Case Study : Loan Default Prediction

Introduction

A consumer finance company which specialises in lending various types of loans to urban customers. When the company receives a loan application, the company has to make a decision for loan approval based on the applicant’s profile. Two types of risks are associated with the bank’s decision:

> - If the applicant is likely to repay the loan, then not approving the loan results in a loss of business to the company

> - If the applicant is not likely to repay the loan, i.e. he/she is likely to default, then approving the loan may lead to a financial loss for the company

Problem statement

> The company wants to understand the driving factors (or driver variables) behind loan default, i.e. the variables which are strong indicators of default. The company can utilise this knowledge for its portfolio and risk assessment.

Exploratory Data Analysis (EDA) as follows:

1. Data Sourcing

2. Data Cleaning

3. Derived Features

4.Univariate Analysis

5.Bivariate/Multivariate Analysis

1. Data Sourcing

Importing python packages and loading Data into dataframe

In [1]:

```
import os
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import datetime as dt
import seaborn as sns
import warnings
import plotly.offline as pyo
import plotly.graph_objs as go
import plotly.figure_factory as ff
import plotly.express as px
from plotly.subplots import make_subplots
warnings.filterwarnings("ignore")
pd.set_option('display.max_columns',2000)
pd.set_option('display.width',120)
pd.set_option('display.max_rows',2000)
```

In [2]:

```
df = pd.read_csv('loan.csv',parse_dates=['earliest_cr_line','issue_d','last_credit_pull_d','last_pymnt_d'])
```

2. Data Cleaning

> ### Dropping Columns with null percenatge is 100 %
>
> ### Dropping Columns with only one value in coloumn

> Droping columns which depends on customer bahavioural(the customer behavior variables are not available at the time of loan application, and thus they cannot be used as predictors for credit approval)
>
> Droping loan_amnt,funded_amnt_inv as they have higher correlation with funded_amnt
>
> Convert columns to numeric
>
> Droping or filling appropriate values for NAN

In [3]:

```
### Droping columns with Higher percentage of null values
df = df.query("loan_status != 'Current'")
df1 = pd.DataFrame(df.isnull().sum().sort_values(),columns=['Null counter'])
df1['Null_Percent'] = df1.iloc[:,0] / len(df.index) * 100
drop_column_list = df1[df1.Null_Percent > 30].index.tolist()
df.drop(labels=df1[df1.Null_Percent > 30].index,axis=1,inplace=True)

### To drop column which has only one value
df2 = pd.DataFrame(df.nunique().sort_values(),columns=['col_with_1_value'])
df.drop(labels=df2[df2.col_with_1_value < 2].index,axis=1,inplace=True)

### To drop unnecessary columns from the dataframe
df.drop(labels=['url','id','member_id','emp_title'],axis=1,inplace=True)

### To drop customer bahavioural variable
df.drop(labels=['delinq_2yrs','earliest_cr_line','inq_last_6mths','open_acc','pub_rec','revol_bal','revol_util','total_acc','tota
                'pub_rec_bankruptcies','title','total_rec_late_fee','recoveries','collection_recovery_fee','last_pymnt_d','last_p

### Droping funded_amnt,funded_amnt_inv as they have higher correlation with funded_amnt
df.drop(labels=['funded_amnt_inv','loan_amnt'],axis=1,inplace=True)

### Replace nan values of emp_length with self employeed(Assumption : Because those individual have taken loan for different purp
df =  df.fillna(value={'emp_length':'Self-employeed'})
```

In [4]:

```
##### To convert data types of the columns
df.int_rate = df.int_rate.apply(lambda x: x.rstrip('%'))
df.int_rate = df.int_rate.astype(float)
df.sub_grade = df.sub_grade.apply(lambda x: x[-1])
df.fillna(value={'pub_rec_bankruptcies': 0},inplace=True)
```

3. Derived Features

> ### Perc_loan_income: Percentage of Loan amount to the Anual Income
>
> ### Default_amount: Default amount(Credit Loss)
>
> ### Credit_loss_perc: Credit loss Percentage

In [5]:

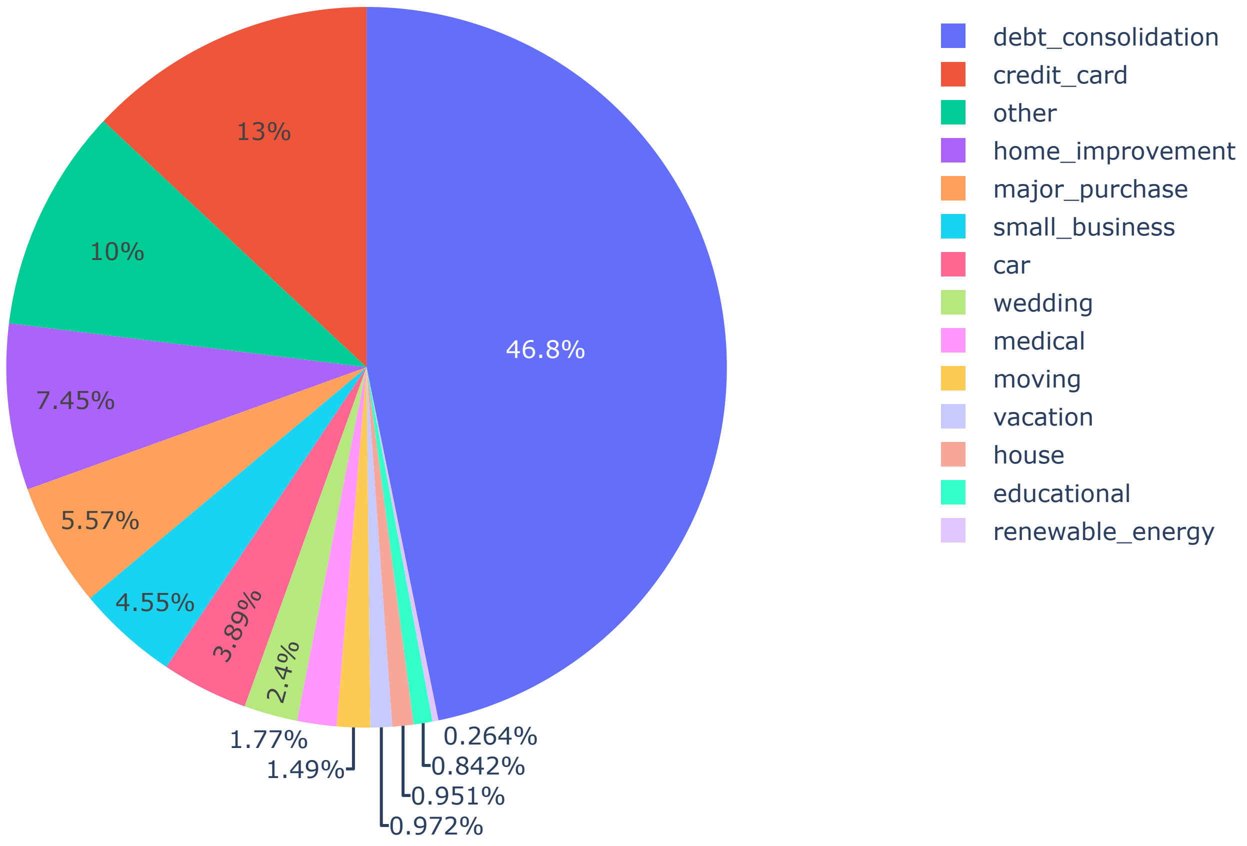
```
df['Perc_loan_income'] = round(df.funded_amnt / df.annual_inc * 100 ,0)
df['Default_amount'] = df.funded_amnt - df.total_rec_prncp
df['Credit_loss_perc'] = round((df.funded_amnt - df.total_rec_prncp) / df.funded_amnt * 100 ,0 )
df['Defaulting_stage'] = df.Credit_loss_perc.apply(lambda x: (100-x) if x > 0 else 0)
```

4.Univariate Analysis

1. Distribution of the loans across categories

In [6]:

```
plot = df['purpose'].value_counts().reset_index()
fig = px.pie(plot, values=plot.purpose, names=plot['index'],
             height=500,width=800)
fig.show()
```



Analysis: Almost half of the loans (47%) are given for the debt consolidation purpose

Debt consolidation is when a borrower takes out a new loan and then uses the loan proceeds to pay off their other debts.

