EDA

1. Imports and Load Cleaned Data

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# --- Setup ---
# Set a consistent style for all our plots
sns.set_style("whitegrid")
plt.rcParams['figure.figsize'] = (12, 7) # Set default figure size

# --- Load Data ---
# Load the cleaned data from the 'processed' folder
file_path = '../data/processed/cleaned_loan_data.csv'
try:
    df = pd.read_csv(file_path)
    print("Cleaned data loaded successfully.")
except FileNotFoundError:
    print(f"ERROR: The file {file_path} was not found. Please run the first note

df.head()
```

```
Cleaned data loaded successfully.
```

Out[1]:

	uniqueid	disbursed_amount	asset_cost	ltv	branch_id	supplier_id	manufacturer_
0	420825	50578	58400	89.55	67	22807	
1	417566	53278	61360	89.63	67	22807	
2	539055	52378	60300	88.39	67	22807	
3	529269	46349	61500	76.42	67	22807	
4	563215	43594	78256	57.50	67	22744	

5 rows × 42 columns

2. Statistical Description

In [2]:
```
# --- Statistical Summary ---
print("Statistical Description of Numerical Variables:")
# .T transposes the output to make it easier to read
df.describe().T
```

```
Statistical Description of Numerical Variables:
```

Out[2]:

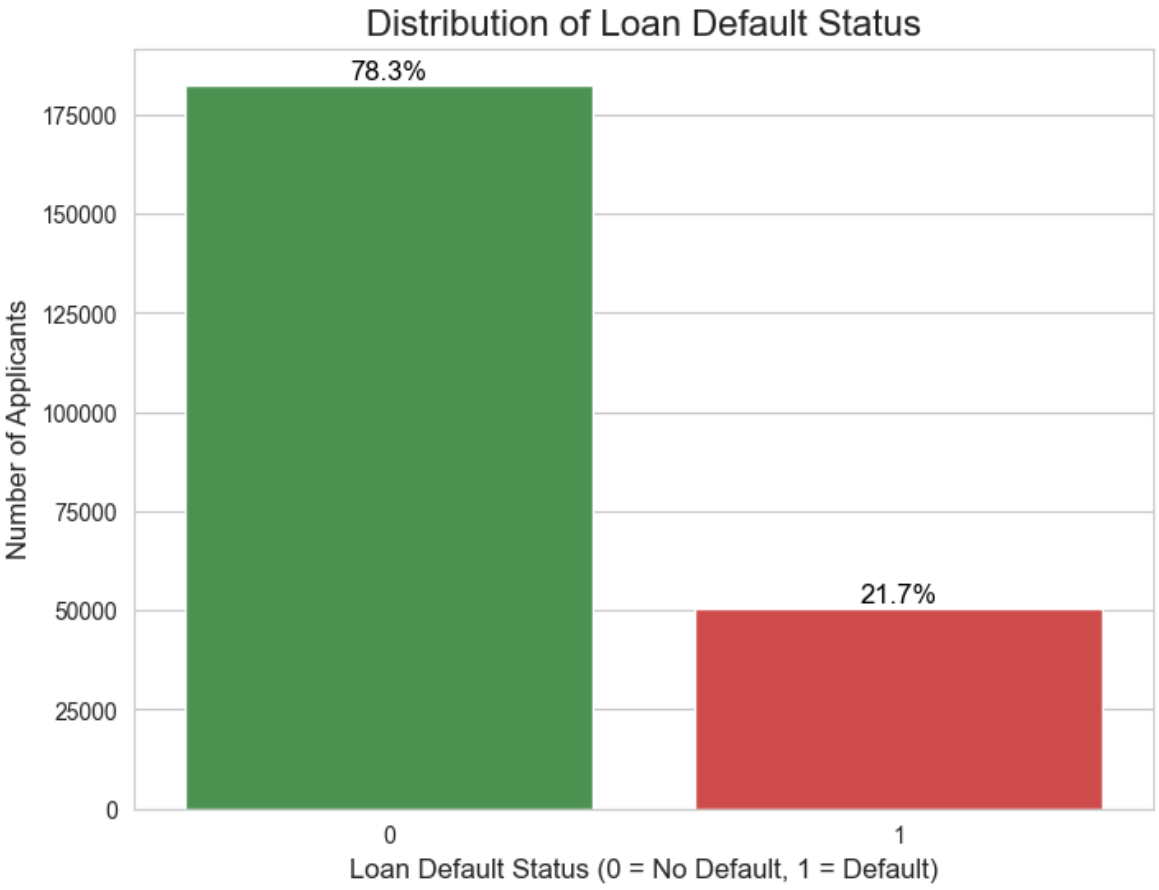
	count	mean	std	min
uniqueid	233154.0	535917.573376	6.831569e+04	417428.00
disbursed_amount	233154.0	54356.993528	1.297131e+04	13320.00
asset_cost	233154.0	75865.068144	1.894478e+04	37000.00
ltv	233154.0	74.746530	1.145664e+01	10.03
branch_id	233154.0	72.936094	6.983499e+01	1.00
supplier_id	233154.0	19638.635035	3.491950e+03	10524.00
manufacturer_id	233154.0	69.028054	2.214130e+01	45.00
current_pincode_id	233154.0	3396.880247	2.238148e+03	1.00
state_id	233154.0	7.262243	4.482230e+00	1.00
employee_code_id	233154.0	1549.477148	9.752613e+02	1.00
mobileno_avl_flag	233154.0	1.000000	0.000000e+00	1.00
