        "      <td>...</td>\n",
        "      <td>81002</td>\n",
        "      <td>Urinalysis, manual test</td>\n",
        "      <td>N</td>\n",
        "      <td>20</td>\n",
        "      <td>18</td>\n",
        "      <td>20</td>\n",
        "      <td>3.5</td>\n",
        "      <td>5</td>\n",
        "      <td>3.43</td>\n",
        "      <td>3.43</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>4</th>\n",
        "      <td>746159</td>\n",
        "      <td>1073627758</td>\n",
        "      <td>PERROTTI</td>\n",
        "      <td>ANTHONY</td>\n",
        "      <td>E</td>\n",
        "      <td>DO</td>\n",
        "      <td>M</td>\n",
        "      <td>I</td>\n",
        "      <td>875 MILITARY TRL</td>\n",
        "      <td>SUITE 200</td>\n",
        "      <td>...</td>\n",
        "      <td>96372</td>\n",
        "      <td>Injection beneath the skin or into muscle for ...</td>\n",
        "      <td>N</td>\n",
        "      <td>33</td>\n",
        "      <td>24</td>\n",
        "      <td>31</td>\n",
        "      <td>26.52</td>\n",
        "      <td>40</td>\n",
        "      <td>19.539393939</td>\n",
        "      <td>19.057575758</td>\n",
        "    </tr>\n",
        "  </tbody>\n",
        "</table>\n",
        "<p>5 rows × 27 columns</p>\n",
        "</div>"
       ],
       "text/plain": [
        "     index  National Provider Identifier  \\\n",
        "0  8774979                    1891106191   \n",
        "1  3354385                    1346202256   \n",
        "2  3001884                    1306820956   \n",
        "3  7594822                    1770523540   \n",
        "4   746159                    1073627758   \n",
        "\n",
        "  Last Name/Organization Name of the Provider First Name of the Provider  \\\n",
        "0                                 UPADHYAYULA                  SATYASREE   \n",
        "1                                       JONES                      WENDY   \n",
        "2                                    DUROCHER                    RICHARD   \n",
        "3                                     FULLARD                     JASPER   \n",
        "4                                    PERROTTI                    ANTHONY   \n",
        "\n",
        "  Middle Initial of the Provider Credentials of the Provider  \\\n",
        "0                            NaN                        M.D.   \n",
        "1                              P                        M.D.   \n",
        "2                              W                         DPM   \n",
        "3                            NaN                          MD   \n",
        "4                              E                          DO   \n",
        "\n",
        "  Gender of the Provider Entity Type of the Provider  \\\n",
        "0                      F                           I   \n",
        "1                      F                           I   \n",
        "2                      M                           I   \n",
        "3                      M                           I   \n",
        "4                      M                           I   \n",
        "\n",
        "  Street Address 1 of the Provider Street Address 2 of the Provider  ...  \\\n",
        "0                1402 S GRAND BLVD                   FDT 14TH FLOOR  ...   \n",
        "1                  2950 VILLAGE DR                              NaN  ...   \n",
        "2                20 WASHINGTON AVE                          STE 212  ...   \n",
        "3               5746 N BROADWAY ST                              NaN  ...   \n",
        "4                 875 MILITARY TRL                        SUITE 200  ...   \n",
        "\n",
        "  HCPCS Code                                  HCPCS Description  \\\n",
        "0      99223  Initial hospital inpatient care, typically 70 ...   \n",
        "1      G0202  Screening mammography, bilateral (2-view study...   \n",
        "2      99348  Established patient home visit, typically 25 m...   \n",
        "3      81002                            Urinalysis, manual test   \n",
        "4      96372  Injection beneath the skin or into muscle for ...   \n",
        "\n",
        "  HCPCS Drug Indicator Number of Services Number of Medicare Beneficiaries  \\\n",
        "0                    N                 27                               24   \n",
        "1                    N                175                              175   \n",
        "2                    N                 32                               13   \n",
