Importing libraries

In [1]:
```
#importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import matplotlib
import plotly
import plotly.graph_objs as go
```

Reading CSV data

In [2]:
```
#importing csv data
df = pd.read_csv('lending_club_loans_copy.csv')
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_8540\1194260924.py:2: DtypeWarning: Columns (0,25,34,36,46,49,55) h
ave mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv('lending_club_loans_copy.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	...	num_tl_90g_dpd_
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	...	
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.83	C	C4	...	
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	84.33	C	C5	...	
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	339.31	C	C1	...	
4	1075358	1311748.0	3000.0	3000.0	3000.0	60 months	12.69%	67.79	B	B5	...	

5 rows × 115 columns

In [4]:
```
# Looking at the top 5 rows to understand data
pd.set_option('display.max_columns', None)

df.head(3)
```

Out[4]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_lengt
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	NaN	10+ yea
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.83	C	C4	Ryder	< 1 ye
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	84.33	C	C5	NaN	10+ yea

In [5]:
```
# Size of the dataset
df.shape
```

Out[5]:
```
(42539, 115)
```

In [6]:
```
# Checking info of the raw dataframe
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42539 entries, 0 to 42538
Columns: 115 entries, id to total_il_high_credit_limit
dtypes: float64(85), object(30)
memory usage: 37.3+ MB
```

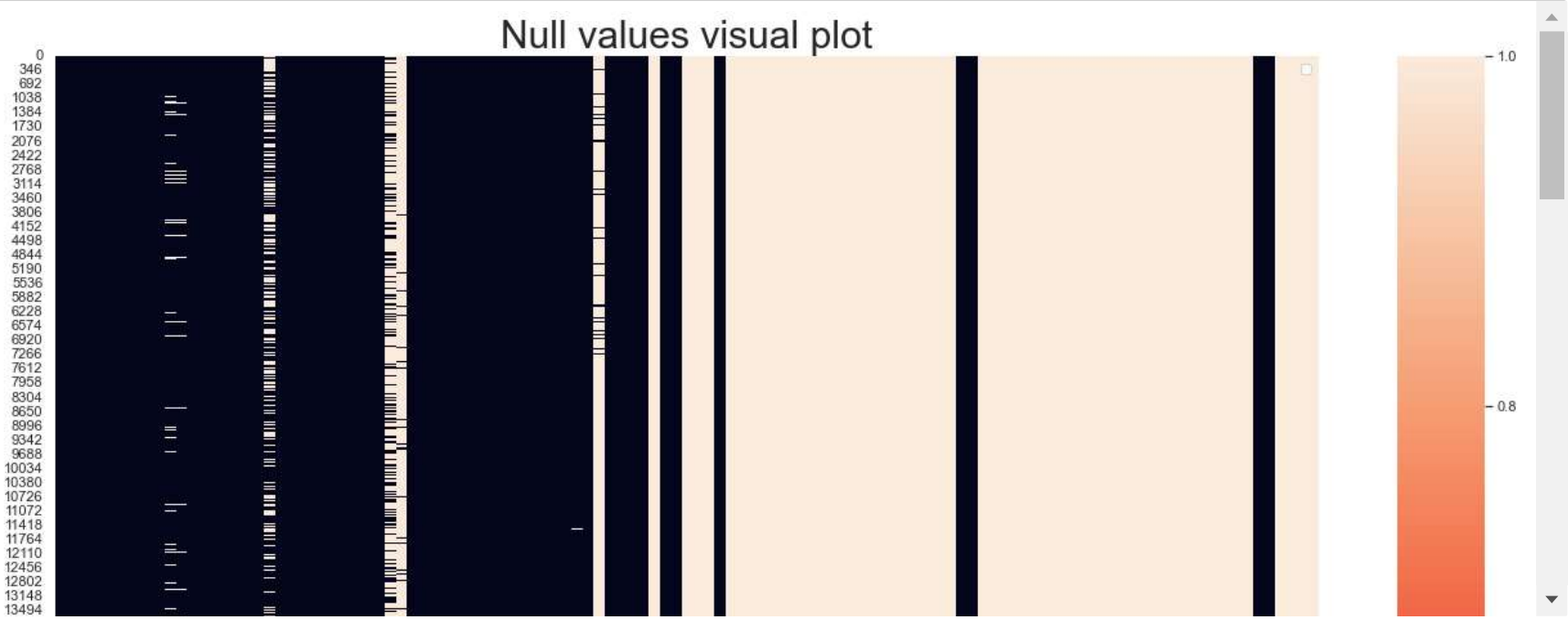
In [7]:
```
df.columns
```

Out[7]:
```
Index(['id', 'member_id', 'loan_amnt', 'funded_amnt', 'funded_amnt_inv',
       'term', 'int_rate', 'installment', 'grade', 'sub_grade',
       ...
       'num_tl_90g_dpd_24m', 'num_tl_op_past_12m', 'pct_tl_nvr_dlq',
       'percent_bc_gt_75', 'pub_rec_bankruptcies', 'tax_liens',
       'tot_hi_cred_lim', 'total_bal_ex_mort', 'total_bc_limit',
       'total_il_high_credit_limit'],
      dtype='object', length=115)
```

In [8]:
```
# Total percentage of null values in the data
(df.isnull().sum().sum())/(df.shape[0]*df.shape[1])
```

Out[8]:
```
0.49501684898870296
```

In [9]:
```
# Checking for null values using a heat map as a visualizing tool
sns.set(rc={'figure.figsize':(22,25)})
sns.set_style('whitegrid')
sns.heatmap(df.isnull())
plt.title('Null values visual plot',fontdict={'fontsize': 30})
plt.legend(df.isnull())
plt.show()
```



In [10]:
```
# plt.figure(figsize=(20,20))
# sns.heatmap(data = df,
#             annot = True,
#             vmin = 0.0,
#             vmax = 1.0,
#             cmap = 'PuBuGn')
```

In [11]:
```
# fig = px.imshow(df, text_auto=True, aspect="auto")
# fig.show()

# cmap = matplotlib.cm.jet
# cmap.set_bad('white',1.)
# px.imshow(df, cmap=cmap)
```

In [12]:
```
df.columns
```

Out[12]:
```
Index(['id', 'member_id', 'loan_amnt', 'funded_amnt', 'funded_amnt_inv',
       'term', 'int_rate', 'installment', 'grade', 'sub_grade',
       ...
       'num_tl_90g_dpd_24m', 'num_tl_op_past_12m', 'pct_tl_nvr_dlq',
       'percent_bc_gt_75', 'pub_rec_bankruptcies', 'tax_liens',
       'tot_hi_cred_lim', 'total_bal_ex_mort', 'total_bc_limit',
       'total_il_high_credit_limit'],
      dtype='object', length=115)
```

In [13]:
```
# Considering only those columns which have null values less than 40% in that particular column
df = pd.read_csv('lending_club_loans_copy.csv')
df = df[df.columns[((df.isnull().sum())/42539) < 0.4]]
df.shape
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_8540\1422710136.py:2: DtypeWarning: Columns (0,25,34,36,46,49,55) h
ave mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv('lending_club_loans_copy.csv')
```

Out[13]:
```
(42539, 58)
```

In [14]:
```
# Creating a dataframe to display percentage of null values in each column
new_df = pd.DataFrame()
new_df['Percentage of null values']=['10% or less','20% or less','30% or less','40% or less','50% or less','60%
                                     '80% or less','90% or less','100% or less']

ten_percent = len(df.columns[((df.isnull().sum())/len(df)) < 0.1])
twenty_percent = len(df.columns[((df.isnull().sum())/len(df)) < 0.2])
thirty_percent = len(df.columns[((df.isnull().sum())/len(df)) < 0.3])
fourty_percent = len(df.columns[((df.isnull().sum())/len(df)) < 0.4])
fifty_percent = len(df.columns[((df.isnull().sum())/len(df)) < 0.5])
sixty_percent = len(df.columns[((df.isnull().sum())/len(df)) < 0.6])
seventy_percent = len(df.columns[((df.isnull().sum())/len(df)) < 0.7])
eighty_percent = len(df.columns[((df.isnull().sum())/len(df)) < 0.8])
ninty_percent = len(df.columns[((df.isnull().sum())/len(df)) < 0.9])
hundread_percent = len(df.columns[((df.isnull().sum())/len(df)) < 1])


new_df['No.of columns'] = [ten_percent,twenty_percent,thirty_percent,fourty_percent,fifty_percent,sixty_percent,
                           eighty_percent,ninty_percent,hundread_percent]
```

In [15]:
```
new_df
```

Out[15]:

	Percentage of null values	No.of columns
0	10% or less	57
1	20% or less	57
2	30% or less	57
3	40% or less	58
4	50% or less	58
5	60% or less	58
6	70% or less	58
7	80% or less	58
8	90% or less	58
9	100% or less	58

Updating dataframe df with columns which have less null values than 40%

In [16]:
```
# Considering only those columns which have null values less than 40% in that particular column
df = pd.read_csv('lending_club_loans_copy.csv')
df = df[df.columns[((df.isnull().sum())/len(df)) < 0.4]]
df.shape
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_8540\1162782033.py:2: DtypeWarning: Columns (0,25,34,36,46,49,55) h
ave mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv('lending_club_loans_copy.csv')
```

Out[16]:
```
(42539, 58)
```

By considering columns with less number of null values, we were able to decrease total number of columns from 115 to 58.

In [17]:
```
df.head()
```

Out[17]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_l
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	NaN	10+
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.83	C	C4	Ryder	<
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	84.33	C	C5	NaN	10+
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	339.31	C	C1	AIR RESOURCES BOARD	10+
4	1075358	1311748.0	3000.0	3000.0	3000.0	60 months	12.69%	67.79	B	B5	University Medical Group	

In [18]:
```
# Checking info of updated dataframe
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42539 entries, 0 to 42538
Data columns (total 58 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   id                          42537 non-null  object 
 1   member_id                   42535 non-null  float64
 2   loan_amnt                   42535 non-null  float64
 3   funded_amnt                 42535 non-null  float64
 4   funded_amnt_inv             42535 non-null  float64
 5   term                        42535 non-null  object 
 6   int_rate                    42535 non-null  object 
 7   installment                 42535 non-null  float64
 8   grade                       42535 non-null  object 
 9   sub_grade                   42535 non-null  object 
 10  emp_title                   39909 non-null  object 
 11  emp_length                  41423 non-null  object 
 12  home_ownership              42535 non-null  object 
 13  annual_inc                  42531 non-null  float64
 14  verification_status         42535 non-null  object 
 15  issue_d                     42535 non-null  object 
 16  loan_status                 42535 non-null  object 
 17  pymnt_plan                  42535 non-null  object 
 18  url                         42535 non-null  object 
 19  desc                        29242 non-null  object 
 20  purpose                     42535 non-null  object 
 21  title                       42522 non-null  object 
 22  zip_code                    42535 non-null  object 
 23  addr_state                  42535 non-null  object 
 24  dti                         42535 non-null  float64
 25  delinq_2yrs                 42506 non-null  object 
 26  earliest_cr_line            42506 non-null  object 
 27  fico_range_low              42535 non-null  float64
 28  fico_range_high             42535 non-null  float64
 29  inq_last_6mths              42506 non-null  float64
 30  open_acc                    42506 non-null  float64
 31  pub_rec                     42506 non-null  float64
 32  revol_bal                   42535 non-null  object 
 33  revol_util                  42445 non-null  object 
 34  total_acc                   42506 non-null  object 
```

```
 35  initial_list_status         42535 non-null  object
 36  out_prncp                   42535 non-null  float64
 37  out_prncp_inv               42535 non-null  float64
 38  total_pymnt                 42535 non-null  float64
 39  total_pymnt_inv             42535 non-null  float64
 40  total_rec_prncp             42535 non-null  float64
 41  total_rec_int               42535 non-null  float64
 42  total_rec_late_fee          42535 non-null  float64
 43  recoveries                  42535 non-null  float64
 44  collection_recovery_fee     42535 non-null  object
 45  last_pymnt_d                42452 non-null  object
 46  last_pymnt_amnt             42534 non-null  float64
 47  last_credit_pull_d          42531 non-null  object
 48  last_fico_range_high        42535 non-null  float64
 49  last_fico_range_low         42535 non-null  float64
 50  collections_12_mths_ex_med  42389 non-null  float64
 51  policy_code                 42535 non-null  object
 52  application_type            42534 non-null  object
 53  acc_now_delinq              42505 non-null  float64
 54  chargeoff_within_12_mths    42390 non-null  float64
 55  delinq_amnt                 42505 non-null  float64
 56  pub_rec_bankruptcies        41170 non-null  float64
 57  tax_liens                   42429 non-null  float64
dtypes: float64(29), object(29)
memory usage: 18.8+ MB
```

Understanding Features

In [19]:
```
# Considering 'term' feature
df.term.value_counts()
```

Out[19]:
```
 36 months    31534
 60 months    11001
Name: term, dtype: int64
```

In [20]:
```
term_data = pd.DataFrame(df.term.value_counts())
term_data
```

Out[20]:

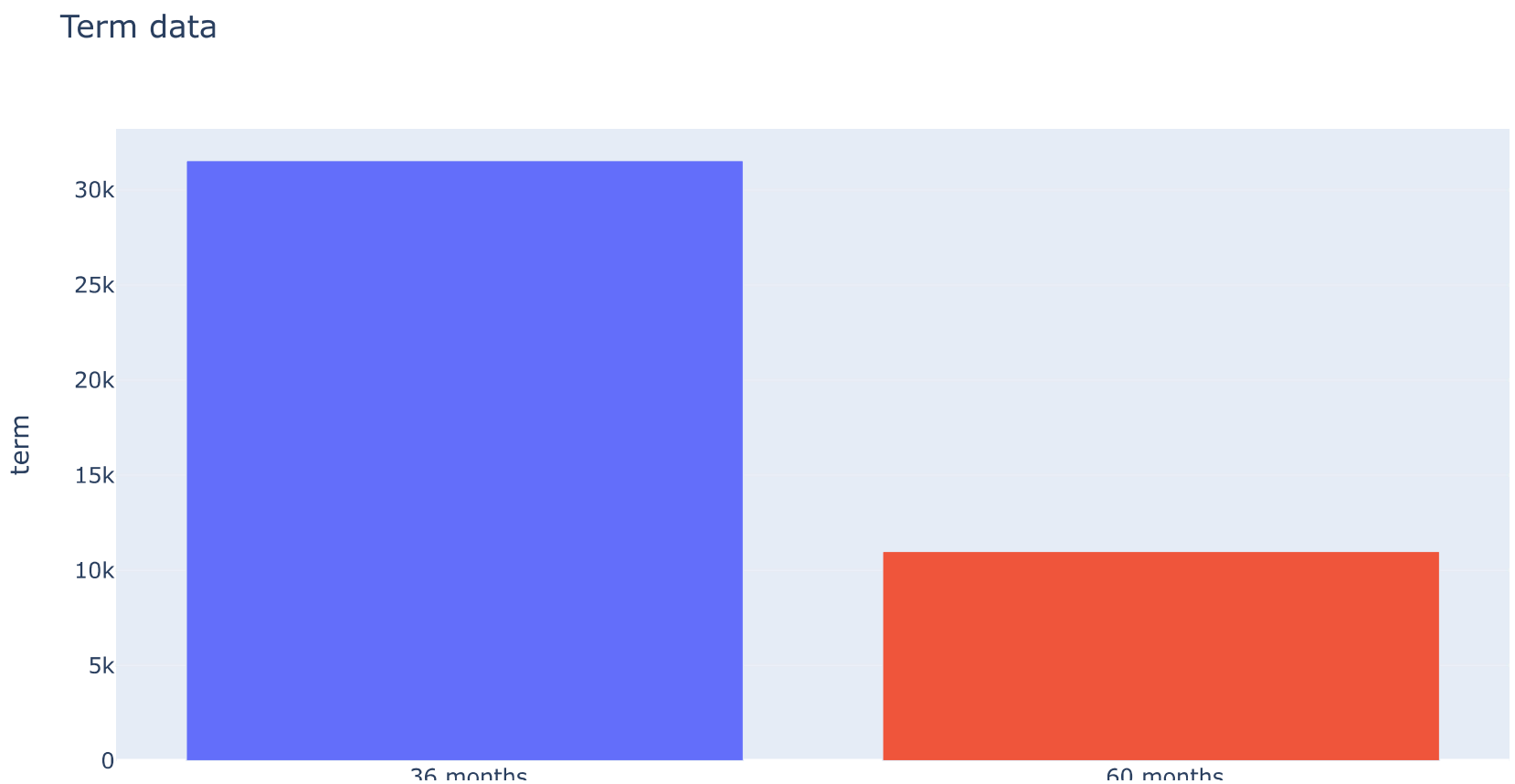
	term
36 months	31534
60 months	11001

In [21]:
```
term_data.index
```

Out[21]:
```
Index([' 36 months', ' 60 months'], dtype='object')
```

In [22]:
```
# sns.set(rc={'figure.figsize':(8,10)})
# sns.countplot(x='term',data=term_data,hue=term_data.index)
# plt.show()
```

In [23]:
```
# sns.set(rc={'figure.figsize':(8,10)})
# sns.barplot(y='term',data=term_data,x=term_data.index)
# plt.show()
labels=dict(x="Terms", y="Count",color=term_data.index)

fig = px.bar(term_data, x=term_data.index, y='term',color=term_data.index,title='Term data')
fig.show()
```



In [24]:
```
# Considering 'grade' feature
df.grade.value_counts()
```

Out[24]:
```
B    12389
A    10183
C     8740
D     6016
E     3394
F     1301
G      512
Name: grade, dtype: int64
```

In [25]:
```
grade_data = pd.DataFrame(df.grade.value_counts())
grade_data
```

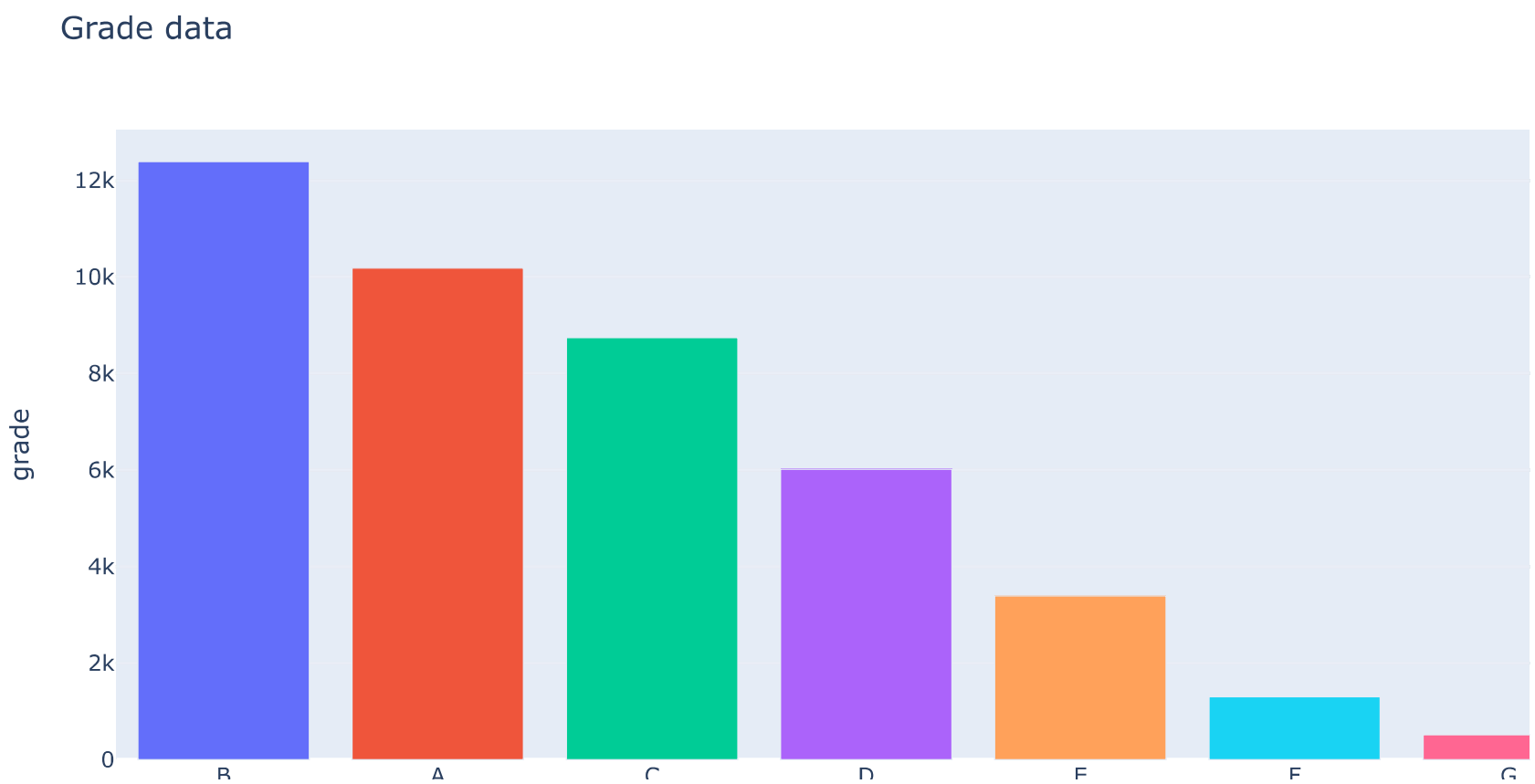
Out[25]:

	grade
B	12389
A	10183
C	8740
D	6016
E	3394
F	1301
G	512

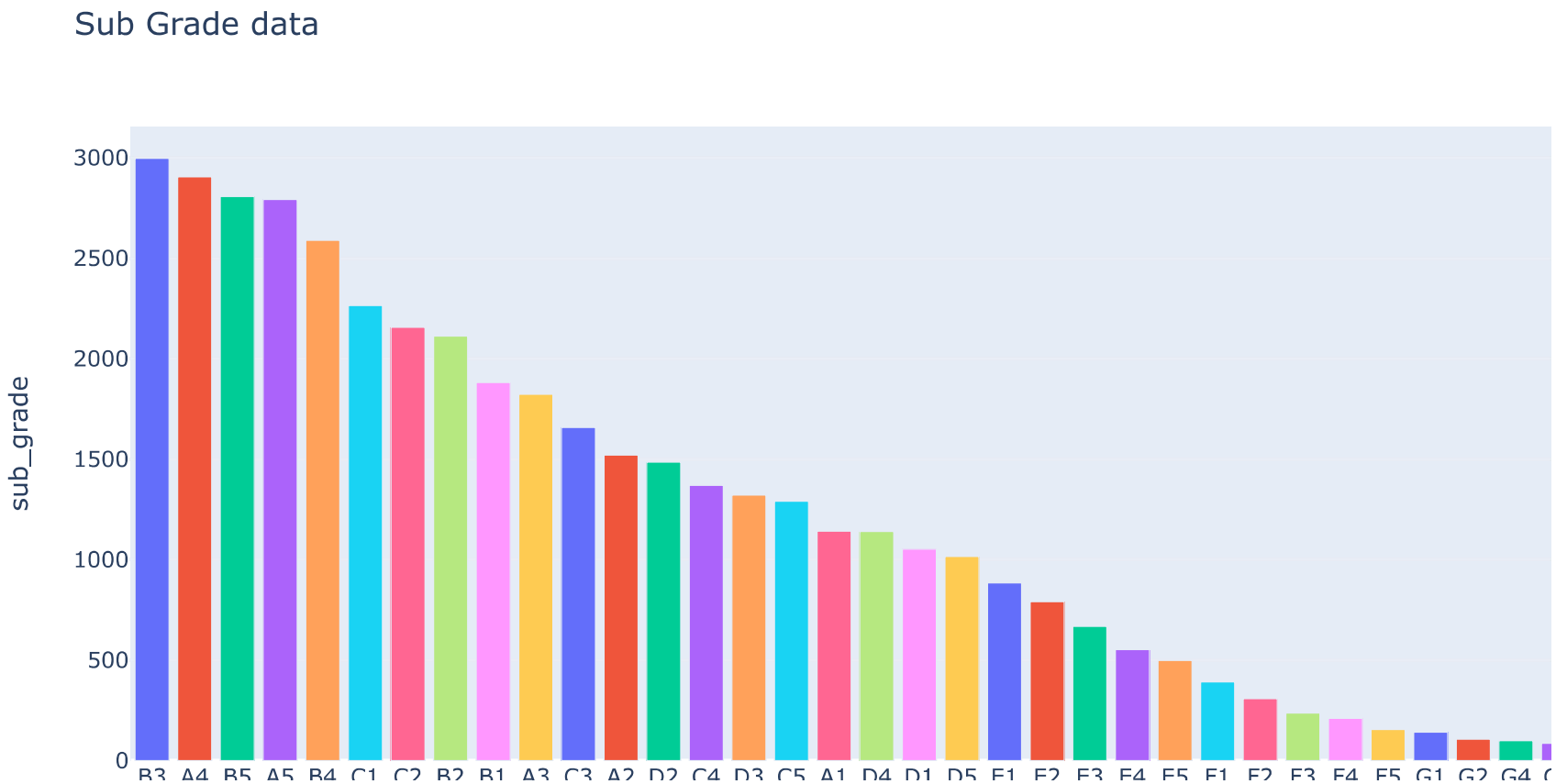
In [26]:
```
grade_data.index
```

Out[26]:
```
Index(['B', 'A', 'C', 'D', 'E', 'F', 'G'], dtype='object')
```

In [27]:
```
fig = px.bar(grade_data, x=grade_data.index, y='grade',color=grade_data.index,title='Grade data')
fig.show()
```



In [28]:
```
# Considering 'sub_grade' feature
df.sub_grade.value_counts()
subgrade_data = pd.DataFrame(df.sub_grade.value_counts())
subgrade_data
fig = px.bar(subgrade_data, x=subgrade_data.index, y='sub_grade',color=subgrade_data.index,title='Sub Grade data
fig.show()
```



In [29]:
```
# Considering 'emp_title' feature
df.emp_title.value_counts()
# emptitle_data = pd.DataFrame(df.emp_title.value_counts())
# emptitle_data
# fig = px.bar(emptitle_data, x=emptitle_data.index, y='emp_title',color=emptitle_data.index,title='emp title da
# fig.show()
```

Out[29]:
```
US Army                                139
Bank of America                        115
IBM                                     72
Kaiser Permanente                       61
AT&T                                    61
                                      ...
Regional Elite Airlines Services         1
Mass General Medical Group               1
Kontera                                  1
Southeast Georgia Health ystem           1
Homemaker                                1
Name: emp_title, Length: 30658, dtype: int64
```

Type *Markdown* and LaTeX: α^2

In [30]:
```
df.emp_title
```

Out[30]:
```
0                            NaN
1                          Ryder
2                            NaN
3              AIR RESOURCES BOARD
4        University Medical Group
                    ...
42534                 Homemaker
42535                        NaN
42536                        NaN
42537                        NaN
42538                        NaN
Name: emp_title, Length: 42539, dtype: object
```

In [31]:
```
# Instead of checking every column of 53 columns like above, here we are running a for loop on top of 'object' c
for i in df.columns[df.dtypes == 'object']:
    print(df[i].value_counts())
    print('************************')
```

