

Total Loan Applications

```
[12]: total_loan_application = df['id'].count()
      print("Total Loan Applications:", total_loan_application)
```

Total Loan Applications: 38576

MTD Total Loan Applications

```
[17]: latest_issue_date = df['issue_date'].max()
      latest_year = latest_issue_date.year
      latest_month = latest_issue_date.month

      mtd_data = df[(df['issue_date'].dt.year == latest_year) & (df['issue_date'].dt.month == latest_month)]
      mtd_loan_applications = mtd_data['id'].count()
      print(f"MTD Loan Applications (for {latest_issue_date.strftime('%B %Y')}) : {mtd_loan_applications}")
```

MTD Loan Applications (for December 2021):4314

Total Funded Amount

```
[29]: total_funded_amount = df['loan_amount'].sum()
      total_funded_amount_millions = total_funded_amount/ 1000000
      print ("Total Funded Amount: ${:.2f}M". format(total_funded_amount_millions))
```

Total Funded Amount: \$435.76M

MTD Total Funded Amount

```
[27]: latest_issue_date = df['issue_date'].max()
      latest_year = latest_issue_date.year
      latest_month = latest_issue_date.month

      mtd_data = df[(df['issue_date'].dt.year == latest_year) & (df['issue_date'].dt.month == latest_month)]

      mtd_total_funded_amount = mtd_data['loan_amount'].sum()
      mtd_total_funded_amount_millions = mtd_total_funded_amount/1000000

      print ("MTD Total Funded Amount: ${:.2f}M" . format(mtd_total_funded_amount_millions))
```

MTD Total Funded Amount: \$53.98M

Total Amount Received

```
[28]: total_amount_recieved = df['total_payment'].sum()
      total_amount_recieved_millions = total_funded_amount/ 1000000
      print ("Total Amount Recieved: ${:.2f}M". format(total_amount_recieved_millions))
```

Total Amount Received: \$435.76M

MTD Total Amount Received

```
[31]: latest_issue_date = df['issue_date'].max()
      latest_year = latest_issue_date.year
      latest_month = latest_issue_date.month

      mtd_data = df[(df['issue_date'].dt.year == latest_year) & (df['issue_date'].dt.month == latest_month)]

      mtd_total_amount_recieved = mtd_data['total_payment'].sum()
      mtd_total_amount_recieved_millions = mtd_total_amount_recieved/1000000

      print ("MTD Total Amount Recieved: ${:.2f}M" . format(mtd_total_amount_recieved_millions))
```

MTD Total Amount Received: \$58.07M

Average Interest Rate

```
[38]: avg_interest_rate = df['int_rate'].mean()
      print("Average Interest Rate: {:.2f}%".format(avg_interest_rate))
```

Average Interest Rate: 0.12%

Average Debt-to-Income Ratio (DTI)

```
[39]: avg_dti = df['dti'].mean()*100
      print("Average Debt-to-Income Ratio (DTI): {:.2f}%".format(avg_dti))
```

Average Debt-to-Income Ratio (DTI): 13.33%

Average Debt to Income Ratio (DTI): 45.93%

Good Loan Metrics

```
[42]: good_loans = df[df['loan_status'].isin(["Fully Paid", "Current"])]

total_loan_applications = df['id'].count()

good_loan_applications = good_loans['id'].count()
good_loan_funded_amount = good_loans['loan_amount'].sum()
good_loan_received = good_loans['total_payment'].sum()

good_loan_funded_amount_millions = good_loan_funded_amount / 1000000
good_loan_received_millions = good_loan_received / 1000000

good_loan_percentage = (good_loan_applications / total_loan_applications) * 100

print("Good Loan Applications:", good_loan_applications)
print("Good Loan Funded Amount (in Millions): {:.2f}M".format(good_loan_funded_amount_millions))
print("Good Loan Total Received (in Millions): {:.2f}M".format(good_loan_received_millions))
print("Percentage of Good Loan Applications: {:.2f}%".format(good_loan_percentage))
```