aadhar_flag	233154.0	0.840320	3.663097e-01	0.00
pan_flag	233154.0	0.075577	2.643201e-01	0.00
voterid_flag	233154.0	0.144943	3.520439e-01	0.00
driving_flag	233154.0	0.023242	1.506720e-01	0.00
passport_flag	233154.0	0.002127	4.607421e-02	0.00
perform_cns_score	233154.0	289.462994	3.383748e+02	0.00
pri_no_of_accts	233154.0	2.440636	5.217233e+00	0.00
pri_active_accts	233154.0	1.039896	1.941496e+00	0.00
pri_overdue_accts	233154.0	0.156549	5.487867e-01	0.00
pri_current_balance	233154.0	165900.076936	9.422736e+05	-6678296.00
pri_sanctioned_amount	233154.0	218503.855323	2.374794e+06	0.00
pri_disbursed_amount	233154.0	218065.898655	2.377744e+06	0.00
sec_no_of_accts	233154.0	0.059081	6.267946e-01	0.00
sec_active_accts	233154.0	0.027703	3.160566e-01	0.00
sec_overdue_accts	233154.0	0.007244	1.110789e-01	0.00
sec_current_balance	233154.0	5427.792819	1.702370e+05	-574647.00
sec_sanctioned_amount	233154.0	7295.923347	1.831560e+05	0.00
sec_disbursed_amount	233154.0	7179.997873	1.825925e+05	0.00
primary_instal_amt	233154.0	13105.481720	1.513679e+05	0.00
sec_instal_amt	233154.0	323.268449	1.555369e+04	0.00
new_accts_in_last_six_months	233154.0	0.381833	9.551067e-01	0.00
delinquent_accts_in_last_six_months	233154.0	0.097481	3.844390e-01	0.00

	count	mean	std	min
no_of_inquiries	233154.0	0.206615	7.064977e-01	0.00
loan_default	233154.0	0.217071	4.122523e-01	0.00
age	233154.0	41.100946	9.805992e+00	25.00

3. Target Variable Distribution

In [4]:
```
# --- Target Variable Analysis ---
plt.figure(figsize=(8, 6))
plt.title('Distribution of Loan Default Status', fontsize=16)

# Create the count plot
ax = sns.countplot(x='loan_default', data=df, palette=['#43a047', '#e53935'])

# Add labels and annotations
plt.xlabel('Loan Default Status (0 = No Default, 1 = Default)', fontsize=12)
plt.ylabel('Number of Applicants', fontsize=12)