```
1077501                                  1
540348                                   1
540300                                   1
540270                                   1
540280                                   1
                                        ..
769217                                   1
770000                                   1
765994                                   1
769973                                   1
Total amount funded in policy code 2: 0    1
Name: id, Length: 42537, dtype: int64
************************
 36 months    31534
 60 months    11001
Name: term, dtype: int64
************************
10.99%    970
11.49%    837
```

In [32]:
```
df.columns[df.dtypes == 'object']
```

Out[32]:
```
Index(['id', 'term', 'int_rate', 'grade', 'sub_grade', 'emp_title',
       'emp_length', 'home_ownership', 'verification_status', 'issue_d',
       'loan_status', 'pymnt_plan', 'url', 'desc', 'purpose', 'title',
       'zip_code', 'addr_state', 'delinq_2yrs', 'earliest_cr_line',
       'revol_bal', 'revol_util', 'total_acc', 'initial_list_status',
       'collection_recovery_fee', 'last_pymnt_d', 'last_credit_pull_d',
       'policy_code', 'application_type'],
      dtype='object')
```

After observing the above output, we are dropping columns which are not important and which doesn't add enough information.

In [33]:
```
# After observing the above output, we are dropping columns which are not important and which doesn't add enough

#df.drop(['emp_title','pymnt_plan','desc','title', 'zip_code', 'addr_state', 'initial_list_status','application_
df.drop(['pymnt_plan','title', 'zip_code', 'initial_list_status','application_type'],axis=1,inplace=True)
```

New dataframe - columns count is changed from 58 to 53

In [34]:
```
df
```

Out[34]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_tit
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	Na
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.83	C	C4	Ryd
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	84.33	C	C5	Na
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	339.31	C	C1	A RESOURCE BOAR

In [35]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42539 entries, 0 to 42538
Data columns (total 53 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   id                          42537 non-null  object 
 1   member_id                   42535 non-null  float64
 2   loan_amnt                   42535 non-null  float64
 3   funded_amnt                 42535 non-null  float64
 4   funded_amnt_inv             42535 non-null  float64
 5   term                        42535 non-null  object 
 6   int_rate                    42535 non-null  object 
 7   installment                 42535 non-null  float64
 8   grade                       42535 non-null  object 
 9   sub_grade                   42535 non-null  object 
 10  emp_title                   39909 non-null  object 
 11  emp_length                  41423 non-null  object 
 12  home_ownership              42535 non-null  object 
 13  annual_inc                  42531 non-null  float64
 14  verification_status         42535 non-null  object 
 15  issue_d                     42535 non-null  object 
 16  loan_status                 42535 non-null  object 
 17  url                         42535 non-null  object 
 18  desc                        29242 non-null  object 
 19  purpose                     42535 non-null  object 
 20  addr_state                  42535 non-null  object 
 21  dti                         42535 non-null  float64
 22  delinq_2yrs                 42506 non-null  object 
 23  earliest_cr_line            42506 non-null  object 
 24  fico_range_low              42535 non-null  float64
 25  fico_range_high             42535 non-null  float64
 26  inq_last_6mths              42506 non-null  float64
 27  open_acc                    42506 non-null  float64
 28  pub_rec                     42506 non-null  float64
 29  revol_bal                   42535 non-null  object 
 30  revol_util                  42445 non-null  object 
 31  total_acc                   42506 non-null  object 
 32  out_prncp                   42535 non-null  float64
 33  out_prncp_inv               42535 non-null  float64
 34  total_pymnt                 42535 non-null  float64
 35  total_pymnt_inv             42535 non-null  float64
```

```
 36  total_rec_prncp             42535 non-null  float64
 37  total_rec_int               42535 non-null  float64
 38  total_rec_late_fee          42535 non-null  float64
 39  recoveries                  42535 non-null  float64
 40  collection_recovery_fee     42535 non-null  object
 41  last_pymnt_d                42452 non-null  object
 42  last_pymnt_amnt             42534 non-null  float64
 43  last_credit_pull_d          42531 non-null  object
 44  last_fico_range_high        42535 non-null  float64
 45  last_fico_range_low         42535 non-null  float64
 46  collections_12_mths_ex_med  42389 non-null  float64
 47  policy_code                 42535 non-null  object
 48  acc_now_delinq              42505 non-null  float64
 49  chargeoff_within_12_mths    42390 non-null  float64
 50  delinq_amnt                 42505 non-null  float64
 51  pub_rec_bankruptcies        41170 non-null  float64
 52  tax_liens                   42429 non-null  float64
dtypes: float64(29), object(24)
memory usage: 17.2+ MB
```

In [36]:
```
#Checking numericals columns
for i in df.columns[df.dtypes == 'float64']:
    print(df[i].value_counts())
    print('************************')
```

```
1296599.0    1
694920.0     1
697615.0     1
697589.0     1
697502.0     1
            ..
971607.0     1
970659.0     1
971558.0     1
966956.0     1
70681.0      1
Name: member_id, Length: 42535, dtype: int64
************************
10000.0    3016
12000.0    2439
5000.0     2260
6000.0     2037
15000.0    2012
           ...
```

In [37]:
```
col = ['acc_now_delinq','chargeoff_within_12_mths','collections_12_mths_ex_med','delinq_amnt','policy_code','tax
```

In [38]:
```
for i in col:
    print(df[i].value_counts())
    print('************************')
```

```
0.0    42501
1.0        4
Name: acc_now_delinq, dtype: int64
************************
0.0    42390
Name: chargeoff_within_12_mths, dtype: int64
************************
0.0    42389
Name: collections_12_mths_ex_med, dtype: int64
************************
0.0       42503
27.0          1
6053.0        1
Name: delinq_amnt, dtype: int64
************************
1             34343
1              8191
INDIVIDUAL        1
Name: policy_code, dtype: int64
************************
0.0    42428
1.0        1
Name: tax_liens, dtype: int64
************************
```

In [39]:
```
# There are also some unnecessary numerical columns as well. We are dropping those numerical columns after exami

df.drop(['acc_now_delinq','chargeoff_within_12_mths','collection_recovery_fee',
         'collections_12_mths_ex_med','delinq_amnt','policy_code','tax_liens'],axis=1,inplace=True)
```

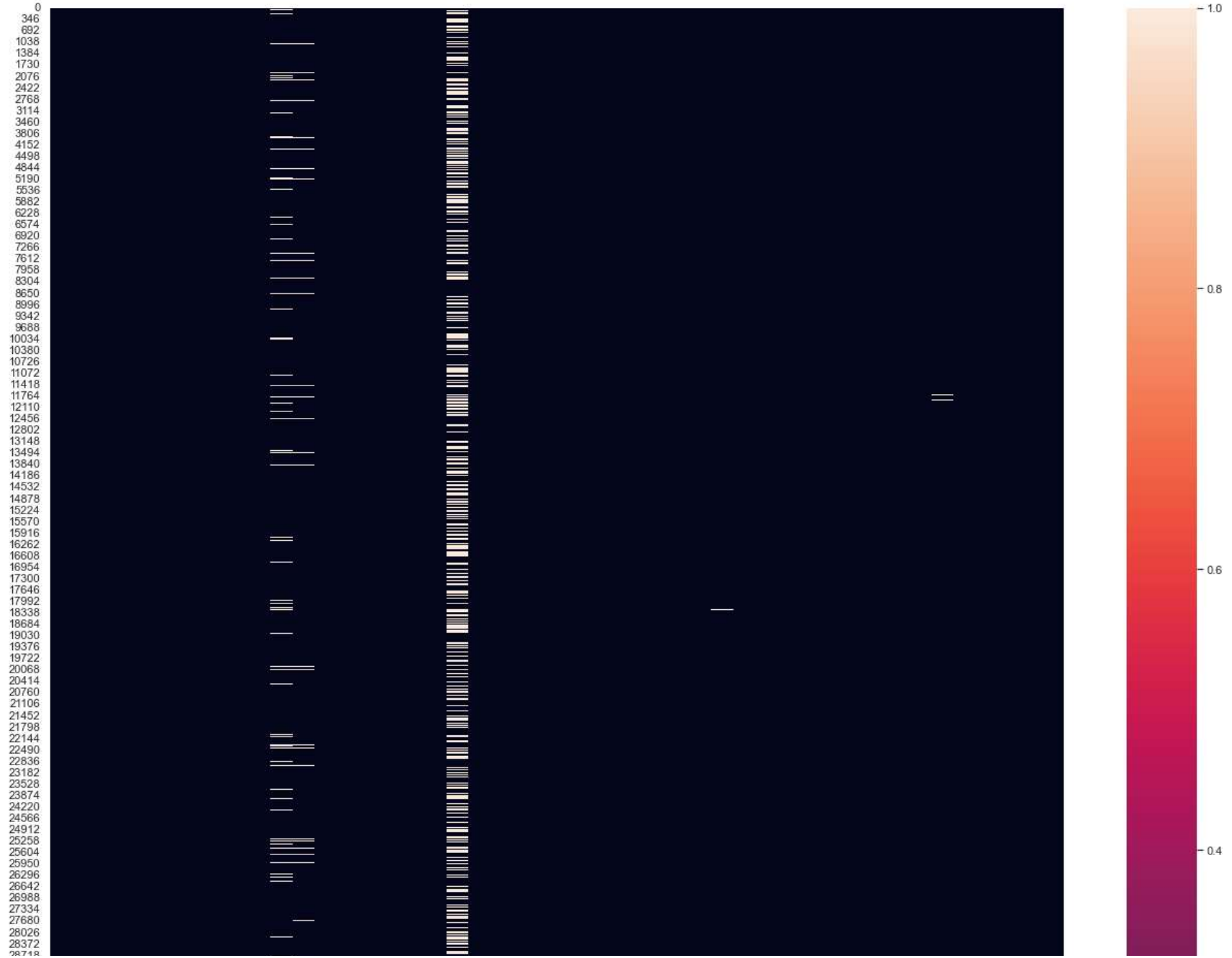
In [40]:
```
df.head(1)
```

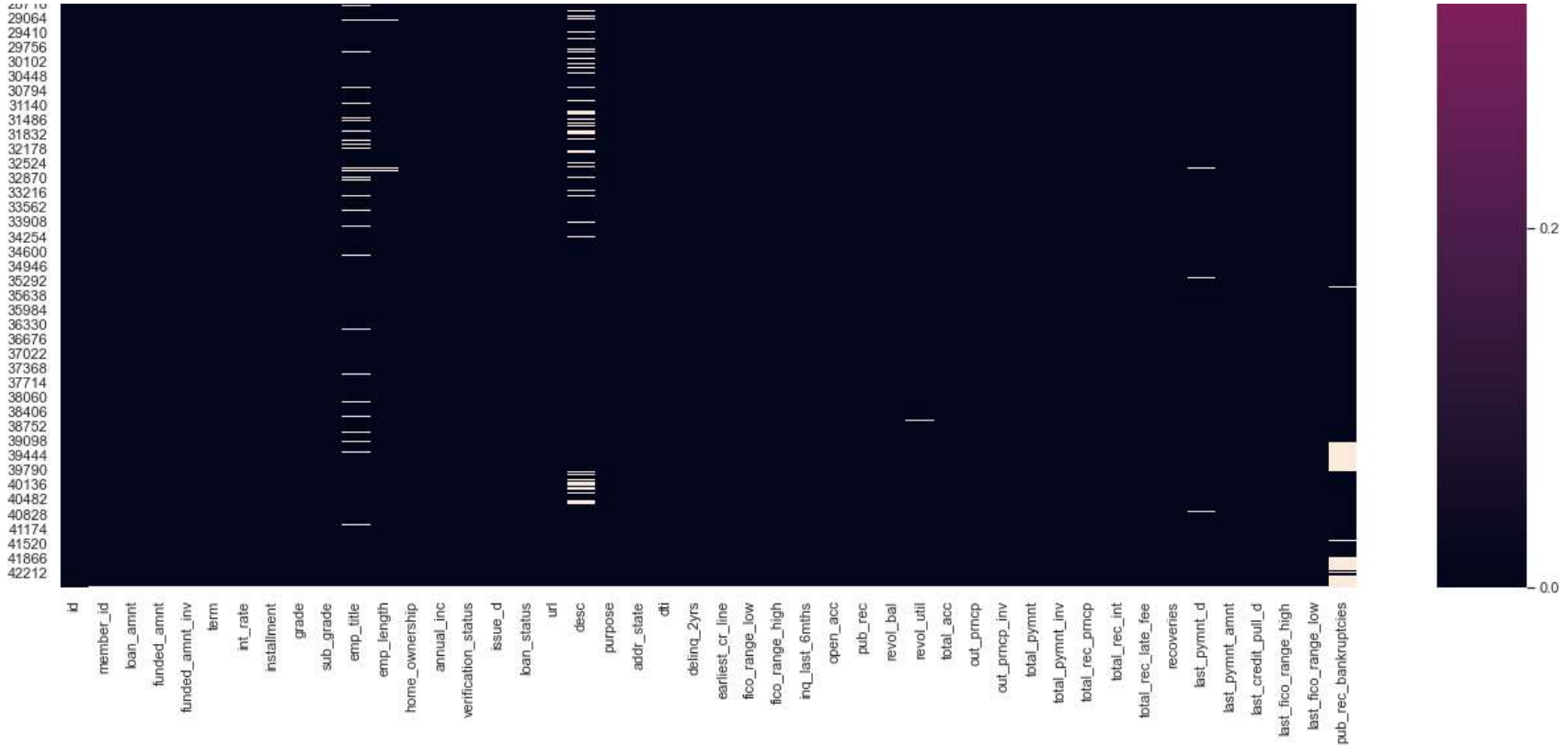
Out[40]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_lengt
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	NaN	10+ yea

Checking for null values in the updated dataframe

In [41]:
```
# Checking for null values in the updated dataframe
sns.heatmap(df.isnull())
plt.show()
```




29064
29410
29756
30102
30448
30794
31140
31486
31832
32178
32524
32870
33216
33562
33908
34254
34600
34946
35292
35638
35984
36330
36676
37022
37368
37714
38060
38406
38752
39098
39444
39790
40136
40482
40828
41174
41520
41866
42212
id
member_id
loan_amnt
funded_amnt
funded_amnt_inv
term
int_rate
installment
grade
sub_grade
emp_title
emp_length
home_ownership
annual_inc
verification_status
issue_d
loan_status
url
desc
purpose
addr_state
dti
delinq_2yrs
earliest_cr_line
fico_range_low
fico_range_high
inq_last_6mths
open_acc
pub_rec
revol_bal
revol_util
total_acc
out_prncp
out_prncp_inv
total_pymnt
total_pymnt_inv
total_rec_prncp
total_rec_int
total_rec_late_fee
recoveries
last_pymnt_d
last_pymnt_amnt
last_credit_pull_d
last_fico_range_high
last_fico_range_low
pub_rec_bankruptcies
0.2
0.0

In [42]:
```
df.isnull().sum()
```

Out[42]:
```
id                          2
member_id                   4
loan_amnt                   4
funded_amnt                 4
funded_amnt_inv             4
term                        4
int_rate                    4
installment                 4
grade                       4
sub_grade                   4
emp_title                2630
emp_length               1116
home_ownership              4
annual_inc                  8
verification_status         4
issue_d                     4
loan_status                 4
url                         4
desc                    13297
purpose                     4
addr_state                  4
dti                         4
delinq_2yrs                33
earliest_cr_line           33
fico_range_low              4
fico_range_high             4
inq_last_6mths             33
open_acc                   33
pub_rec                    33
revol_bal                   4
revol_util                 94
total_acc                  33
out_prncp                   4
out_prncp_inv               4
total_pymnt                 4
total_pymnt_inv             4
total_rec_prncp             4
total_rec_int               4
total_rec_late_fee          4
recoveries                  4
last_pymnt_d               87
```

```
last_pymnt_amnt             5
last_credit_pull_d          8
last_fico_range_high        4
last_fico_range_low         4
pub_rec_bankruptcies     1369
dtype: int64
```

In [43]:
```
# Displaying all the columns

pd.set_option('display.max_columns', None)

df.head(2)
```

Out[43]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_lengt
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	NaN	10+ yea
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.83	C	C4	Ryder	< 1 ye

Decreased no.of columns from 53 to 46

In [44]:
```
df.shape
# We were able to decrease no.of columns from 50 to 43 and we will still try to decrease the no.of columns.
```

Out[44]:
```
(42539, 46)
```

In [45]:
```
# Checking two columns which are the same.
df[df['out_prncp'] != df['out_prncp_inv']] # Both are same
```

Out[45]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title
32	1065420	1299514.0	10000.0	10000.0	9975.00000	60 months	15.96%	242.97	C	C5	US Legal Support
39	1069346	1304237.0	12500.0	12500.0	12475.00000	60 months	12.69%	282.44	B	B5	United States Infrastructure Corporation
86	1063958	1297940.0	14000.0	14000.0	13975.00000	60 months	17.27%	349.98	D	D3	community colleges of spokane
95	1068575	1303001.0	15300.0	15300.0	15275.00000	60 months	22.06%	423.10	F	F4	OSSI
242	1064063	1296651.0	18825.0	18825.0	18800.00000	60 months	17.58%	473.75	D	D4	executive plaza
...	...	...	...	...	...	...	...	...	...	...	...
38040	367791	381665.0	10000.0	10000.0	6592.91596	36 months	14.74%	345.37	D	D3	CBD college
42535	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
42536	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title
42537	Total amount funded in policy code 1: 460296150	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
42538	Total amount funded in policy code 2: 0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

276 rows × 46 columns

Dropping same columns

In [46]:
```
# Dropping those columns to avoid multi-collinearity
df.drop('out_prncp_inv',axis=1,inplace = True)
df.drop('out_prncp',axis=1,inplace = True)
```

In [47]:
```
df
```

Out[47]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	NaN
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.83	C	C4	Ryder
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	84.33	C	C5	NaN
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	339.31	C	C1	AIR RESOURCES BOARD
4	1075358	1311748.0	3000.0	3000.0	3000.0	60 months	12.69%	67.79	B	B5	University Medical Group
...	...	...	...	...	...	...	...	...	...	...	...
42534	70686	70681.0	5000.0	5000.0	0.0	36 months	7.75%	156.11	A	A3	Homemaker
42535	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
42536	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
42537	Total amount funded in policy code 1: 460296150	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title
42538	Total amount funded in policy code 2: 0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

42539 rows × 44 columns

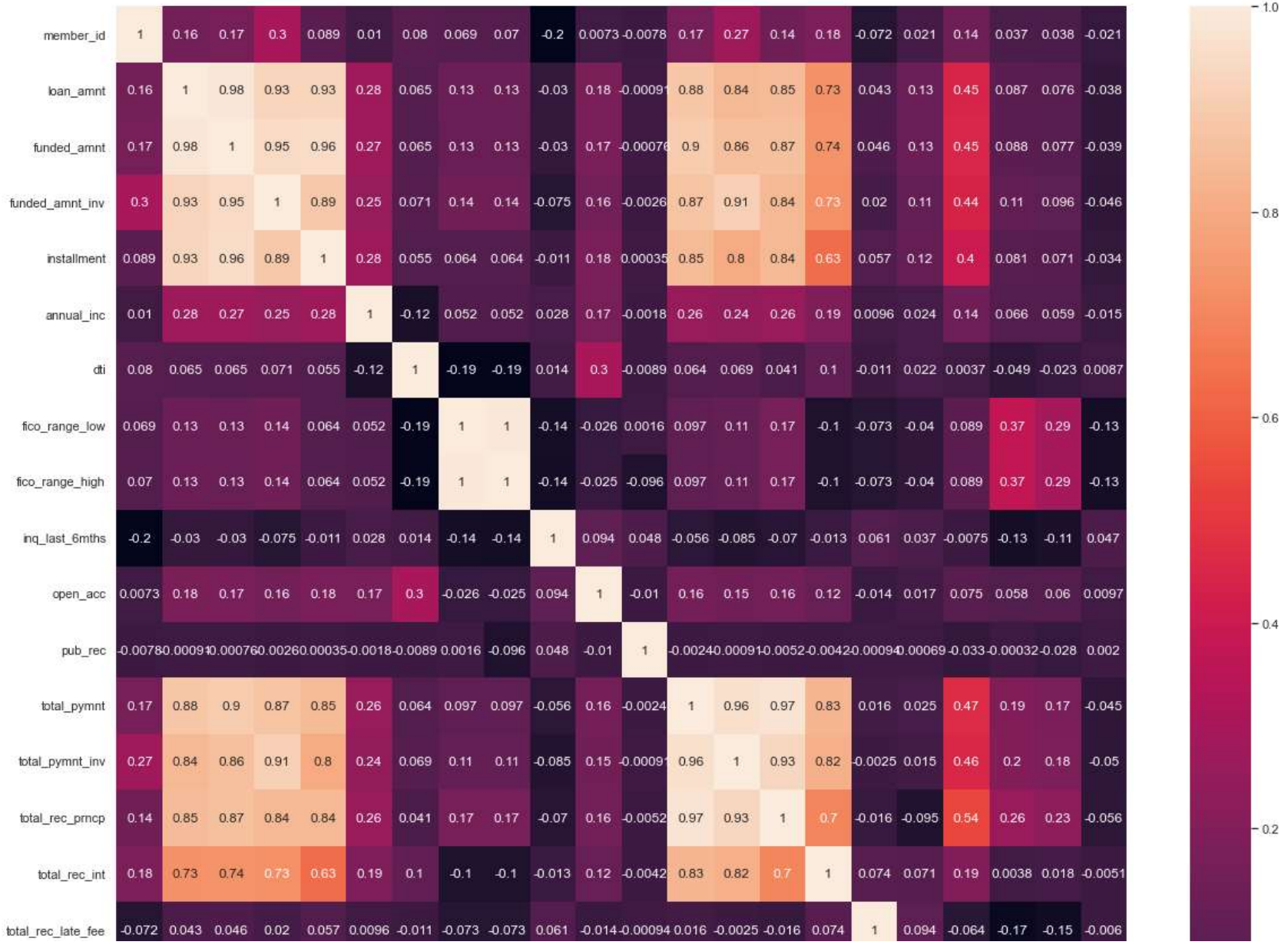
In [48]:
```
#df[df['out_prncp'] != df['out_prncp_inv']]   #dropped
```

