

DATA 6320 - Scenario 2

Python Videos

- [Importing data from a Database](#)
- [Formatting Data](#)
- [Explore the Data](#)
- [Drop columns that are not needed](#)
- [Remove rows with null values](#)
- [Clean observations - Fill NaNs with a specific value](#)
- [Clean Observations - Fill in NaNs with mean or median](#)
- [Recode Features with Replacement](#)
- [Clean Observations - Fill in NaNs with grouped mean](#)
- [Record with a translation table](#)

Different techniques that this is getting prepped for:

- Regression
 - Simple regression
 - Multiple regression with backwards elimination
 - Lasso regression
 - Ridge regression
- Classification
 - Logistic Regression
 - Decision Tree
 - Random Forest

Summary of Notebook

Contents

- **Business Understanding**
 - Available resources, problems, goals
- **Data Understanding**
 - What data do you have available to you?
 - Install or Load tools or applications
 - Programming – Jupyter notebooks for Python or R Studio for R
 - BI/spreadsheets – Excel – PowerPivot - Tableau

Data Preprocessing -----

- - Import or download the data
 - Format the data
 - View, explore, and summarize the data
- **Data Preparation**
 - Remove Columns
 - Remove Rows
 - Fill in null values
 - Replace or remove mistakes
 - Remove outliers
 - Recode categorical or numerical features
 - Construct new data feature engineering
 - For Supervised Learning, create X and Y
- **Modeling**
 - Split the data (Train/Test Split)
 - Transform the data

Data Preprocessing -----

- - Setup models for machine learning/AI processes
 - Can also include developing the outline for visuals, dashboards or reports
- **Evaluation**
 - Hyper-parameter tuning
- **Deployment of models**

BUSINESS UNDERSTANDING

Business Objective

- What is the relationship between annual income and loan amount. Is it a good predictor?
- Can we predict the amount for a loan?
- What features are most important in predicting loan amount?

Technical Objective

- Review different data cleansing techniques to continue to improve
- Conduct a simple regression using scikit-learn
- Conduct a multiple regression using statsmodels
- Learn and perform Lasso and Ridge regression and tune its parameters

DATA UNDERSTANDING

[Top](#)

Importing data and viewing its contents

Tables from Database

- customers (Customers from 2018 to current)
- loanstatus (loanstatus with coding for bad loan)
- pre2018 (Customers before 2018)
- reason (reason for loan with coding so that all categories are the same)

To do with database

- import tables
- set the data types
- concatenate customers and pre2018

Import Libraries

```
In [1]: #Code Block 01
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

pd.set_option('display.max_columns',500)

plt.style.use('seaborn-colorblind') #a style that can be used for plots

sns.set_style('whitegrid')
```

Import the data and create dataframes that can then be cleansed, recoded, transformed, and split.

Import Data

Use SQLAlchemy

SQLAlchemy is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL.

It provides a full suite of well known enterprise-level persistence patterns, designed for efficient and high-performing database access, adapted into a simple and Pythonic domain language.

To import a SQL file:

- create an engine
- create a connection
- execute SQL statement to retrieve data
 - create a DataFrame to store the data
 - pull in the keys to use as column names
- close connection

<https://www.sqlalchemy.org/library.html#tutorials> (<https://www.sqlalchemy.org/library.html#tutorials>)