Good Loan Applications: 33243
Good Loan Funded Amount (in Millions): \$370.22M
Good Loan Total Received (in Millions): \$435.79M
Percentage of Good Loan Applications: 86.18%

Bad Loan Metrics

```
[43]: bad_loans = df[df['loan_status'].isin(["Charged Off"])]

total_loan_applications = df['id'].count()

bad_loan_applications = bad_loans['id'].count()
bad_loan_funded_amount = bad_loans['loan_amount'].sum()
bad_loan_received = bad_loans['total_payment'].sum()

bad_loan_funded_amount_millions = bad_loan_funded_amount / 1000000
bad_loan_received_millions = bad_loan_received / 1000000

bad_loan_percentage = (bad_loan_applications / total_loan_applications) * 100

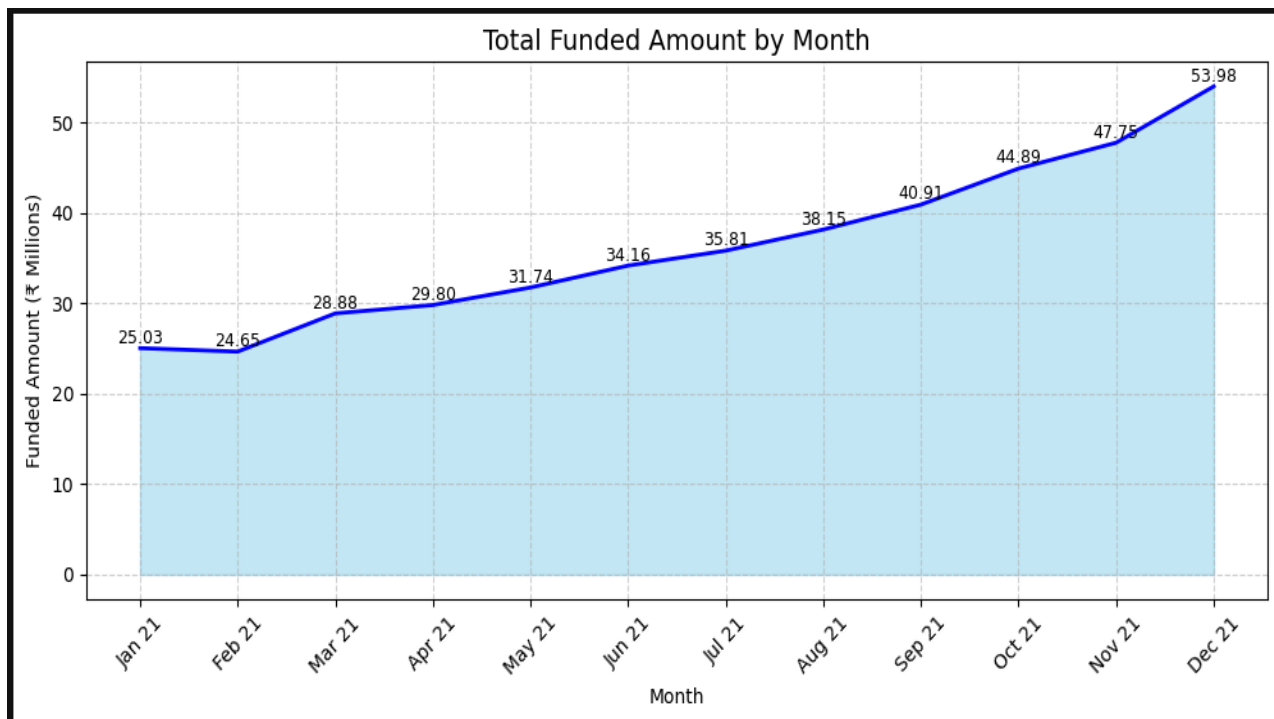
print("Bad Loan Applications:", bad_loan_applications)
print("Bad Loan Funded Amount (in Millions): {:.2f}M".format(bad_loan_funded_amount_millions))
print("Bad Loan Total Received (in Millions): {:.2f}M".format(bad_loan_received_millions))
print("Percentage of Bad Loan Applications: {:.2f}%".format(bad_loan_percentage))
```