        "3                    N                 20                               18   \n",
        "4                    N                 33                               24   \n",
        "\n",
        "  Number of Distinct Medicare Beneficiary/Per Day Services  \\\n",
        "0                                                 27         \n",
        "1                                                175         \n",
        "2                                                 32         \n",
        "3                                                 20         \n",
        "4                                                 31         \n",
        "\n",
        "  Average Medicare Allowed Amount Average Submitted Charge Amount  \\\n",
        "0                    200.58777778                    305.21111111   \n",
        "1                          123.73                           548.8   \n",
        "2                           90.65                             155   \n",
        "3                             3.5                               5   \n",
       "4                           26.52                             40   \n",
       "\n",
       "  Average Medicare Payment Amount Average Medicare Standardized Amount  \n",
       "0                    157.26222222                        160.90888889  \n",
       "1                          118.83                        135.31525714  \n",
       "2                      64.4396875                          60.5959375  \n",
       "3                            3.43                                 3.43  \n",
       "4                    19.539393939                        19.057575758  \n",
       "\n",
       "[5 rows x 27 columns]"
      ]
     },
     "execution_count": 3,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "# Display basic information about the dataset\n",
    "print(\"Dataset Information:\")\n",
    "df.info()\n",
    "\n",
    "print(\"\\nFirst few rows of the dataset:\")\n",
    "df.head()\n"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "\n",
    "Check and handle any missing values in the dataset.\n",
    "df.isnull().sum() counts the number of missing values in each column.\n",
    "df.dropna() removes rows with missing values. You might choose to fill missing values instead depending on the dataset.\n",
    "\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 4,
   "metadata": {},
   "outputs": [],
   "source": [
    "# Handle missing values (e.g., dropping rows with missing values)\n",
    "df = df.dropna()\n"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "Generate basic statistics for numerical columns.\n",
    "df.describe() provides summary statistics for numerical columns, including count, mean, standard deviation, min, max, and quartiles.\n",
    "\n",
    "\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 5,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "\n",
      "Basic Statistics:\n"
     ]
    },
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: right;\n",
       "    }\n",
       "</style>\n",
       "<table border=\"1\" class=\"dataframe\">\n",
       "  <thead>\n",
       "    <tr style=\"text-align: right;\">\n",
       "      <th></th>\n",
       "      <th>index</th>\n",
       "      <th>National Provider Identifier</th>\n",
       "      <th>Zip Code of the Provider</th>\n",
       "    </tr>\n",
       "  </thead>\n",
       "  <tbody>\n",
       "    <tr>\n",
       "      <th>count</th>\n",
       "      <td>2.934600e+04</td>\n",
       "      <td>2.934600e+04</td>\n",
       "      <td>2.934600e+04</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>mean</th>\n",
       "      <td>4.894577e+06</td>\n",
       "      <td>1.497140e+09</td>\n",
       "      <td>4.201327e+08</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>std</th>\n",
       "      <td>2.845657e+06</td>\n",
       "      <td>2.880431e+08</td>\n",
       "      <td>3.143447e+08</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>min</th>\n",
       "      <td>3.900000e+02</td>\n",
       "      <td>1.003002e+09</td>\n",
       "      <td>6.120000e+02</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>25%</th>\n",
      "      <td>2.431374e+06</td>\n",
      "      <td>1.245407e+09</td>\n",
      "      <td>1.521325e+08</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>50%</th>\n",
      "      <td>4.885643e+06</td>\n",
      "      <td>1.497782e+09</td>\n",
      "      <td>3.580146e+08</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>75%</th>\n",
      "      <td>7.321879e+06</td>\n",
      "      <td>1.740266e+09</td>\n",