# Calculate and display the percentage on the bars
total = len(df)
for p in ax.patches:
    percentage = f'{100 * p.get_height() / total:.1f}%'
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    ax.annotate(percentage, (x, y), ha='center', va='bottom', fontsize=12, color

# --- SAVE THE FIGURE ---
# This line saves the plot to the specified file before displaying it.
plt.savefig('../reports/figures/target_variable_distribution.png', bbox_inches='

plt.show()

# Print the exact default rate
default_rate = df['loan_default'].value_counts(normalize=True).loc[1] * 100
print(f"Overall Default Rate: {default_rate:.2f}%")
```

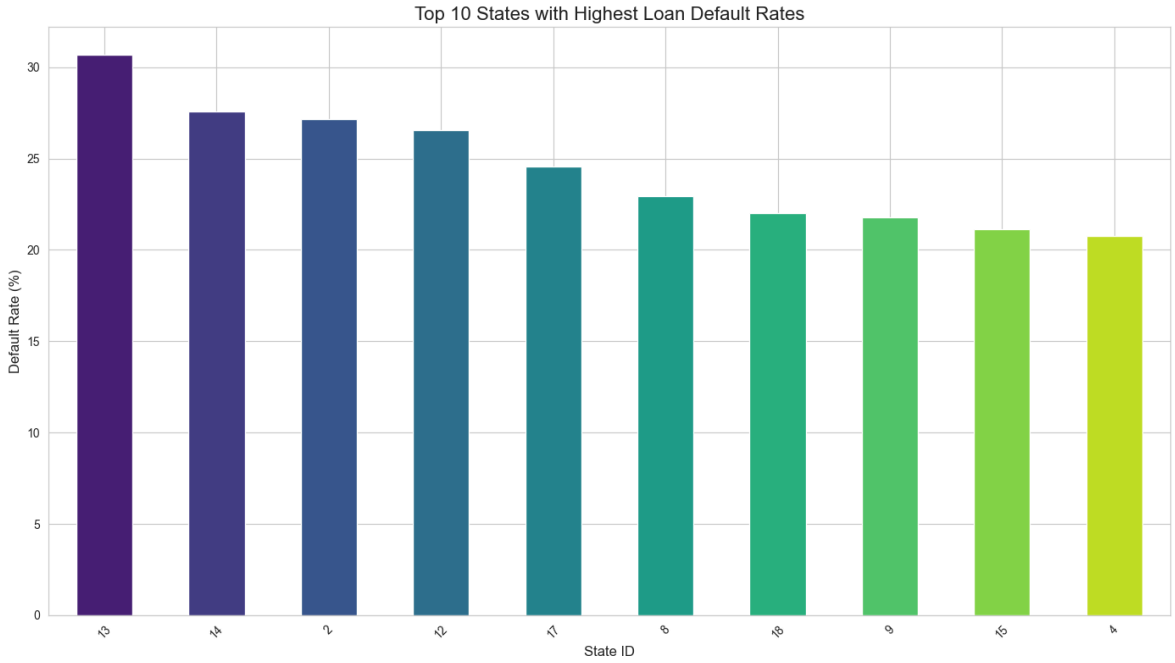
```
C:\Users\ThapeloMasebe\AppData\Local\Temp\ipykernel_16100\3021288222.py:6: Future
Warning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  ax = sns.countplot(x='loan_default', data=df, palette=['#43a047', '#e53935'])
```