Bad Loan Applications: 5333
Bad Loan Funded Amount (in Millions): \$65.53M
Bad Loan Total Received (in Millions): \$37.28M
Percentage of Bad Loan Applications: 13.82%

Monthly Trends by Issue Date for Total Funded Amount

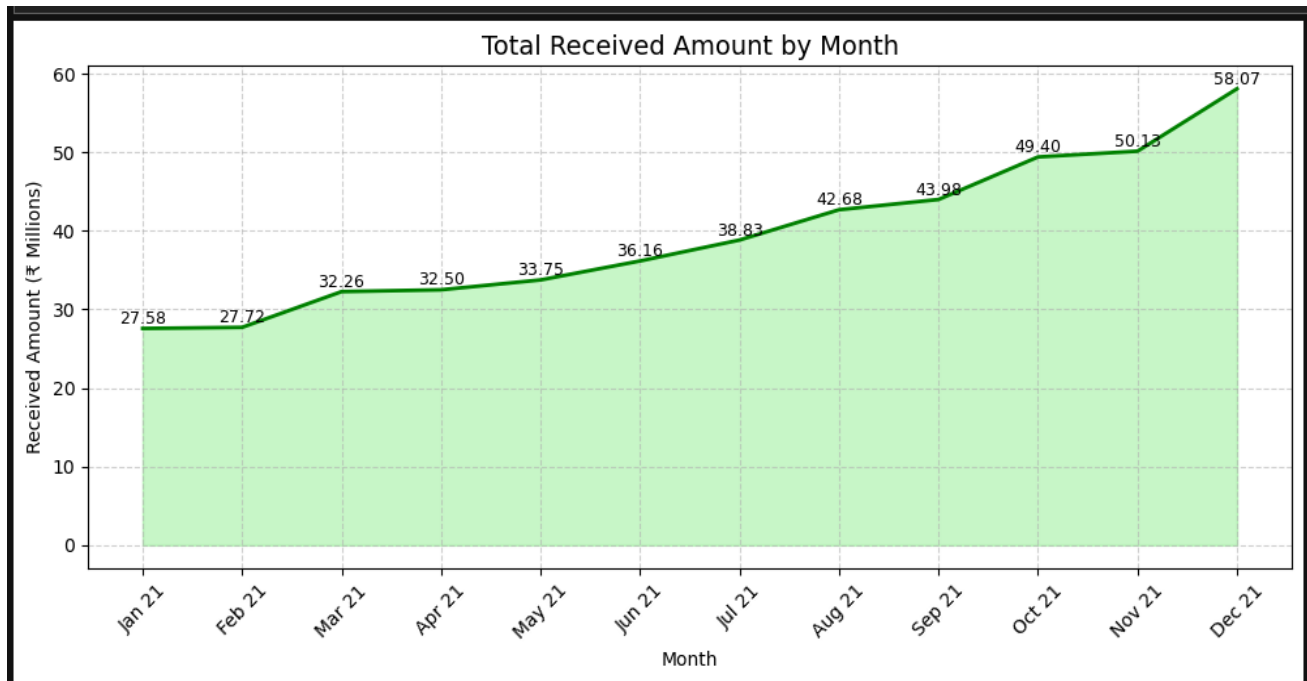
Monthly Trends by Issue Date for Total Funded Amount