<https://docs.sqlalchemy.org/en/13/dialects/sqlite.html> (<https://docs.sqlalchemy.org/en/13/dialects/sqlite.html>)

```
In [2]: #Code Block 02
from sqlalchemy import create_engine
```

```
In [3]: #Code Block 03
engine = create_engine('sqlite:///data/Appleton.db')
```

```
In [4]: #Code Block 04
con = engine.connect()
print('connection is ok')

connection is ok
```

```
In [5]: #Code Block 05
print(engine.table_names())

['customers', 'loanstatus', 'pre2018', 'reason']
```

```
In [6]: #Code Block 06
rs = con.execute("SELECT * FROM customers")
```

```
In [7]: #Code Block 07
```

```
df_customers = pd.DataFrame(rs.fetchall()) ##fetches all data from the customers table
display(df_customers.head(2))
df_customers.info()
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	1581986	9000	12/22/18	36	12.12	299.45	45000	3		0	11	0	7341	15	0	10696.3	9000	9595.47	1696.3	
1	1751708	6625	4/14/18	36	11.14	217.34	28000	1	0	23	0	8	0	3493	14	0	6302.35	5164.37	4215.79	1137.98

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18262 entries, 0 to 18261
Data columns (total 42 columns):
 #   Column  Non-Null Count  Dtype  
---  --  
 0   0        18262 non-null   object 
 1   1        18262 non-null   object 
 2   2        18262 non-null   object 
 3   3        18262 non-null   object 
 4   4        18262 non-null   object 
 5   5        18262 non-null   object 
 6   6        18262 non-null   object 
 7   7        18262 non-null   object 
 8   8        18262 non-null   object 
 9   9        18262 non-null   object 
 10  10       18262 non-null   object 
 11  11       18262 non-null   object 
 12  12       18262 non-null   object 
 13  13       18262 non-null   object 
 14  14       18262 non-null   object 
 15  15       18262 non-null   object 
 16  16       18262 non-null   object 
 17  17       18262 non-null   object 
 18  18       18262 non-null   object 
 19  19       18262 non-null   object 
 20  20       18262 non-null   object 
 21  21       18262 non-null   object 
 22  22       18262 non-null   object 
 23  23       18262 non-null   object 
 24  24       18262 non-null   object 
 25  25       18262 non-null   object 
 26  26       18262 non-null   object 
 27  27       18262 non-null   object 
 28  28       18262 non-null   object 
 29  29       18262 non-null   object 
 30  30       18262 non-null   object 
 31  31       18262 non-null   object 
 32  32       18262 non-null   object 
 33  33       18262 non-null   object 
 34  34       18262 non-null   object 
 35  35       18262 non-null   object 
 36  36       18262 non-null   object 
 37  37       18262 non-null   object 
 38  38       18262 non-null   object 
 39  39       18262 non-null   object 
 40  40       18262 non-null   object 
 41  41       18262 non-null   object 
dtypes: object(42)
memory usage: 5.9+ MB
```

```
In [8]: #Code Block 08
```

```
##adds column names to df
df_customers.columns = rs.keys()
```

```
In [9]: #Code Block 09
```

```
display(df_customers.head(2))
df_customers.info()
```

	member_id	loan_amnt	orig_date	term	int_rate	installment	annual_inc	delinq_2yrs	inq_last_6mths	mths_since_last_delinq
0	1581986	9000	12/22/18	36	12.12	299.45	45000			3
1	1751708	6625	4/14/18	36	11.14	217.34	28000		1	0


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18262 entries, 0 to 18261
Data columns (total 42 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        18262 non-null   object 
 1   loan_amnt       18262 non-null   object 
 2   orig_date        18262 non-null   object 
 3   term             18262 non-null   object 
 4   int_rate         18262 non-null   object 
 5   installment      18262 non-null   object 
 6   annual_inc       18262 non-null   object 
 7   delinq_2yrs      18262 non-null   object 
 8   inq_last_6mths   18262 non-null   object 
 9   mths_since_last_delinq  18262 non-null   object 
 10  mths_since_last_record 18262 non-null   object 
 11  open_acc         18262 non-null   object 
 12  pub_rec          18262 non-null   object 
 13  revol_bal        18262 non-null   object 
 14  total_acc        18262 non-null   object 
 15  out_prncp        18262 non-null   object 
 16  total_pymnt      18262 non-null   object 
 17  total_rec_prncp  18262 non-null   object 
 18  total_debt_paid  18262 non-null   object 
 19  total_rec_int    18262 non-null   object 
 20  princ_int_ratio  18262 non-null   object 
 21  total_rec_late_fee 18262 non-null   object 
 22  recoveries       18262 non-null   object 
 23  collection_recovery_fee 18262 non-null   object 
 24  last_pymnt_amnt 18262 non-null   object 
 25  collections_12_mths_ex_med 18262 non-null   object 
 26  mths_since_last_major_derog 18262 non-null   object 
 27  acc_now_delinq   18262 non-null   object 
 28  tot_coll_amt    18262 non-null   object 
 29  tot_cur_bal     18262 non-null   object 
 30  total_credit_rv 18262 non-null   object 
 31  revol_util      18262 non-null   object 
 32  sub_grade        18262 non-null   object 
 33  emp_length       18262 non-null   object 
 34  home_ownership   18262 non-null   object 
 35  loan_status      18262 non-null   object 
 36  initial_list_status 18262 non-null   object 
 37  months_since_issue 18262 non-null   object 
 38  months_since_payment 18262 non-null   object 
 39  months_since_last_credit_pull 18262 non-null   object 
 40  months_since_earliest_credit 18262 non-null   object 
 41  reason           18262 non-null   object 

