# Financial

July 13, 2025

[7]:
```
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
```

[8]:
```
df=pd.read_csv("financial_loan.csv")
```

[9]:
```
df
```

[9]:
```
            id address_state application_type emp_length  \
0      1077430            GA       INDIVIDUAL   < 1 year
1      1072053            CA       INDIVIDUAL    9 years
2      1069243            CA       INDIVIDUAL    4 years
3      1041756            TX       INDIVIDUAL   < 1 year
4      1068350            IL       INDIVIDUAL  10+ years
…         …           …              …        …
38571   803452            NJ       INDIVIDUAL   < 1 year
38572   970377            NY       INDIVIDUAL    8 years
38573   875376            CA       INDIVIDUAL    5 years
38574   972997            NY       INDIVIDUAL    5 years
38575   682952            NY       INDIVIDUAL    4 years

                             emp_title grade home_ownership  issue_date  \
0                                Ryder     C           RENT  11-02-2021
1                       MKC Accounting     E           RENT  01-01-2021
2                Chemat Technology Inc     C           RENT  05-01-2021
3                  barnes distribution     B       MORTGAGE  25-02-2021
4                        J&J Steel Inc     A       MORTGAGE  01-01-2021
…                                 …   …            …         …
38571         Joseph M Sanzari Company     C       MORTGAGE  11-07-2021
38572                        Swat Fame     C           RENT  11-10-2021
38573  Anaheim Regional Medical Center     D           RENT  11-09-2021
38574               Brooklyn Radiology     D           RENT  11-10-2021
38575                    Allen Edmonds     F           RENT  11-07-2021

      last_credit_pull_date last_payment_date  … sub_grade        term  \
0                13-09-2021        13-04-2021  …        C4   60 months
1                14-12-2021        15-01-2021  …        E1   36 months
```

```
2                12-12-2021         09-01-2021  …        C5   36 months
3                12-12-2021         12-03-2021  …        B2   60 months
4                14-12-2021         15-01-2021  …        A1   36 months
…                     …               …  …       …         …
38571            16-05-2021         16-05-2021  …        C1   60 months
38572            16-04-2021         16-05-2021  …        C1   60 months
38573            16-05-2021         16-05-2021  …        D5   60 months
38574            16-05-2021         16-05-2021  …        D5   60 months
38575            16-05-2021         16-05-2021  …        F3   60 months

       verification_status annual_income     dti installment int_rate  \
0          Source Verified       30000.0  0.0100       59.83   0.1527
1          Source Verified       48000.0  0.0535      109.43   0.1864
2             Not Verified       50000.0  0.2088      421.65   0.1596
3          Source Verified       42000.0  0.0540       97.06   0.1065
4                 Verified       83000.0  0.0231      106.53   0.0603
…                    …           …     …         …      …
38571             Verified      100000.0  0.1986      551.64   0.1299
38572             Verified       50000.0  0.0458      579.72   0.1349
38573             Verified       65000.0  0.1734      627.93   0.1749
38574             Verified      368000.0  0.0009      612.72   0.1825
38575             Verified       80000.0  0.0600      486.86   0.2099

       loan_amount  total_acc  total_payment
0             2500          4           1009
1             3000          4           3939
2            12000         11           3522
3             4500          9           4911
4             3500         28           3835
…            …        …            …
38571        24250         33          31946
38572        25200         18          31870
38573        25000         20          35721
38574        24000          9          33677
38575        18000          7          27679

[38576 rows x 24 columns]
```

[10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38576 entries, 0 to 38575
Data columns (total 24 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   id                     38576 non-null  int64
 1   address_state          38576 non-null  object
 2   application_type       38576 non-null  object
 3   emp_length             38576 non-null  object
 4   emp_title              37138 non-null  object
 5   grade                  38576 non-null  object
 6   home_ownership         38576 non-null  object
 7   issue_date             38576 non-null  object
 8   last_credit_pull_date  38576 non-null  object
 9   last_payment_date      38576 non-null  object
 10  loan_status            38576 non-null  object
 11  next_payment_date      38576 non-null  object
 12  member_id              38576 non-null  int64
 13  purpose                38576 non-null  object
 14  sub_grade              38576 non-null  object
 15  term                   38576 non-null  object
 16  verification_status    38576 non-null  object
 17  annual_income          38576 non-null  float64
 18  dti                    38576 non-null  float64
 19  installment            38576 non-null  float64
 20  int_rate               38576 non-null  float64
 21  loan_amount            38576 non-null  int64
 22  total_acc              38576 non-null  int64
 23  total_payment          38576 non-null  int64
dtypes: float64(4), int64(5), object(15)
memory usage: 7.1+ MB
```

[11]:

```python
df['issue_date']=pd.to_datetime(df['issue_date'],format="%d-%m-%Y")
```

[12]:

```python
df.dtypes
```

[12]:

```
id                                int64
address_state                    object
application_type                 object
emp_length                       object
emp_title                        object
grade                            object
home_ownership                   object
issue_date               datetime64[ns]
last_credit_pull_date            object
last_payment_date                object
loan_status                      object
next_payment_date                object
member_id                         int64
purpose                          object
sub_grade                        object
term                             object
verification_status              object
annual_income                   float64
dti                             float64
```

```
installment                       float64
int_rate                          float64
loan_amount                         int64
total_acc                           int64
total_payment                       int64
dtype: object
```

[13]:

```
df['last_credit_pull_date']=pd.
 ↪to_datetime(df['last_credit_pull_date'],format="%d-%m-%Y")
```

[14]:

```
df.dtypes
```

[14]:

```
id                                  int64
address_state                      object
application_type                   object
emp_length                         object
emp_title                          object
grade                              object
home_ownership                     object
issue_date                 datetime64[ns]
last_credit_pull_date      datetime64[ns]
last_payment_date                  object
loan_status                        object
next_payment_date                  object
member_id                           int64
purpose                            object
sub_grade                          object
term                               object
verification_status                object
annual_income                     float64
dti                               float64
installment                       float64
int_rate                          float64
loan_amount                         int64
total_acc                           int64
total_payment                       int64
dtype: object
```

[15]:

```
df['last_payment_date']=pd.
 ↪to_datetime(df['last_payment_date'],format="%d-%m-%Y")
```

[16]:

```
df['next_payment_date']=pd.
 ↪to_datetime(df['next_payment_date'],format="%d-%m-%Y")
```

[17]:

```
df.dtypes
```

[17]:

```
id                                int64
address_state                    object
application_type                 object
emp_length                       object
emp_title                        object
grade                            object
home_ownership                   object
issue_date               datetime64[ns]
last_credit_pull_date    datetime64[ns]
last_payment_date        datetime64[ns]
loan_status                      object
next_payment_date        datetime64[ns]
member_id                         int64
purpose                          object
sub_grade                        object
term                             object
verification_status              object
annual_income                   float64
dti                             float64
installment                     float64
int_rate                        float64
loan_amount                       int64
total_acc                         int64
total_payment                     int64
dtype: object
```

[18]:

```
df.isnull().sum()
```

[18]:

```
id                          0
address_state               0
application_type            0
emp_length                  0
emp_title                1438
grade                       0
home_ownership              0
issue_date                  0
last_credit_pull_date       0
last_payment_date           0
loan_status                 0
next_payment_date           0
member_id                   0
purpose                     0
sub_grade                   0
term                        0
verification_status         0
annual_income               0
dti                         0
```

```
installment              0
int_rate                 0
loan_amount              0
total_acc                0
total_payment            0
dtype: int64
```