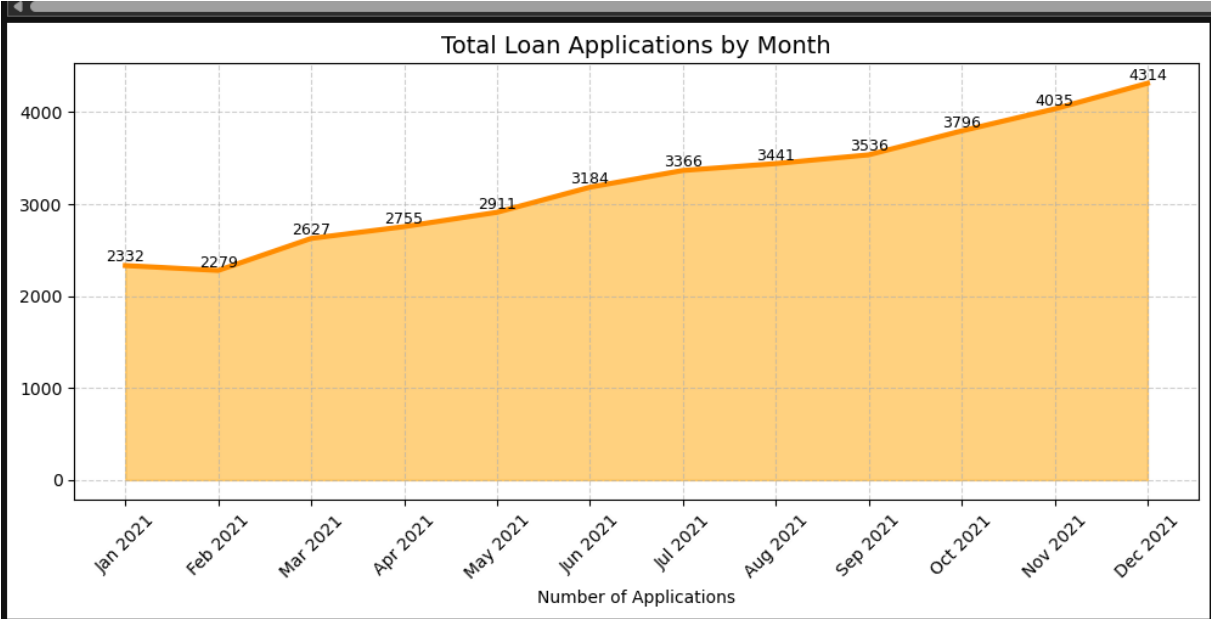
```
[45]: monthly_funded = (  
    df.sort_values('issue_date')  
    .assign(month_name=lambda x: x['issue_date'].dt.strftime('%b %y'))  
    .groupby('month_name', sort=False)['loan_amount']  
    .sum()  
    .div(1000000)  
    .reset_index(name='loan_amount_millions')  
)  
  
plt.figure(figsize=(10, 5))  
plt.fill_between(monthly_funded['month_name'], monthly_funded['loan_amount_millions'], color='skyblue', alpha=0.5)  
plt.plot(monthly_funded['month_name'], monthly_funded['loan_amount_millions'], color='blue', linewidth=2)  
  
for i, row in monthly_funded.iterrows():  
    plt.text(i, row['loan_amount_millions'] + 0.1, f"{row['loan_amount_millions']:.2f}",  
            ha='center', va='bottom', fontsize=9, rotation=0, color='black')  
  
plt.title('Total Funded Amount by Month', fontsize=14)  
plt.xlabel('Month')  
plt.ylabel('Funded Amount (₹ Millions)')  
plt.xticks(ticks=range(len(monthly_funded)), labels=monthly_funded['month_name'], rotation=45)  
plt.grid(True, linestyle='--', alpha=0.6)  
plt.tight_layout()
```