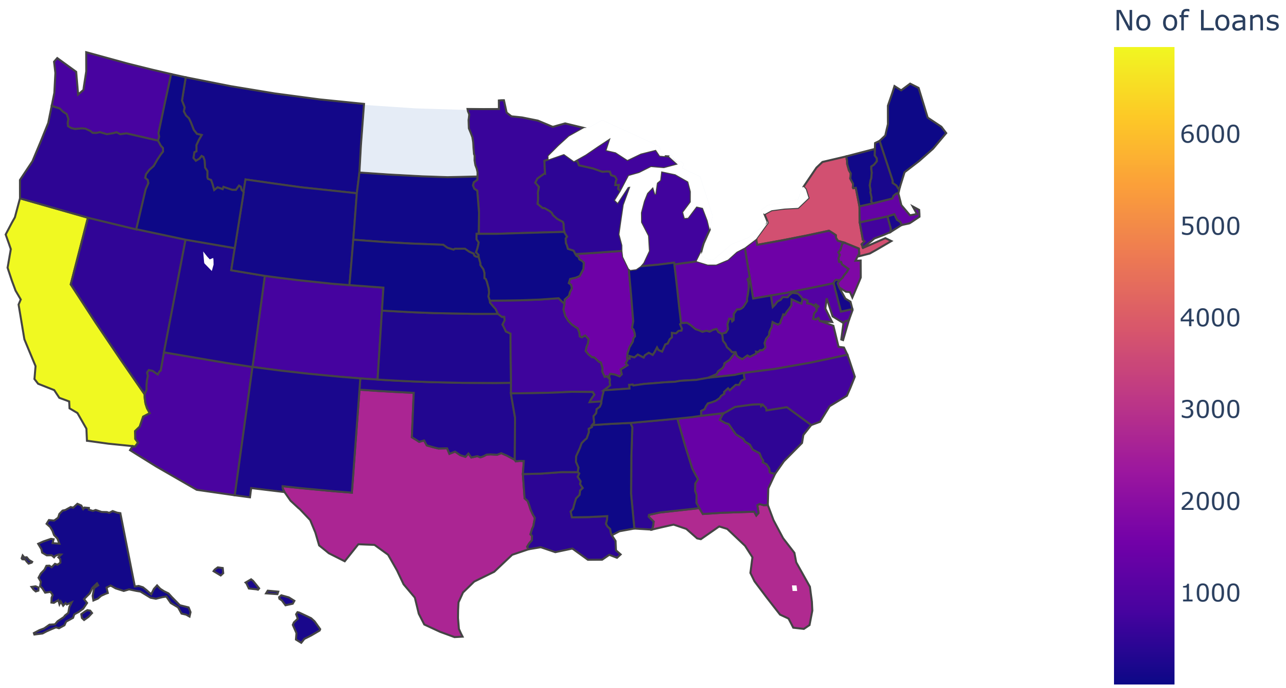
2. Distribution of the loans across the States

In [7]:

```
df_plot = pd.DataFrame(df['addr_state'].value_counts()).reset_index()
df_plot.rename(columns = {'index':'State','addr_state':'No of Loans'}, inplace = True)
```

```
fig = px.choropleth(df_plot,
                    locations=df_plot['State'],
                    locationmode='USA-states',
                    scope='usa',
                    color=df_plot['No of Loans'],
                    hover_name=df_plot['State'],
                   height=500,width=800)

fig.show()
```

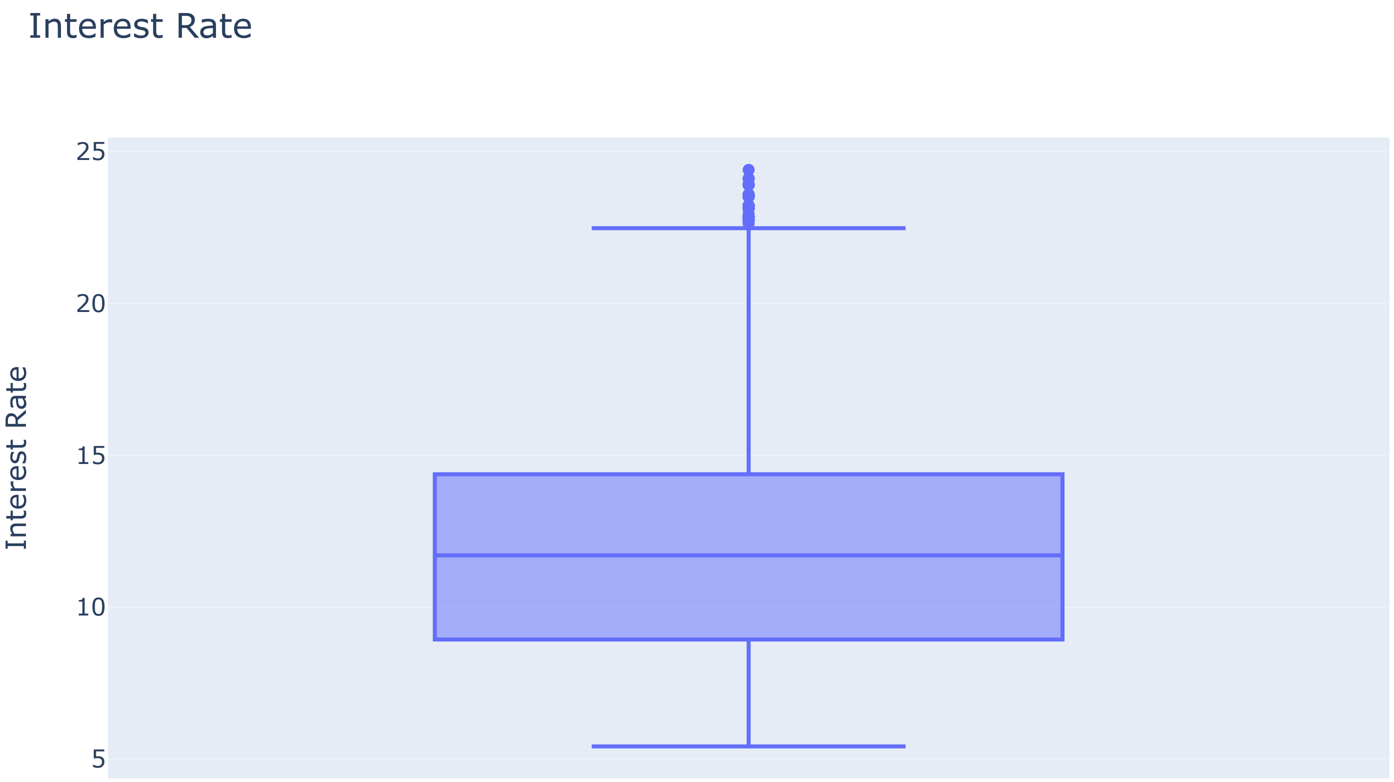


Analysis: Maximum number of loans are taken from coastal states like California, New York, Florida, and Texas.