```
[19]: df['emp_title']=df['emp_title'].fillna('fill Allen')
```

```
[20]: df.isnull().sum()
```

```
[20]: id                      0
      address_state           0
      application_type        0
      emp_length              0
      emp_title               0
      grade                   0
      home_ownership          0
      issue_date              0
      last_credit_pull_date   0
      last_payment_date       0
      loan_status             0
      next_payment_date       0
      member_id               0
      purpose                 0
      sub_grade               0
      term                    0
      verification_status     0
      annual_income           0
      dti                     0
      installment             0
      int_rate                0
      loan_amount             0
      total_acc               0
      total_payment           0
      dtype: int64
```

```
[21]: df.duplicated().sum()
```

```
[21]: np.int64(0)
```

```
[22]: df.tail()
```

```
[22]:            id address_state application_type emp_length  \
      38571  803452            NJ       INDIVIDUAL   < 1 year
      38572  970377            NY       INDIVIDUAL    8 years
      38573  875376            CA       INDIVIDUAL    5 years
```

```
38574  972997            NY       INDIVIDUAL    5 years
38575  682952            NY       INDIVIDUAL    4 years

                             emp_title grade home_ownership issue_date  \
38571         Joseph M Sanzari Company     C       MORTGAGE 2021-07-11
38572                        Swat Fame     C           RENT 2021-10-11
38573  Anaheim Regional Medical Center     D           RENT 2021-09-11
38574               Brooklyn Radiology     D           RENT 2021-10-11
38575                    Allen Edmonds     F           RENT 2021-07-11

      last_credit_pull_date last_payment_date  … sub_grade        term  \
38571            2021-05-16        2021-05-16  …        C1   60 months
38572            2021-04-16        2021-05-16  …        C1   60 months
38573            2021-05-16        2021-05-16  …        D5   60 months
38574            2021-05-16        2021-05-16  …        D5   60 months
38575            2021-05-16        2021-05-16  …        F3   60 months

       verification_status annual_income     dti installment int_rate  \
38571             Verified      100000.0  0.1986      551.64    0.1299
38572             Verified       50000.0  0.0458      579.72    0.1349
38573             Verified       65000.0  0.1734      627.93    0.1749
38574             Verified      368000.0  0.0009      612.72    0.1825
38575             Verified       80000.0  0.0600      486.86    0.2099

       loan_amount  total_acc  total_payment
38571        24250         33          31946
38572        25200         18          31870
38573        25000         20          35721
38574        24000          9          33677
38575        18000          7          27679

[5 rows x 24 columns]
```

[23]:
```
df['emp_length']=df['emp_length'].replace('< 1 year','1 year')
```

[24]:
```
df.head()
```

[24]:
```
        id address_state application_type emp_length              emp_title  \
0  1077430            GA       INDIVIDUAL     1 year                  Ryder
1  1072053            CA       INDIVIDUAL    9 years         MKC Accounting
2  1069243            CA       INDIVIDUAL    4 years  Chemat Technology Inc
3  1041756            TX       INDIVIDUAL     1 year    barnes distribution
4  1068350            IL       INDIVIDUAL  10+ years          J&J Steel Inc

  grade home_ownership issue_date last_credit_pull_date last_payment_date  \
0     C           RENT 2021-02-11            2021-09-13        2021-04-13
1     E           RENT 2021-01-01            2021-12-14        2021-01-15
```

```
2     C            RENT  2021-01-05            2021-12-12         2021-01-09
3     B        MORTGAGE  2021-02-25            2021-12-12         2021-03-12
4     A        MORTGAGE  2021-01-01            2021-12-14         2021-01-15

  … sub_grade        term  verification_status annual_income     dti  \
0 …        C4   60 months     Source Verified         30000.0  0.0100
1 …        E1   36 months     Source Verified         48000.0  0.0535
2 …        C5   36 months        Not Verified         50000.0  0.2088
3 …        B2   60 months     Source Verified         42000.0  0.0540
4 …        A1   36 months            Verified         83000.0  0.0231

  installment int_rate  loan_amount  total_acc  total_payment
0       59.83   0.1527         2500          4           1009
1      109.43   0.1864         3000          4           3939
2      421.65   0.1596        12000         11           3522
3       97.06   0.1065         4500          9           4911
4      106.53   0.0603         3500         28           3835

[5 rows x 24 columns]
```

[25]:

```
df['emp_length']=df['emp_length'].replace('10+ years','10 year')
```

[26]:

```
df.head()
```

[26]:

```
        id address_state application_type emp_length              emp_title  \
0  1077430            GA       INDIVIDUAL     1 year                  Ryder
1  1072053            CA       INDIVIDUAL    9 years         MKC Accounting
2  1069243            CA       INDIVIDUAL    4 years  Chemat Technology Inc
3  1041756            TX       INDIVIDUAL     1 year    barnes distribution
4  1068350            IL       INDIVIDUAL    10 year          J&J Steel Inc

  grade home_ownership issue_date last_credit_pull_date last_payment_date  \
0     C            RENT  2021-02-11            2021-09-13         2021-04-13
1     E            RENT  2021-01-01            2021-12-14         2021-01-15
2     C            RENT  2021-01-05            2021-12-12         2021-01-09
3     B        MORTGAGE  2021-02-25            2021-12-12         2021-03-12
4     A        MORTGAGE  2021-01-01            2021-12-14         2021-01-15

  … sub_grade        term  verification_status annual_income     dti  \
0 …        C4   60 months     Source Verified         30000.0  0.0100
1 …        E1   36 months     Source Verified         48000.0  0.0535
2 …        C5   36 months        Not Verified         50000.0  0.2088
3 …        B2   60 months     Source Verified         42000.0  0.0540
4 …        A1   36 months            Verified         83000.0  0.0231

  installment int_rate  loan_amount  total_acc  total_payment
0       59.83   0.1527         2500          4           1009
```

```
1      109.43   0.1864          3000           4            3939
2      421.65   0.1596         12000          11            3522
3       97.06   0.1065          4500           9            4911
4      106.53   0.0603          3500          28            3835

[5 rows x 24 columns]
```

[27]:

```
df.columns
```

[27]:

```
Index(['id', 'address_state', 'application_type', 'emp_length', 'emp_title',
       'grade', 'home_ownership', 'issue_date', 'last_credit_pull_date',
       'last_payment_date', 'loan_status', 'next_payment_date', 'member_id',
       'purpose', 'sub_grade', 'term', 'verification_status', 'annual_income',
       'dti', 'installment', 'int_rate', 'loan_amount', 'total_acc',
       'total_payment'],
      dtype='object')
```

[28]:

```
avg_annual_income=df.groupby('annual_income')['loan_amount'].mean()
```

[29]:

```
avg_annual_income
```

[29]:

```
annual_income
4000.0        2000.0
4080.0        1400.0
4200.0        2750.0
4800.0        1800.0
4888.0        1400.0
               …
1782000.0    12025.0
1900000.0     1500.0
2039784.0     8450.0
3900000.0    25000.0
6000000.0     5000.0
Name: loan_amount, Length: 5096, dtype: float64
```

[30]:

```
result=df['annual_income'].mean()
```

[31]:

```
result
```

[31]:

```
np.float64(69644.54031003732)
```

[32]:

```
Score=df['loan_amount'].mean()
Score
```

[32]:

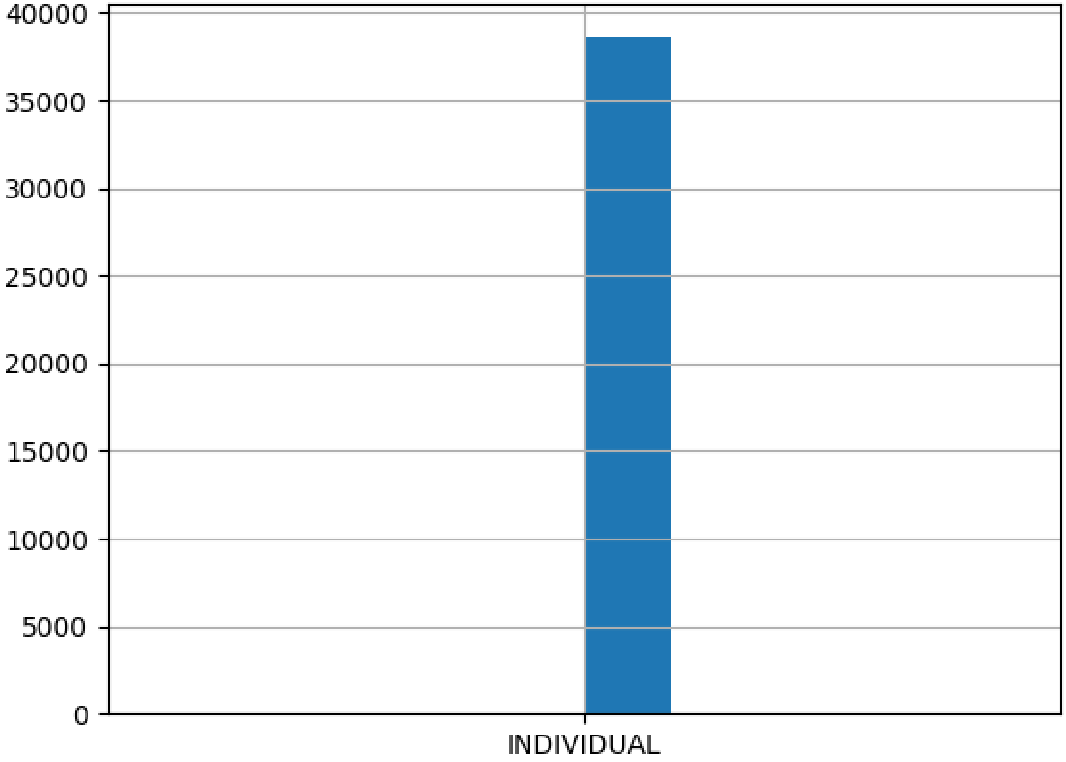
```
np.float64(11296.066855039402)
```

[33]:

```
payment=df['total_payment'].mean()
payment
```

[33]:
```
np.float64(12263.348532766488)
```

[34]:
```
df['application_type'].hist()
plt.show()
```



[35]:
```
grouped=df.groupby('purpose')['annual_income'].mean()
grouped
```

[35]:
```
purpose
Debt consolidation    67846.561670
car                   62681.970969
credit card           70994.439396
educational           54042.668508
home improvement      91042.380056
house                 78125.224481
major purchase        67063.463502
medical               69315.476147
moving                62587.266816
other                 64099.074995
renewable_energy      81331.049255
small business        75414.329229
vacation              60710.732756
```

```
wedding                 68901.525377
Name: annual_income, dtype: float64
```

[36]:

```
grp=df.groupby('grade')['annual_income'].mean()
grp
```

[36]:

```
grade
A    67533.991024
B    68320.577551
C    68482.656662
D    69092.806177
E    78328.377943
F    85115.680263
G    94724.937732
Name: annual_income, dtype: float64
```

[37]:

```
gp=df.groupby('home_ownership')['loan_amount'].mean()
gp
```

[37]:

```
home_ownership
MORTGAGE    12753.177695
NONE         5600.000000
OTHER       10663.010204
OWN         10429.060958
RENT        10074.758664
Name: loan_amount, dtype: float64
```

[38]:

```
group=df.groupby('loan_status')['loan_amount'].mean()
group
```

[38]:

```
loan_status
Charged Off    12288.060191
Current        17182.604736
Fully Paid     10930.419972
Name: loan_amount, dtype: float64
```

[39]:

```
groups=df.groupby('loan_status')['total_payment'].mean()
groups
```

[39]:

```
loan_status
Charged Off     6991.330021
Current        22039.994536
Fully Paid     12804.052139
Name: total_payment, dtype: float64
```

[40]:

```
Group=df.groupby('loan_status')['annual_income'].mean()
Group
```

[40]:

```
loan_status
Charged Off    63515.728245
Current        76487.821366
Fully Paid     70427.586885
Name: annual_income, dtype: float64
```

[41]:

```
test=df.groupby('term').agg({'loan_amount':'sum'})
test
```

[41]:

```
           loan_amount
term
36 months    273041225
60 months    162715850
```

[42]:

```
txt=df.groupby('sub_grade').agg({'total_payment':'mean'})
txt
```

[42]:

```
           total_payment
sub_grade
A1           7767.437262
A2           7696.325694
A3           8184.326437
A4           9842.420978
A5          10161.445365
B1          10716.402627
B2          11462.107035
B3          12530.377205
B4          12309.474949
B5          12658.907337
C1          12294.992820
C2          12721.889959
C3          11920.932886
C4          11531.500000
C5          11797.270200
D1          11074.312158
D2          12965.495434
D3          13883.784091
D4          15147.344792
D5          15572.130435
E1          16153.324000
E2          16921.167187
E3          17894.905204
E4          18486.071429
E5          20269.248780
F1          20489.221538
F2          20139.304527
F3          21082.615385
```

```
F4          19694.920245
F5          21013.756522
G1          23603.950495
G2          22519.692308
G3          21453.729167
G4          25822.125000
G5          21656.800000
```

[43]:

```
text=df.groupby('address_state').agg({'annual_income':'mean'})
text
```

[43]:

```
               annual_income
address_state
AK              78759.090256
AL              63085.690880
AR              60025.896695
AZ              68087.739784
CA              72558.934922
CO              68102.769753
CT              76426.845932
DC              77772.504673
DE              66564.545455
FL              65331.953588
GA              69585.382258
HI              63018.079059
IA              50599.200000
ID              57792.933333
IL              69911.666191
IN              35416.111111
KS              63501.781077
KY              61106.057750
LA              74543.005822
MA              73542.916595
MD              78314.101626
ME              23866.666667
MI              66900.522891
MN              60902.009274
MO              61738.638758
MS              55121.526316
MT              55274.124051
NC              65702.779381
NE              49624.000000
NH              69741.089565
NJ              75149.568968
NM              84666.168962
NV              69878.393776
NY              72159.510008
```