      "      <td>7.111093e+08</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>max</th>\n",
      "      <td>9.847437e+06</td>\n",
      "      <td>1.993000e+09</td>\n",
      "      <td>9.990166e+08</td>\n",
      "    </tr>\n",
      "  </tbody>\n",
      "</table>\n",
      "</div>"
     ],
     "text/plain": [
      "              index  National Provider Identifier  Zip Code of the Provider\n",
      "count  2.934600e+04                  2.934600e+04              2.934600e+04\n",
      "mean   4.894577e+06                  1.497140e+09              4.201327e+08\n",
      "std    2.845657e+06                  2.880431e+08              3.143447e+08\n",
      "min    3.900000e+02                  1.003002e+09              6.120000e+02\n",
      "25%    2.431374e+06                  1.245407e+09              1.521325e+08\n",
      "50%    4.885643e+06                  1.497782e+09              3.580146e+08\n",
      "75%    7.321879e+06                  1.740266e+09              7.111093e+08\n",
      "max    9.847437e+06                  1.993000e+09              9.990166e+08"
     ]
    },
    "execution_count": 5,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "# Basic statistics\n",
   "print(\"\\nBasic Statistics:\")\n",
   "df.describe()\n"
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {},
  "source": [
   "Visualize the distribution of numerical features using histograms.\n",
   "This step selects numerical columns and plots their distributions using histograms and KDE plots.\n"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 6,
  "metadata": {},
  "outputs": [
   {
    "data": {
     "image/png": "iVBORw0KGgoAAAANSUhEUgAAA1sAAAIjCAYAAAD1OgEdAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
     "text/plain": [
      "<Figure size 1000x600 with 1 Axes>"
     ]
    },
    "metadata": {},
    "output_type": "display_data"
   },
   {
    "data": {
     "image/png": "iVBORw0KGgoAAAANSUhEUgAAA1sAAAIjCAYAAAD1OgEdAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
     "text/plain": [
      "<Figure size 1000x600 with 1 Axes>"
     ]
    },
    "metadata": {},
    "output_type": "display_data"
   },
   {
    "data": {
     "image/png": "iVBORw0KGgoAAAANSUhEUgAAA1sAAAIjCAYAAAD1OgEdAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
     "text/plain": [
      "<Figure size 1000x600 with 1 Axes>"
     ]
    },
    "metadata": {},
    "output_type": "display_data"
   }
  ],
  "source": [
   "# Distribution of numerical features\n",
   "numerical_features = df.select_dtypes(include=[np.number]).columns.tolist()\n",
   "\n",
   "if numerical_features:\n",
   "    for feature in numerical_features:\n",
   "        plt.figure(figsize=(10, 6))\n",
   "        sns.histplot(df[feature],kde=True)\n",
   "        plt.title(f'Distribution of {feature}')\n",
   "        plt.xlabel(feature)\n",
   "        plt.ylabel('Frequency')\n",
   "        plt.show()\n",
   "else:\n",
   "    print(\"No numerical features found in the dataset.\")\n"
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {},
  "source": [
   "Check for outliers in numerical features using boxplots.\n",
   "Boxplots help identify outliers in numerical data.\n"
  ]
 },
  {
   "cell_type": "code",
   "execution_count": 7,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "image/png": "iVBORw0KGgoAAAANSUhEUgAAAx8AAAIjCAYAAABia6bHAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
      "text/plain": [
       "<Figure size 1000x600 with 1 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    },
    {
     "data": {
      "image/png": "iVBORw0KGgoAAAANSUhEUgAAAx8AAAIjCAYAAABia6bHAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
      "text/plain": [
       "<Figure size 1000x600 with 1 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    },
    {
     "data": {
      "image/png": "iVBORw0KGgoAAAANSUhEUgAAAx8AAAIjCAYAAABia6bHAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
      "text/plain": [
       "<Figure size 1000x600 with 1 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    }
   ],
   "source": [
    "# Boxplots for numerical features to check for outliers\n",
    "if numerical_features:\n",
    "    for feature in numerical_features:\n",
    "        plt.figure(figsize=(10, 6))\n",
    "        sns.boxplot(x=df[feature])\n",
    "        plt.title(f'Boxplot of {feature}')\n",
    "        plt.xlabel(feature)\n",
    "        plt.show()\n",
    "else:\n",
    "    print(\"No numerical features found in the dataset.\")\n"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "Visualize the distribution of categorical features using count plots.\n",