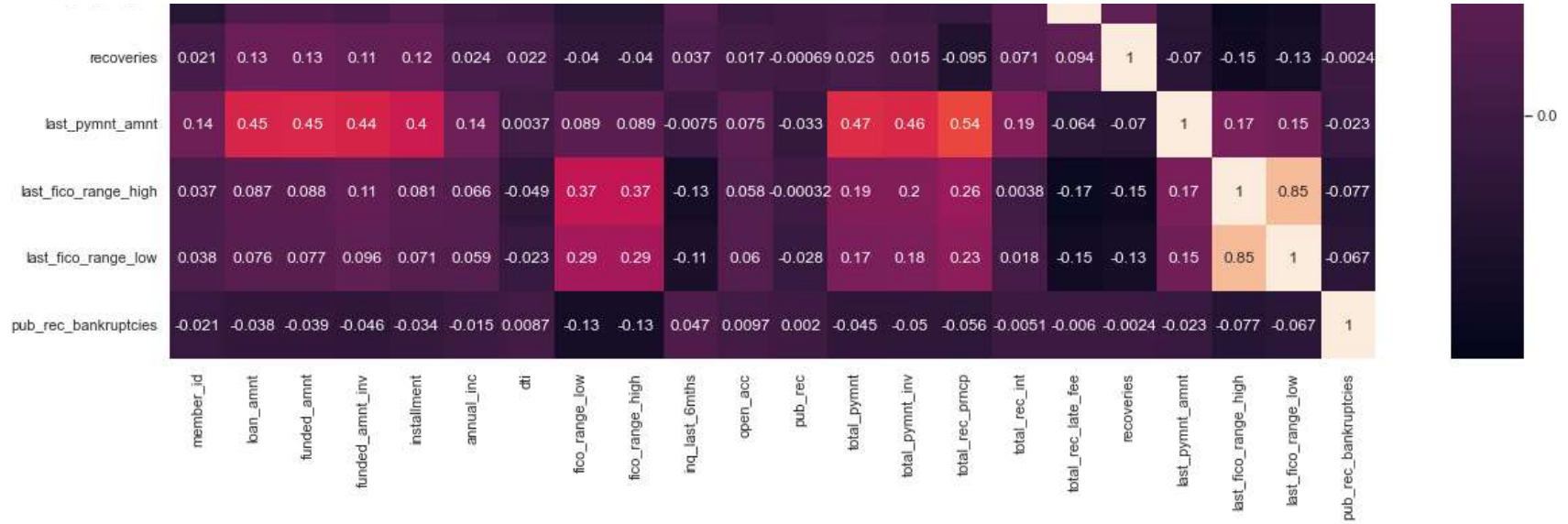
In [49]:
```
df.head()
```

Out[49]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_l
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	NaN	10+
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.83	C	C4	Ryder	<
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	84.33	C	C5	NaN	10+
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	339.31	C	C1	AIR RESOURCES BOARD	10+
4	1075358	1311748.0	3000.0	3000.0	3000.0	60 months	12.69%	67.79	B	B5	University Medical Group	

In [50]:
```
# Checking correlation  and using heatmap to visualise it.
sns.set(rc={'figure.figsize':(20,20)})
sns.set_style('whitegrid')
# Heatmap
sns.heatmap(df.corr(),annot=True)
plt.show()
```





Type *Markdown* and LaTeX: α^2

Dropping rows which have only null values in all the columns

In [51]:
```
# Dropping rows which have only null values in all the columns
df.dropna(how='all',inplace=True)
```

In [52]:
```
df
```

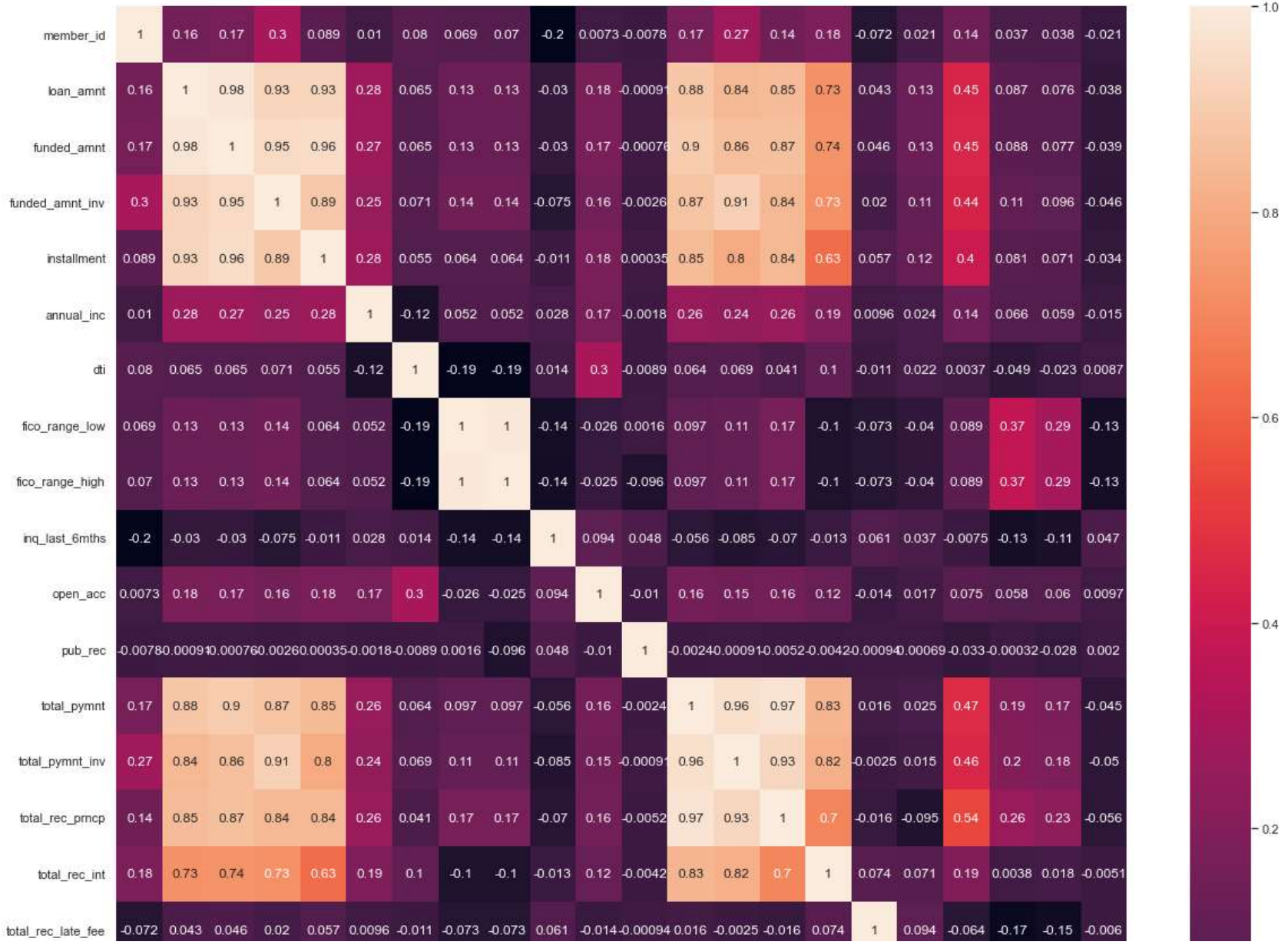
Out[52]:

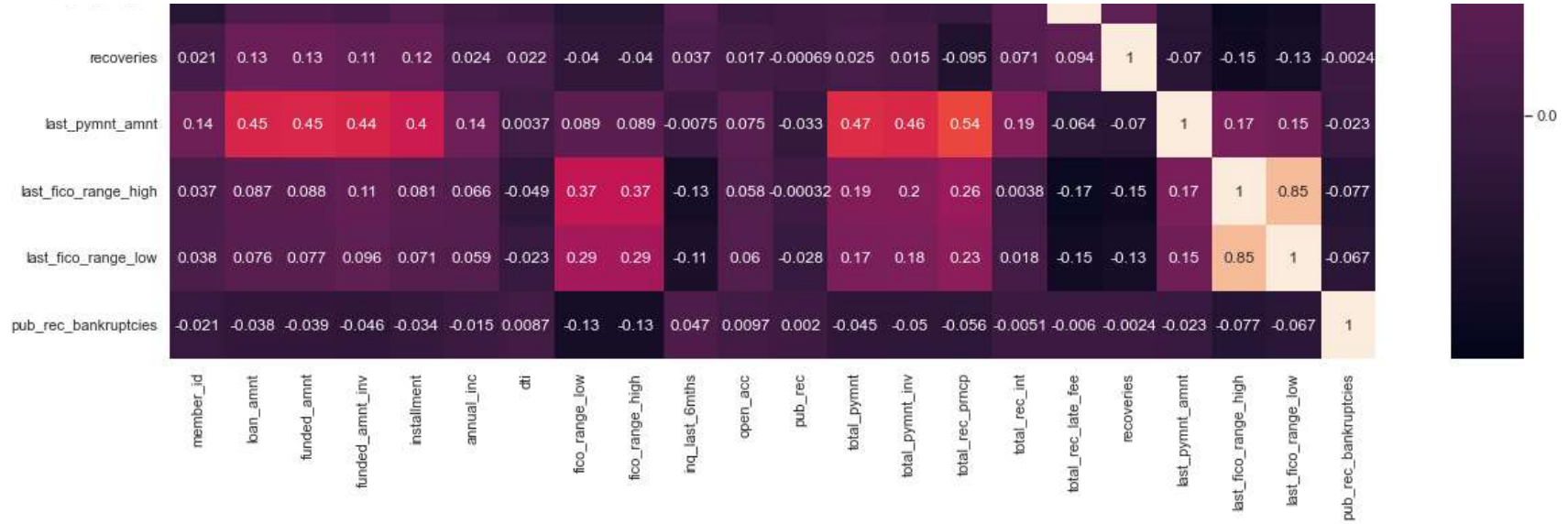
	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	NaN
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.83	C	C4	Ryder
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	84.33	C	C5	NaN
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	339.31	C	C1	AIR RESOURCES BOARD
4	1075358	1311748.0	3000.0	3000.0	3000.0	60 months	12.69%	67.79	B	B5	University Medical Group
...	...	...	...	...	...	...	...	...	...	...	...
42532	72176	70868.0	2525.0	2525.0	225.0	36 months	9.33%	80.69	B	B3	NaN
42533	71623	70735.0	6500.0	6500.0	0.0	36 months	8.38%	204.84	A	A5	NaN
42534	70686	70681.0	5000.0	5000.0	0.0	36 months	7.75%	156.11	A	A3	Homemaker

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title
42537	Total amount funded in policy code 1: 460296150	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
42538	Total amount funded in policy code 2: 0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

42537 rows × 44 columns

In [53]:
```
# Checking correlation  and using heatmap to visualise it.
sns.set(rc={'figure.figsize':(20,20)})
sns.set_style('whitegrid')
# Heatmap
sns.heatmap(df.corr(),annot=True)
plt.show()
```





In [54]:
```
df.shape
```

Out[54]: (42537, 44)

In [55]:
```
df.head()
```

Out[55]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_l
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	NaN	10+
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.83	C	C4	Ryder	<
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	84.33	C	C5	NaN	10+
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	339.31	C	C1	AIR RESOURCES BOARD	10+
4	1075358	1311748.0	3000.0	3000.0	3000.0	60 months	12.69%	67.79	B	B5	University Medical Group	

Decreased no.of columns from 46 to 44

In [56]:
```
df.shape
```

Out[56]: (42537, 44)

Converting categorical columns to numerical columns

In [57]:
```
df.describe()
```

Out[57]:

	member_id	loan_amnt	funded_amnt	funded_amnt_inv	installment	annual_inc	dti	fico_range_low	fico_range
count	4.253500e+04	42535.000000	42535.000000	42535.000000	42535.000000	4.253100e+04	42535.000000	42535.000000	42535.
mean	8.257026e+05	11089.722581	10821.585753	10139.830603	322.623063	6.913656e+04	13.372683	713.052639	717.
std	2.795409e+05	7410.938391	7146.914675	7131.686447	208.927216	6.409635e+04	6.726567	36.188475	36.
min	7.047300e+04	500.000000	500.000000	0.000000	15.670000	1.896000e+03	0.000000	610.000000	1.
25%	6.384795e+05	5200.000000	5000.000000	4950.000000	165.520000	4.000000e+04	8.200000	685.000000	689.
50%	8.241780e+05	9700.000000	9600.000000	8500.000000	277.690000	5.900000e+04	13.470000	710.000000	714.
75%	1.033946e+06	15000.000000	15000.000000	14000.000000	428.180000	8.250000e+04	18.680000	740.000000	744.
max	1.314167e+06	35000.000000	35000.000000	35000.000000	1305.190000	6.000000e+06	29.990000	825.000000	829.

Updating dates format

In [58]:
```
col = ['issue_d','earliest_cr_line','last_pymnt_d','last_credit_pull_d']
for i in col:
    print(df[i].value_counts())
    print('####################')
```

```
Dec-11    2267
Nov-11    2232
Oct-11    2118
Sep-11    2067
Aug-11    1934
Jul-11    1875
Jun-11    1835
May-11    1704
Apr-11    1563
Mar-11    1448
Jan-11    1380
Dec-10    1335
Feb-11    1298
Oct-10    1232
Nov-10    1224
Jul-10    1204
Sep-10    1189
Aug-10    1175
Jun-10    1105
May-10     989
Apr-10     912
Mar-10     828
Feb-10     682
Jan-10     662
Nov-09     662
Dec-09     658
Oct-09     604
Sep-09     507
Aug-09     446
Jul-09     411
Jun-09     406
Mar-08     402
May-09     359
Apr-09     333
Mar-09     324
Feb-08     306
Jan-08     305
Feb-09     302
```

```
Jan-09     269
Apr-08     259
Dec-08     253
Nov-08     209
Dec-07     172
Jul-08     141
Jun-08     124
Oct-08     122
May-08     115
Nov-07     112
Oct-07     105
Aug-08     100
Aug-07      74
Jul-07      63
Sep-08      57
Sep-07      53
Jun-07      24
Name: issue_d, dtype: int64
####################
Oct-99    393
Nov-98    390
Oct-00    370
Dec-98    366
Dec-97    348
          ...
Mar-63      1
Oct-64      1
Feb-66      1
Dec-61      1
Nov-59      1
Name: earliest_cr_line, Length: 531, dtype: int64
####################
Mar-13    1070
Dec-14     949
May-13     943
Feb-13     906
Mar-12     893
          ...
Mar-08      18
Jan-08      11
Feb-08       8
Dec-07       2
14.99        1
```

```
Name: last_pymnt_d, Length: 107, dtype: int64
####################
Sep-16    16232
Mar-16      859
Aug-16      771
Apr-16      700
Feb-13      696
           ...
Jul-08        1
Jun-08        1
694           1
Jul-07        1
May-08        1
Name: last_credit_pull_d, Length: 112, dtype: int64
####################
```

In [59]:
```
df['issue_d'].value_counts()
```

Out[59]:
```
Dec-11    2267
Nov-11    2232
Oct-11    2118
Sep-11    2067
Aug-11    1934
Jul-11    1875
Jun-11    1835
May-11    1704
Apr-11    1563
Mar-11    1448
Jan-11    1380
Dec-10    1335
Feb-11    1298
Oct-10    1232
Nov-10    1224
Jul-10    1204
Sep-10    1189
Aug-10    1175
Jun-10    1105
May-10     989
Apr-10     912
Mar-10     828
Feb-10     682
Jan-10     662
Nov-09     662
Dec-09     658
Oct-09     604
Sep-09     507
Aug-09     446
Jul-09     411
Jun-09     406
Mar-08     402
May-09     359
Apr-09     333
Mar-09     324
Feb-08     306
Jan-08     305
Feb-09     302
Jan-09     269
Apr-08     259
Dec-08     253
```

```
Nov-08     209
Dec-07     172
Jul-08     141
Jun-08     124
Oct-08     122
May-08     115
Nov-07     112
Oct-07     105
Aug-08     100
Aug-07      74
Jul-07      63
Sep-08      57
Sep-07      53
Jun-07      24
Name: issue_d, dtype: int64
```

In [60]:
```
#Changing date format
df['issue_d'] = pd.to_datetime(df['issue_d'], format='%b-%y', yearfirst=False)
df['issue_d']
```

Out[60]:
```
0       2011-12-01
1       2011-12-01
2       2011-12-01
3       2011-12-01
4       2011-12-01
              ...
42532   2007-06-01
42533   2007-06-01
42534   2007-06-01
42537          NaT
42538          NaT
Name: issue_d, Length: 42537, dtype: datetime64[ns]
```

In [61]:
```
#Changing date format
df['earliest_cr_line'].value_counts()
```

Out[61]:
```
Oct-99    393
Nov-98    390
Oct-00    370
Dec-98    366
Dec-97    348
         ...
Mar-63      1
Oct-64      1
Feb-66      1
Dec-61      1
Nov-59      1
Name: earliest_cr_line, Length: 531, dtype: int64
```

In [62]:
```
df[df['earliest_cr_line']=='725']
```

Out[62]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_l
38040	367791	381665.0	10000.0	10000.0	6592.91596	36 months	14.74%	345.37	D	D3	CBD college	9

In [63]:
```
#Dropping row with wrong date value
df.drop(labels=38040,axis=0,inplace=True)
```

In [64]:
```
df[df['earliest_cr_line']=='725']
```

Out[64]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_length	home_

In [65]:
```
#Changing date format
df['earliest_cr_line'] = pd.to_datetime(df['earliest_cr_line'], format='%b-%y', yearfirst=False)
df['earliest_cr_line']
```

Out[65]:
```
0       1985-01-01
1       1999-04-01
2       2001-11-01
3       1996-02-01
4       1996-01-01
           ...
42532          NaT
42533          NaT
42534          NaT
42537          NaT
42538          NaT
Name: earliest_cr_line, Length: 42536, dtype: datetime64[ns]
```

In [66]:
```
df['last_pymnt_d'].value_counts()
```

Out[66]:
```
Mar-13    1070
Dec-14     949
May-13     943
Feb-13     906
Mar-12     893
          ...
Jun-08      20
Mar-08      18
Jan-08      11
Feb-08       8
Dec-07       2
Name: last_pymnt_d, Length: 106, dtype: int64
```

In [67]:
```
#Changing date format
df['last_pymnt_d'] = pd.to_datetime(df['last_pymnt_d'], format='%b-%y', yearfirst=False)
df['last_pymnt_d']
```

Out[67]:
```
0       2015-01-01
1       2013-04-01
2       2014-06-01
3       2015-01-01
4       2016-09-01
              ...
42532   2010-06-01
42533   2010-06-01
42534   2010-06-01
42537          NaT
42538          NaT
Name: last_pymnt_d, Length: 42536, dtype: datetime64[ns]
```

In [68]:
```
df['last_credit_pull_d'].value_counts()
```

Out[68]:
```
Sep-16    16232
Mar-16      859
Aug-16      771
Apr-16      700
Feb-13      696
           ...
Nov-07        3
May-08        1
Jul-08        1
Jun-08        1
Jul-07        1
Name: last_credit_pull_d, Length: 111, dtype: int64
```

In [69]:
```
#Changing date format
df['last_credit_pull_d'] = pd.to_datetime(df['last_credit_pull_d'], format='%b-%y', yearfirst=False)
df['last_credit_pull_d']
```

Out[69]:
```
0       2016-09-01
1       2016-09-01
2       2016-09-01
3       2016-04-01
4       2016-09-01
           ...
42532   2007-05-01
42533   2007-08-01
42534   2015-02-01
42537          NaT
42538          NaT
Name: last_credit_pull_d, Length: 42536, dtype: datetime64[ns]
```

In [70]:
```
# # Converting objects to datetime columns
# col = ['issue_d','earliest_cr_line','last_pymnt_d','last_credit_pull_d']
# for i in col:
#     df[i] = pd.to_datetime(df[i].str.upper(), format=[['%y-%m-%d','%b-%y']])
```

In [71]:
```
# Adding new column for Year of joining for 'earliest_cr_line' column.
df['earliest_cr_line_year'] = pd.DatetimeIndex(df['earliest_cr_line']).year
df['earliest_cr_line_year']
```

Out[71]:
```
0        1985.0
1        1999.0
2        2001.0
3        1996.0
4        1996.0
          ...
42532       NaN
42533       NaN
42534       NaN
42537       NaN
42538       NaN
Name: earliest_cr_line_year, Length: 42536, dtype: float64
```

Adding new features by getting month and year from

issue_d,last_pymnt_d and last_credit_pull_d columns

In [72]:
```
# Adding new features by getting month and year from issue_d,last_pymnt_d and last_credit_pull_d columns
df['issue_d_year'] = pd.DatetimeIndex(df['issue_d']).year
df['issue_d_month'] = pd.DatetimeIndex(df['issue_d']).month
df['last_pymnt_d_year'] = pd.DatetimeIndex(df['last_pymnt_d']).year
df['last_pymnt_d_month'] = pd.DatetimeIndex(df['last_pymnt_d']).month
df['last_credit_pull_d_year'] = pd.DatetimeIndex(df['last_credit_pull_d']).year
df['last_credit_pull_d_month'] = pd.DatetimeIndex(df['last_credit_pull_d']).month
```

In [73]:
```
df['issue_d_year']
```

Out[73]:
```
0        2011.0
1        2011.0
2        2011.0
3        2011.0
4        2011.0
          ...
42532    2007.0
42533    2007.0
42534    2007.0
42537       NaN
42538       NaN
Name: issue_d_year, Length: 42536, dtype: float64
```

In [74]:
```
df.head(1)
```

Out[74]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_lengt
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	NaN	10+ yea

In [75]:
```
df['earliest_cr_line_year']
```

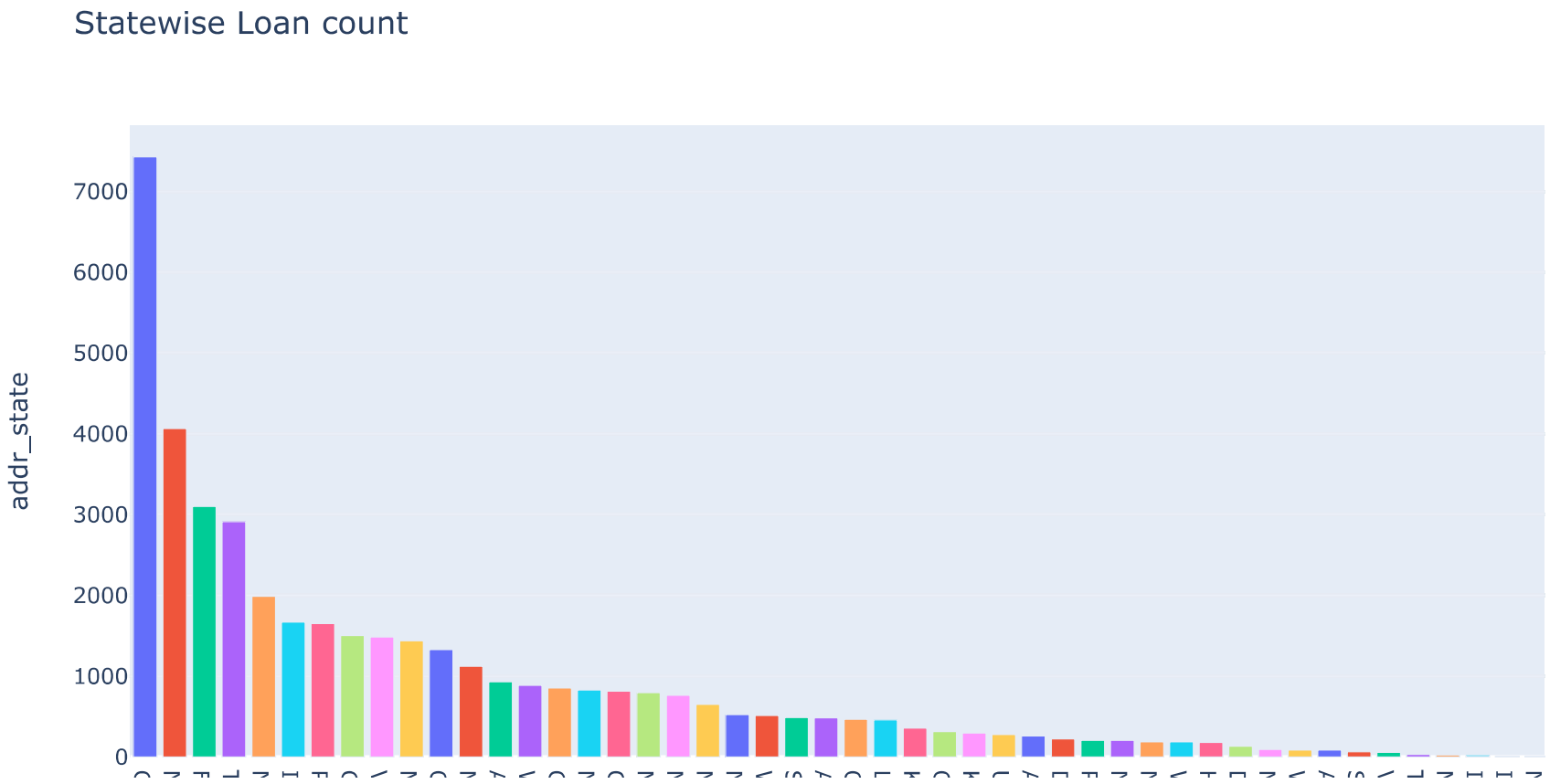
Out[75]:
```
0        1985.0
1        1999.0
2        2001.0
3        1996.0
4        1996.0
           ...
42532       NaN
42533       NaN
42534       NaN
42537       NaN
42538       NaN
Name: earliest_cr_line_year, Length: 42536, dtype: float64
```