```
OH              60785.244234
OK              62369.447645
OR              58386.422156
PA              63097.620020
RI              69035.616327
SC              63218.079655
SD              49522.078730
TN              61530.058824
TX              74066.193780
UT              66146.671111
VA              74341.230276
VT              53985.955741
WA              66887.277267
WI              63445.098453
WV              56216.736287
WY              58220.557975
```

[44]:
```
df['address_state'].unique()
```

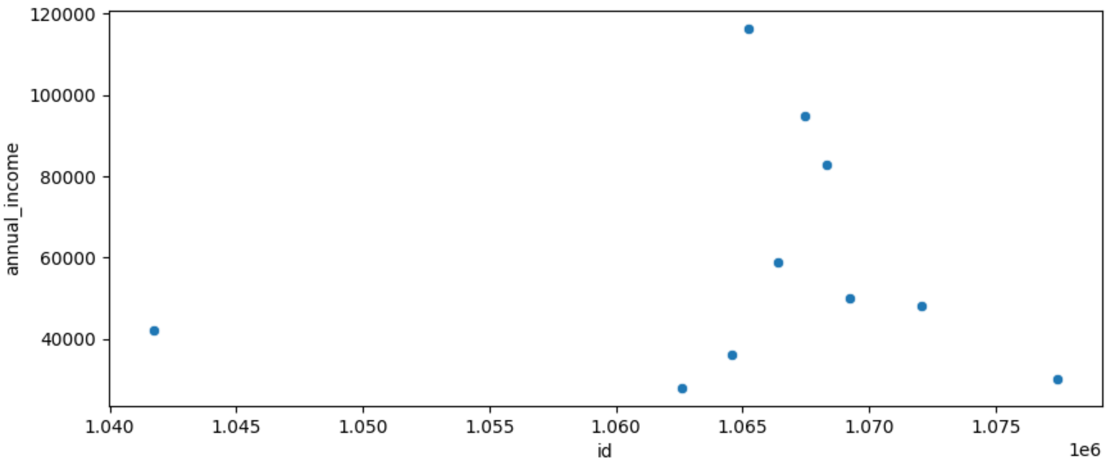
[44]:
```
array(['GA', 'CA', 'TX', 'IL', 'PA', 'FL', 'MI', 'RI', 'NY', 'MD', 'WI',
       'NV', 'UT', 'WA', 'NH', 'HI', 'MA', 'OK', 'NJ', 'OH', 'AZ', 'CT',
       'MN', 'CO', 'TN', 'VA', 'MO', 'DE', 'NM', 'LA', 'AR', 'KY', 'NC',
       'SC', 'WV', 'KS', 'WY', 'OR', 'AL', 'VT', 'MS', 'DC', 'MT', 'SD',
       'AK', 'IN', 'ME', 'ID', 'NE', 'IA'], dtype=object)
```

[45]:
```
df.columns
```

[45]:
```
Index(['id', 'address_state', 'application_type', 'emp_length', 'emp_title',
       'grade', 'home_ownership', 'issue_date', 'last_credit_pull_date',
       'last_payment_date', 'loan_status', 'next_payment_date', 'member_id',
       'purpose', 'sub_grade', 'term', 'verification_status', 'annual_income',
       'dti', 'installment', 'int_rate', 'loan_amount', 'total_acc',
       'total_payment'],
      dtype='object')
```

[46]:
```
plt.figure(figsize=(10,4))
sb.scatterplot(data=df.head(10),x='id',y='annual_income')
```

[46]:
```
<Axes: xlabel='id', ylabel='annual_income'>
```



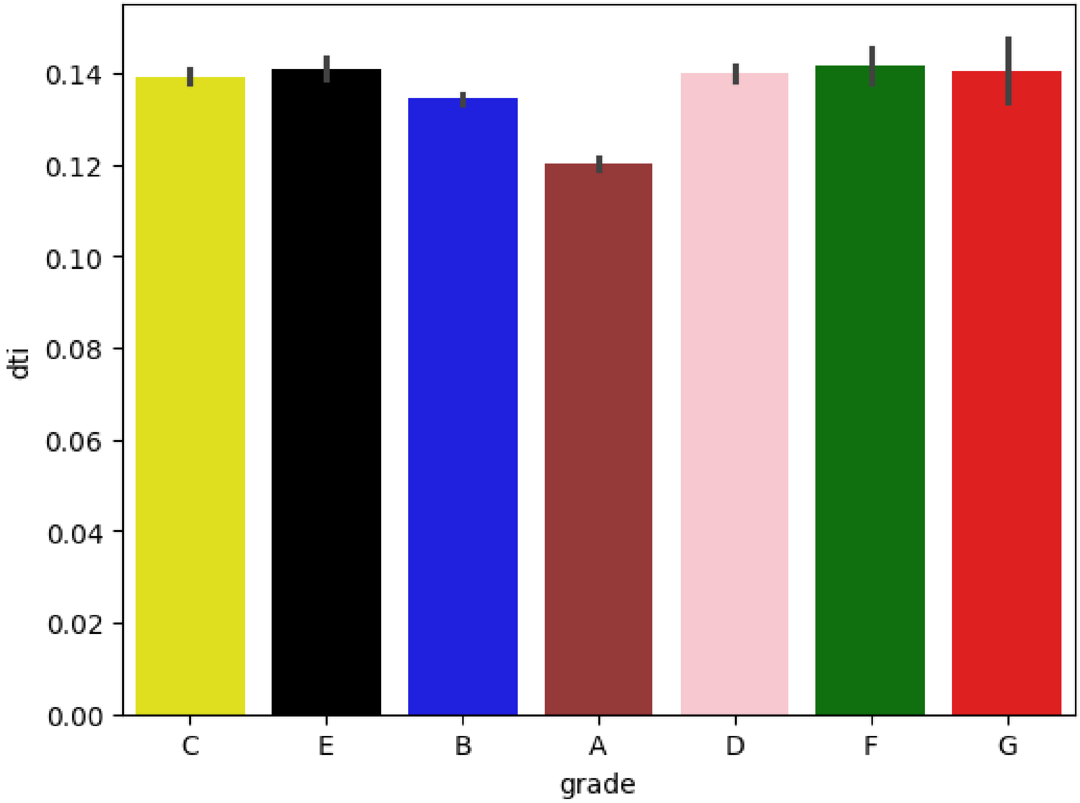
[47]:

```
colors=('yellow','black','blue','brown','pink','green','red','skyblue','voilet','darkgreen')
sb.barplot(data=df,x='grade',y='dti',palette=colors)
```

```
C:\Users\admin\AppData\Local\Temp\ipykernel_17780\390487513.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sb.barplot(data=df,x='grade',y='dti',palette=colors)
```

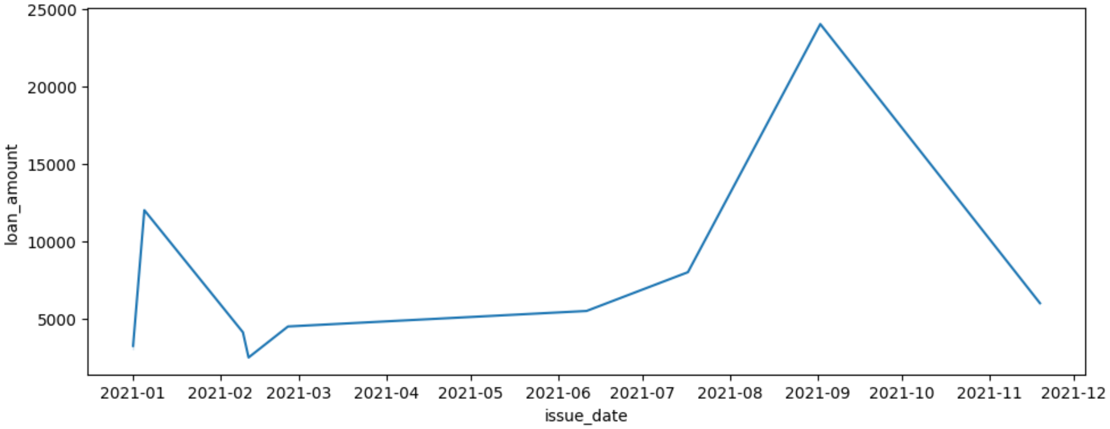
[47]:

```
<Axes: xlabel='grade', ylabel='dti'>
```



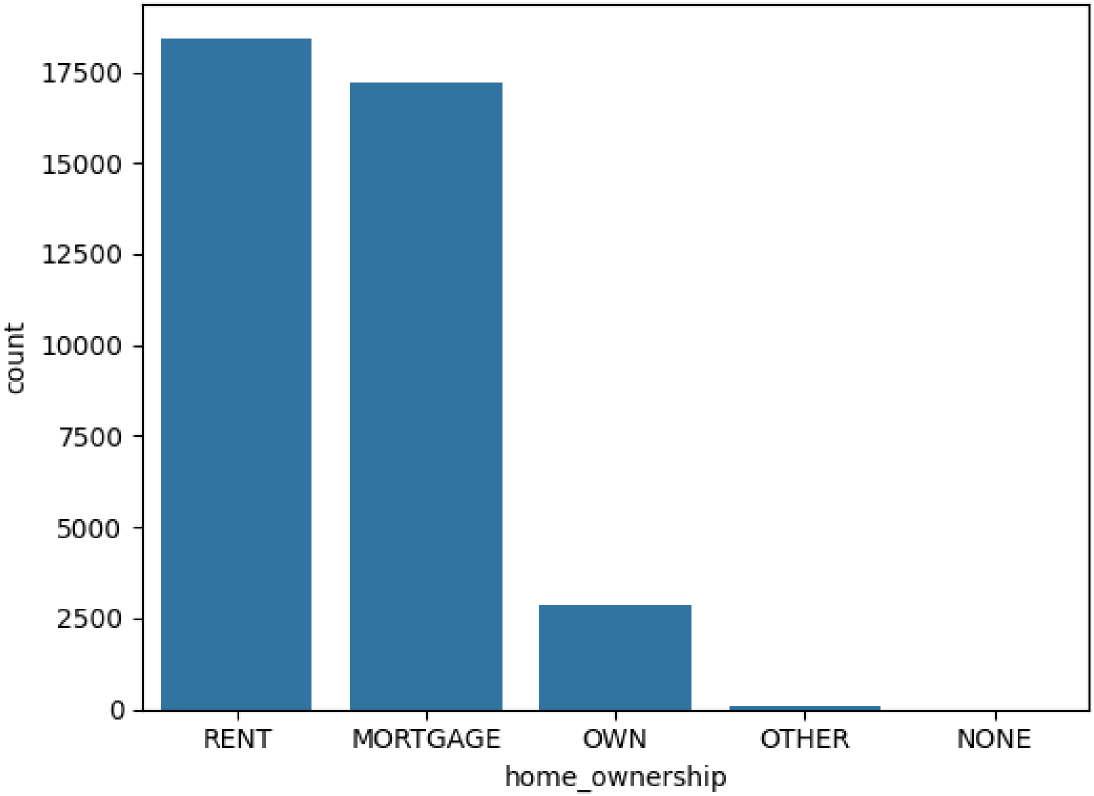
[48]:

```
plt.figure(figsize=(10,4))
sb.lineplot(data=df.head(10),x='issue_date',y='loan_amount')
plt.tight_layout()
plt.show()
```



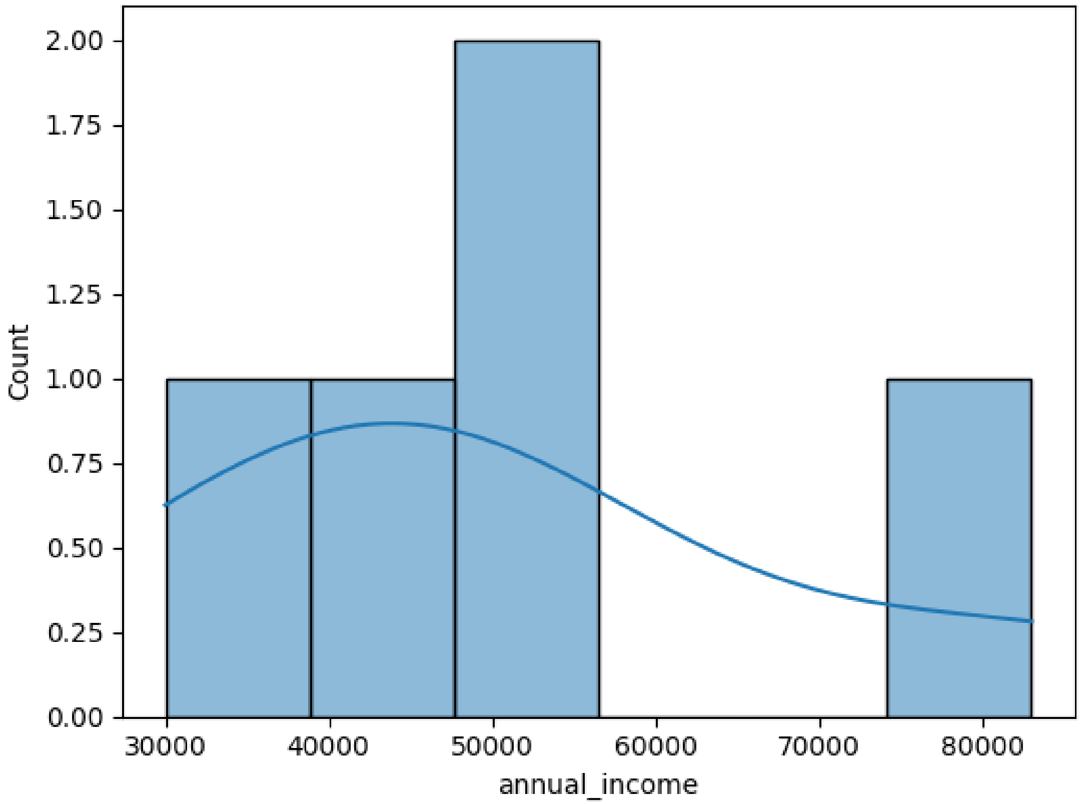
```
[49]: sb.countplot(data=df,x='home_ownership')
```

```
[49]: <Axes: xlabel='home_ownership', ylabel='count'>
```



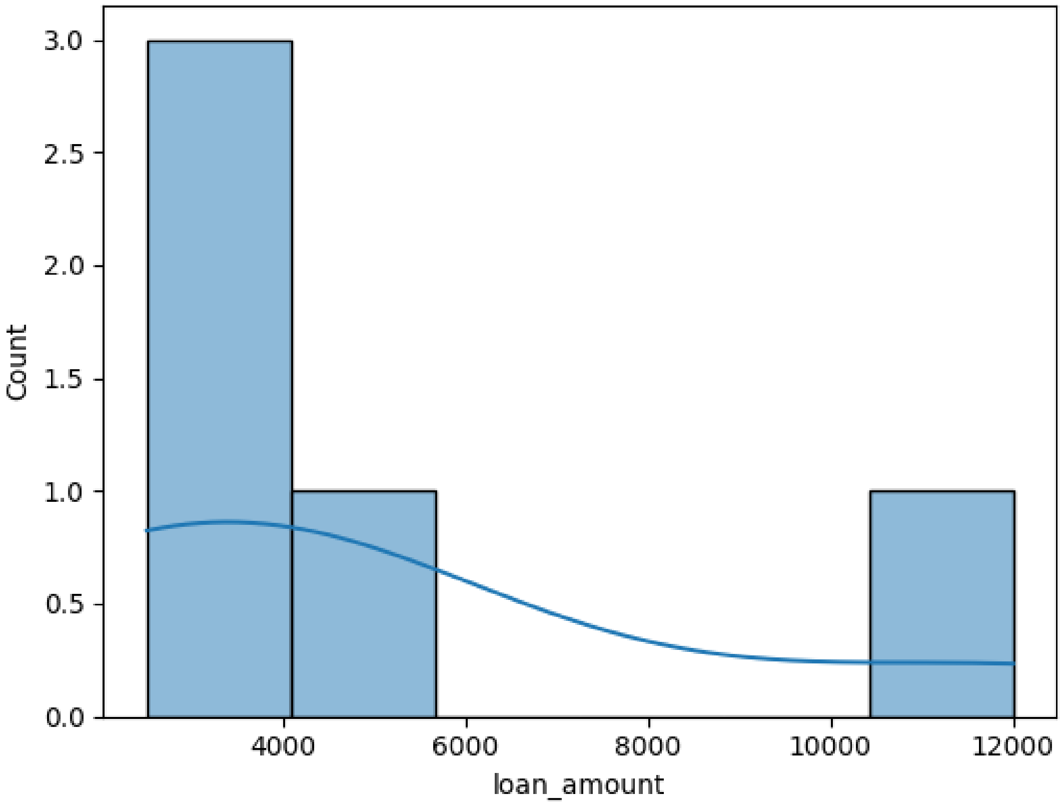
```
[50]: sb.histplot(data=df.head(),x='annual_income',kde='term')
```