dtypes: object(42)
memory usage: 5.9+ MB
```

Formatting Data

What happened to the null values?

In [10]: #Code Block 10

```
df_customers = df_customers.replace(' ', np.nan)
df_customers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18262 entries, 0 to 18261
Data columns (total 42 columns):
 #   Column           Non-Null Count  Dtype  
 ---  --  
 0   member_id        18258 non-null   float64
 1   loan_amnt        18258 non-null   float64
 2   orig_date         18258 non-null   object 
 3   term              18258 non-null   float64
 4   int_rate          18258 non-null   float64
 5   installment       18258 non-null   float64
 6   annual_inc        18258 non-null   float64
 7   delinq_2yrs       2228  non-null   float64
 8   inq_last_6mths    18258 non-null   float64
 9   mths_since_last_delinq  6676  non-null   float64
 10  mths_since_last_record 18258 non-null   float64
 11  open_acc          18253 non-null   float64
 12  pub_rec           18258 non-null   float64
 13  revol_bal         18258 non-null   float64
 14  total_acc          18258 non-null   float64
 15  out_prncp         18258 non-null   float64
 16  total_pymnt        18258 non-null   float64
 17  total_rec_prncp    18258 non-null   float64
 18  total_debt_paid    18258 non-null   float64
 19  total_rec_int       18258 non-null   float64
 20  princ_int_ratio     18258 non-null   float64
 21  total_rec_late_fee  18258 non-null   float64
 22  recoveries         18258 non-null   float64
 23  collection_recovery_fee 18258 non-null   float64
 24  last_pymnt_amnt    18258 non-null   float64
 25  collections_12_mths_ex_med 18258 non-null   float64
 26  mths_since_last_major_derog 18258 non-null   float64
 27  acc_now_delinq      18258 non-null   float64
 28  tot_coll_amt        18258 non-null   float64
 29  tot_cur_bal          18258 non-null   float64
 30  total_credit_rv      18258 non-null   float64
 31  revol_util          18258 non-null   float64
 32  sub_grade            18258 non-null   object 
 33  emp_length           18258 non-null   float64
 34  home_ownership       18258 non-null   object 
 35  loan_status           18258 non-null   object 
 36  initial_list_status    18258 non-null   object 
 37  months_since_issue     18258 non-null   float64
 38  months_since_payment    18258 non-null   float64
 39  months_since_last_credit_pull 18258 non-null   float64
 40  months_since_earliest_credit 18258 non-null   float64
 41  reason               18258 non-null   object 

dtypes: float64(36), object(6)
memory usage: 5.9+ MB
```

```
In [11]: #Code Block 11
```

```
#import all columns and records
rs = con.execute("SELECT * FROM pre2018")

#Changes to DataFrame
df_pre2018 = pd.DataFrame(rs.fetchall())

#Adds headers to each
df_pre2018.columns = rs.keys()