In [76]:
```
df['addr_state'].value_counts()
```

Out[76]:
```
CA    7428
NY    4065
FL    3104
TX    2915
NJ    1988
IL    1672
PA    1651
GA    1503
VA    1487
MA    1438
OH    1329
MD    1125
AZ     933
WA     888
CO     857
NC     830
CT     816
MI     796
MO     765
MN     652
NV     527
WI     516
SC     489
AL     484
OR     468
LA     461
KY     359
OK     317
KS     298
UT     278
AR     261
DC     224
RI     208
NM     205
NH     188
WV     187
HI     181
DE     136
MT      96
WY      87
AK      86
```

```
SD      67
VT      57
TN      32
MS      26
IN      19
IA      12
NE      11
ID       9
ME       3
Name: addr_state, dtype: int64
```

In [77]:
```
# Considering 'State data' feature
df.addr_state.value_counts()
addr_state_data = pd.DataFrame(df.addr_state.value_counts())
addr_state_data
fig = px.bar(addr_state_data, x=addr_state_data.index, y='addr_state',color=addr_state_data.index,title='Statewi
#fig.xticks(rotation=180)
fig.show()
```



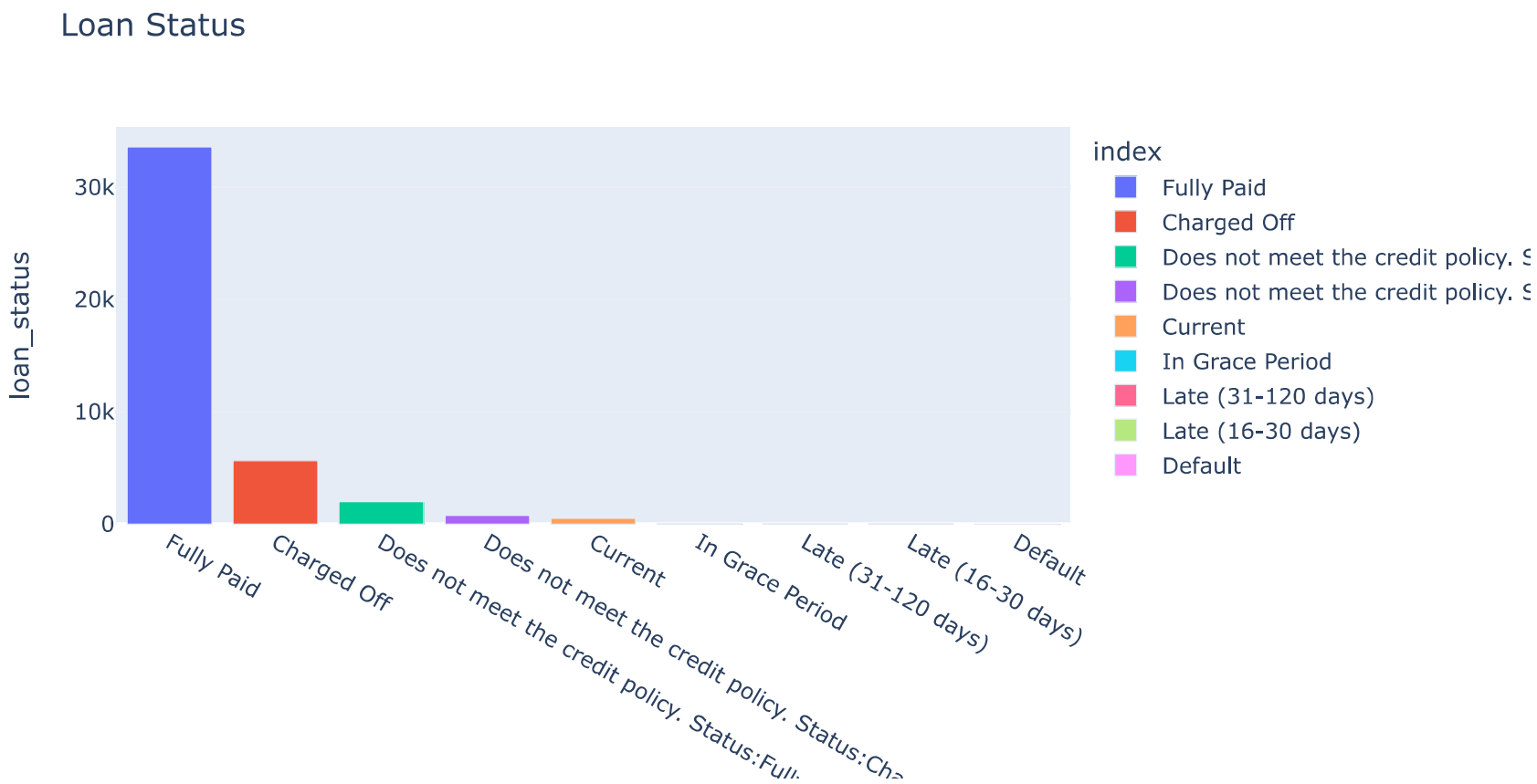
In [78]:
```
# fig = px.choropleth(df,locations=addr_state_data, color='unemp',
#                             color_continuous_scale="Viridis",
#                             range_color=(0, 12),
#                             scope="usa",
#                             labels={'unemp':'unemployment rate'}
#                            )
# fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
# fig.show()
```

In [79]:
```
df.loan_status.value_counts()
```

Out[79]:
```
Fully Paid                                             33585
Charged Off                                             5653
Does not meet the credit policy. Status:Fully Paid      1988
Does not meet the credit policy. Status:Charged Off      761
Current                                                  513
In Grace Period                                           16
Late (31-120 days)                                        12
Late (16-30 days)                                          5
Default                                                    1
Name: loan_status, dtype: int64
```

Loan status of our DataFrame

In [80]:
```
# plt.figure(figsize=(15,9))
# sns.countplot(x='loan_status',data=df)
# plt.show()


df.loan_status.value_counts()
loan_status_data = pd.DataFrame(df.loan_status.value_counts())
loan_status_data
fig = px.bar(loan_status_data, x=loan_status_data.index, y='loan_status',color=loan_status_data.index,title='Loa
#fig.xticks(rotation=180)
fig.show()
```



Selecting "Fully Paid" , "Charged Off" & "Current" loans only

In [81]:
```
fullypaid_data = df[df['loan_status']=='Fully Paid']
fullypaid_data
```

Out[81]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	e
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	NaN	
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	84.33	C	C5	NaN	
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	339.31	C	C1	AIR RESOURCES BOARD	
5	1075269	1311441.0	5000.0	5000.0	5000.0	36 months	7.90%	156.46	A	A4	Veolia Transportaton	
6	1069639	1304742.0	7000.0	7000.0	7000.0	60 months	15.96%	170.08	C	C5	Southern Star Photography	
...	...	...	...	...	...	...	...	...	...	...	...	...
39781	92187	92174.0	2500.0	2500.0	1075.0	36 months	8.07%	78.42	A	A4	FiSite Research	
39782	90665	90607.0	8500.0	8500.0	875.0	36 months	10.28%	275.38	C	C1	Squarewave Solutions, Ltd.	
39783	90395	90390.0	5000.0	5000.0	1325.0	36 months	8.07%	156.84	A	A4	NaN	
39784	90376	89243.0	5000.0	5000.0	650.0	36 months	7.43%	155.38	A	A2	NaN	

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	e
39785	87023	86999.0	7500.0	7500.0	800.0	36 months	13.75%	255.43	E	E2	Evergreen Center	

33585 rows × 51 columns

In [82]:
```
chargedoff_data = df[df['loan_status']=='Charged Off']
chargedoff_data
```

Out[82]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	em
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.83	C	C4	
8	1071795	1306957.0	5600.0	5600.0	5600.0	60 months	21.28%	152.39	F	F2	
9	1071570	1306721.0	5375.0	5375.0	5350.0	60 months	12.69%	121.45	B	B5	Sta
12	1064687	1298717.0	9000.0	9000.0	9000.0	36 months	13.49%	305.38	C	C1	Va. I Conservation/Rec
14	1069057	1303503.0	10000.0	10000.0	10000.0	36 months	10.65%	325.74	B	B2	S
...	...	...	...	...	...	...	...	...	...	...	
39736	118823	118026.0	2500.0	2500.0	675.0	36 months	12.80%	84.00	D	D4	Nebraska Occup T
39737	118533	117783.0	2500.0	2500.0	825.0	36 months	9.64%	80.26	B	B4	AMZ Ma
39738	118523	118519.0	6500.0	6500.0	225.0	36 months	15.01%	225.37	F	F1	Universal Adve
39747	113179	113093.0	1000.0	1000.0	950.0	36 months	10.59%	32.55	C	C2	Invision Power Se
39757	111227	111223.0	20000.0	20000.0	2800.0	36 months	13.43%	678.08	E	E1	Auto motors of

5653 rows × 51 columns

In [83]:
```
currentloan_data = df[df['loan_status']=='Current']
currentloan_data
```

Out[83]:

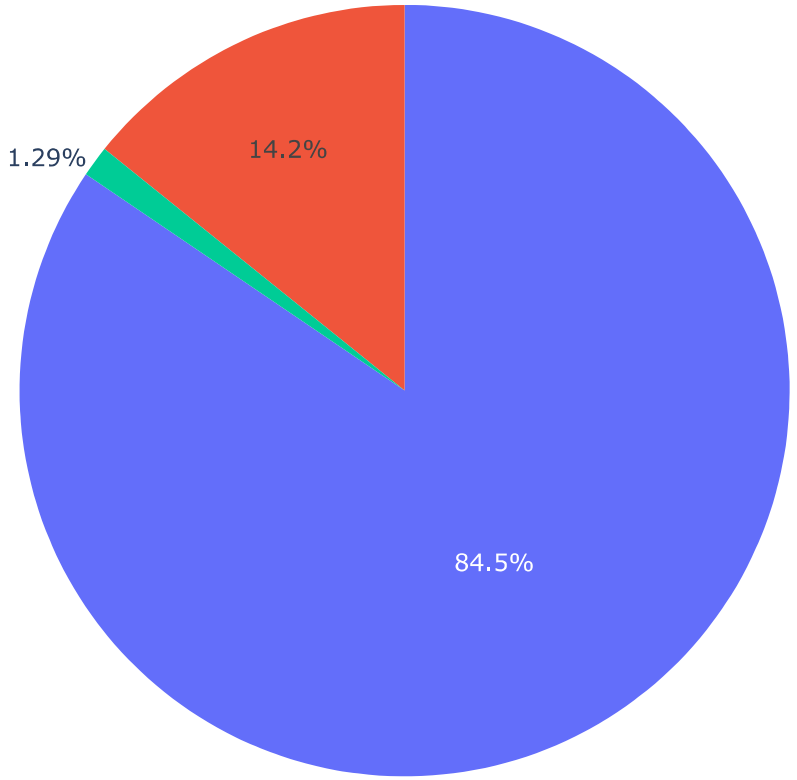
	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp
4	1075358	1311748.0	3000.0	3000.0	3000.00000	60 months	12.69%	67.79	B	B5	University Medical Group	
32	1065420	1299514.0	10000.0	10000.0	9975.00000	60 months	15.96%	242.97	C	C5	US Legal Support	
39	1069346	1304237.0	12500.0	12500.0	12475.00000	60 months	12.69%	282.44	B	B5	United States Infrastructure Corporation	
86	1063958	1297940.0	14000.0	14000.0	13975.00000	60 months	17.27%	349.98	D	D3	community colleges of spokane	
95	1068575	1303001.0	15300.0	15300.0	15275.00000	60 months	22.06%	423.10	F	F4	OSSI	
...	...	...	...	...	...	...	...	...	...	...	...	
8096	871278	1085327.0	16000.0	16000.0	15820.87104	60 months	12.99%	363.97	C	C1	Abt Associates Inc.	
8161	869981	1083884.0	15000.0	12575.0	12575.00000	60 months	15.99%	305.74	D	D2	reed lallier	
8370	863089	1076155.0	30000.0	23100.0	23075.00000	60 months	12.99%	525.48	C	C1	Tasc	

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp
8473	863251	1076373.0	18000.0	17450.0	17450.00000	60 months	16.49%	428.91	D	D3	Infor	
8835	855539	1067857.0	5100.0	5100.0	5100.00000	60 months	11.49%	112.14	B	B4	brightwood	

513 rows × 51 columns

Loan Status - pie chart

In [84]:
```
value = [len(fullypaid_data),len(chargedoff_data),len(currentloan_data)]
names = ['Fully Paid','Charged Off','Current']
fig = px.pie(values=value, names=names)
fig.show()
```



In [ ]:
```

```

Late/Delay Loan Status

In [85]:
```
# Late loan status data for 0 to 30 days period
late_30days = df[df.loan_status == 'Late (16-30 days)']
late_30days
```

Out[85]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_
2799	1031659	1249389.0	5200.0	5200.0	5200.0	60 months	8.90%	107.70	A	A5	NaN	
3056	1026310	1255475.0	7800.0	7800.0	7650.0	60 months	9.91%	165.39	B	B1	colon and digestive health specialists	2
5540	984837	1107578.0	15000.0	15000.0	15000.0	60 months	18.25%	382.95	D	D5	TransFirst	3
8486	863011	1076078.0	10500.0	10500.0	10475.0	60 months	11.49%	230.87	B	B4	morries automotive group	4
8546	864440	1077716.0	35000.0	22525.0	22375.0	60 months	11.49%	495.28	B	B4	united healthcare group	5

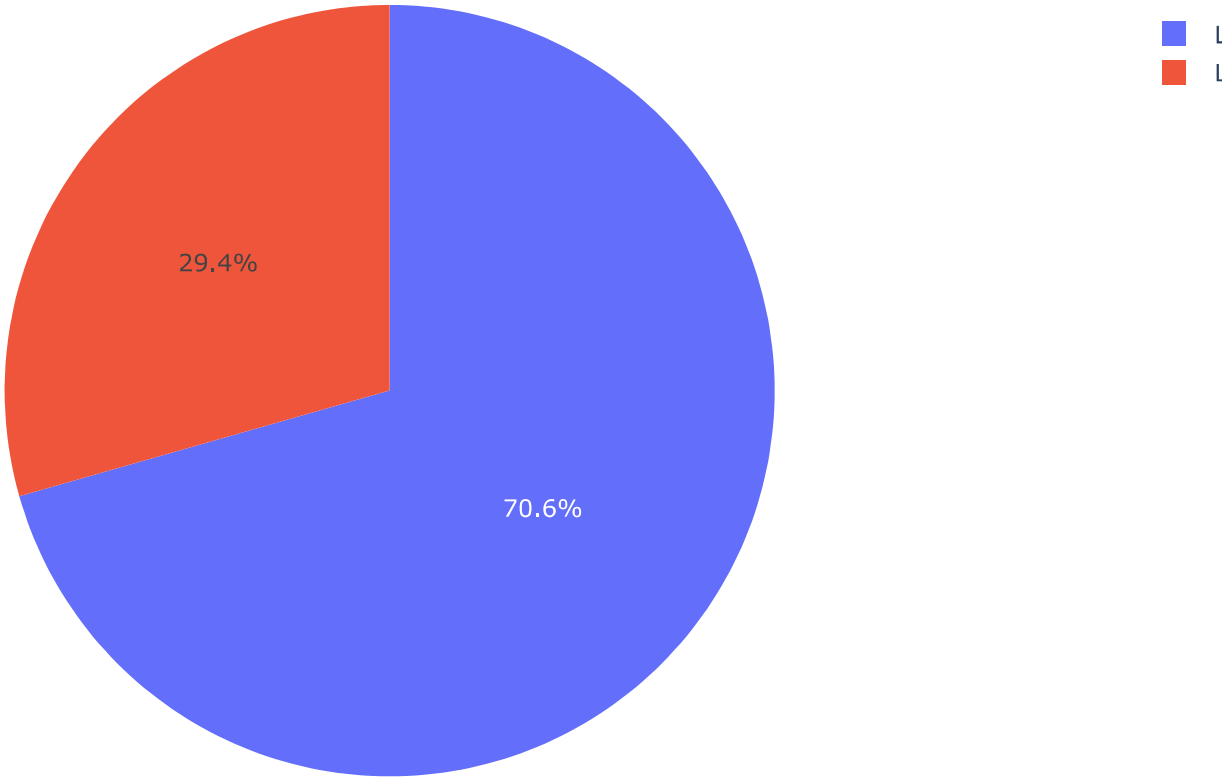
In [86]:
```
# Late loan status data for 30 to 120 days period
late_120days = df[df.loan_status == 'Late (31-120 days)']
late_120days
```

Out[86]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title
488	1063653	1296216.0	25000.0	25000.0	24975.0000	60 months	19.42%	654.31	E	E3	hudson national golf club
1625	1048015	1279149.0	16000.0	16000.0	16000.0000	60 months	12.42%	359.32	B	B4	Canon
2841	1031189	1260595.0	14125.0	14125.0	14100.0000	60 months	19.42%	369.69	E	E3	Department of Defense
3231	1022731	1251722.0	30000.0	30000.0	29725.0000	60 months	11.71%	662.95	B	B3	Clear Channel Communication
3285	1015398	1242849.0	12000.0	12000.0	12000.0000	60 months	15.96%	291.57	C	C5	US Air Force Research Laboratory
6366	967338	1188265.0	12000.0	12000.0	11750.0000	60 months	12.42%	269.49	B	B4	walmart
10923	822971	1031499.0	1500.0	1500.0	1500.0000	60 months	16.49%	36.87	D	D3	Hilton Anatole
11051	821938	1030335.0	13600.0	13600.0	13600.0000	60 months	16.89%	337.20	D	D4	Acapulco
11891	788059	991688.0	19750.0	19750.0	19297.9479	60 months	19.69%	519.86	E	E5	Connetquot Central School District

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title
11951	805996	1012150.0	33950.0	22975.0	22925.0000	60 months	21.74%	631.16	F	F5	Telstra Inc.
12100	803608	1009390.0	12000.0	12000.0	11950.0000	60 months	20.99%	324.58	F	F3	CBP
14058	771984	973816.0	35000.0	35000.0	34950.0000	60 months	15.23%	836.88	C	C5	US Postal Service

In [87]:
```
value = [len(late_30days),len(late_120days)]
names = ['Late (16-30 days)','Late (31-120 days)']
fig = px.pie(values=value, names=names)
fig.show()
```



In [88]:
```
df.head(1)
```

Out[88]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_lengt
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	NaN	10+ yea

Maximum Loan amount -> 35000.0

In [89]:
```
df['loan_amnt'].max()
```

Out[89]:
```
35000.0
```

Minimum Loan amount -> 500.0

In [90]:
```
df['loan_amnt'].min()
```

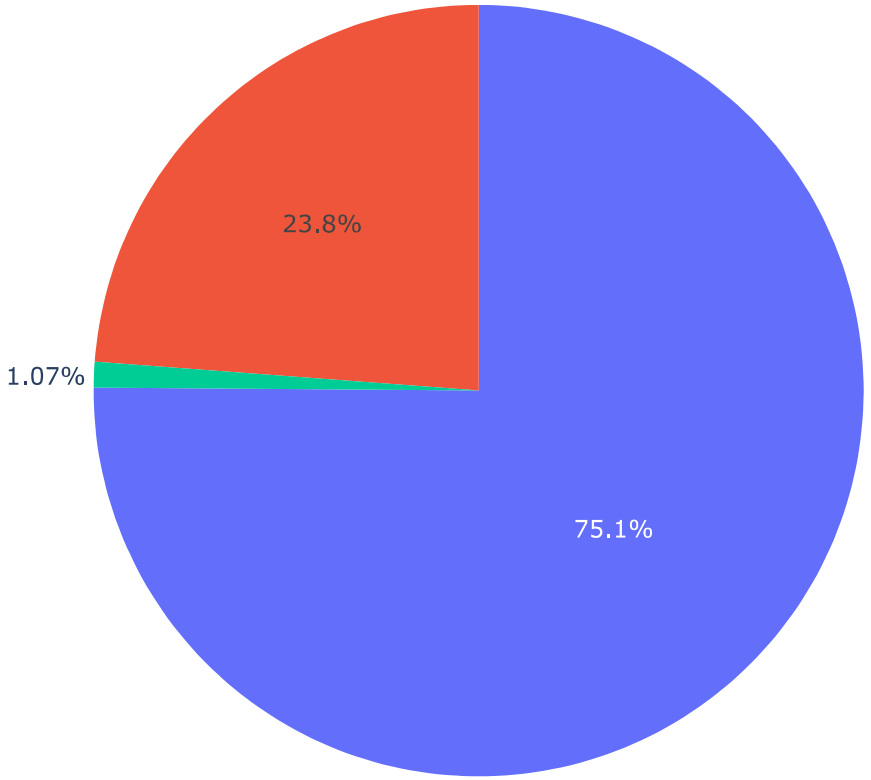
Out[90]:
```
500.0
```

Average Loan amount -> 11089.74820143885

In [91]:
```
df['loan_amnt'].mean()
```

Out[91]:
```
11089.74820143885
```

In [92]:
```
value = [df['loan_amnt'].max(),df['loan_amnt'].min(),df['loan_amnt'].mean()]
names = ['Max','Min','Avg']
fig = px.pie(values=value, names=names)
fig.show()
```



Statewise Avg. Loan amount

In [93]:
```
df.addr_state.value_counts()
```

Out[93]:
```
CA    7428
NY    4065
FL    3104
TX    2915
NJ    1988
IL    1672
PA    1651
GA    1503
VA    1487
MA    1438
OH    1329
MD    1125
AZ     933
WA     888
CO     857
NC     830
CT     816
MI     796
MO     765
MN     652
NV     527
WI     516
SC     489
AL     484
OR     468
LA     461
KY     359
OK     317
KS     298
UT     278
AR     261
DC     224
RI     208
NM     205
NH     188
WV     187
HI     181
DE     136
MT      96
WY      87
AK      86
```

```
SD      67
VT      57
TN      32
MS      26
IN      19
IA      12
NE      11
ID       9
ME       3
Name: addr_state, dtype: int64
```

In [94]:
```
statewise_avg_loan_amt = df['loan_amnt'].groupby(df['addr_state']).mean()
statewise_avg_loan_amt
# addr_state_data = pd.DataFrame(df.addr_state.value_counts())
# addr_state_data
```

Out[94]:
```
addr_state
AK    12913.953488
AL    11173.295455
AR    10369.827586
AZ    10889.040729
CA    11283.774233
CO    11190.985998
CT    11320.955882
DC    12329.575893
DE    10181.801471
FL    10597.857603
GA    11189.587492
HI    10817.265193
IA     8345.833333
ID     8038.888889
IL    11380.786483
IN     7582.894737
KS    10666.778523
KY    10850.278552
LA    10380.585683
MA    11244.297636
MD    11365.333333
ME     3066.666667
MI    10997.456030
MN    10394.900307
MO    10413.267974
MS     7325.000000
MT    10496.093750
NC    11265.512048
NE     8086.363636
NH    11615.691489
NJ    11614.097082
NM    10410.487805
NV    10771.299810
NY    11161.469865
OH    10697.498119
OK    11228.154574
OR    10591.079060
```