```
[50]: <Axes: xlabel='annual_income', ylabel='Count'>
```



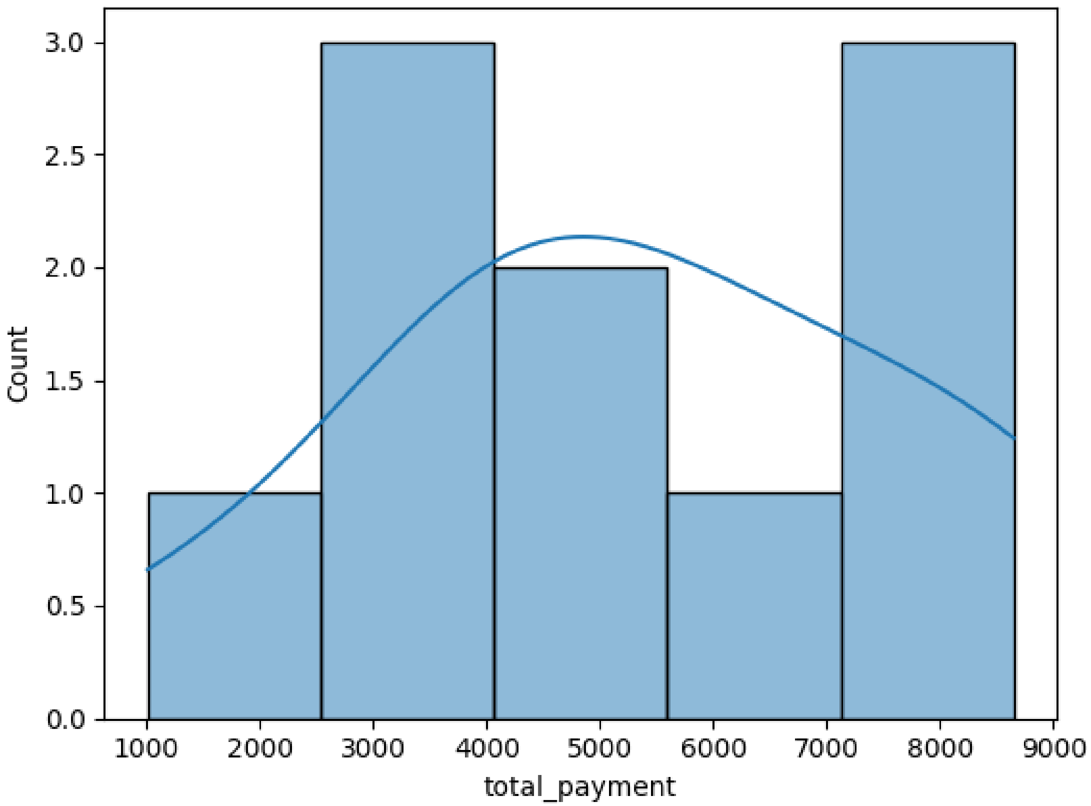
```
[51]: sb.histplot(data=df.head(),x='loan_amount',kde='sub_grade')
```

```
[51]: <Axes: xlabel='loan_amount', ylabel='Count'>
```



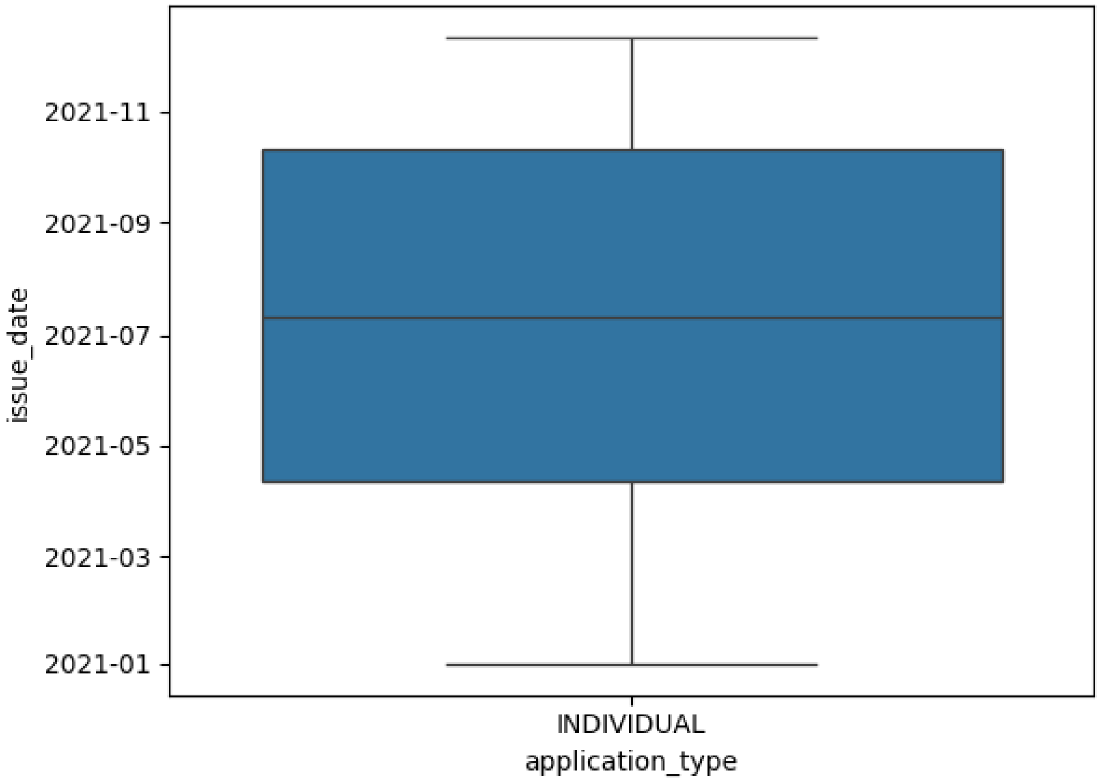
```
[52]: sb.histplot(data=df.head(10),x='total_payment',kde='grade')
```

