```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime



data = pd.read_csv('data.csv')


data.head()


data.describe()


# prompt: cheack null value in all data

data.isnull().sum()



data = data.rename(columns={'Unnamed: 0': 'index'})


data = data.set_index('index')


data.head(

)


# prompt: perform eda process on data

import matplotlib.pyplot as plt
# Check for missing values
missing_values = data.isnull().sum()

# Check for duplicate rows
duplicate_rows = data[data.duplicated()]







print(duplicate_rows)


data.shape


data.info()


district_counts = data['district'].value_counts().reset_index()
district_counts.columns = ['district', 'count']
print(district_counts)


pincode_counts = data['pincode'].value_counts().reset_index()
pincode_counts.columns = ['pincode', 'count']
print(pincode_counts)



status_counts = data['status'].value_counts().reset_index()
status_counts.columns = ['status', 'count']
print(status_counts)


promoter_counts = data['promoter_name'].value_counts().reset_index()
promoter_counts.columns = ['promoter_name', 'count']
print(promoter_counts)
```

```
project_type_counts = data['project_type'].value_counts().reset_index()
project_type_counts.columns = ['project_type', 'count']
print(project_type_counts)


number_of_plots_counts = data['number_of_plots'].value_counts().reset_index()
number_of_plots_counts.columns = ['number_of_plots', 'count']
print(number_of_plots_counts)


# Plot histogram for a single numerical variable
plt.figure(figsize=(8, 6))
sns.histplot(data['projectarea_sqmts'], bins=20, kde=True)
plt.title('Distribution of Project Area (sqmts)')
plt.xlabel('Project Area (sqmts)')
plt.ylabel('Frequency')
plt.show()


# Plot histogram for another numerical variable
plt.figure(figsize=(8, 6))
sns.histplot(data['sanctioned_fsi'], bins=20, kde=True)
plt.title('Distribution of Sanctioned FSI')
plt.xlabel('Sanctioned FSI')
plt.ylabel('Frequency')
plt.show()


# Plot count plot for a categorical variable
plt.figure(figsize=(8, 6))
sns.countplot(data=data, x='status')
plt.title('Distribution of Project Status')
plt.xlabel('Status')
plt.ylabel('Count')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.show()


# Plot bar plot for top promoters
plt.figure(figsize=(12, 8))
top_promoters = data['promoter_name'].value_counts().head(10)
sns.barplot(x=top_promoters.values, y=top_promoters.index, palette='viridis')
plt.title('Top 10 Promoters by Project Count')
plt.xlabel('Number of Projects')
plt.ylabel('Promoter Name')
plt.show()


# Plot box plot for project area across different project types
plt.figure(figsize=(12, 8))
sns.boxplot(data=data, x='project_type', y='projectarea_sqmts', palette='Set3')
plt.title('Distribution of Project Area across Project Types')
plt.xlabel('Project Type')
plt.ylabel('Project Area (sqmts)')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.show()



# Plot scatter plot for project area vs. number of plots
plt.figure(figsize=(10, 6))
sns.scatterplot(data=data, x='projectarea_sqmts', y='number_of_plots', alpha=0.5)
plt.title('Project Area vs. Number of Plots')
plt.xlabel('Project Area (sqmts)')
plt.ylabel('Number of Plots')
plt.show()



# Calculate correlation matrix
correlation_matrix = data.corr()

# Plot heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('Correlation Heatmap of Numerical Variables')
plt.show()
```

```
# Plot bar plot for count of projects in each district
plt.figure(figsize=(12, 8))
sns.countplot(data=data, x='district', palette='muted', order=data['district'].value_counts().index)
plt.title('Count of Projects in Each District')
plt.xlabel('District')
plt.ylabel('Count of Projects')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.show()
```

```
# Create a crosstab of district and status
district_status_cross = pd.crosstab(data['district'], data['status'])

# Plot stacked bar plot
plt.figure(figsize=(72, 72))
district_status_cross.plot(kind='bar', stacked=True, colormap='viridis')
plt.title('Distribution of Project Status by District')
plt.xlabel('District')
plt.ylabel('Count')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.legend(title='Status', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

```
# Create a crosstab of project name and status
project_status_cross = pd.crosstab(data['project_name'], data['status'])

# Plot stacked bar plot
plt.figure(figsize=(15, 8))
project_status_cross.plot(kind='bar', stacked=True, colormap='viridis')
plt.title('Distribution of Project Status by Project Name')
plt.xlabel('Project Name')
plt.ylabel('Count')
plt.xticks(rotation=90)  # Rotate x-axis labels for better readability
plt.legend(title='Status', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

```
# Group by 'district' and 'status' columns and count occurrences
district_status_counts = data.groupby(['district', 'status']).size().reset_index(name='count')

# Display the result
print(district_status_counts)
```

```
pune_district_status_counts = district_status_counts[district_status_counts['district'] == 'Pune']

# Display the result
print(pune_district_status_counts)
```

```
thane_district_status_counts = district_status_counts[district_status_counts['district'] == 'Thane']

# Display the result
print(thane_district_status_counts)
```

```
mumbai_suburban_district_status_counts = district_status_counts[district_status_counts['district'] == 'Mumbai Suburban']

# Display the result
print(mumbai_suburban_district_status_counts)
```

```
raigarh_district_status_counts = district_status_counts[district_status_counts['district'] == 'Raigarh']

# Display the result
print(raigarh_district_status_counts)
```

```
nashik_district_status_counts = district_status_counts[district_status_counts['district'] == 'Nashik']

# Display the result
print(nashik_district_status_counts)
```

```
mumbai_city_district_status_counts = district_status_counts[district_status_counts['district'] == 'Mumbai City']

# Display the result
print(mumbai_city_district_status_counts)
```

```
nagpur_district_status_counts = district_status_counts[district_status_counts['district'] == 'Nagpur']

# Display the result
print(nagpur_district_status_counts)
```