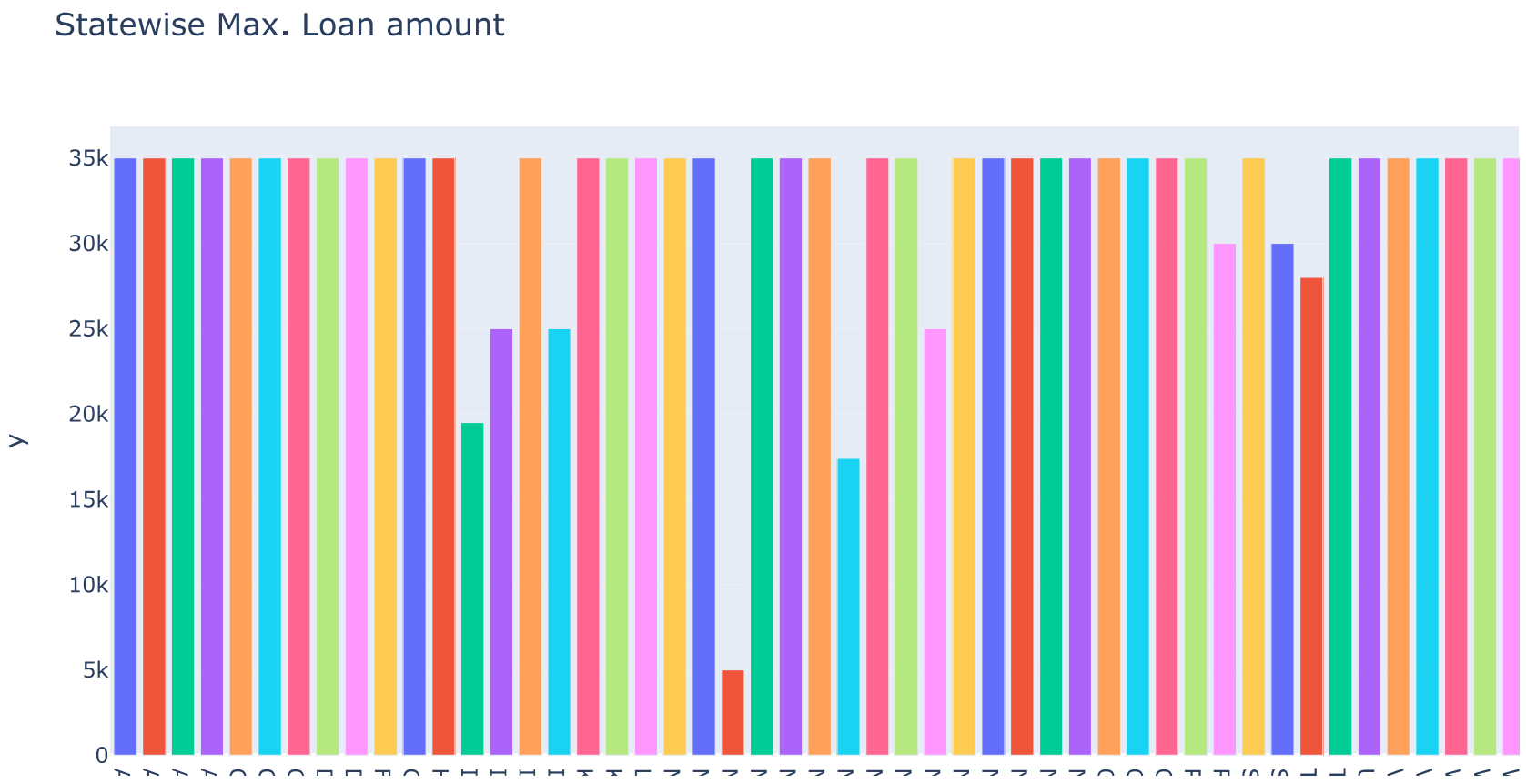
```
PA    10500.348274
RI     9562.259615
SC    10895.092025
SD     9761.567164
TN     7904.687500
TX    11578.773585
UT    11099.910072
VA    11479.236718
VT     9021.052632
WA    10893.158784
WI    11102.034884
WV    10701.336898
WY    11129.310345
Name: loan_amnt, dtype: float64
```

In [95]:
```
# statewise_avg_loan_amt
```

In [96]:
```
fig = px.bar(statewise_avg_loan_amt, x=statewise_avg_loan_amt.index, y=statewise_avg_loan_amt,color=statewise_av
#fig.xticks(rotation=180)
fig.show()
```

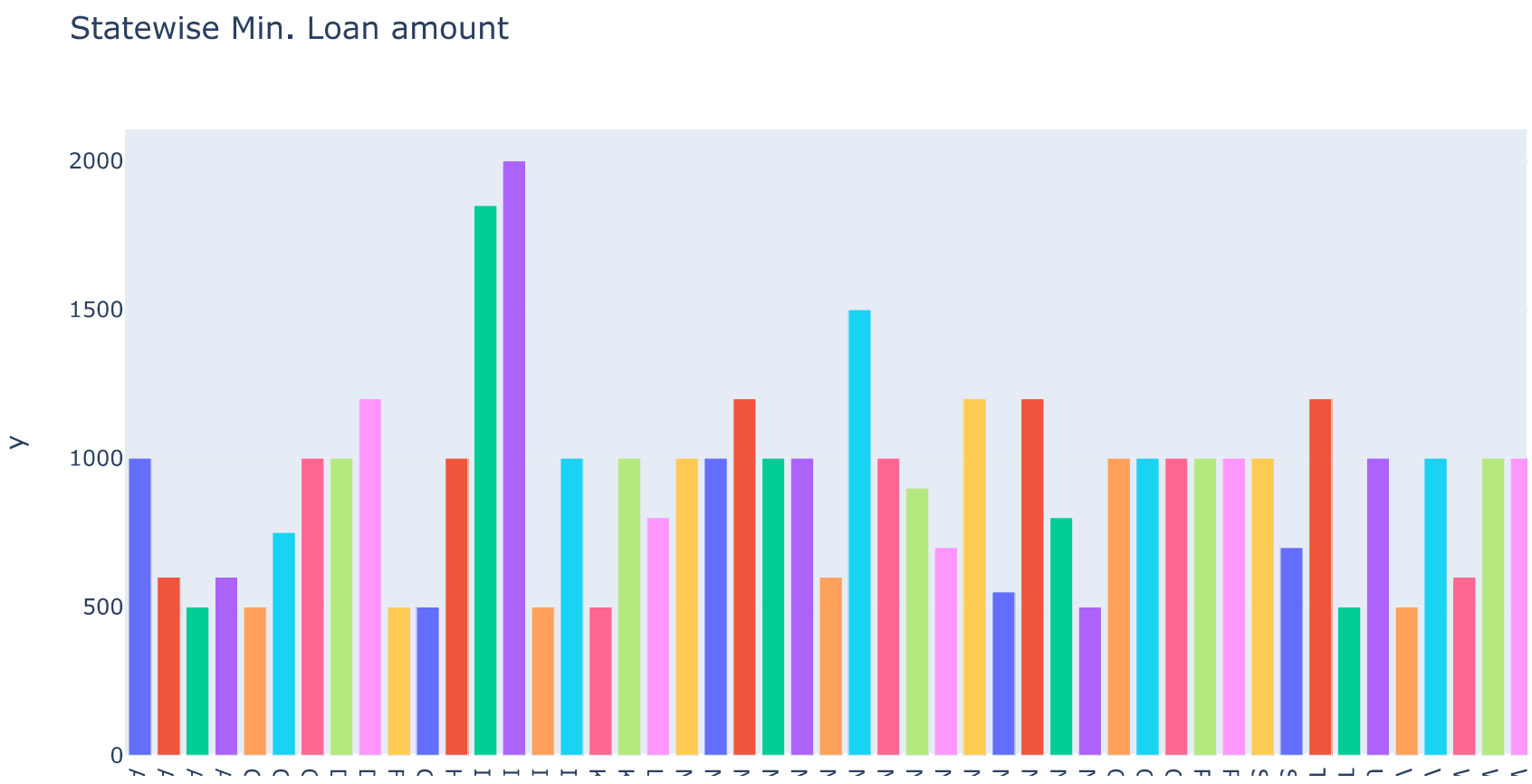
Statewise Max Loan amount

In [97]:
```
statewise_max_loan_amt = df['loan_amnt'].groupby(df['addr_state']).max()
fig = px.bar(statewise_max_loan_amt, x=statewise_max_loan_amt.index, y=statewise_max_loan_amt,color=statewise_ma
#fig.xticks(rotation=180)
fig.show()
```



Statewise Min Loan amount

In [98]:
```
statewise_min_loan_amt = df['loan_amnt'].groupby(df['addr_state']).min()
fig = px.bar(statewise_min_loan_amt, x=statewise_min_loan_amt.index, y=statewise_min_loan_amt,color=statewise_mi
#fig.xticks(rotation=180)
fig.show()
```



In [99]:
```
df.columns
```

Out[99]:
```
Index(['id', 'member_id', 'loan_amnt', 'funded_amnt', 'funded_amnt_inv',
       'term', 'int_rate', 'installment', 'grade', 'sub_grade', 'emp_title',
       'emp_length', 'home_ownership', 'annual_inc', 'verification_status',
       'issue_d', 'loan_status', 'url', 'desc', 'purpose', 'addr_state', 'dti',
       'delinq_2yrs', 'earliest_cr_line', 'fico_range_low', 'fico_range_high',
       'inq_last_6mths', 'open_acc', 'pub_rec', 'revol_bal', 'revol_util',
       'total_acc', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp',
       'total_rec_int', 'total_rec_late_fee', 'recoveries', 'last_pymnt_d',
       'last_pymnt_amnt', 'last_credit_pull_d', 'last_fico_range_high',
       'last_fico_range_low', 'pub_rec_bankruptcies', 'earliest_cr_line_year',
       'issue_d_year', 'issue_d_month', 'last_pymnt_d_year',
       'last_pymnt_d_month', 'last_credit_pull_d_year',
       'last_credit_pull_d_month'],
      dtype='object')
```

Maximum Funded amount -> 35000.0

In [100]:
```
df['funded_amnt'].max()
```

Out[100]:
```
35000.0
```

Minimum Funded amount -> 500.0

In [101]:
```
df['funded_amnt'].min()
```

Out[101]:
```
500.0
```

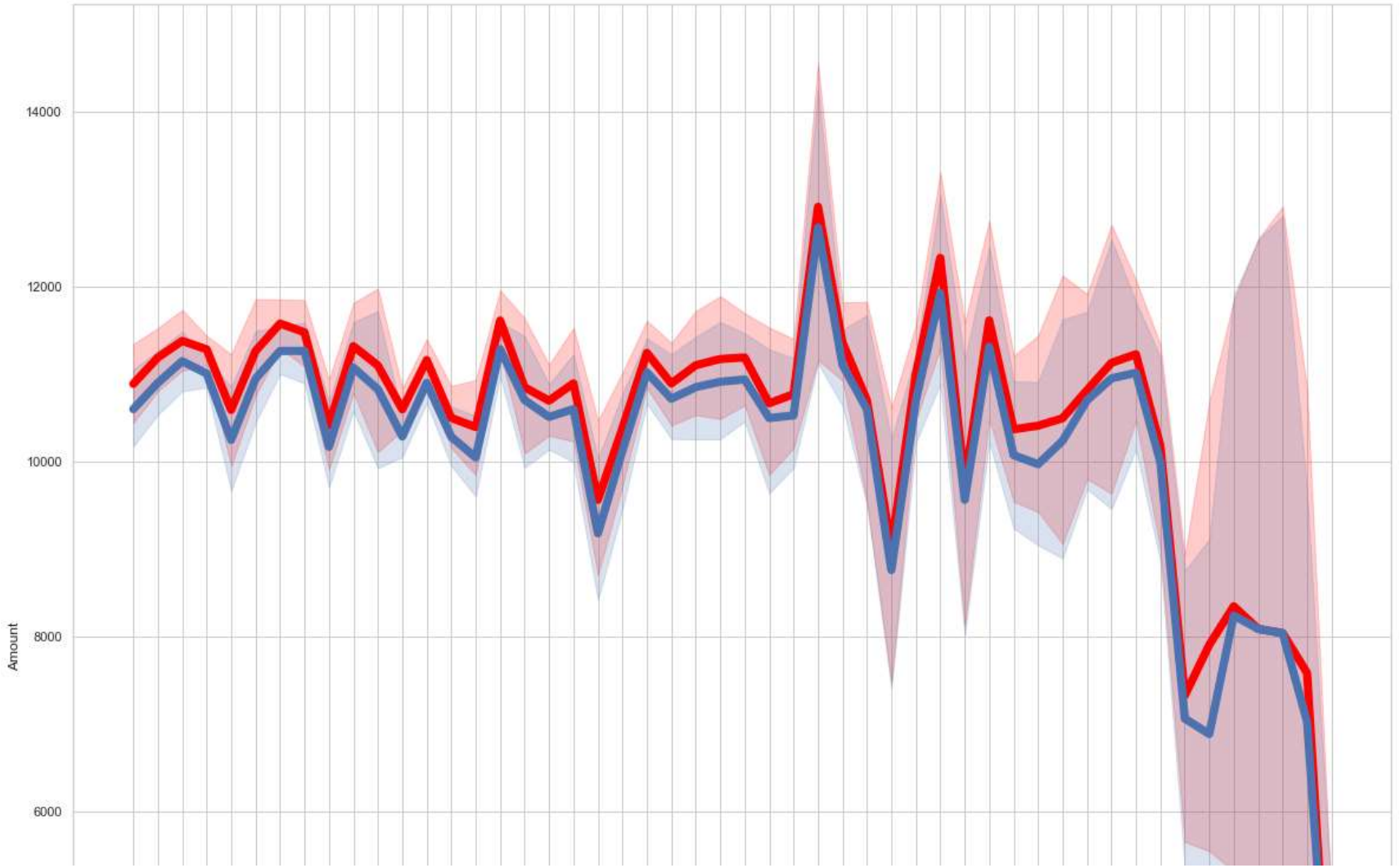
Average Funded amount -> 10821.605068886067

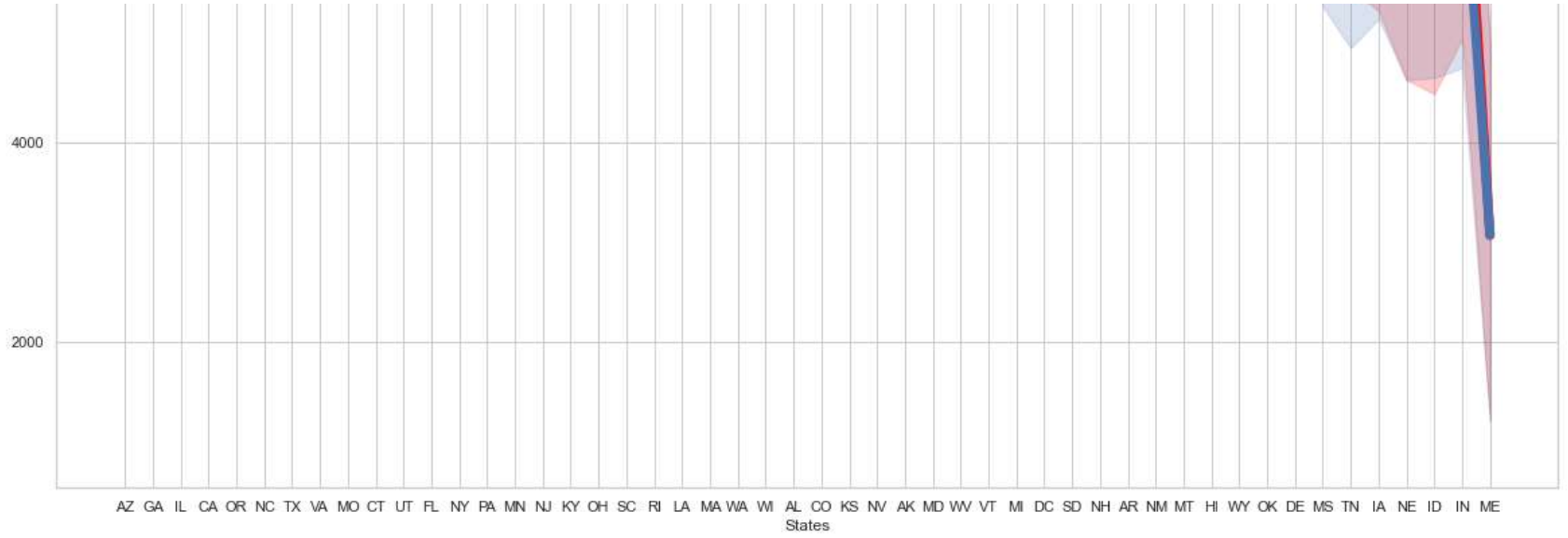
In [102]:
```
df['funded_amnt'].mean()
```

Out[102]:
```
10821.605068886067
```

Comparing Statewise Loan amount and Funded amount

In [103]:
```
# px.line(x=df['addr_state'], y=df['loan_amnt'])
# fig.show()

sns.lineplot(x=df['addr_state'], y=df['loan_amnt'],lw=7,color='red')
sns.lineplot(x=df['addr_state'], y=df['funded_amnt'],lw=7)
plt.xlabel('States')
plt.ylabel('Amount')
#ax.legend()
#plt.legend(labels=["Legend_Day1","Legend_Day2"], title = "Title_Legend")
#plt.legend(labels=['loan_amnt','funded_amnt'])
plt.show()
```





In [104]:
```
# fig = px.line(df,x='addr_state', y='loan_amnt')
# fig.show()
```

Maximum Funded amount -> 35000.0

In [105]:
```
df['funded_amnt_inv'].max()
```

Out[105]: 35000.0

Minimum Funded amount -> 0.0

In [106]:
```
df['funded_amnt_inv'].min()
```

Out[106]: 0.0

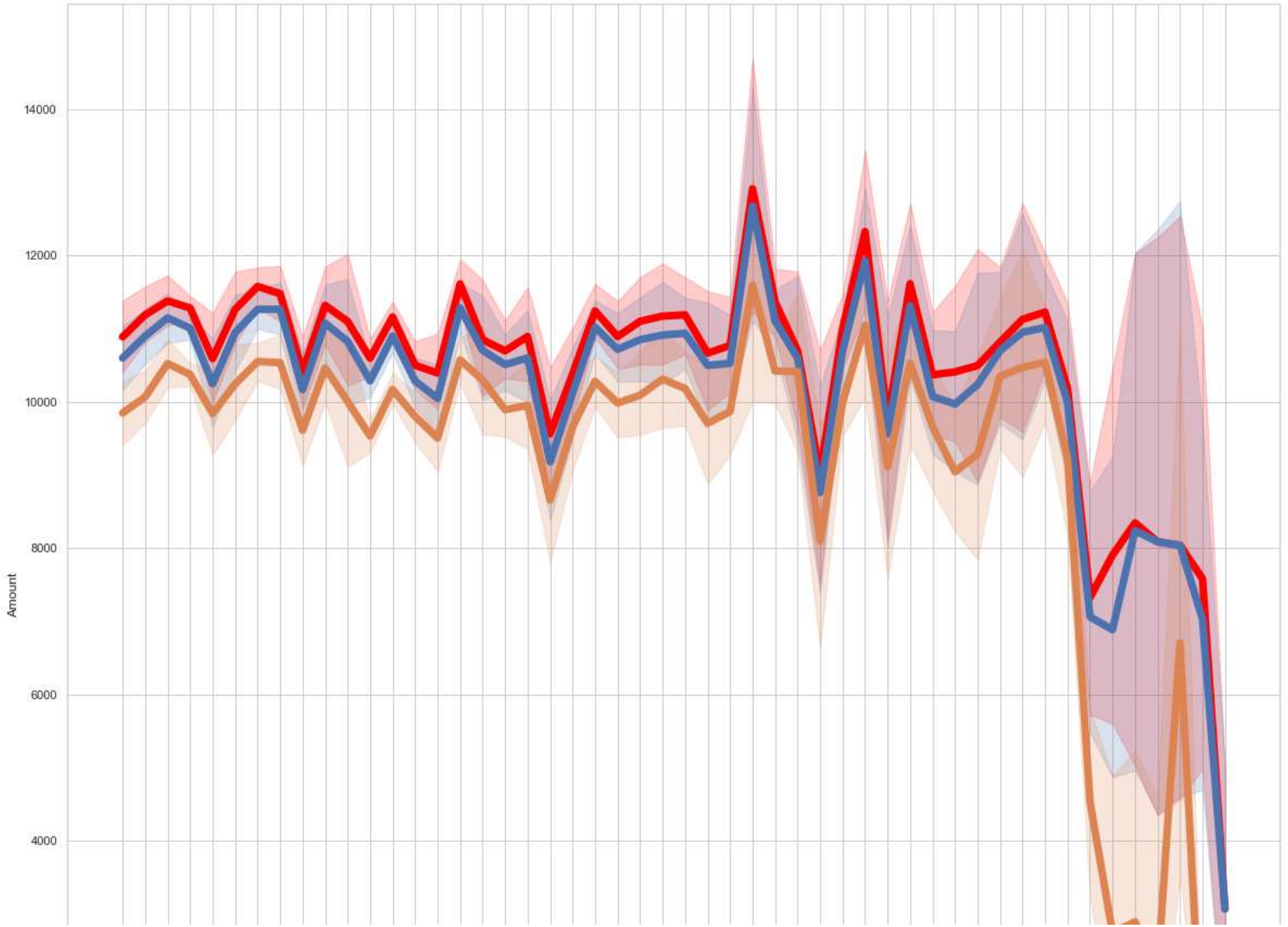
Average Funded amount -> 10139.913992908563

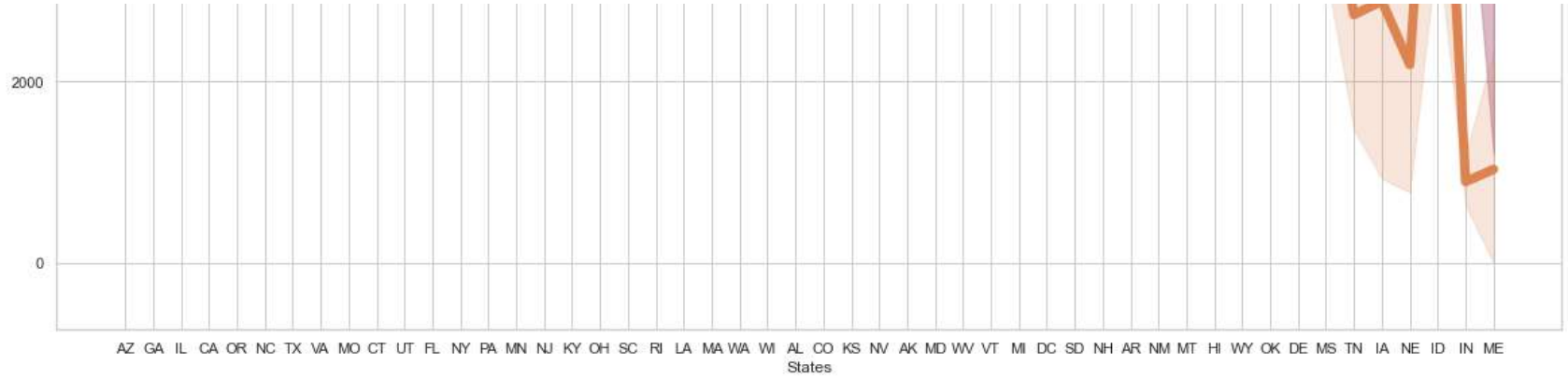
In [107]:
```
df['funded_amnt_inv'].mean()
```