3. Interest Rate

In [8]:

```
fig = px.box(df, y="int_rate",title="Interest Rate")
fig.update_layout(yaxis_title="Interest Rate",height=500,width=800)
fig.show()
```

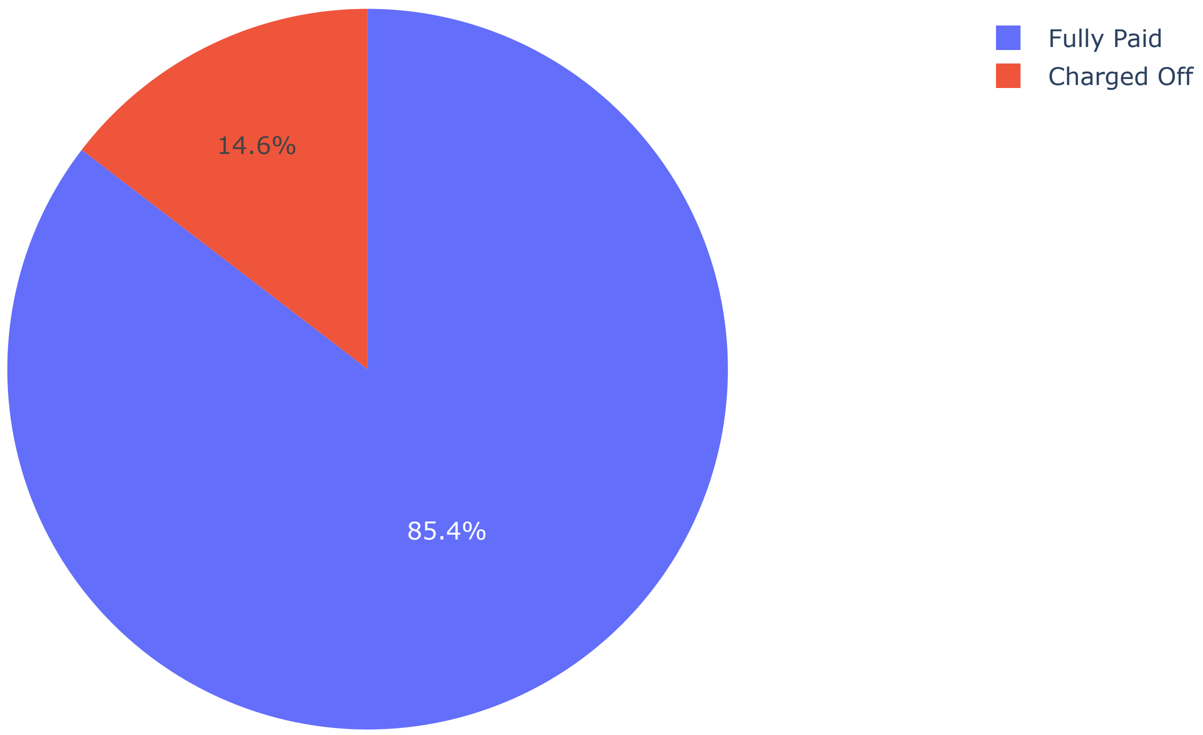


Analysis : Most of the loans Interest rate is distributed between 8.94 and 14.38. There is small portion of the risky loan which are given at higher interest rate

4. Default Rate

In [9]:

```
plot = df['loan_status'].value_counts().reset_index()
fig = px.pie(plot, values=plot.loan_status, names=plot['index'],
             height=500,width=800)
fig.show()
```



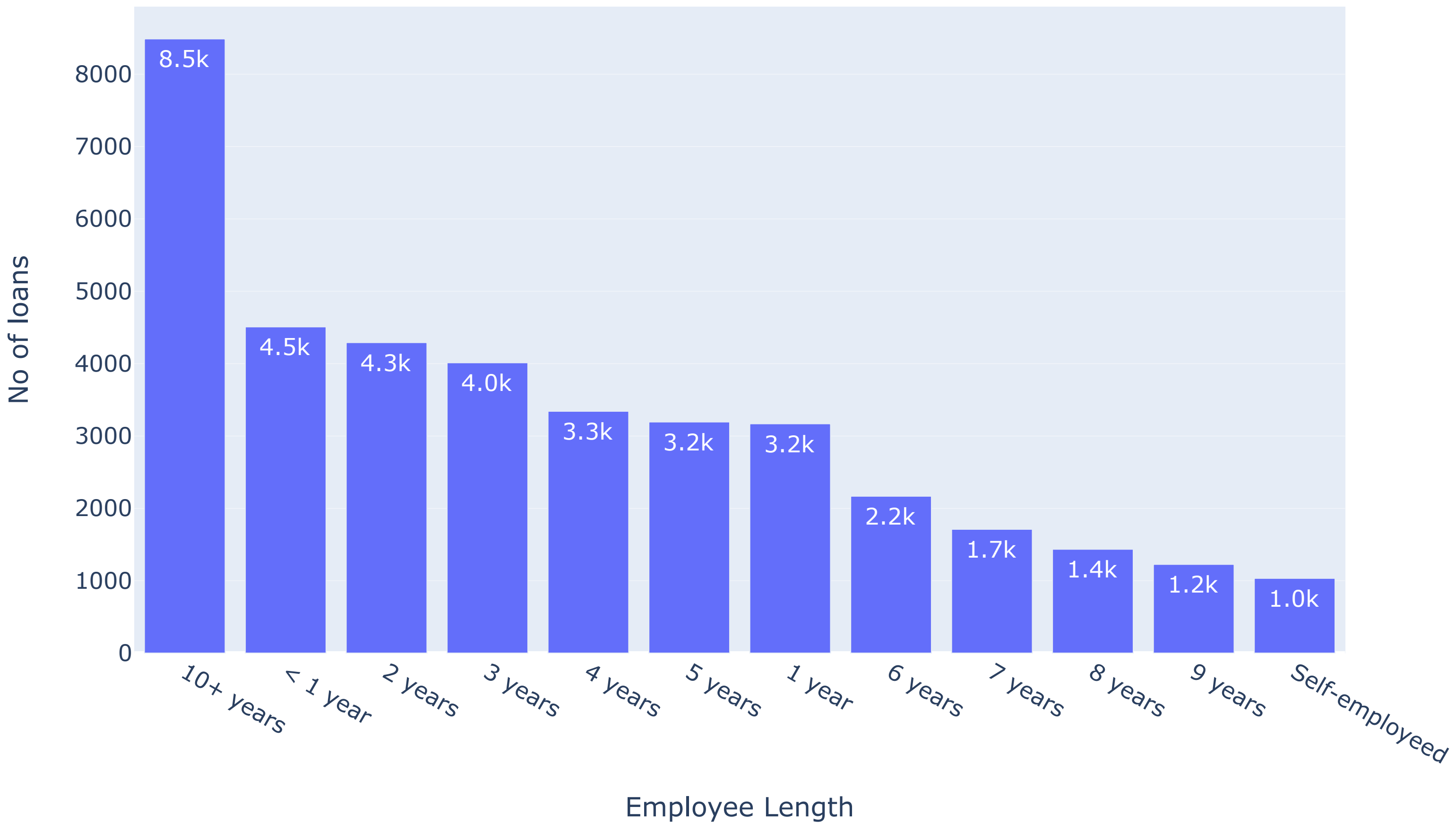
Analysis : The current default Rate of the Loans is about 15 Percent.