```
[52]: <Axes: xlabel='total_payment', ylabel='Count'>
```

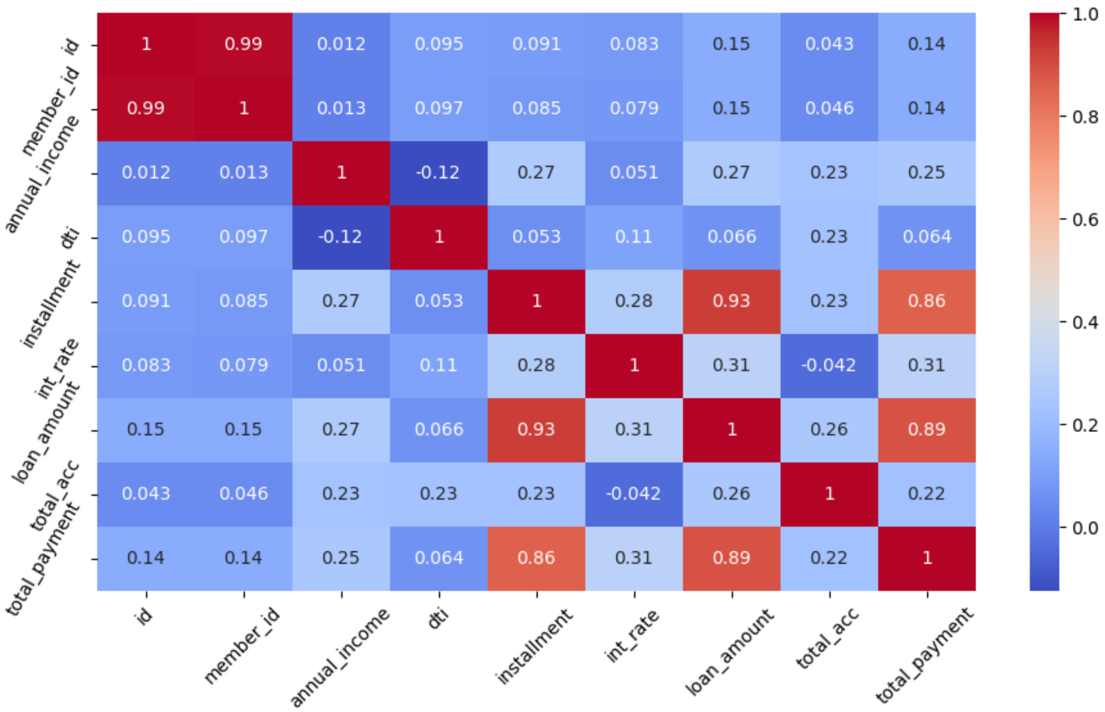


```
[53]: sb.boxplot(data=df,x='application_type',y='issue_date')
```

```
[53]: <Axes: xlabel='application_type', ylabel='issue_date'>
```



[54]:
```
plt.figure(figsize=(10,6))
corr=df.corr(numeric_only=True)
sb.heatmap(corr,annot=True,cmap="coolwarm")
plt.tight_layout()
plt.xticks(rotation=45)
plt.yticks(rotation=60)
plt.show()
```



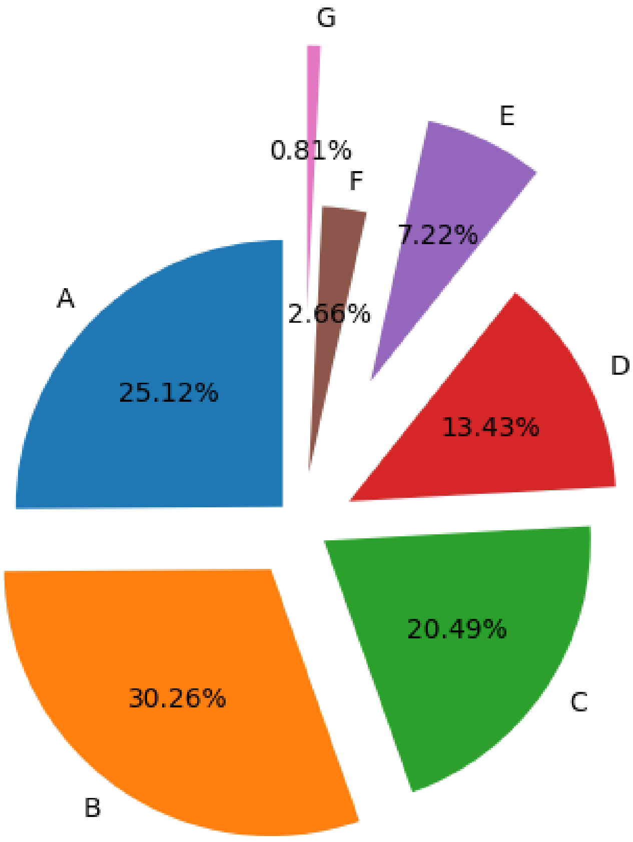
[55]:
```
df.columns
```

[55]:
```
Index(['id', 'address_state', 'application_type', 'emp_length', 'emp_title',
       'grade', 'home_ownership', 'issue_date', 'last_credit_pull_date',
       'last_payment_date', 'loan_status', 'next_payment_date', 'member_id',
       'purpose', 'sub_grade', 'term', 'verification_status', 'annual_income',
       'dti', 'installment', 'int_rate', 'loan_amount', 'total_acc',
       'total_payment'],
      dtype='object')
```

[57]:
```
df['grade'].unique()
```

[57]:
```
array(['C', 'E', 'B', 'A', 'D', 'F', 'G'], dtype=object)
```

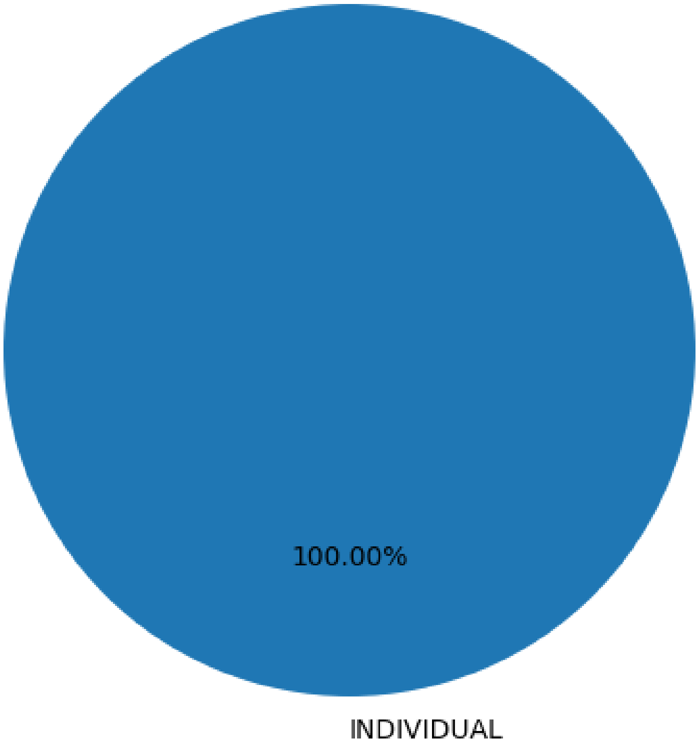
[65]:
```
counts=df['grade'].value_counts()
counts=counts.sort_index()
explodes=[0.1,0.2,0.1,0.2,0.6,0.2,0.8]
plt.pie(counts.values,labels=counts.index,autopct="%1.
  ↪2f%%",startangle=90,explode=explodes)