Out[107]:
```
10139.913992908563
```

Comparing Statewise Loan amount, Funded amount & Funded amount Invoice

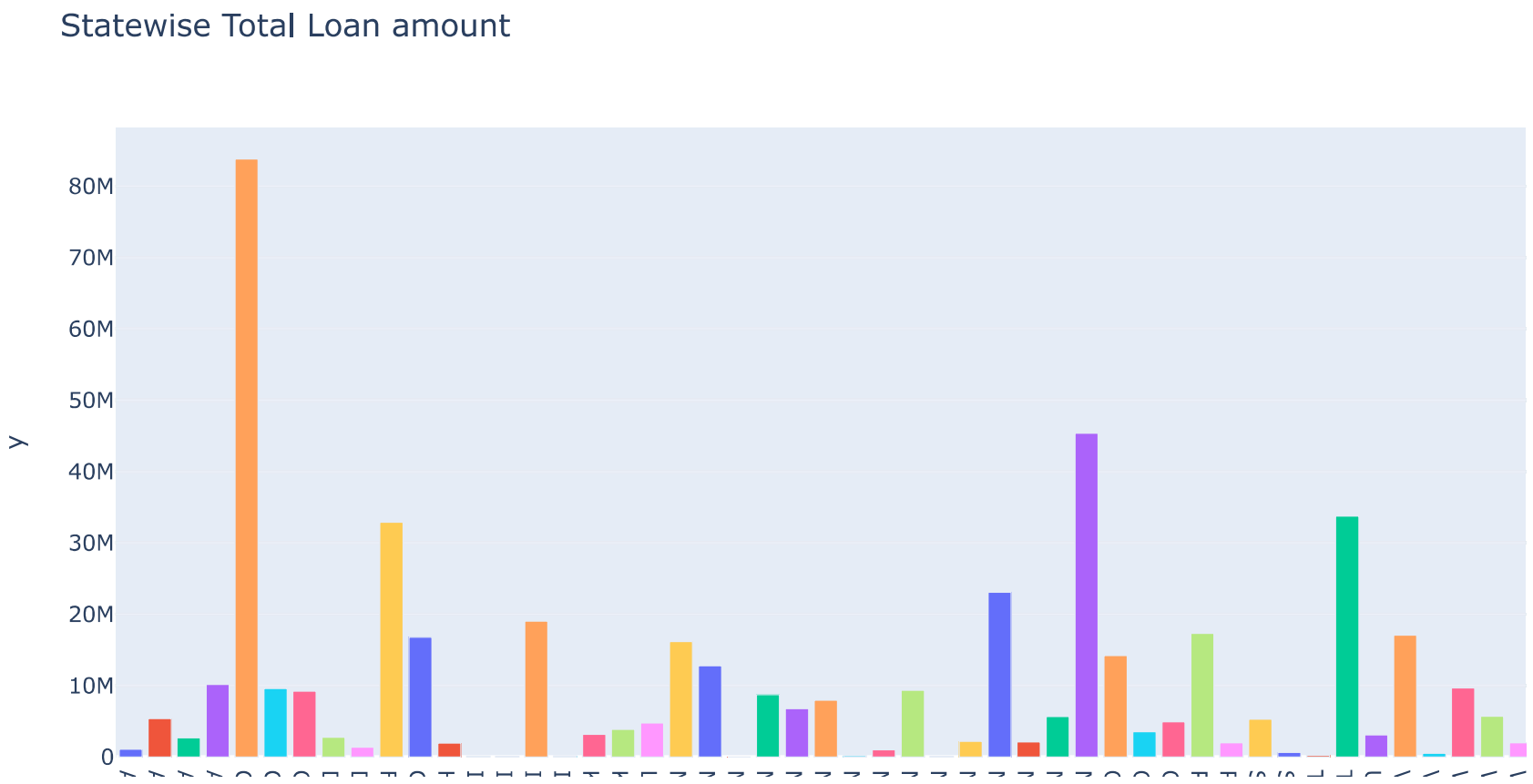
In [108]:
```
sns.lineplot(x=df['addr_state'], y=df['loan_amnt'],lw=7,color='red')
sns.lineplot(x=df['addr_state'], y=df['funded_amnt'],lw=7)
sns.lineplot(x=df['addr_state'], y=df['funded_amnt_inv'],lw=7)
plt.xlabel('States')
plt.ylabel('Amount')
plt.show()
```





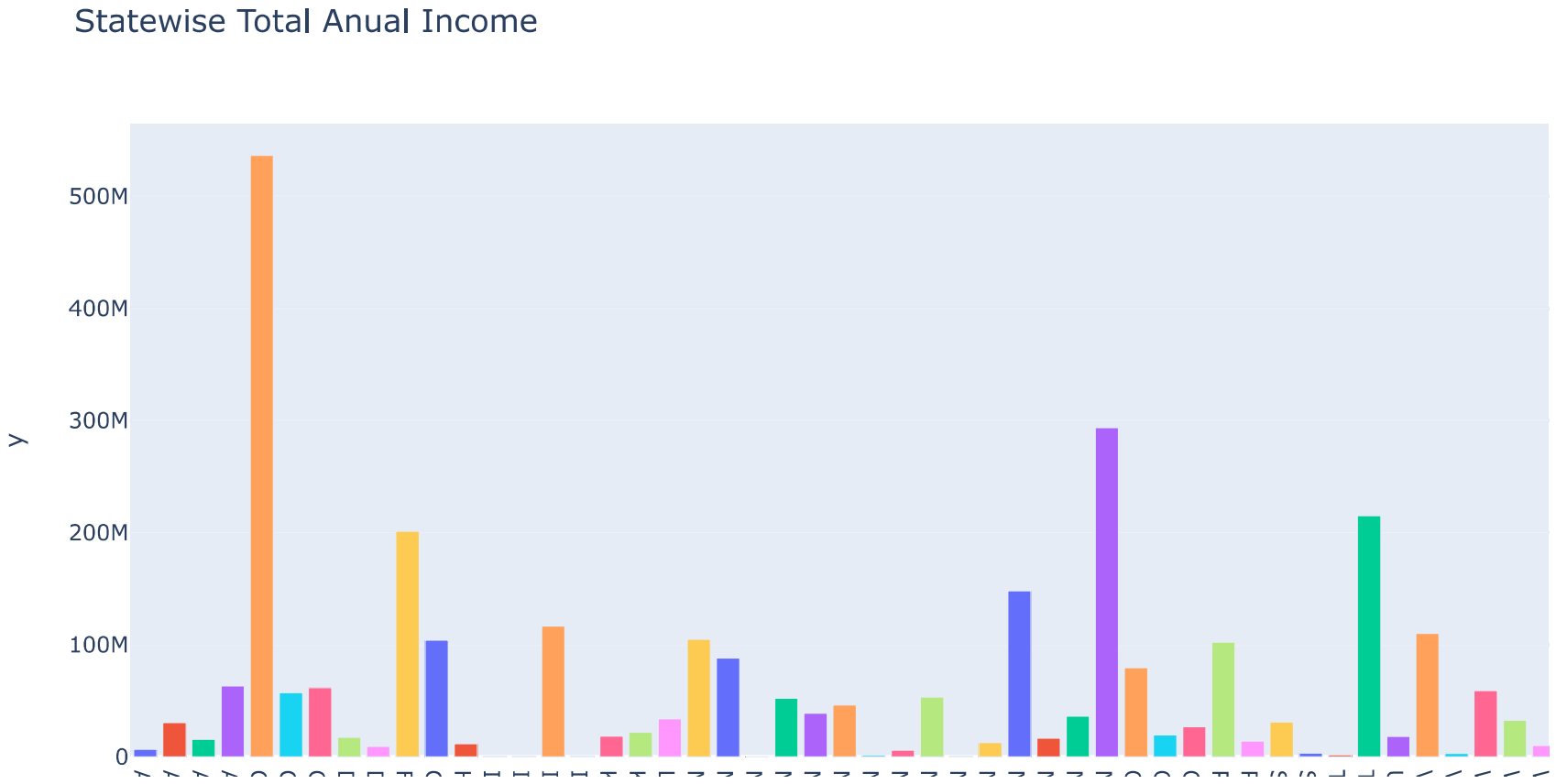
Statewise Total Loan Amount

In [109]:
```
statewise_total_loan_amt = df['loan_amnt'].groupby(df['addr_state']).sum()
fig = px.bar(statewise_total_loan_amt, x=statewise_total_loan_amt.index, y=statewise_total_loan_amt,color=statew
fig.show()
```



Statewise Total Anual Income

In [110]:
```
statewise_total_anual_loan = df['annual_inc'].groupby(df['addr_state']).sum()
fig = px.bar(statewise_total_anual_loan, x=statewise_total_anual_loan.index, y=statewise_total_anual_loan,color=
fig.show()
```



In [111]:
```
df.head(1)
```

Out[111]:

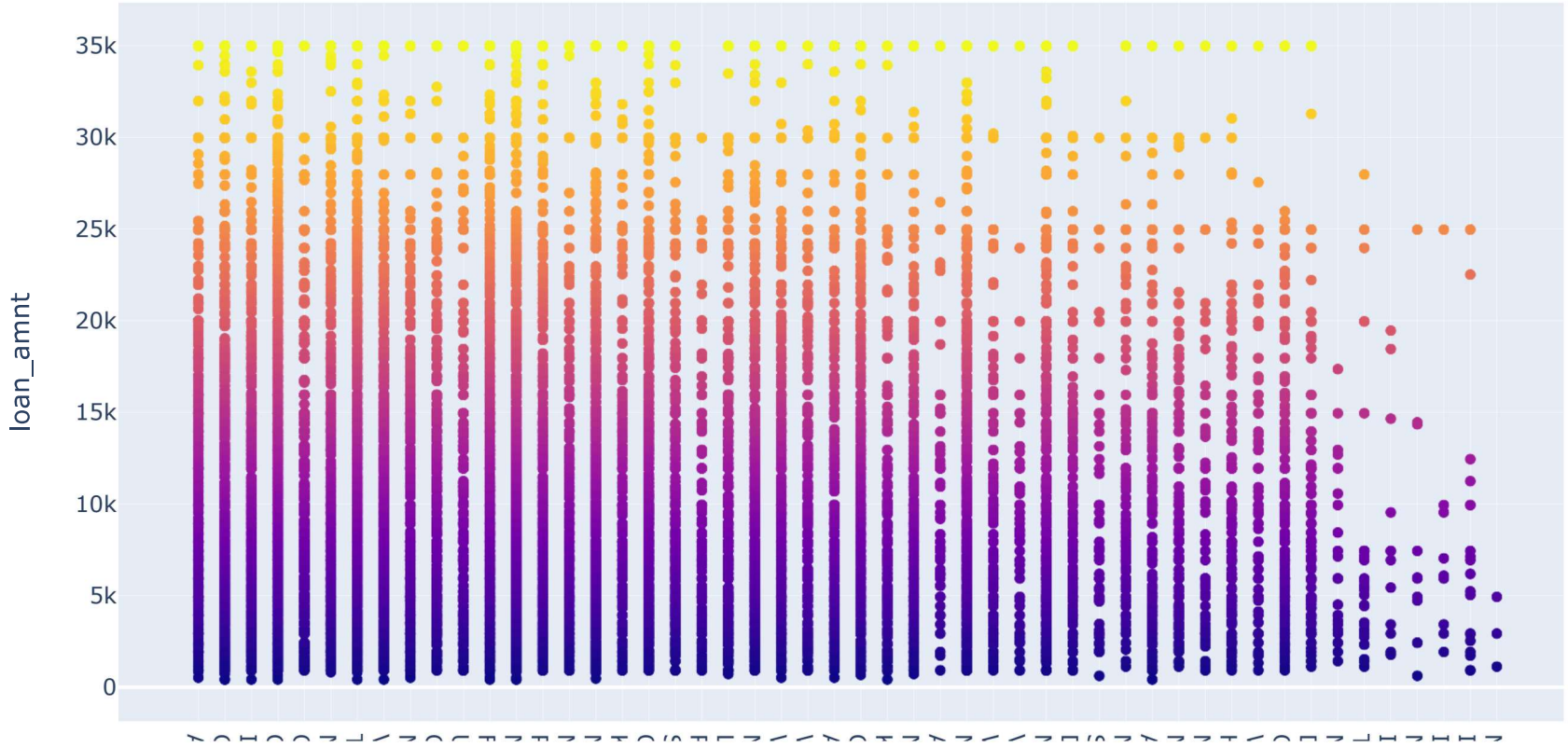
	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_lengt
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	NaN	10+ yea

Anual Income vs Loan amount

In [112]:
```
sns.lineplot(x=df['addr_state'], y=df['loan_amnt'],lw=7,color='red')
sns.lineplot(x=df['addr_state'], y=df['annual_inc'],lw=7,color='green')
plt.xlabel('States')
plt.ylabel('Amount')
plt.show()
```


20000
0
AZ GA IL CA OR NC TX VA MO CT UT FL NY PA MN NJ KY OH SC RI LA MA WA WI AL CO KS NV AK MD WV VT MI DC SD NH AR NM MT HI WY OK DE MS TN IA NE ID IN ME
States

In [113]:
```
fig = px.scatter(df,x='addr_state', y='loan_amnt',color='loan_amnt')
fig.show()
```



In [114]:
```
# #sns.barplot(x=df['loan_status'], y=df['loan_amnt'],lw=7)
# fig = px.bar(df, x=loan_status, y=loan_amnt,color=loan_status,title='Loan Status vs Loan amount')
# fig.show()
```

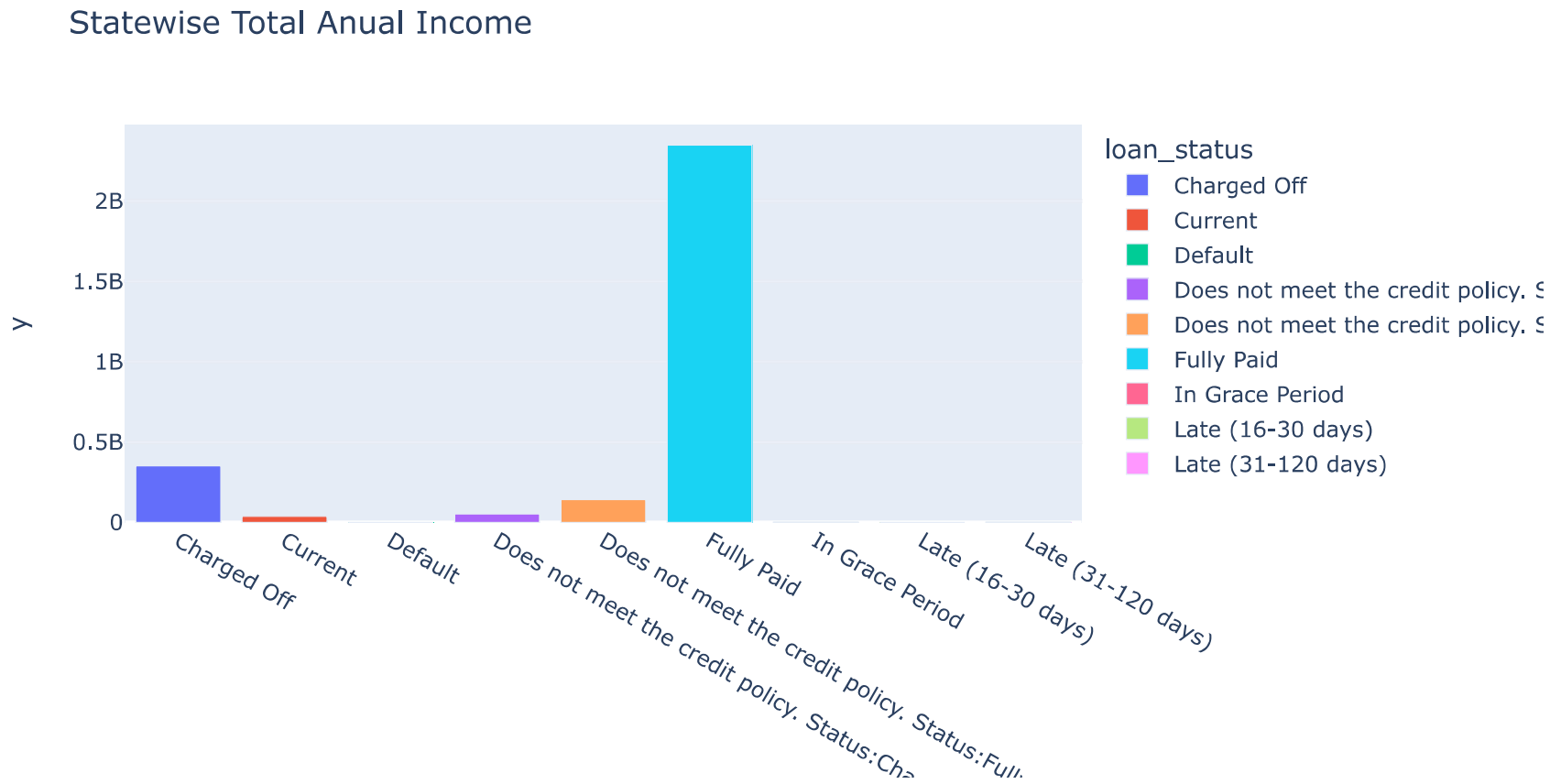
In [115]:
```
# n=len(df['loan_amnt'])
# r = np.arange(n)
# width = 0.25
# plt.bar(r, df['loan_amnt'], color = 'b',
#         width = width, edgecolor = 'black',
#         label='Loan Amount')
# plt.bar(r + width, df['annual_inc'], color = 'g',
#         width = width, edgecolor = 'black',
#         label='Anual Income')

# plt.xlabel("Year")
# plt.ylabel("Number of people voted")
# plt.title("Number of people voted in each year")

# plt.grid(linestyle='--')
# #plt.xticks(r + width/2,['2018','2019','2020','2021'])
# plt.legend()
# plt.show()
```

In [116]:
```
anual_inc_loan_status = df['annual_inc'].groupby(df['loan_status']).sum()
```

In [117]:
```
anual_inc_loan_status
```

Out[117]:
```
loan_status
Charged Off                                         3.532143e+08
Current                                             3.881916e+07
Default                                             7.000000e+04
Does not meet the credit policy. Status:Charged Off 5.290922e+07
Does not meet the credit policy. Status:Fully Paid  1.431365e+08
Fully Paid                                          2.349692e+09
In Grace Period                                     1.109582e+06
Late (16-30 days)                                   4.375200e+05
Late (31-120 days)                                  1.012920e+06
Name: annual_inc, dtype: float64
```

In [118]:
```
fig = px.bar(anual_inc_loan_status, x=anual_inc_loan_status.index, y=anual_inc_loan_status,color=anual_inc_loan_
fig.show()
```



In [119]:
```
df['addr_state'].value_counts()
#canada have maximum no of data
```

Out[119]:
```
CA    7428
NY    4065
FL    3104
TX    2915
NJ    1988
IL    1672
PA    1651
GA    1503
VA    1487
MA    1438
OH    1329
MD    1125
AZ     933
WA     888
CO     857
NC     830
CT     816
MI     796
MO     765
MN     652
NV     527
WI     516
SC     489
AL     484
OR     468
LA     461
KY     359
OK     317
KS     298
UT     278
AR     261
DC     224
RI     208
NM     205
NH     188
WV     187
HI     181
DE     136
MT      96
WY      87
```

```
AK      86
SD      67
VT      57
TN      32
MS      26
IN      19
IA      12
NE      11
ID       9
ME       3
Name: addr_state, dtype: int64
```

Analyzing Canada data

In [120]:
```
canada_data = df[df['addr_state']=='CA']
```

In [121]:
```
canada_data
```

Out[121]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	e
3	1076863	1277178.0	10000.0	10000.0	10000.000000	36 months	13.49%	339.31	C	C1	AIR RESOURCES BOARD	
7	1072053	1288686.0	3000.0	3000.0	3000.000000	36 months	18.64%	109.43	E	E1	MKC Accounting	
8	1071795	1306957.0	5600.0	5600.0	5600.000000	60 months	21.28%	152.39	F	F2	NaN	
11	1069908	1305008.0	12000.0	12000.0	12000.000000	36 months	12.69%	402.54	B	B5	UCLA	
14	1069057	1303503.0	10000.0	10000.0	10000.000000	36 months	10.65%	325.74	B	B2	SFMTA	
...	...	...	...	...	...	...	...	...	...	...	...	
42164	197214	197210.0	18500.0	18500.0	3300.000000	36 months	17.22%	661.61	G	G3	L.A. County Dept. of Public Works	
42176	193979	190581.0	16000.0	16000.0	4902.640000	36 months	16.28%	564.73	F	F5	Rex Moore Electrical	
42181	192193	191767.0	6000.0	6000.0	1250.000000	36 months	13.12%	202.52	D	D5	Lucas Arts	
42188	188014	188001.0	14550.0	14550.0	925.000000	36 months	13.75%	495.52	E	E2	E & J Hauling	

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	e
42191	187383	128788.0	25000.0	25000.0	3824.993774	36 months	17.86%	902.06	G	G5	James M Ballard LLC	

7428 rows × 51 columns

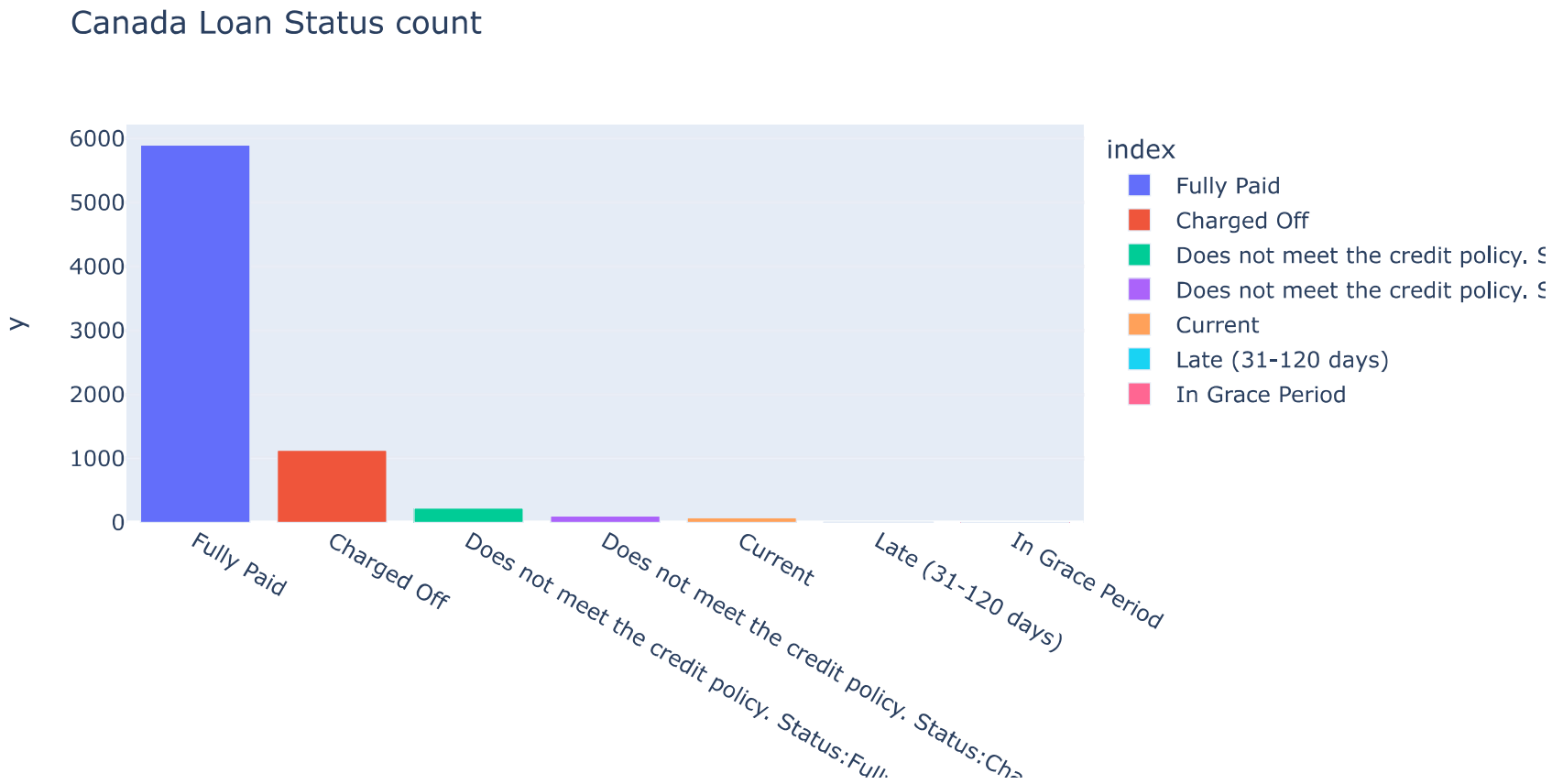
Canada loan status

In [122]:
```
ca_loan_status = canada_data['loan_status'].value_counts()
```

In [123]:
```
ca_loan_status.index
```

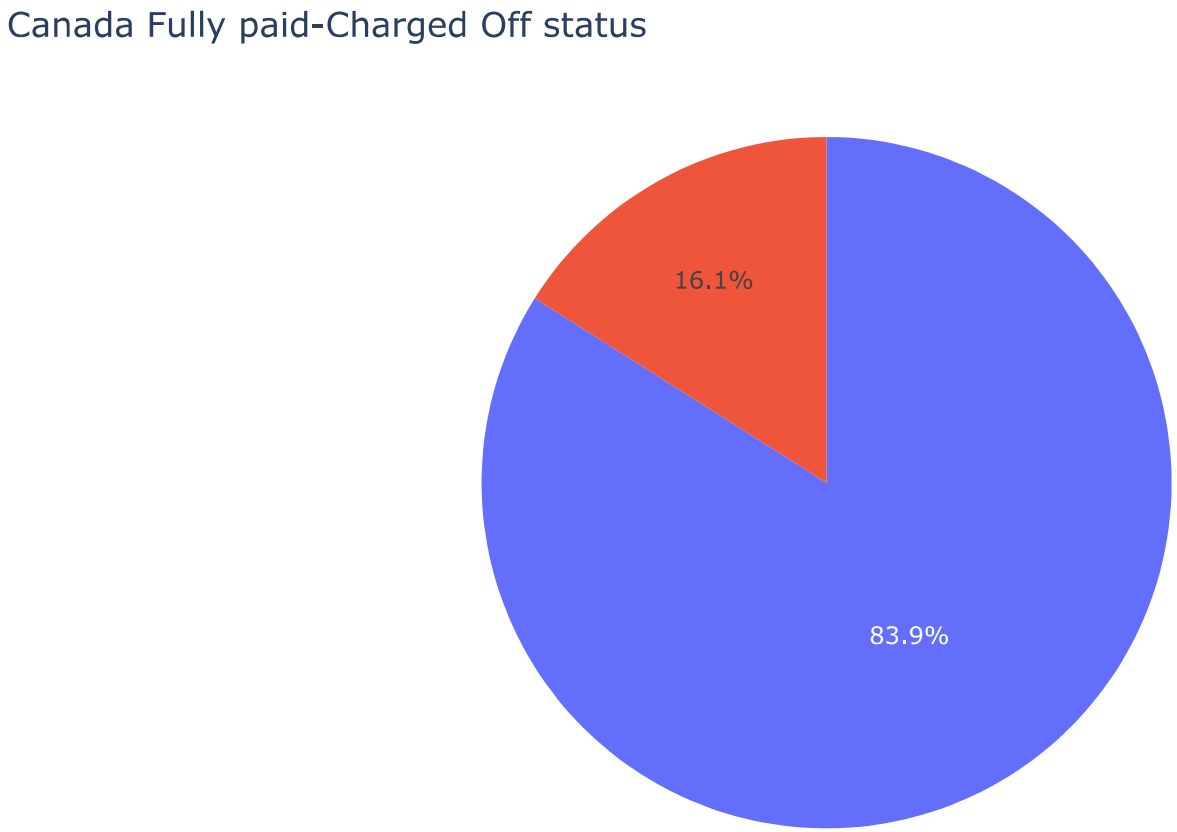
Out[123]:
```
Index(['Fully Paid', 'Charged Off',
       'Does not meet the credit policy. Status:Fully Paid',
       'Does not meet the credit policy. Status:Charged Off', 'Current',
       'Late (31-120 days)', 'In Grace Period'],
      dtype='object')
```