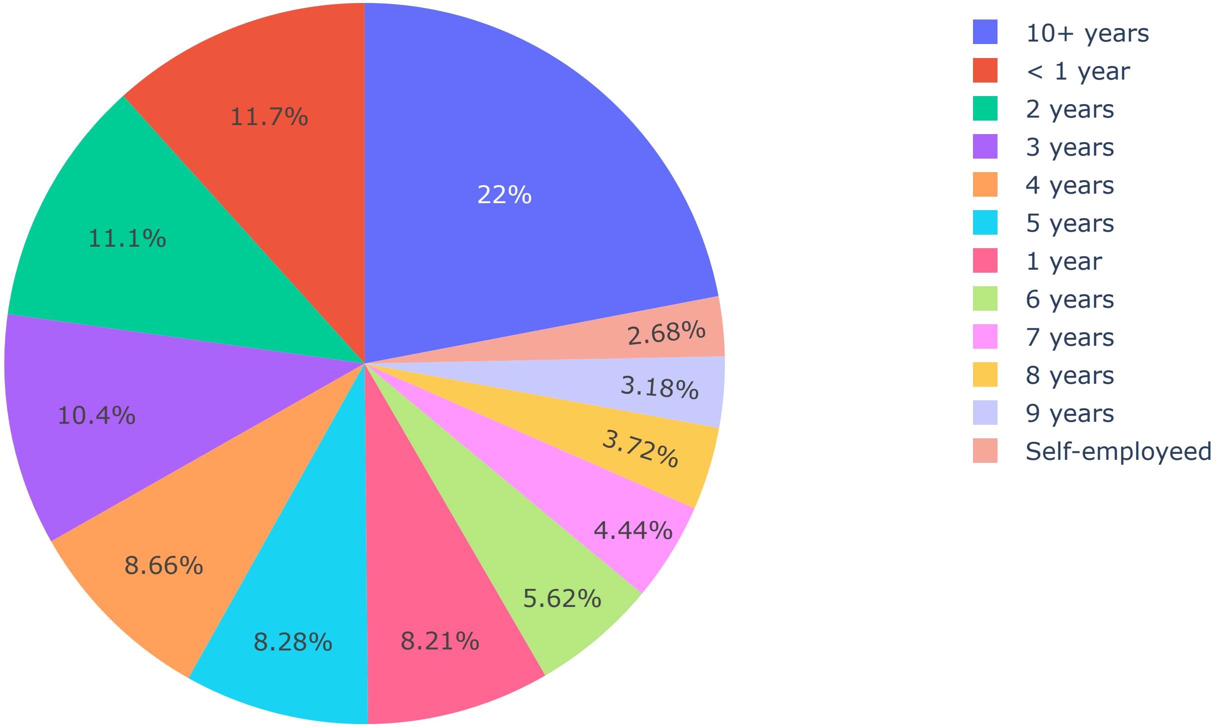
5. Distribution across the Employee Length

In [10]:

```
plot = df['emp_length'].value_counts().reset_index()
fig = px.bar(plot, x=plot['index'], y=plot.emp_length,text_auto='.2s',labels={'emp_length':'No of loans','index':'Employee Length
             height=500,width=800)
fig.show()

plot = df['emp_length'].value_counts().reset_index()
fig = px.pie(plot, values=plot.emp_length, names=plot['index'],
             height=500,width=800)
fig.show()
```


No of loans
8000
7000
6000
5000
4000
3000
2000
1000
0
8.5k
4.5k
4.3k
4.0k
3.3k
3.2k
3.2k
2.2k
1.7k
1.4k
1.2k
1.0k
10+ years
< 1 year
2 years
3 years
4 years
5 years
1 year
6 years
7 years
8 years
9 years
Self-employeed
Employee Length



Analysis : There is inverse relation between Employee length and number of loans.

```
** 10+ years contain all the loans given to the employee length more than 10 years
```