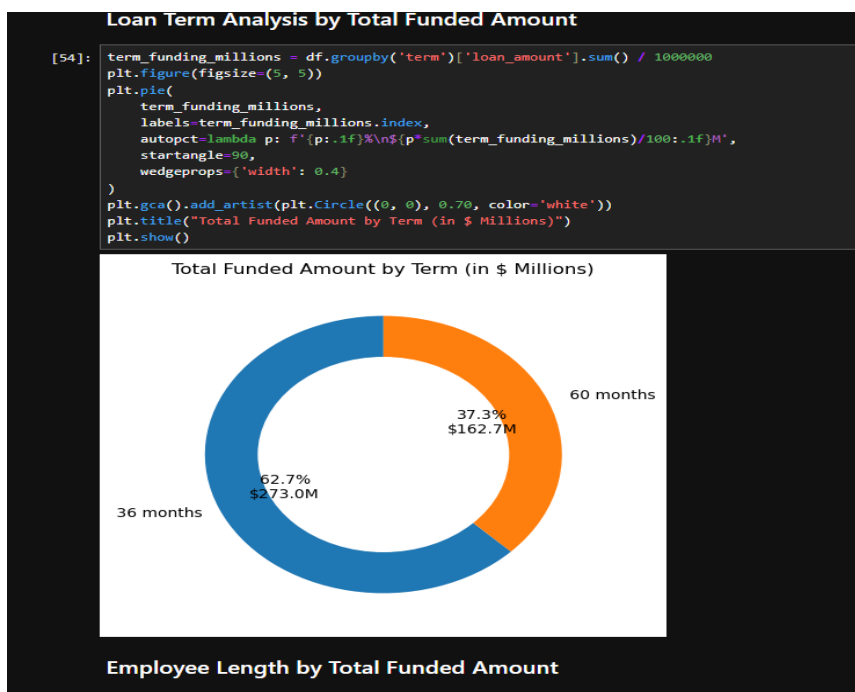
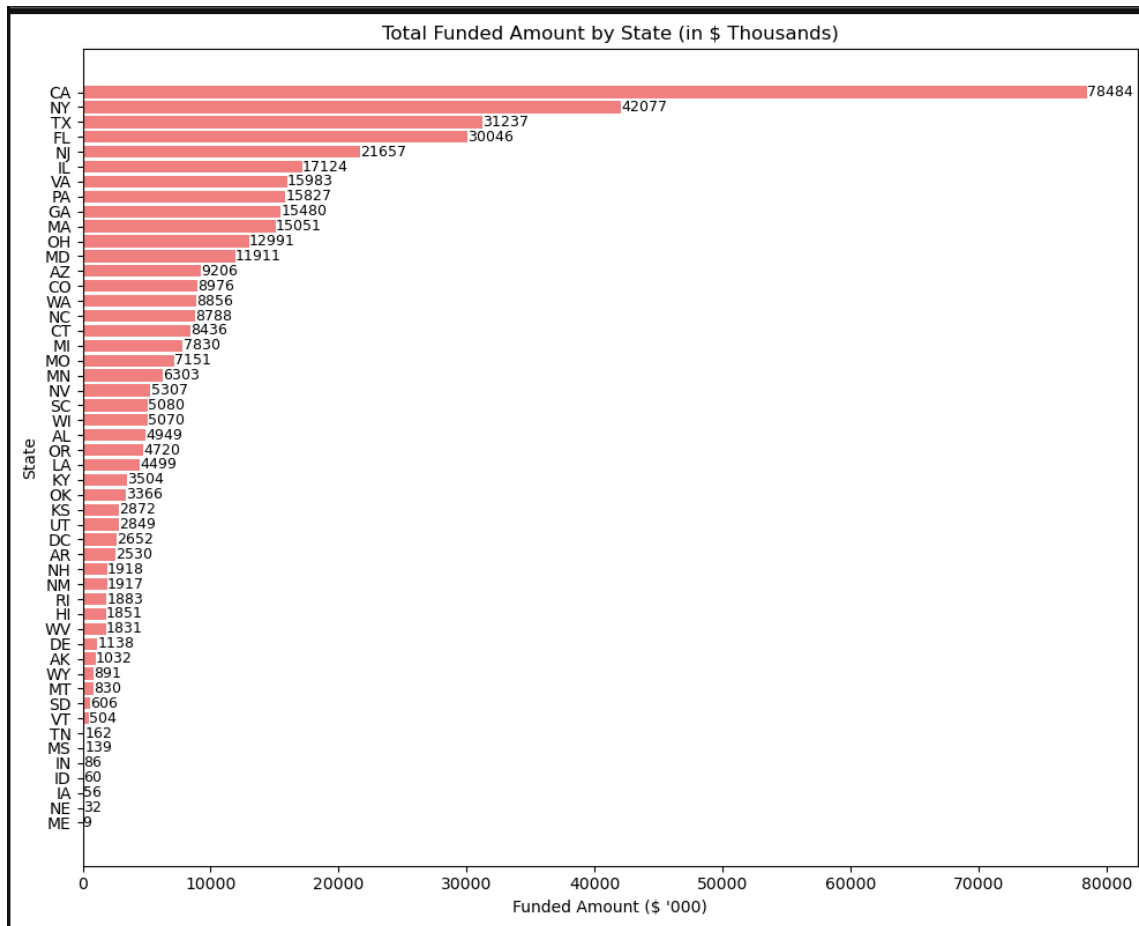