plt.tight_layout()
plt.show()
```



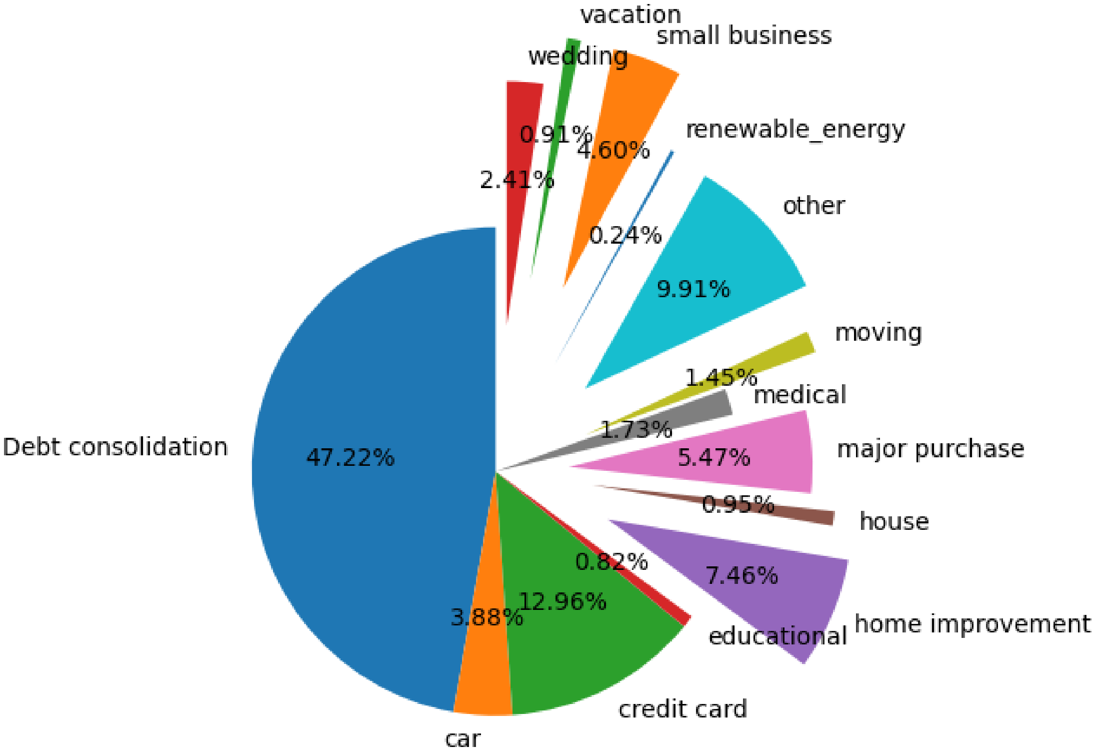
[66]:
```
df.columns
```

[66]:
```
Index(['id', 'address_state', 'application_type', 'emp_length', 'emp_title',
       'grade', 'home_ownership', 'issue_date', 'last_credit_pull_date',
       'last_payment_date', 'loan_status', 'next_payment_date', 'member_id',
       'purpose', 'sub_grade', 'term', 'verification_status', 'annual_income',
       'dti', 'installment', 'int_rate', 'loan_amount', 'total_acc',
       'total_payment'],
      dtype='object')
```

[68]:
```
counts=df['application_type'].value_counts()
counts=counts.sort_index()
plt.pie(counts.values,labels=counts.index,autopct="%1.2f%%",startangle=90)

plt.tight_layout()
plt.show()
```



[82]:
```
counts=df['purpose'].value_counts()
counts=counts.sort_index()
explodes=[0.0,0.0,0.0,0.0,0.5,0.4,0.3,0.0,0.4,0.5,0.5,0.8,0.8,0.6]
plt.pie(counts.values,labels=counts.index,autopct="%1.
  ↪2f%%",startangle=90,explode=explodes)
plt.tight_layout()
plt.show()
```



```
[83]: df.columns
```

```
[83]: Index(['id', 'address_state', 'application_type', 'emp_length', 'emp_title',
             'grade', 'home_ownership', 'issue_date', 'last_credit_pull_date',
             'last_payment_date', 'loan_status', 'next_payment_date', 'member_id',
             'purpose', 'sub_grade', 'term', 'verification_status', 'annual_income',
             'dti', 'installment', 'int_rate', 'loan_amount', 'total_acc',
             'total_payment'],
            dtype='object')
```

```
[84]: df['diff_Amt']=df['total_payment']-df['loan_amount']
```

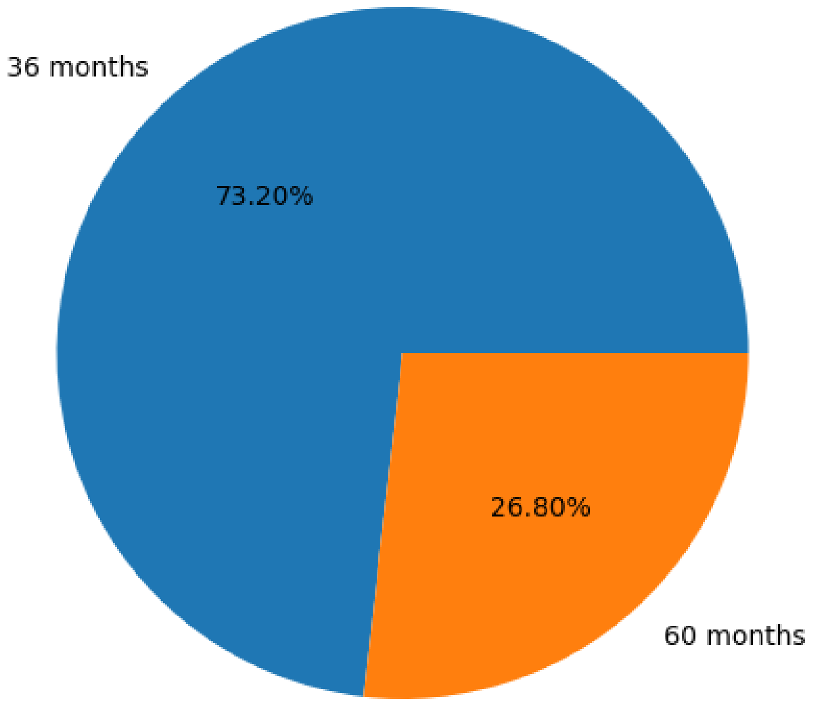
```
[85]: df.head()
```

```
[85]:         id address_state application_type emp_length            emp_title  \
      0  1077430            GA       INDIVIDUAL     1 year                Ryder
      1  1072053            CA       INDIVIDUAL    9 years       MKC Accounting
      2  1069243            CA       INDIVIDUAL    4 years  Chemat Technology Inc
      3  1041756            TX       INDIVIDUAL     1 year    barnes distribution
      4  1068350            IL       INDIVIDUAL    10 year          J&J Steel Inc

        grade home_ownership issue_date last_credit_pull_date last_payment_date  \
0     C            RENT 2021-02-11           2021-09-13        2021-04-13
1     E            RENT 2021-01-01           2021-12-14        2021-01-15
2     C            RENT 2021-01-05           2021-12-12        2021-01-09
3     B        MORTGAGE 2021-02-25           2021-12-12        2021-03-12
4     A        MORTGAGE 2021-01-01           2021-12-14        2021-01-15

   …        term verification_status  annual_income     dti installment  \
0  …   60 months     Source Verified        30000.0  0.0100       59.83
1  …   36 months     Source Verified        48000.0  0.0535      109.43
2  …   36 months        Not Verified        50000.0  0.2088      421.65
3  …   60 months     Source Verified        42000.0  0.0540       97.06
4  …   36 months            Verified        83000.0  0.0231      106.53

  int_rate loan_amount  total_acc  total_payment  diff_Amt
0   0.1527        2500          4           1009     -1491
1   0.1864        3000          4           3939       939
2   0.1596       12000         11           3522     -8478
3   0.1065        4500          9           4911       411
4   0.0603        3500         28           3835       335

[5 rows x 25 columns]
```

[97]:
```
cat_cols=df[['term']]
for col in cat_cols:
    print(f"value counts for '{col}':")
    display(df[col].value_counts().plot(kind='pie',autopct="%1.2f%%"))
plt.tight_layout()
plt.xlabel('')
plt.ylabel(" ")
```

```
value counts for 'term':

<Axes: ylabel='count'>
```

[97]:
```
Text(137.644444444444445, 0.5, ' ')
```


36 months
73.20%
26.80%
60 months

```
[ ]:
```