In [124]:
```
fig = px.bar(ca_loan_status, x=ca_loan_status.index, y=ca_loan_status,color=ca_loan_status.index,title='Canada L
fig.show()
```



Canada Fully paid vs Charged off status

In [125]:
```
value = [ca_loan_status[0],ca_loan_status[1]]
names = ['Fully Paid','Charged Off']
fig = px.pie(values=value, names=names,title='Canada Fully paid-Charged Off status')
fig.show()
```



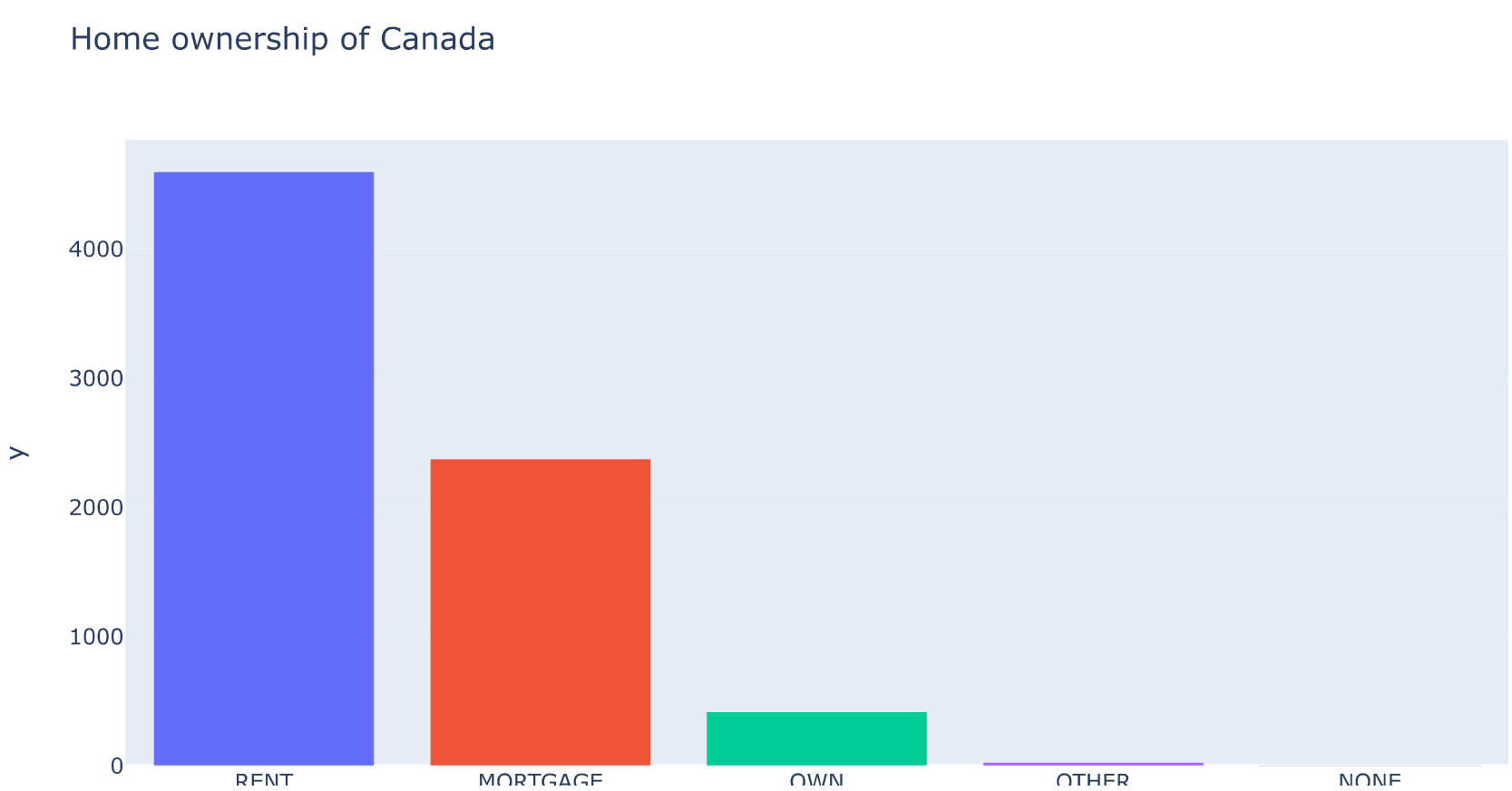
Home ownership count of Canada

In [126]:
```
#Home ownership count of Canada
ca_home_own = canada_data['home_ownership'].value_counts()
```

In [127]:
```
ca_home_own
```

Out[127]:
```
RENT        4601
MORTGAGE    2377
OWN          420
OTHER         29
NONE           1
Name: home_ownership, dtype: int64
```

In [128]:
```
fig = px.bar(ca_home_own, x=ca_home_own.index, y=ca_home_own,color=ca_home_own.index,title='Home ownership of Ca
fig.show()
```



canada ppl who lives in rental place took loans

In [129]:
```
#canada ppl who lives in rental place took loans
rent_home_ca = canada_data[canada_data['home_ownership']=='RENT']
```

In [130]:
```
rent_home_ca.head(3)
```

Out[130]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	339.31	C	C1	AIR RESOURCES BOARD	10
7	1072053	1288686.0	3000.0	3000.0	3000.0	36 months	18.64%	109.43	E	E1	MKC Accounting	
14	1069057	1303503.0	10000.0	10000.0	10000.0	36 months	10.65%	325.74	B	B2	SFMTA	

In [131]:
```
#Rental ppl who fully paid the loan
rent_home_ca_fp=rent_home_ca[rent_home_ca['loan_status']=='Fully Paid']
rent_home_ca_fp
```

Out[131]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title
3	1076863	1277178.0	10000.0	10000.0	10000.00	36 months	13.49%	339.31	C	C1	AIR RESOURCES BOARD
7	1072053	1288686.0	3000.0	3000.0	3000.00	36 months	18.64%	109.43	E	E1	MKC Accounting
16	1065775	1299699.0	10000.0	10000.0	10000.00	36 months	15.27%	347.98	C	C4	Chin's Restaurant
19	1069742	1304855.0	9200.0	9200.0	9200.00	36 months	6.03%	280.01	A	A1	Network Interpreting Service
23	1069700	1304810.0	10000.0	10000.0	10000.00	36 months	11.71%	330.76	B	B3	Wells Fargo Bank
...	...	...	...	...	...	...	...	...	...	...	...
39526	194565	194314.0	12000.0	12000.0	1614.03	36 months	9.33%	383.45	B	B3	Roth Staffing
39529	193452	191101.0	10000.0	10000.0	2696.64	36 months	8.38%	315.12	A	A5	Telasic Communications
39532	192299	190578.0	16000.0	16000.0	780.46	36 months	12.80%	537.57	D	D4	United States Air Force
39534	191730	187320.0	20000.0	20000.0	2055.50	36 months	14.38%	687.25	E	E4	Groovemasters
39541	187464	187430.0	10000.0	10000.0	1673.16	36 months	9.64%	320.99	B	B4	Ink

3656 rows × 51 columns

In [132]:
```
#Rental ppl who fully paid the loan
rent_home_ca_co=rent_home_ca[rent_home_ca['loan_status']=='Charged Off']
rent_home_ca_co
```

Out[132]:

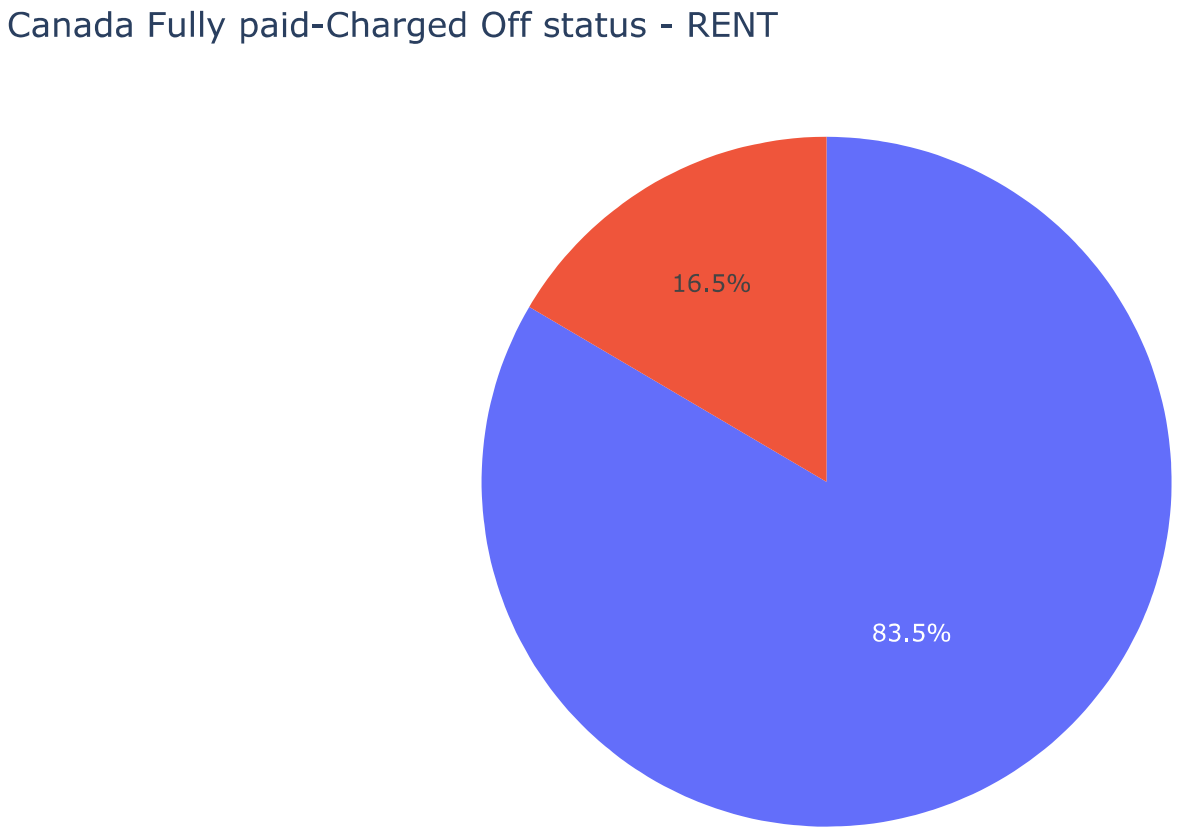
	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp
14	1069057	1303503.0	10000.0	10000.0	10000.000000	36 months	10.65%	325.74	B	B2	SFMTA	
24	1069559	1304634.0	6000.0	6000.0	6000.000000	36 months	11.71%	198.46	B	B3	bmg-educational	
53	1069243	1304116.0	12000.0	12000.0	12000.000000	36 months	15.96%	421.65	C	C5	Chemat Technology Inc	
68	1068906	1303528.0	8200.0	8200.0	8200.000000	60 months	21.28%	223.14	F	F2	autozone	
71	1060981	1292558.0	6400.0	6400.0	6400.000000	36 months	16.77%	227.45	D	D2	Riverside County, California	
...	...	...	...	...	...	...	...	...	...	...	...	
39458	213732	209340.0	20000.0	20000.0	575.001865	36 months	12.99%	673.79	D	D3	California Pizza Kitchen	
39501	200805	200788.0	5000.0	5000.0	1199.999616	36 months	13.93%	170.72	E	E1	First Federal Bank of California	
39519	196428	189204.0	4725.0	4725.0	1049.988203	36 months	11.22%	155.19	C	C4	Mel Cotton's	
39540	187671	187665.0	25000.0	25000.0	1150.003931	36 months	14.70%	862.97	E	E5	Sundance Technology, Inc	
39667	140561	140555.0	2225.0	2225.0	225.000000	36 months	10.59%	72.42	C	C2	Sanctuary	

725 rows × 51 columns

In [133]:
```
len(rent_home_ca_fp)
```

Out[133]: 3656

In [134]:
```
value = [len(rent_home_ca_fp),len(rent_home_ca_co)]
names = ['Fully Paid','Charged Off']
fig = px.pie(values=value, names=names,title='Canada Fully paid-Charged Off status - RENT ')
fig.show()
```



Canada fully paid & charged off data

In [135]:
```
#Canada Fully paid data
ca_fullypaid= canada_data[canada_data['loan_status']=='Fully Paid']
ca_fullypaid.head(2)
```

Out[135]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_l
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	339.31	C	C1	AIR RESOURCES BOARD	10+
7	1072053	1288686.0	3000.0	3000.0	3000.0	36 months	18.64%	109.43	E	E1	MKC Accounting	9

In [136]:
```
#Canada Charged Off data
ca_chargedoff = canada_data[canada_data['loan_status']=='Charged Off']
ca_chargedoff.head(2)
```

Out[136]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_leng
8	1071795	1306957.0	5600.0	5600.0	5600.0	60 months	21.28%	152.39	F	F2	NaN	4 ye
14	1069057	1303503.0	10000.0	10000.0	10000.0	36 months	10.65%	325.74	B	B2	SFMTA	3 ye

Canada fully paid -> Home ownership -> Rent data

In [137]:
```
ca_fullypaid_rent = ca_fullypaid[ca_fullypaid['home_ownership']=='RENT']
ca_fullypaid_rent.head(2)
```

Out[137]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_l
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	339.31	C	C1	AIR RESOURCES BOARD	10+
7	1072053	1288686.0	3000.0	3000.0	3000.0	36 months	18.64%	109.43	E	E1	MKC Accounting	9

Canada fully paid -> Home ownership -> Own data

In [138]:
```
ca_fullypaid_own = ca_fullypaid[ca_fullypaid['home_ownership']=='OWN']
ca_fullypaid_own.head(2)
```

Out[138]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_le
11	1069908	1305008.0	12000.0	12000.0	12000.0	36 months	12.69%	402.54	B	B5	UCLA	10+ y
363	1060082	1291917.0	25000.0	25000.0	25000.0	36 months	15.27%	869.95	C	C4	Chaffey Joint Union High School District	8 y

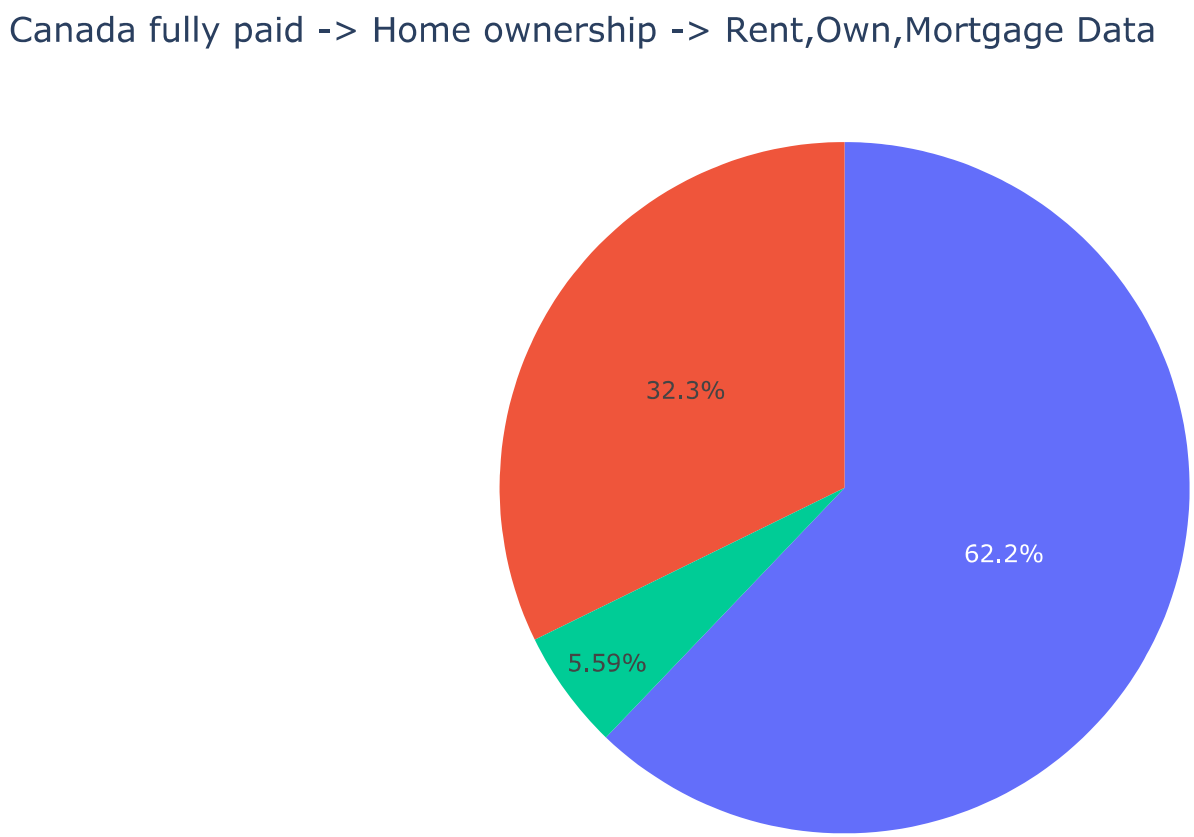
Canada fully paid -> Home ownership -> Mortgage Data

In [139]:
```
ca_fullypaid_mortgage = ca_fullypaid[ca_fullypaid['home_ownership']=='MORTGAGE']
ca_fullypaid_mortgage.head(2)
```

Out[139]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_le
128	1068159	1302729.0	35000.0	35000.0	35000.0	36 months	8.90%	1111.37	A	A5	City of Los Angeles	6 y
138	1067655	1302044.0	10000.0	10000.0	10000.0	60 months	12.69%	225.95	B	B5	Barnes and Noble	2 y

In [140]:
```
value = [len(ca_fullypaid_rent),len(ca_fullypaid_own),len(ca_fullypaid_mortgage)]
names = ['Rent','Own','Mortgage']
fig = px.pie(values=value, names=names,title='Canada fully paid -> Home ownership -> Rent,Own,Mortgage Data')
fig.show()
```



Average anual income who lives in RENT & loan status is Fully paid

In [141]:
```
ca_fullypaid_rent['annual_inc'].mean()
```

Out[141]:
```
61827.76807986871
```

Maximum anual income who lives in RENT & loan status is Fully paid

In [142]:
```
ca_fullypaid_rent['annual_inc'].max()
```

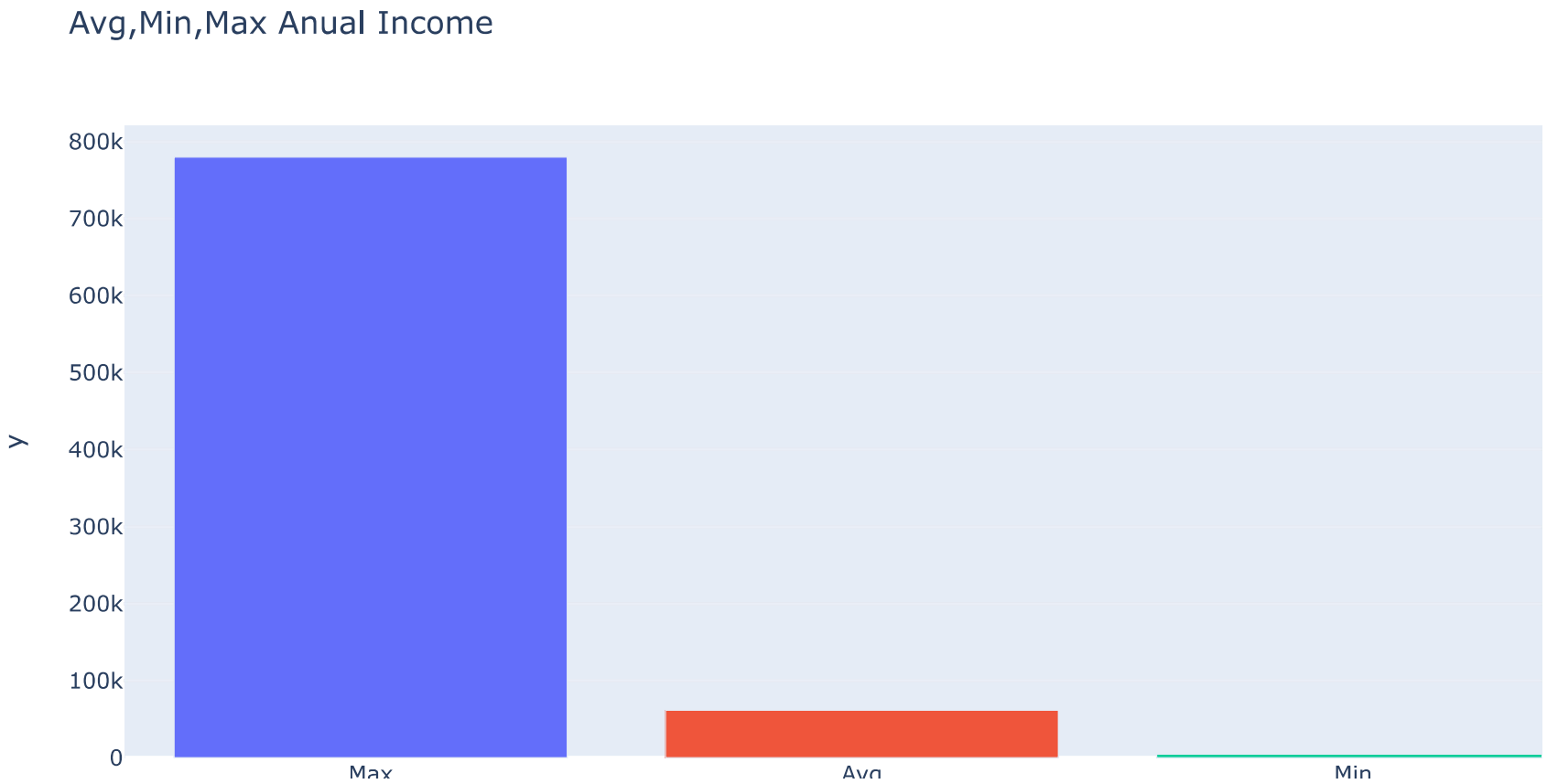
Out[142]:
```
780000.0
```

Minimum anual income who lives in RENT & loan status is Fully paid

In [143]:
```
ca_fullypaid_rent['annual_inc'].min()
```

Out[143]:
```
4800.0
```

In [144]:
```
x_data = ['Max','Avg','Min']
y_data = [ca_fullypaid_rent['annual_inc'].max(),ca_fullypaid_rent['annual_inc'].mean(),ca_fullypaid_rent['annual
fig = px.bar(x=x_data, y=y_data,color=x_data,title='Avg,Min,Max Anual Income')
fig.show()
```