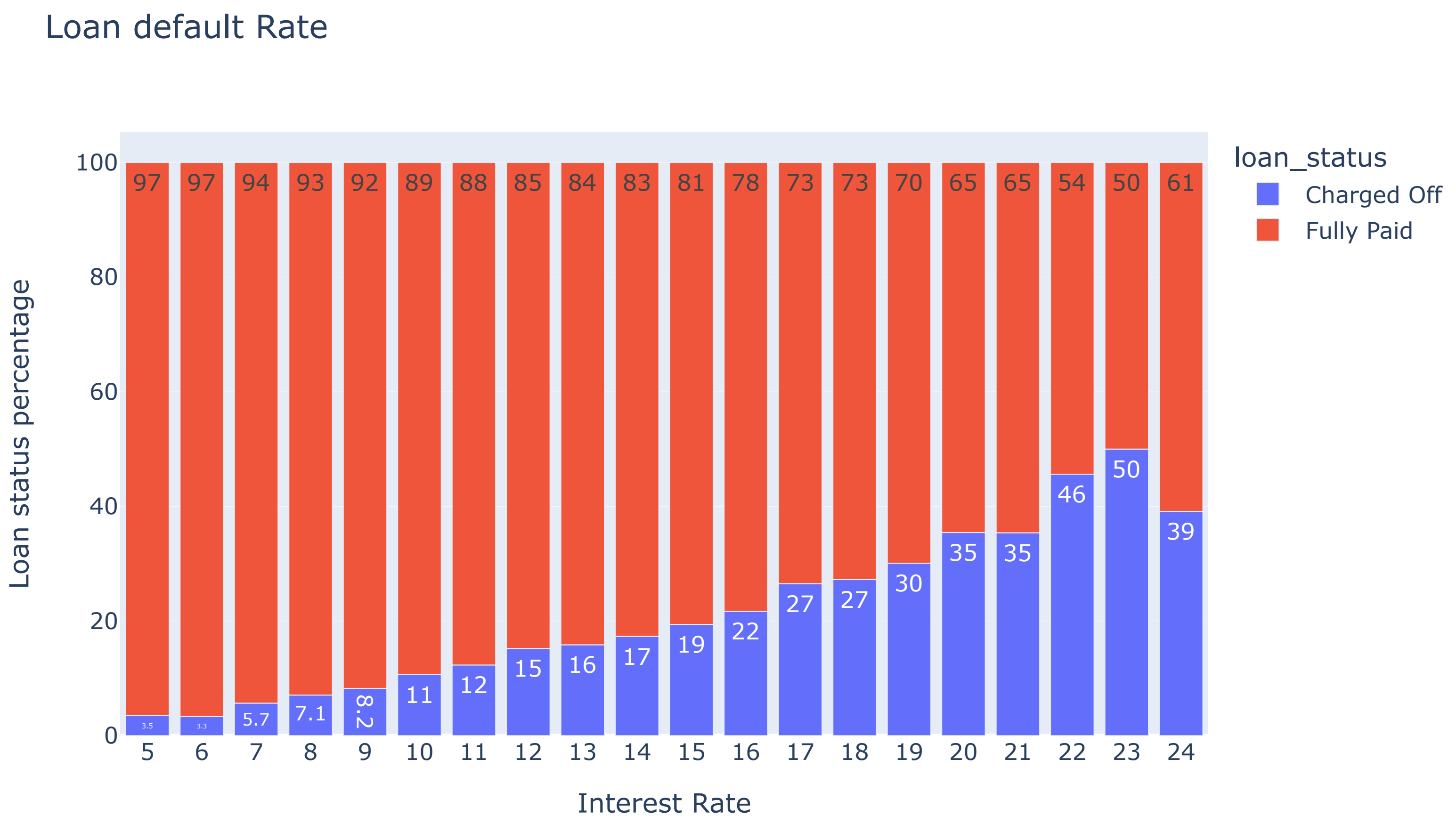
5.Bivariate/Multivariate Analysis

1. Distribution of the Default Loans against Interest rate.

In [11]:

```
df_1 = df
df_1.int_rate = df.int_rate.apply(lambda x: round(x,0))
df_plot = df_1.groupby(['int_rate','loan_status'])["loan_status"].size().groupby(level=0).apply(
    lambda x: 100 * x / x.sum()
).unstack().sort_index()

fig = px.bar(df_plot, x=df_plot.index, y=df_plot.columns,text_auto='.2s',height=500,width=800,
             labels={'value':'Loan status percentage','int_rate':"Interest Rate"},title="Loan default Rate")
fig.update_xaxes(
    tickvals=df_plot.index
)
fig.show()
```



Analysis : As Rate of Interest increase default Rate is also going up. Risky loans are given at higher interest have higher default rate. Reducing risky loans default rate can be controlled.

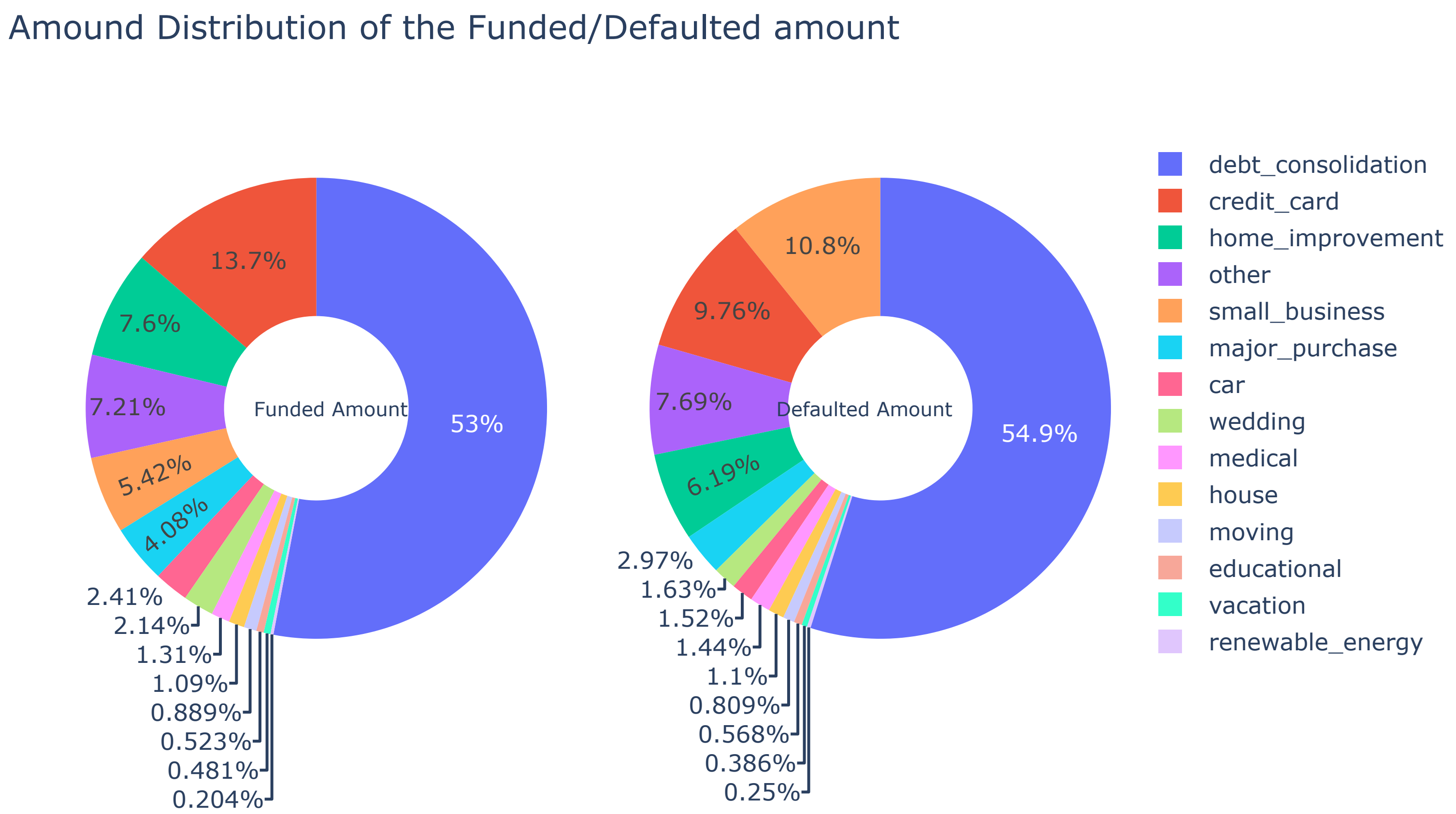
2. Distribution of the Funded and Defaulted amounts across the Purpose