```
Overall Default Rate: 21.71%
```

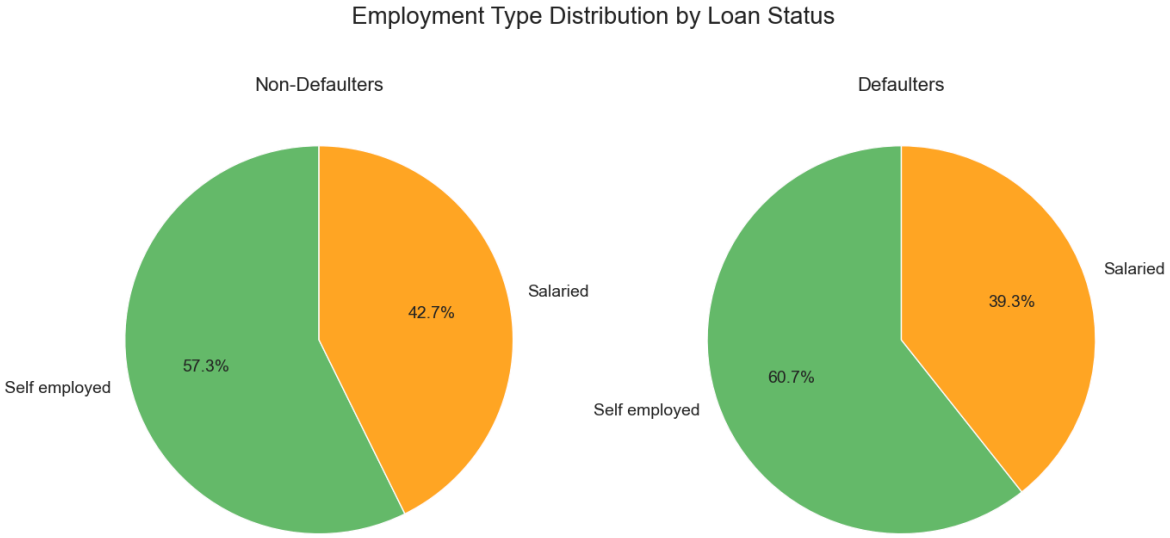
4. Default Rate Across Categories

In [5]:
```
# Calculate the default rate for each state
state_default_rate = df.groupby('state_id')['loan_default'].mean().sort_values(a

# Plot the top 10 states with the highest default rates
plt.figure(figsize=(14, 8))
state_default_rate.head(10).plot(kind='bar', color=sns.color_palette("viridis",
plt.title('Top 10 States with Highest Loan Default Rates', fontsize=16)
plt.xlabel('State ID', fontsize=12)
plt.ylabel('Default Rate (%)', fontsize=12)
plt.xticks(rotation=45)
plt.tight_layout() # Adjusts plot to ensure everything fits without overlapping

# --- SAVE THE FIGURE ---
plt.savefig('../reports/figures/top10_states_by_default_rate.png', bbox_inches='

plt.show()
```



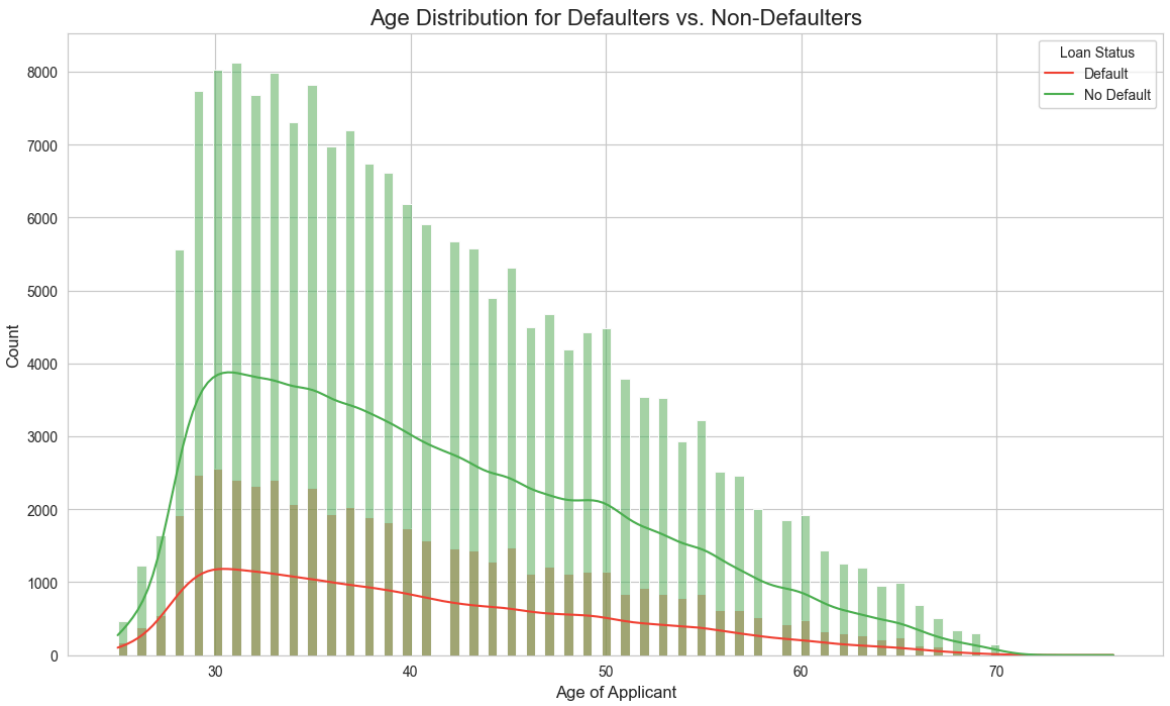
5. Employment Type and Defaulting

In [6]:
```
# --- Employment Type vs. Default Status ---
fig, axes = plt.subplots(1, 2, figsize=(16, 8))
plt.suptitle('Employment Type Distribution by Loan Status', fontsize=20)