Canada Charged Off -> Home ownership -> Rent data

In [145]:
```
ca_chargedoff_rent = ca_chargedoff[ca_chargedoff['home_ownership']=='RENT']
ca_chargedoff_rent.head(2)
```

Out[145]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_le
14	1069057	1303503.0	10000.0	10000.0	10000.0	36 months	10.65%	325.74	B	B2	SFMTA	3 y
24	1069559	1304634.0	6000.0	6000.0	6000.0	36 months	11.71%	198.46	B	B3	bmg-educational	1

Canada Charged Off -> Home ownership -> Own data

In [146]:
```
ca_chargedoff_own = ca_chargedoff[ca_chargedoff['home_ownership']=='OWN']
ca_chargedoff_rent.head(2)
```

Out[146]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_le
14	1069057	1303503.0	10000.0	10000.0	10000.0	36 months	10.65%	325.74	B	B2	SFMTA	3 y
24	1069559	1304634.0	6000.0	6000.0	6000.0	36 months	11.71%	198.46	B	B3	bmg-educational	1

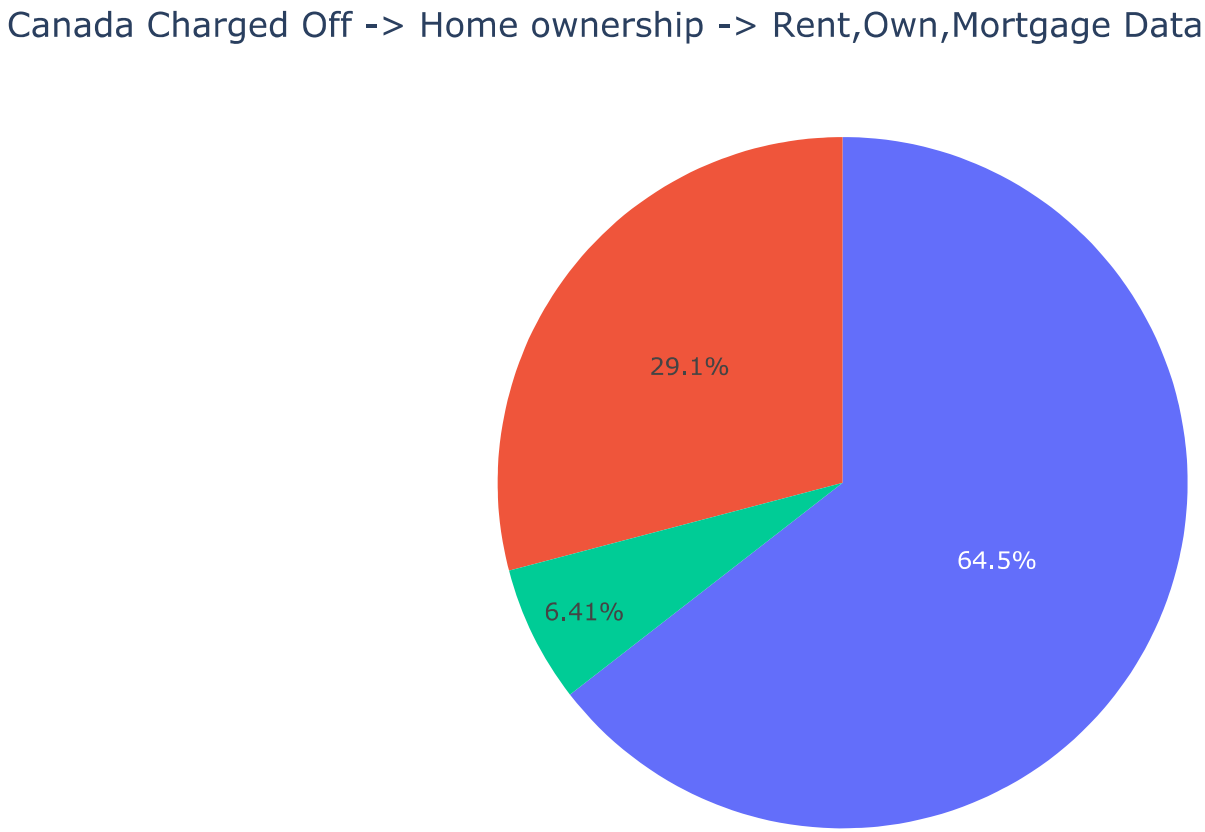
Canada Charged Off -> Home ownership -> Mortgage data

In [147]:
```
ca_chargedoff_mortgage = ca_chargedoff[ca_chargedoff['home_ownership']=='MORTGAGE']
ca_chargedoff_rent.head(2)
```

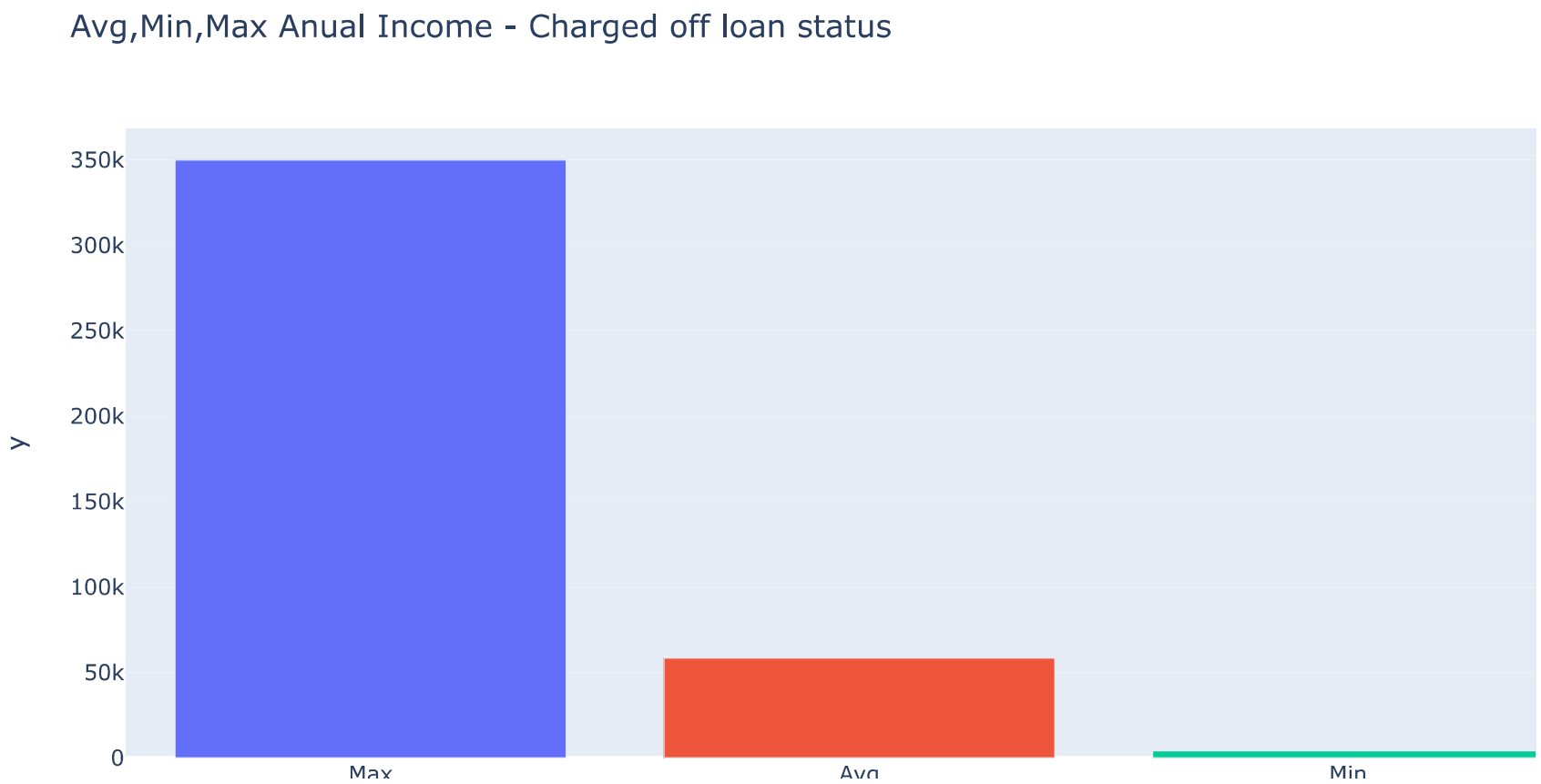
Out[147]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_le
14	1069057	1303503.0	10000.0	10000.0	10000.0	36 months	10.65%	325.74	B	B2	SFMTA	3 y
24	1069559	1304634.0	6000.0	6000.0	6000.0	36 months	11.71%	198.46	B	B3	bmg-educational	1

In [148]:
```
value = [len(ca_chargedoff_rent),len(ca_chargedoff_own),len(ca_chargedoff_mortgage)]
names = ['Rent','Own','Mortgage']
fig = px.pie(values=value, names=names,title='Canada Charged Off -> Home ownership -> Rent,Own,Mortgage Data')
fig.show()
```

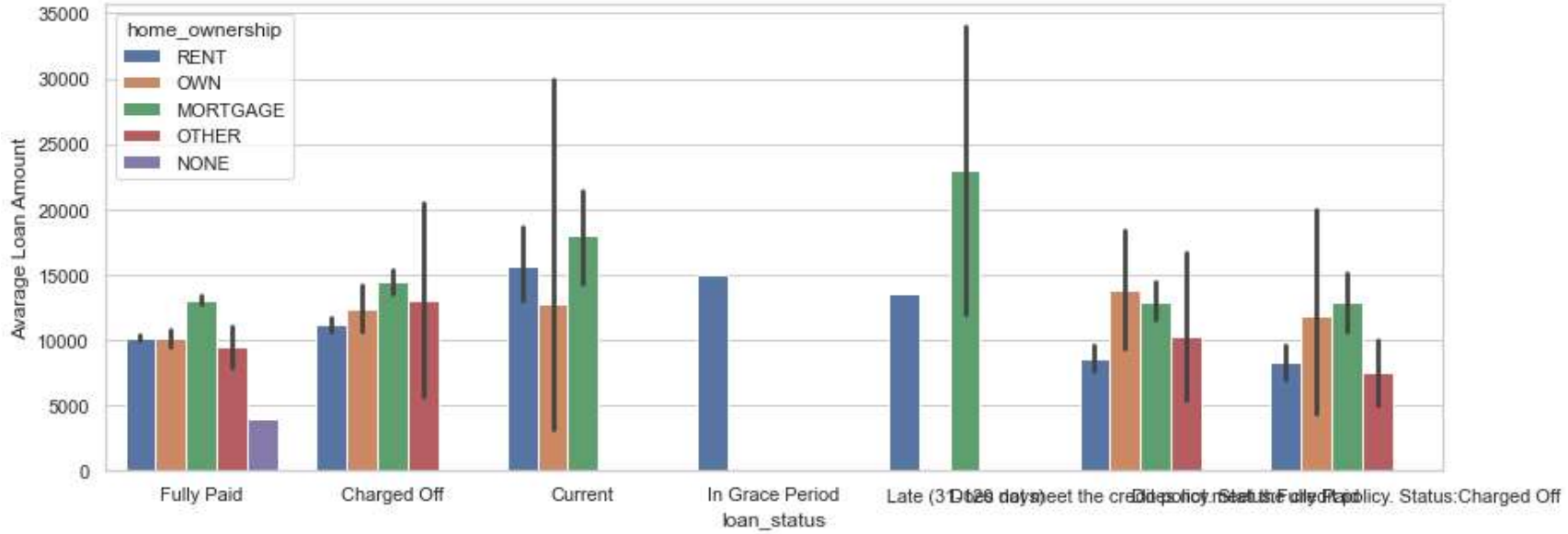


In [149]:
```
x_data = ['Max','Avg','Min']
y_data = [ca_chargedoff_rent['annual_inc'].max(),ca_chargedoff_rent['annual_inc'].mean(),ca_chargedoff_rent['anr
fig = px.bar(x=x_data, y=y_data,color=x_data,title='Avg,Min,Max Anual Income - Charged off loan status')
fig.show()
```



In [150]:
```
plt.figure(figsize=(14,5))
sns.barplot('loan_status','loan_amnt',data=canada_data,hue='home_ownership')
plt.ylabel('Avarage Loan Amount')
plt.show()
```

```
C:\Users\user\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\_decorators.py:36: FutureWarni
ng:

Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will
be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretatio
n.
```



In [151]:
```
#Updating 2 statuses of loan status for UI
#canada_data[canada_data['loan_status']=='Does not meet the credit policy. Status:Fully Paid'].replace('DNMCP FP
```

In [152]:
```
#canada_data.replace(canada_data['loan_status']=='Does not meet the credit policy. Status:Fully Paid','DNMCP FP'
```

In [153]:
```
canada_data.replace(to_replace ="Does not meet the credit policy. Status:Fully Paid",
                    value ="DNMCP FP",inplace=True)
canada_data.replace(to_replace ="Does not meet the credit policy. Status:Charged Off",
                    value ="DNMCP CO",inplace=True)
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_8540\1065859821.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

C:\Users\user\AppData\Local\Temp\ipykernel_8540\1065859821.py:3: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
```

In [154]:
```
df
```

Out[154]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87	B	B2	NaN
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.83	C	C4	Ryder
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	84.33	C	C5	NaN
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	339.31	C	C1	AIR RESOURCES BOARD
4	1075358	1311748.0	3000.0	3000.0	3000.0	60 months	12.69%	67.79	B	B5	University Medical Group
...	...	...	...	...	...	...	...	...	...	...	...
42532	72176	70868.0	2525.0	2525.0	225.0	36 months	9.33%	80.69	B	B3	NaN
42533	71623	70735.0	6500.0	6500.0	0.0	36 months	8.38%	204.84	A	A5	NaN
42534	70686	70681.0	5000.0	5000.0	0.0	36 months	7.75%	156.11	A	A3	Homemaker

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title
42537	Total amount funded in policy code 1: 460296150	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
42538	Total amount funded in policy code 2: 0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

42536 rows × 51 columns

In [155]:
```
df.columns
```

Out[155]:
```
Index(['id', 'member_id', 'loan_amnt', 'funded_amnt', 'funded_amnt_inv',
       'term', 'int_rate', 'installment', 'grade', 'sub_grade', 'emp_title',
       'emp_length', 'home_ownership', 'annual_inc', 'verification_status',
       'issue_d', 'loan_status', 'url', 'desc', 'purpose', 'addr_state', 'dti',
       'delinq_2yrs', 'earliest_cr_line', 'fico_range_low', 'fico_range_high',
       'inq_last_6mths', 'open_acc', 'pub_rec', 'revol_bal', 'revol_util',
       'total_acc', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp',
       'total_rec_int', 'total_rec_late_fee', 'recoveries', 'last_pymnt_d',
       'last_pymnt_amnt', 'last_credit_pull_d', 'last_fico_range_high',
       'last_fico_range_low', 'pub_rec_bankruptcies', 'earliest_cr_line_year',
       'issue_d_year', 'issue_d_month', 'last_pymnt_d_year',
       'last_pymnt_d_month', 'last_credit_pull_d_year',
       'last_credit_pull_d_month'],
      dtype='object')
```

Removing Unnecessary columns for ML model

In [156]:
```
new_df = df.drop(['id', 'member_id','emp_title',
       'emp_length','verification_status','url', 'desc', 'purpose', 'addr_state', 'dti',
       'delinq_2yrs', 'earliest_cr_line', 'fico_range_low', 'fico_range_high',
       'inq_last_6mths', 'open_acc', 'pub_rec', 'revol_bal', 'revol_util',
       'total_acc', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp',
       'total_rec_int', 'total_rec_late_fee', 'recoveries', 'last_pymnt_d',
       'last_pymnt_amnt', 'last_credit_pull_d', 'last_fico_range_high',
       'last_fico_range_low', 'pub_rec_bankruptcies', 'earliest_cr_line_year',
       'issue_d_year', 'issue_d_month', 'last_pymnt_d_year',
       'last_pymnt_d_month', 'last_credit_pull_d_year',
       'last_credit_pull_d_month','term','issue_d','int_rate'],axis=1)
```

In [157]:
```
new_df=new_df.dropna()
new_df
```

Out[157]:

	loan_amnt	funded_amnt	funded_amnt_inv	installment	grade	sub_grade	home_ownership	annual_inc	loan_status
0	5000.0	5000.0	4975.0	162.87	B	B2	RENT	24000.0	Fully Paid
1	2500.0	2500.0	2500.0	59.83	C	C4	RENT	30000.0	Charged Off
2	2400.0	2400.0	2400.0	84.33	C	C5	RENT	12252.0	Fully Paid
3	10000.0	10000.0	10000.0	339.31	C	C1	RENT	49200.0	Fully Paid
4	3000.0	3000.0	3000.0	67.79	B	B5	RENT	80000.0	Current
...	...	...	...	...	...	...	...	...	...
42529	6500.0	6500.0	0.0	208.66	B	B4	RENT	20000.0	Does not meet the credit policy. Status:Charge...
42530	3500.0	3500.0	225.0	113.39	C	C1	RENT	180000.0	Does not meet the credit policy. Status:Fully ...
42531	1000.0	1000.0	0.0	32.11	B	B4	RENT	12000.0	Does not meet the credit policy. Status:Fully ...
42532	2525.0	2525.0	225.0	80.69	B	B3	RENT	110000.0	Does not meet the credit policy. Status:Fully ...
42534	5000.0	5000.0	0.0	156.11	A	A3	MORTGAGE	70000.0	Does not meet the credit policy. Status:Fully ...

42530 rows × 9 columns

Replacing values for classification model

In [158]:
```
new_df=new_df.replace(['A','B','C','D','E','F','G'],
                      [1,2,3,4,5,6,7])
```

In [159]:
```
new_df['grade']
```

Out[159]:
```
0        2
1        3
2        3
3        3
4        2
        ..
42529    2
42530    3
42531    2
42532    2
42534    1
Name: grade, Length: 42530, dtype: int64
```

In [160]:
```
A = new_df['sub_grade'].value_counts()
for i in range(0,len(A.index)):
    new_df=new_df.replace(A.index[i],i)
print(new_df['sub_grade'])
print('-------------------------------------------')
B = new_df['home_ownership'].value_counts()
for i in range(0,len(B.index)):
    new_df=new_df.replace(B.index[i],i)
print(new_df['home_ownership'])
print('-------------------------------------------')
```

```
0         7
1        13
2        15
3         5
4         2
         ..
42529     4
42530     5
42531     4
42532     0
42534     9
Name: sub_grade, Length: 42530, dtype: int64
-------------------------------------------
0        0
1        0
2        0
3        0
4        0
        ..
42529    0
42530    0
42531    0
42532    0
42534    1
Name: home_ownership, Length: 42530, dtype: int64
-------------------------------------------
```

In [161]:
```
# drop1=new_df[(new_df['loan_status']=='Does not meet the credit policy. Status:Fully Paid')
#and (new_df['loan_status']=='Does not meet the credit policy. Status:Charged Off') ]
# drop2=new_df[(new_df['loan_status']=='Does not meet the credit policy. Status:Charged Off')]
# drop3=new_df[(new_df['loan_status']=='Current')]
# drop4=new_df[(new_df['loan_status']=='In Grace Period')]
# drop5=new_df[(new_df['loan_status']=='Late (31-120 days)')]
# drop6=new_df[(new_df['loan_status']=='Late (16-30 days)')]
# drop7=new_df[(new_df['loan_status']=='Default')]
```

In [171]:
```
# latest_df=new_df[(new_df['loan_status']=='Fully Paid')
# and (new_df['loan_status']=='Charged Off') ]
```

In [162]:
```
#new_df.drop(index=39786,inplace = True)
# new_df.drop(new_df.iloc[0])
```

In [163]:
```
# for i in drop1.index:
#     print(i)
#     new_df.drop(new_df.iloc[i],axis=1)
```

In [164]:
```
# print(len(new_df))
# new_df.drop(drop1.index[4])
# # new_df.drop(drop2.index)
# # new_df.drop(drop3.index)
# # new_df.drop(drop4.index)
# # new_df.drop(drop5.index)
# # new_df.drop(drop6.index)
# # new_df.drop(drop7.index)
# print(len(new_df))
```

In [165]:
```
# new_df=new_df.replace(['Fully Paid','Charged Off'],
#                       [1,0])
# new_df['loan_status'].value_counts()
```

In [166]:
```
len(new_df)
```

Out[166]: 42530

In [189]:
```
#new_df[((new_df['loan_status']=='Fully Paid') and (new_df['loan_status']=='Charged Off'))]

#new_df[(new_df['loan_status']=='Charged Off')]
```

Updating new Dataframe for ML model

In [187]:
```
latest_df = new_df[(new_df['loan_status']=='Fully Paid') | (new_df['loan_status']=='Charged Off')]
latest_df
```

Out[187]:

	loan_amnt	funded_amnt	funded_amnt_inv	installment	grade	sub_grade	home_ownership	annual_inc	loan_status
0	5000.0	5000.0	4975.0	162.87	2	7	0	24000.0	Fully Paid
1	2500.0	2500.0	2500.0	59.83	3	13	0	30000.0	Charged Off
2	2400.0	2400.0	2400.0	84.33	3	15	0	12252.0	Fully Paid
3	10000.0	10000.0	10000.0	339.31	3	5	0	49200.0	Fully Paid
5	5000.0	5000.0	5000.0	156.46	1	1	0	36000.0	Fully Paid
...	...	...	...	...	...	...	...	...	...
39781	2500.0	2500.0	1075.0	78.42	1	1	1	110000.0	Fully Paid
39782	8500.0	8500.0	875.0	275.38	3	5	0	18000.0	Fully Paid
39783	5000.0	5000.0	1325.0	156.84	1	1	1	100000.0	Fully Paid
39784	5000.0	5000.0	650.0	155.38	1	11	1	200000.0	Fully Paid
39785	7500.0	7500.0	800.0	255.43	5	21	2	22000.0	Fully Paid

39238 rows × 9 columns

In [191]:
```
latest_df=latest_df.replace(['Fully Paid','Charged Off'],
                            [1,0])
latest_df
```

Out[191]:

	loan_amnt	funded_amnt	funded_amnt_inv	installment	grade	sub_grade	home_ownership	annual_inc	loan_status
0	5000.0	5000.0	4975.0	162.87	2	7	0	24000.0	1
1	2500.0	2500.0	2500.0	59.83	3	13	0	30000.0	0
2	2400.0	2400.0	2400.0	84.33	3	15	0	12252.0	1
3	10000.0	10000.0	10000.0	339.31	3	5	0	49200.0	1
5	5000.0	5000.0	5000.0	156.46	1	1	0	36000.0	1
...	...	...	...	...	...	...	...	...	...
39781	2500.0	2500.0	1075.0	78.42	1	1	1	110000.0	1
39782	8500.0	8500.0	875.0	275.38	3	5	0	18000.0	1
39783	5000.0	5000.0	1325.0	156.84	1	1	1	100000.0	1
39784	5000.0	5000.0	650.0	155.38	1	11	1	200000.0	1
39785	7500.0	7500.0	800.0	255.43	5	21	2	22000.0	1

39238 rows × 9 columns

Convert Dataframe to CSV file for ML model

In [192]:
```
latest_df.to_csv('updated_lending_club_loans.csv')
```

In [ ]:
```

```