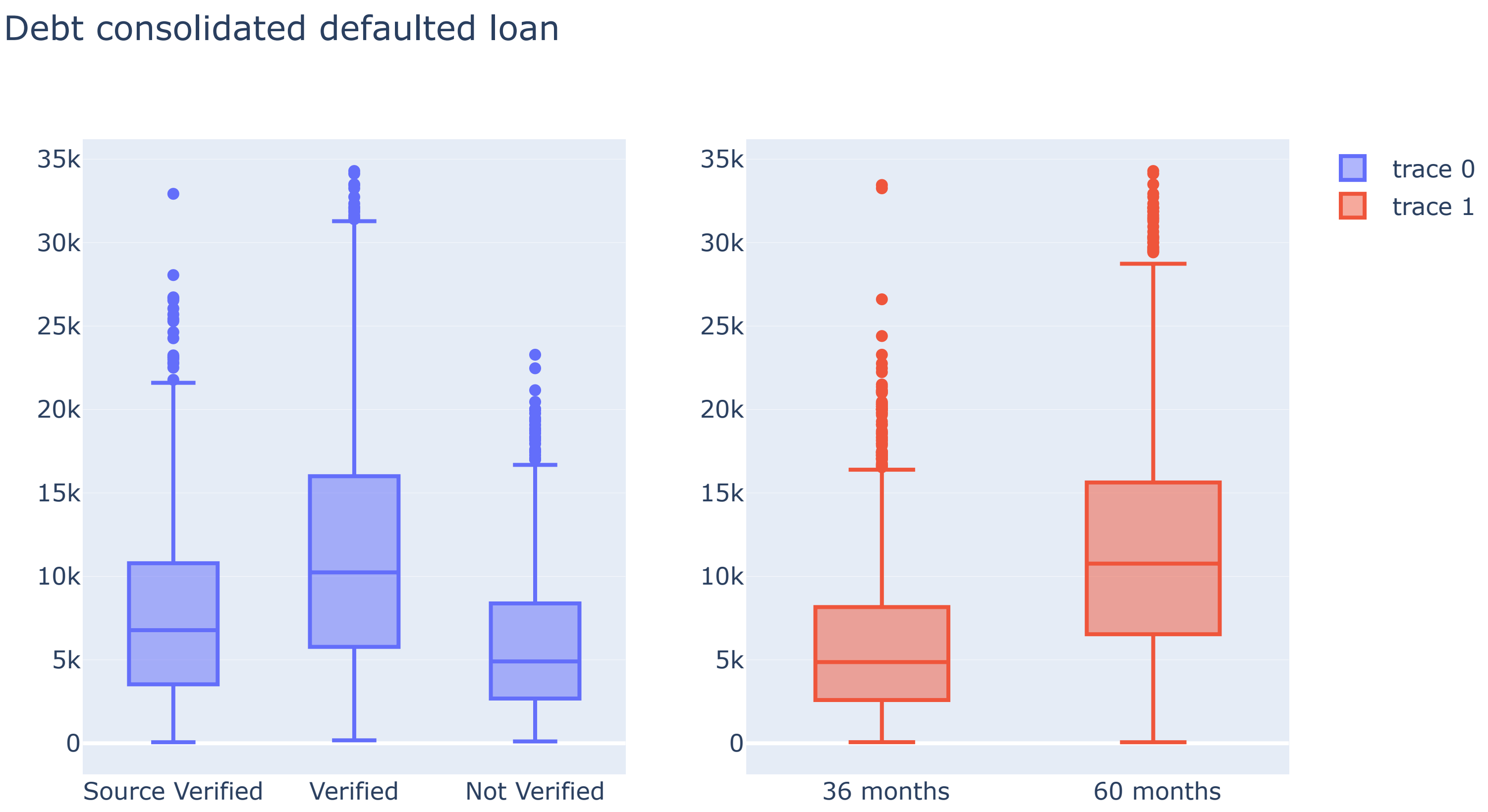
In [12]:

```
### Amound Distribution of the Funded/Defaulted amount
fig = make_subplots(rows=1, cols=2, specs=[[{'type':'domain'}, {'type':'domain'}]])
fig.add_trace(go.Pie(labels=df.purpose, values=df.funded_amnt, name="Funded Amount"),
              1, 1)
```

```
fig.add_trace(go.Pie(labels=df.purpose, values=df.Default_amount, name="Defaulted Amount"),
              1, 2)
fig.update_traces(hole=.4, hoverinfo="label+percent+name")
fig.update_layout(
    title_text="Amound Distribution of the Funded/Defaulted amount",
    height=500,width=800,
    annotations=[dict(text='Funded Amount', x=0.16, y=0.5, font_size=10, showarrow=False),
                 dict(text='Defaulted Amount', x=0.85, y=0.5, font_size=10, showarrow=False)])
fig.show()

### Debt consolidated defaulted loan
fig = make_subplots(rows=1, cols=2)
df_plot1 = df.query("purpose == 'debt_consolidation' and loan_status == 'Charged Off'")
df_plot2 = df.query("purpose == 'debt_consolidation' and loan_status == 'Charged Off'")
fig.add_trace(
    go.Box(y=df_plot1.Default_amount,x=df_plot1.verification_status),
    row=1, col=1
)
fig.add_trace(
    go.Box(x=df_plot2.term, y=df_plot2.Default_amount),
    row=1, col=2
)
fig.update_layout(title_text="Debt consolidated defaulted loan",height=500,width=800)
fig.show()
```


Amound Distribution of the Funded/Defaulted amount
Funded Amount
53%
13.7%
7.6%
7.21%
5.42%
4.08%
2.41%
2.14%
1.31%
1.09%
0.889%
0.523%
0.481%
0.204%
Defaulted Amount
54.9%
10.8%
9.76%
7.69%
6.19%
2.97%
1.63%
1.52%
1.44%
1.1%
0.809%
0.568%
0.386%
0.25%
debt_consolidation
credit_card
home_improvement
other
small_business
major_purchase
car
wedding
medical
house
moving
educational
vacation
renewable_energy



Analysis :

> A. There are debt consolidated loans above 10,500 that are defaulting after bank verification compared to the Source verified. There higher scope for improvement for bank verification so that credit loss can be minimized.

> B. Loans taken for longer duration are defaulting with higher amount compared to the short term loan. Credit loss can be minimized by giving more short term loans compared to long term for the debt consolidation.

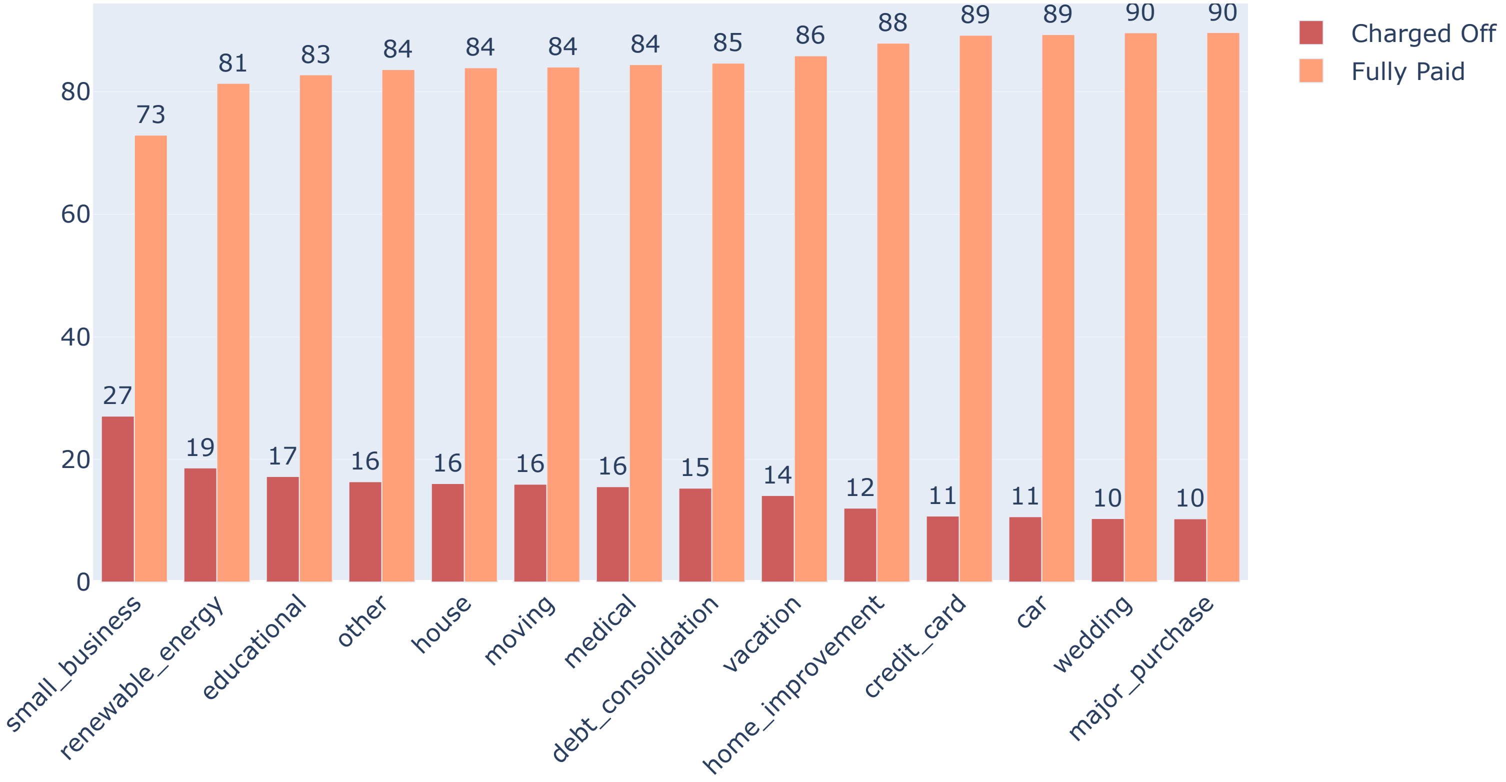
Debt consolidated loans are the major part of the total loans and considering other loans with debt consolidation for analysis may change the analysis output for consolidated loans.