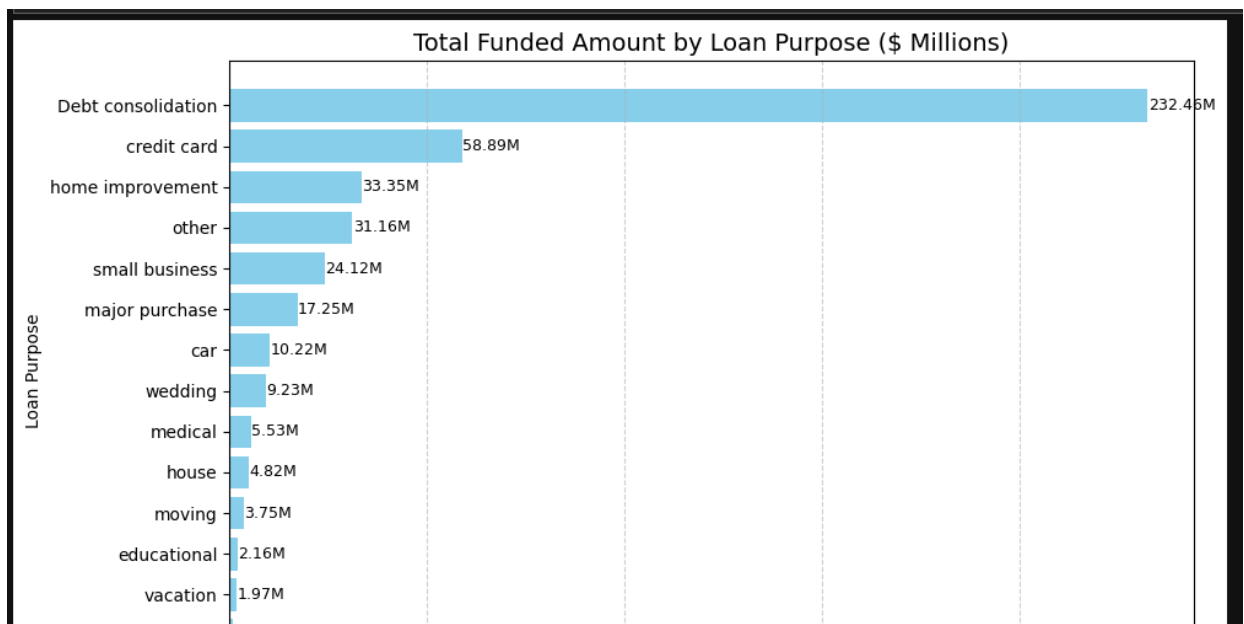
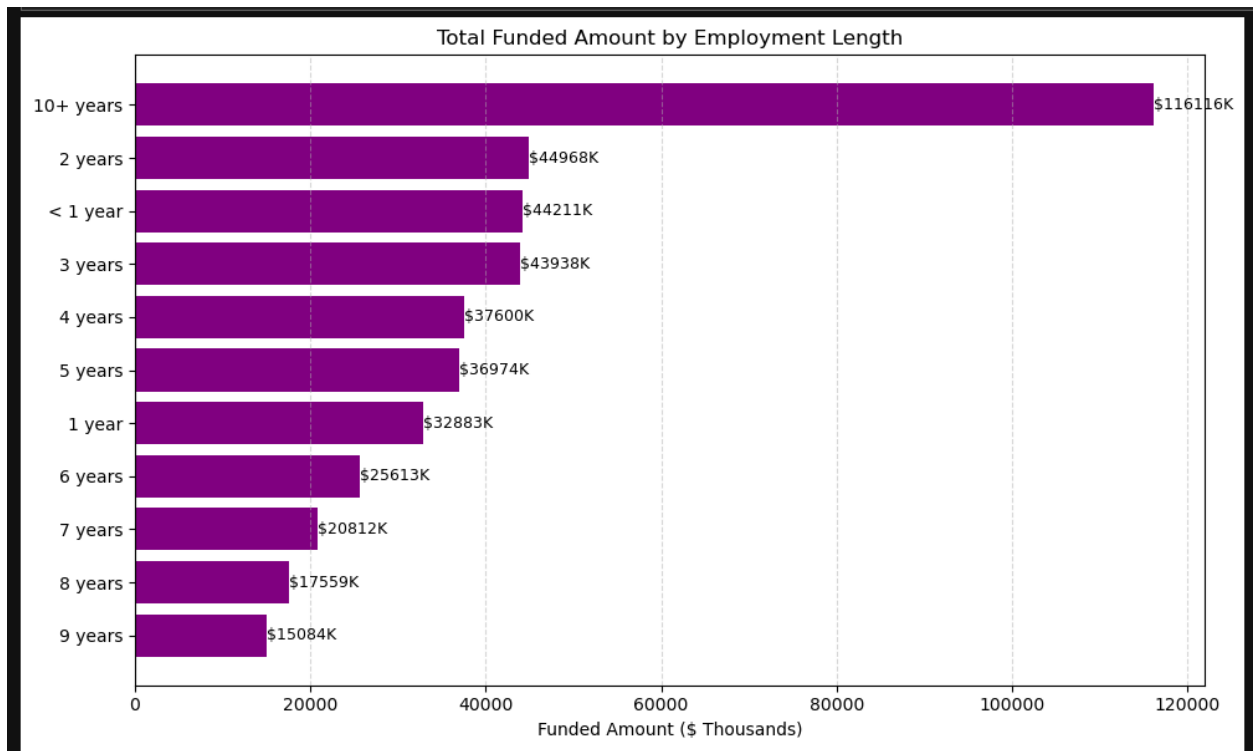
Monthly Trends by Issue Date for Total Amount Received