    "This step selects categorical columns and plots their value counts.\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 8,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "image/png":
      "text/plain": [
       "<Figure size 1000x600 with 1 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    },
    {
     "data": {
      "image/png": "iVBORw0KGgoAAAANSUhEUgAAA4YAAAIjCAYAAABf1QXkAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
      "text/plain": [
       "<Figure size 1000x600 with 1 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    },
    {
     "data": {
      "image/png": "iVBORw0KGgoAAAANSUhEUgAAA0YAAAIjCAYAAAAnagtFAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
      "text/plain": [
       "<Figure size 1000x600 with 1 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    },
    {
     "data": {
      "image/png": "iVBORw0KGgoAAAANSUhEUgAAA1cAAAIjCAYAAADvBuGTAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
      "text/plain": [
       "<Figure size 1000x600 with 1 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    },
    {
     "data": {
      "image/png": "iVBORw0KGgoAAAANSUhEUgAAA0QAAAIjCAYAAAAAjn9t4AAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
      "text/plain": [
       "<Figure size 1000x600 with 1 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    },
    {
     "data": {
      "image/png": "iVBORw0KGgoAAAANSUhEUgAAA0MAAAIjCAYAAADBQ8ABAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
      "text/plain": [
       "<Figure size 1000x600 with 1 Axes>"
      ]
     },
     "metadata": {},
   "output_type": "display_data"
  },
  {
   "data": {
    "image/png":
    "text/plain": [
     "<Figure size 1000x600 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  },
  {
   "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAABAEAAAIjCAYAAAByNwIfAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
    "text/plain": [
     "<Figure size 1000x600 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  },
  {
   "data": {
    "image/png":
    "text/plain": [
     "<Figure size 1000x600 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  },
  {
   "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAAA08AAAIjCAYAAADbfyCPAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
    "text/plain": [
     "<Figure size 1000x600 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  },
  {
   "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAAA0sAAAIjCAYAAADSlID1AAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
    "text/plain": [
     "<Figure size 1000x600 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  },
  {
   "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAABDgAAAIjCAYAAADm0avtAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
    "text/plain": [
     "<Figure size 1000x600 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  },
  {
   "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAAA0AAAAIjCAYAAAAqdHsCAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
    "text/plain": [
     "<Figure size 1000x600 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  },
  {
   "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAAA0MAAAIjCAYAAADBQ8ABAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
    "text/plain": [
     "<Figure size 1000x600 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  },
  {
   "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAAA2YAAAIjCAYAAABoNwiVAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
    "text/plain": [
     "<Figure size 1000x600 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  },
  {
   "data": {
    "image/png":
    "text/plain": [
     "<Figure size 1000x600 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  },
  {
   "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAAA0IAAAIjCAYAAAAugas/AAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
    "text/plain": [
     "<Figure size 1000x600 with 1 Axes>"
    ]
   },
   "metadata": {},
   "output_type": "display_data"
  },
  {
   "data": {
    "image/png": "iVBORw0KGgoAAAANSUhEUgAAA1IAAAIjCAYAAAAJLyrXAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
    "text/plain": [
     "<Figure size 1000x600 with 1 Axes>"
    ]
   },
   "metadata": {},
     "output_type": "display_data"
    },
    {
     "data": {
      "image/png": "iVBORw0KGgoAAAANSUhEUgAAA1IAAAIjCAYAAAAJLyrXAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
      "text/plain": [
       "<Figure size 1000x600 with 1 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    },
    {
     "data": {