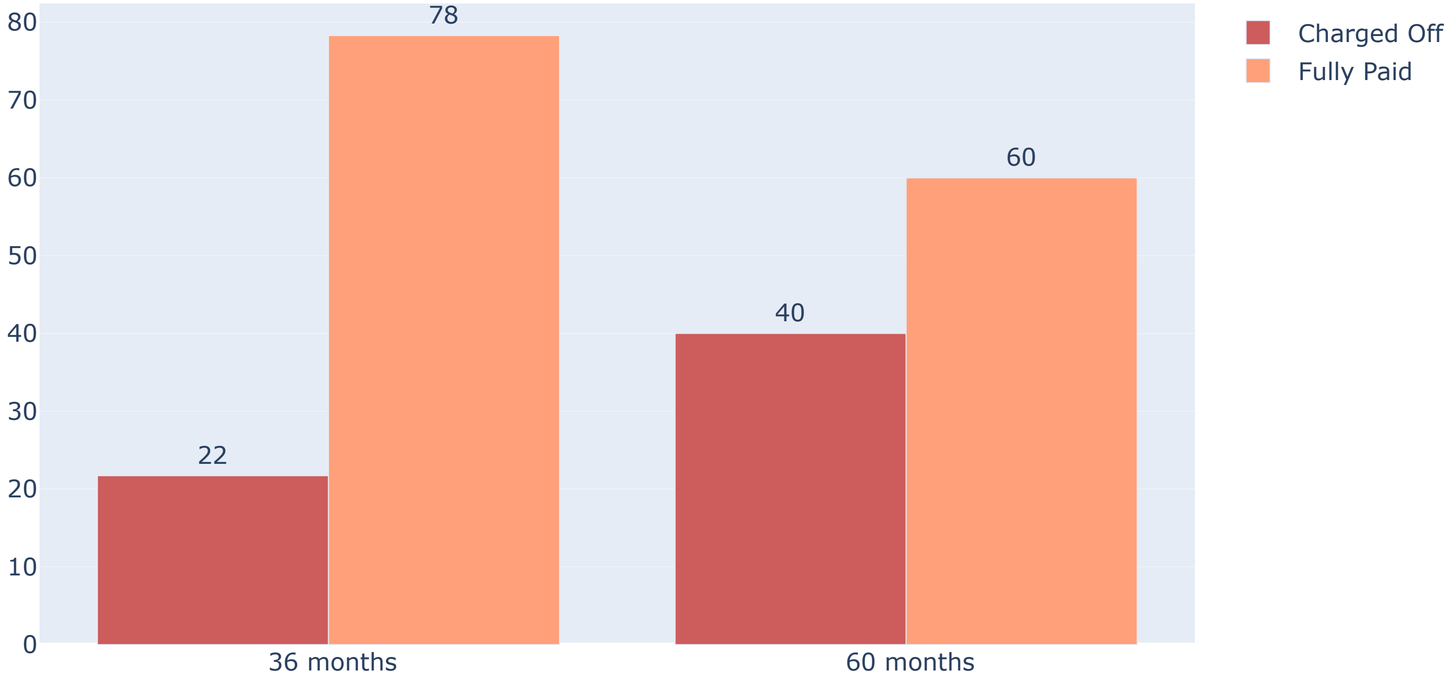
3. Default Percentage across the Purpose of the loan

In [13]:

```
df_plot = df.groupby(['purpose','loan_status'])["loan_status"].size().groupby(level=0).apply(
        lambda x: round(100 * x / x.sum(),2)
    ).unstack().sort_values(by=['Charged Off'],ascending=False)
fig = go.Figure(data=[
    go.Bar(name='Charged Off', x=df_plot.index, y=df_plot['Charged Off'],text=df_plot['Charged Off'],marker_color='indianred'),
    go.Bar(name='Fully Paid', x=df_plot.index, y=df_plot['Fully Paid'],text=df_plot['Fully Paid'],marker_color='lightsalmon')
])
fig.update_traces(texttemplate='%{text:.2s}', textposition='outside')
fig.update_layout(uniformtext_minsize=8, uniformtext_mode='hide',xaxis_tickangle=-45,height=500,width=800)
fig.update_layout(barmode='group')
fig.show()

df_2 = df[df.purpose == 'small_business']
df_plot1 = df_2.groupby(['term','loan_status'])["loan_status"].size().groupby(level=0).apply(
    lambda x: 100 * x / x.sum()
).unstack().sort_index()
df_plot
fig = go.Figure(data=[
    go.Bar(name='Charged Off', x=df_plot1.index, y=df_plot1['Charged Off'],text=df_plot1['Charged Off'],marker_color='indianred')
    go.Bar(name='Fully Paid', x=df_plot1.index, y=df_plot1['Fully Paid'],text=df_plot1['Fully Paid'],marker_color='lightsalmon')
])
fig.update_traces(texttemplate='%{text:.2s}', textposition='outside')
fig.update_layout(uniformtext_minsize=8, uniformtext_mode='hide',xaxis_tickangle=0,height=500,width=800)
fig.update_layout(barmode='group')
fig.show()
```


Charged Off
Fully Paid
0
20
40
60
80
27
73
19
81
17
83
16
84
16
84
16
84
16
84
15
85
14
86
12
88
11
89
11
89
10
90
10
90
small_business
renewable_energy
educational
other
house
moving
medical
debt_consolidation
vacation
home_improvement
credit_card
car
wedding
major_purchase