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

# Set the style of the plot
sns.set_style("whitegrid")

# Create a bar plot
plt.figure(figsize=(12, 8))
sns.barplot(data=district_status_counts, x='count', y='district', hue='status', palette='viridis')

# Add title and labels
plt.title('Distribution of Project Status by District')
plt.xlabel('Count')
plt.ylabel('District')

# Show the plot
plt.show()
```

```
# Filter data for district "Pune"
pune_district_status_counts = district_status_counts[district_status_counts['district'] == 'Pune']

# Set the style of the plot
sns.set_style("whitegrid")

# Create a bar plot
plt.figure(figsize=(8, 6))
sns.barplot(data=pune_district_status_counts, x='count', y='status', palette='viridis')

# Add title and labels
plt.title('Distribution of Project Status in Pune')
plt.xlabel('Count')
plt.ylabel('Status')

# Show the plot
plt.show()
```

```
# Filter data for district "Thane"
thane_district_status_counts = district_status_counts[district_status_counts['district'] == 'Thane']

# Set the style of the plot
sns.set_style("whitegrid")

# Create a bar plot
plt.figure(figsize=(8, 6))
sns.barplot(data=thane_district_status_counts, x='count', y='status', palette='viridis')



# Filter data for district "Mumbai City"
mumbai_city_district_status_counts = district_status_counts[district_status_counts['district'] == 'Mumbai City']

# Set the style of the plot
sns.set_style("whitegrid")

# Create a bar plot
plt.figure(figsize=(8, 6))
sns.barplot(data=mumbai_city_district_status_counts, x='count', y='status', palette='viridis')

# Add title and labels
plt.title('Distribution of Project Status in Mumbai City')
plt.xlabel('Count')
plt.ylabel('Status')

# Show the plot
plt.show()


data.head()


# Group by project_name, promoter_name, district, and project_status and count occurrences
grouped_data = data.groupby(['project_name',  'district', 'status']).size()

# Display the grouped data
print(grouped_data)


grouped_data.head()


import seaborn as sns
import matplotlib.pyplot as plt

# Check the structure of the grouped data DataFrame
print(grouped_data.head())

# If the 'count' column is not present, we need to rename the Series before converting it to a DataFrame
# Ensure the 'count' column exists in the DataFrame
grouped_data_df = grouped_data.rename('count').reset_index()

# Pivot the grouped data to create a matrix of project counts
pivot_data = grouped_data_df.pivot_table(index='project_name', columns=['district', 'status'], values='count', fill_value=0)

# Set the style of the plot
sns.set_style("whitegrid")

# Create a heatmap
plt.figure(figsize=(16, 10))
sns.heatmap(pivot_data, cmap='viridis', annot=True, fmt='d', linewidths=0.5)
plt.title('Project Count by Project Name, District, and Status')
plt.xlabel('District - Status')
plt.ylabel('Project Name')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
project_name                                          district         status
"Laxminath" Project by "Swastika Landmarks"           Pune             New Project    1
"TAKSH REGENCY" a Project by "Reliable Construction"  Pune             New Project    1
1 BY URBANONE                                         Mumbai Suburban  New Project    1
1 GOLDLEAF                                            Pune             New Project    1
1 Hallmark Aveneu Phase II                            Pune             New Project    1
dtype: int64
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-131-ad0890fdd92b> in <cell line: 19>()
     17 # Create a heatmap
     18 plt.figure(figsize=(16, 10))
---> 19 sns.heatmap(pivot_data, cmap='viridis', annot=True, fmt='d', linewidths=0.5)
     20 plt.title('Project Count by Project Name, District, and Status')
     21 plt.xlabel('District - Status')

                                12 frames
/usr/local/lib/python3.10/dist-packages/numpy/core/fromnumeric.py in _wrapit(obj, method, *args, **kwds)
     44         wrap = None
     45     result = getattr(asarray(obj), method)(*args, **kwds)
---> 46     if wrap:
     47         if not isinstance(result, mu.ndarray):
     48             result = asarray(result)

KeyboardInterrupt:

Error in callback <function _draw_all_if_interactive at 0x783921724ee0> (for post_execute):
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/matplotlib/pyplot.py in _draw_all_if_interactive()
    118 def _draw_all_if_interactive():
    119     if matplotlib.is_interactive():
--> 120         draw_all()
    121
    122

                                11 frames
/usr/local/lib/python3.10/dist-packages/matplotlib/collections.py in draw(self, renderer)
   2096                 gc, triangles, colors, transform.frozen())
   2097         else:
-> 2098             renderer.draw_quad_mesh(
   2099                 gc, transform.frozen(),
   2100                 coordinates.shape[1] - 1, coordinates.shape[0] - 1,

KeyboardInterrupt:
```