# Pie chart for Non-Defaulters (loan_default = 0)
df[df['loan_default'] == 0]['employment_type'].value_counts().plot.pie(
autopct='%1.1f%%', ax=axes[0], startangle=90,
colors=['#66bb6a', '#ffa726'], textprops={'fontsize': 14}
)
axes[0].set_title('Non-Defaulters', fontsize=16)
axes[0].set_ylabel('') # Hide the y-label

# Pie chart for Defaulters (loan_default = 1)
df[df['loan_default'] == 1]['employment_type'].value_counts().plot.pie(
autopct='%1.1f%%', ax=axes[1], startangle=90,
colors=['#66bb6a', '#ffa726'], textprops={'fontsize': 14}
)
axes[1].set_title('Defaulters', fontsize=16)
axes[1].set_ylabel('') # Hide the y-label

# --- SAVE THE FIGURE ---
plt.savefig('../reports/figures/employment_type_pie_charts.png', bbox_inches='ti

plt.show()
```



6. Age and Defaulting

In [7]:
```
# --- Age Distribution by Default Status ---
plt.figure(figsize=(14, 8))
# Use a histogram with a Kernel Density Estimate (KDE) for a smooth line
sns.histplot(data=df, x='age', hue='loan_default', multiple='layer', kde=True, p
plt.title('Age Distribution for Defaulters vs. Non-Defaulters', fontsize=16)
plt.xlabel('Age of Applicant', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.legend(title='Loan Status', labels=['Default', 'No Default'])

# --- SAVE THE FIGURE ---
plt.savefig('../reports/figures/age_distribution_by_default.png', bbox_inches='t

plt.show()
```



7. ID Proof Analysis

In [8]:
```
# --- ID Proof Analysis ---
# The ID columns are flags (1 = submitted, 0 = not submitted)
id_flags = ['aadhar_flag', 'pan_flag', 'voterid_flag', 'driving_flag', 'passport
id_counts = df[id_flags].sum().sort_values(ascending=False)

# Create a bar plot
plt.figure(figsize=(10, 6))
sns.barplot(x=id_counts.index, y=id_counts.values, palette='plasma')
plt.title('Number of Applicants by ID Proof Submitted', fontsize=16)
plt.xlabel('Type of ID Proof', fontsize=12)
plt.ylabel('Total Number of Applicants', fontsize=12)
plt.xticks(ticks=range(len(id_flags)), labels=[flag.replace('_flag', '').title()
plt.tight_layout()

# --- SAVE THE FIGURE ---
plt.savefig('../reports/figures/id_proof_counts.png', bbox_inches='tight')

plt.show()
```

```
C:\Users\ThapeloMasebe\AppData\Local\Temp\ipykernel_16100\353651389.py:8: FutureW
arning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.barplot(x=id_counts.index, y=id_counts.values, palette='plasma')
```