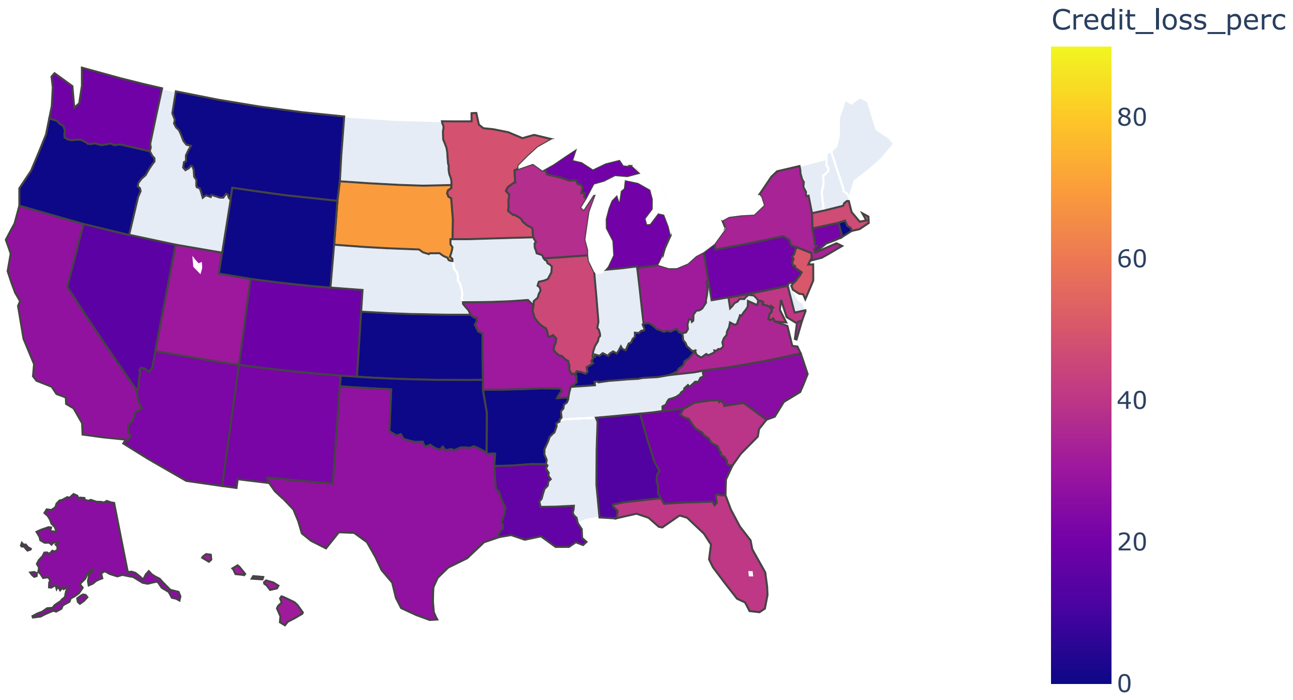
Analysis :

> Default Rate of the small buisness is 27% which higher compared to other purpose loans. Reducing long-term buisness loans(5 year) default rate can be reduced as 4 out of 10 long term loans are getting defaulted.
>
> Loans for weddings, credit cards, and cars have lower default rates. Giving more of this kind of loan default rate can be reduced.

4. Credit loss of the Risky loans(Loans with higher Interest Rate)

In [14]:

```
df_1 = df
df_1 = df_1[df_1.int_rate > 20]
df_plot = df_1.groupby('addr_state').aggregate({'Credit_loss_perc':'mean'}).reset_index()
fig = px.choropleth(df_plot,
                locations=df_plot['addr_state'],
                locationmode='USA-states',
                scope='usa',height=500,width=800,
                range_color=[0,90],
                color=df_plot['Credit_loss_perc'],
                hover_name=df_plot['addr_state'])
plt.figure(figsize=(8,6))
fig.show()
```



```
<Figure size 576x432 with 0 Axes>
```

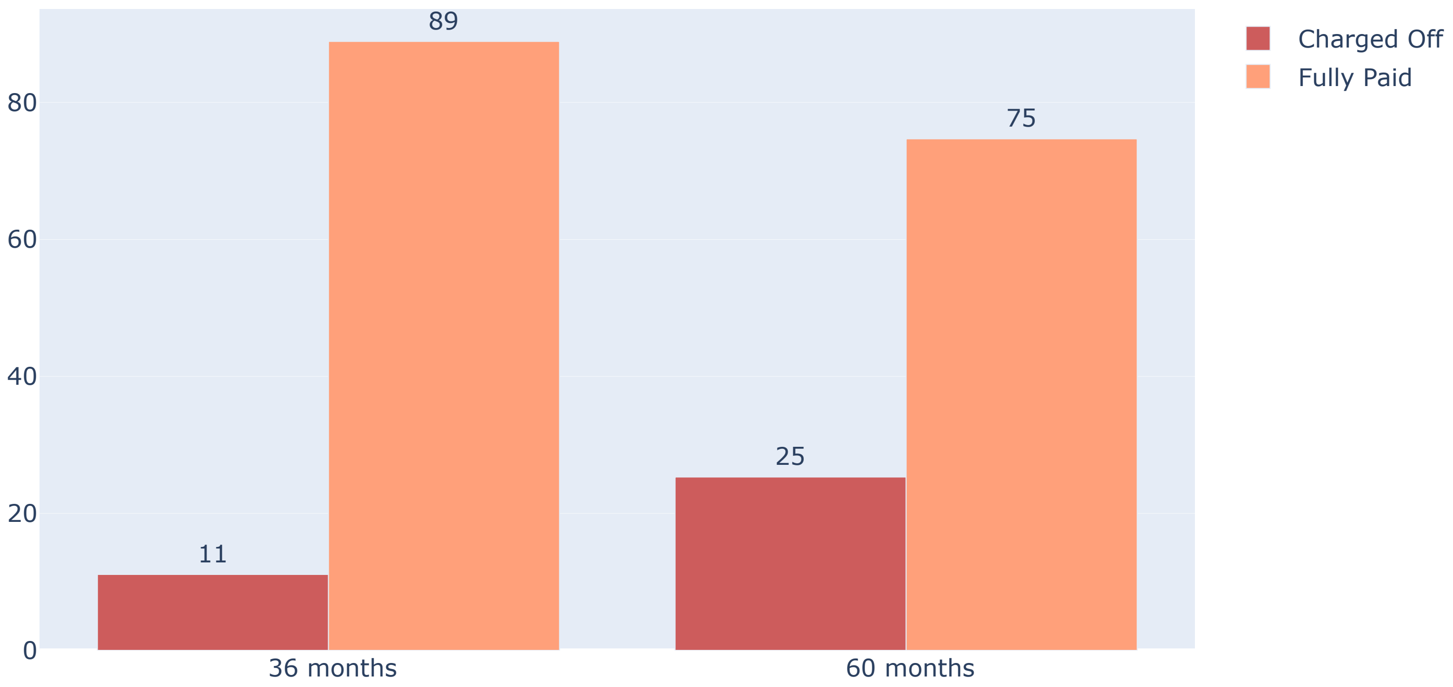
Analysis : Major credit loss is happening in eastern part compared to the western part for risky loans. It can be minimized by changing the proportion towards eastern states like California.

5. Default rate across the loan term.

In [15]:

```
df_plot = df.groupby(['term','loan_status'])["loan_status"].size().groupby(level=0).apply(
    lambda x: 100 * x / x.sum()
).unstack().sort_index()
```

```python
df_plot
fig = go.Figure(data=[
    go.Bar(name='Charged Off', x=df_plot.index, y=df_plot['Charged Off'],text=df_plot['Charged Off'],marker_color='indianred'),
    go.Bar(name='Fully Paid', x=df_plot.index, y=df_plot['Fully Paid'],text=df_plot['Fully Paid'],marker_color='lightsalmon')
])
fig.update_traces(texttemplate='%{text:.2s}', textposition='outside')
fig.update_layout(uniformtext_minsize=8, uniformtext_mode='hide',xaxis_tickangle=0,height=500,width=800)
fig.update_layout(barmode='group')
fig.show()
```



Analysis : Default rate of the long-term(5 years) is 25 Percent which is much higher than the short-term(3 years) loans. Credit loss/Loan default rate can be reduced by increasing portion of the short-term loan.

Important features which are affecting the default rate :

> Term: Loans with a longer term are more likely to default.
>
> Interest Rate: Loans with a higher interest rate have a higher default rate.
>
> Region: Eastern part of the USA has a higher percentage of credit loss for the risky loans.
>
> Purpose: Small business loans have a higher rate of default.