      "image/png": "iVBORw0KGgoAAAANSUhEUgAAA1IAAAIjCAYAAAAJLyrXAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
      "text/plain": [
       "<Figure size 1000x600 with 1 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    }
   ],
   "source": [
    "# Step 3: Select 10 Categorical Features\n",
    "categorical_features = df_subset.select_dtypes(include=[object]).columns.tolist()[:20]\n",
    "\n",
    "# Step 4: Plot Count Plots for Selected Categorical Features\n",
    "if categorical_features:\n",
    "    for feature in categorical_features:\n",
    "        plt.figure(figsize=(10, 6))\n",
    "        sns.countplot(y=df_subset[feature], order=df_subset[feature].value_counts().index)\n",
    "        plt.title(f'Count Plot of {feature}')\n",
    "        plt.xlabel('Count')\n",
    "        plt.ylabel(feature)\n",
    "        plt.show()\n",
    "else:\n",
    "    print(\"No categorical features found in the dataset.\")"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "Generate and visualize the correlation matrix for numerical features.\n",
    "The correlation matrix shows relationships between numerical features.\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 9,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "image/png": "iVBORw0KGgoAAAANSUhEUgAAA3wAAAJdCAYAAABpk3UFAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjkuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy80BEi2AAAACXBIWXMAAA9hAAAPYQGoP6dp
      "text/plain": [
       "<Figure size 1200x700 with 2 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    }
   ],
   "source": [
    "# Correlation matrix for numerical features\n",
    "if numerical_features:\n",
    "    plt.figure(figsize=(12, 7))\n",
    "    correlation_matrix = df[numerical_features].corr()\n",
    "    sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.2)\n",
    "    plt.title('Correlation Matrix')\n",
    "    plt.show()\n",
    "else:\n",
    "    print(\"No numerical features found for correlation matrix.\")\n"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "Visualize relationships between numerical features using pairplots.\n",
    "Pairplots help visualize pairwise relationships between numerical features.\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 10,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "image/png":
      "text/plain": [
       "<Figure size 750x750 with 12 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    }
   ],
   "source": [
    "# Pairplot to visualize relationships between numerical features\n",
    "if len(numerical_features) > 1:\n",
    "    sns.pairplot(df[numerical_features])\n",
    "    plt.title('Pairplot of Numerical Features')\n",
    "    plt.show()\n",
    "else:\n",
    "    print(\"Not enough numerical features for pairplot.\")\n"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "Create scatter plots for specific feature pairs to examine relationships.\n",
    "Replace 'feature1' and 'feature2' with actual column names to examine their relationship.\n",
    "\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 11,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Scatter plot features 'feature1' and 'feature2' not found in the dataset.\n"
     ]
    }
   ],
   "source": [
    "# Scatter plot for selected features (replace 'feature1' and 'feature2' with actual column names)\n",
    "if 'feature1' in df.columns and 'feature2' in df.columns:\n",
    "    plt.figure(figsize=(10, 6))\n",
    "    sns.scatterplot(x='feature1', y='feature2', data=df)\n",
    "    plt.title('Scatter Plot of feature1 vs feature2')\n",
    "    plt.xlabel('feature1')\n",
    "    plt.ylabel('feature2')\n",
    "    plt.show()\n",
    "else:\n",
    "    print(\"Scatter plot features 'feature1' and 'feature2' not found in the dataset.\")\n"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "isualize the distribution of missing values in the dataset.\n",
    "The heatmap shows where missing values are located in the dataset."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 12,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "image/png":
      "text/plain": [
       "<Figure size 1200x800 with 1 Axes>"
      ]
     },
     "metadata": {},
     "output_type": "display_data"
    }
   ],
   "source": [
    "# Heatmap for missing values\n",
    "plt.figure(figsize=(12, 8))\n",
    "sns.heatmap(df.isnull(), cbar=False, cmap='viridis')\n",
    "plt.title('Heatmap of Missing Values')\n",
    "plt.show()\n"
   ]
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.12.2"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 2
}
```