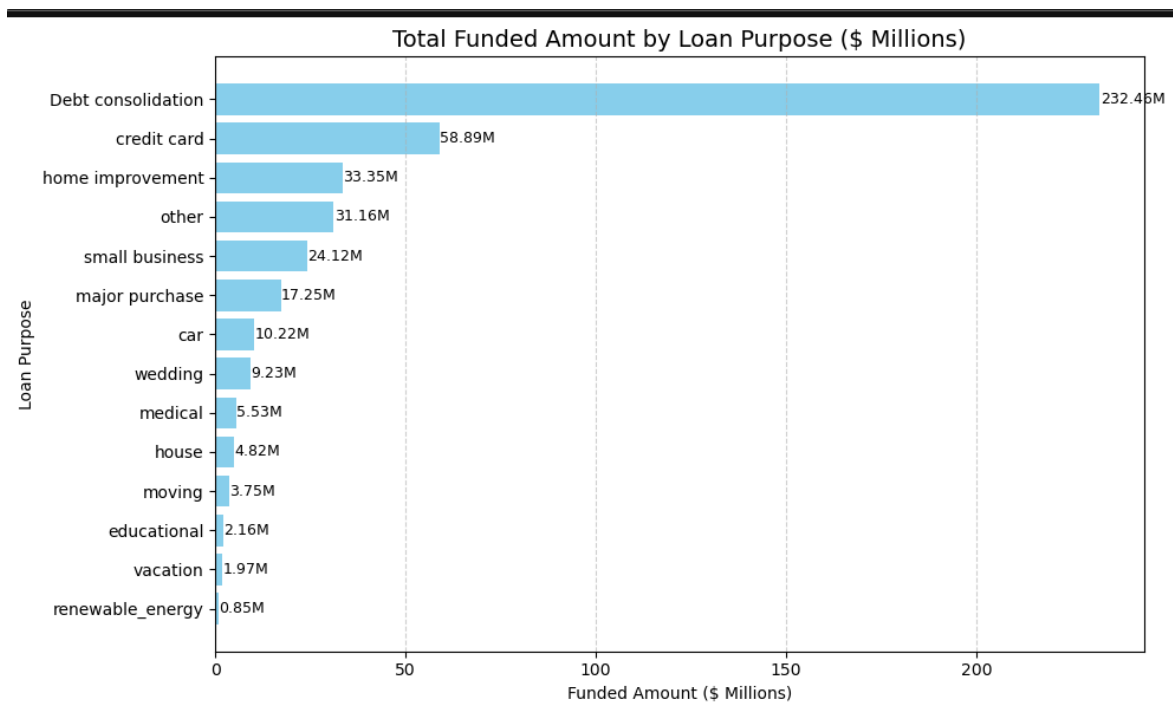
```
[46]: monthly_received = (  
    df.sort_values('issue_date')  
    .assign(month_name=lambda x: x['issue_date'].dt.strftime('%b %y'))  
    .groupby('month_name', sort=False)['total_payment']  
    .sum()  
    .div(1000000)  
    .reset_index(name='received_amount_millions')  
)  
  
plt.figure(figsize=(10, 5))  
plt.fill_between(monthly_received['month_name'], monthly_received['received_amount_millions'],  
                color='lightgreen', alpha=0.5)  
plt.plot(monthly_received['month_name'], monthly_received['received_amount_millions'],  
         color='green', linewidth=2)  
  
for i, row in monthly_received.iterrows():  
    plt.text(i, row['received_amount_millions'] + 0.1, f"{row['received_amount_millions']:.2f}",  
            ha='center', va='bottom', fontsize=9, rotation=0, color='black')  
  
plt.title('Total Received Amount by Month', fontsize=14)  
plt.xlabel('Month')  
plt.ylabel('Received Amount (₹ Millions)')  
plt.xticks(ticks=range(len(monthly_received)), labels=monthly_received['month_name'], rotation=45)  
plt.grid(True, linestyle='--', alpha=0.6)  
plt.tight_layout()  
plt.show()
```











Home ownership by Total Funded Amount

```
[57]: home_funding = df.groupby('home_ownership')['loan_amount'].sum().reset_index()
home_funding['loan_amount_millions'] = home_funding['loan_amount'] / 1_000_000

fig = px.treemap(
    home_funding,
    path=['home_ownership'],
    values='loan_amount_millions',
    color='loan_amount_millions',
    color_continuous_scale='Blues',
    title='Total Funded Amount by Home Ownership ($ Millions)'
)

fig.show()
```

Total Funded Amount by Home Ownership (\$ Millions)