#Set all "" values to NaN
df_pre2018 = df_pre2018.replace(' ', np.nan)

display(df_pre2018.head(2))
df_pre2018.info()
```

	member_id	loan_amnt	orig_date	term	int_rate	installment	annual_inc	delinq_2yrs	inq_last_6mths	mths_since_last_delinq	
0	507531	35000.0	8/8/17	36.0	10.16	1131.99	130000.0	NaN	1.0	NaN	
1	513904	21000.0	1/21/17	36.0	6.03	639.15	120000.0	NaN	0.0	NaN	

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18880 entries, 0 to 18879
Data columns (total 42 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        18880 non-null   int64  
 1   loan_amnt       18878 non-null   float64 
 2   orig_date        18878 non-null   object  
 3   term             18878 non-null   float64 
 4   int_rate          18878 non-null   float64 
 5   installment       18878 non-null   float64 
 6   annual_inc       18878 non-null   float64 
 7   delinq_2yrs       3587  non-null   float64 
 8   inq_last_6mths    18874 non-null   float64 
 9   mths_since_last_delinq  10070 non-null   float64 
 10  mths_since_last_record 18874 non-null   float64 
 11  open_acc          18865 non-null   float64 
 12  pub_rec            18874 non-null   float64 
 13  revol_bal          18874 non-null   float64 
 14  total_acc          18843 non-null   float64 
 15  out_prncp          18874 non-null   float64 
 16  total_pymnt         18874 non-null   float64 
 17  total_rec_prncp     18874 non-null   float64 
 18  total_debt_paid      18874 non-null   float64 
 19  total_rec_int         18874 non-null   float64 
 20  princ_int_ratio       18874 non-null   float64 
 21  total_rec_late_fee     18874 non-null   float64 
 22  recoveries          18874 non-null   float64 
 23  collection_recovery_fee 18874 non-null   float64 
 24  last_pymnt_amnt       18874 non-null   float64 
 25  collections_12_mths_ex_med 18874 non-null   float64 
 26  mths_since_last_major_derog 18874 non-null   float64 
 27  acc_now_delinq        18874 non-null   float64 
 28  tot_coll_amt          18874 non-null   float64 
 29  tot_cur_bal            18874 non-null   float64 
 30  total_credit_rv        18874 non-null   float64 
 31  revol_util            18874 non-null   float64 
 32  sub_grade              18874 non-null   object  
 33  emp_length             18874 non-null   float64 
 34  home_ownership          18874 non-null   object  
 35  loan_status             18874 non-null   object  
 36  initial_list_status      18874 non-null   object  
 37  months_since_issue       18874 non-null   float64 
 38  months_since_payment      18874 non-null   float64 
 39  months_since_last_credit_pull 18874 non-null   float64 
 40  months_since_earliest_credit 18874 non-null   float64 
 41  reason                  18874 non-null   object  
dtypes: float64(35), int64(1), object(6)
memory usage: 6.0+ MB

```

```
In [12]: #Code Block 12
```

```
rs = con.execute("SELECT * FROM loanstatus")
df_loanstatus = pd.DataFrame(rs.fetchall())
df_loanstatus.columns = rs.keys()
display(df_loanstatus)
df_loanstatus.info()
```

	loan_status	loan_is_bad
0	Charged Off	1
1	Current	0
2	Fully Paid	0
3	In Grace Period	0
4	Late (16-30 days)	0
5	Late (31-120 days)	1
6	Default	1

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   loan_status    7 non-null       object 
 1   loan_is_bad    7 non-null       int64  
dtypes: int64(1), object(1)
memory usage: 240.0+ bytes
```

```
In [13]: #Code Block 13
```

```
rs = con.execute("SELECT * FROM reason")
df_reason = pd.DataFrame(rs.fetchall())
df_reason.columns = rs.keys()
display(df_reason.head(9))
df_reason.info()
```

	reason_old	reason_recode
0	cc	credit_card
1	debtcon	debt_consolidation
2	other	other
3	pers	personal
4	med	medical
5	credit_card	credit_card
6	debt_consolidation	debt_consolidation
7	medical	medical
8	personal	personal

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   reason_old    9 non-null       object 
 1   reason_recode 9 non-null       object 
dtypes: object(2)
memory usage: 272.0+ bytes
```

```
In [14]: #Code Block 14
```

```
df_loandata = pd.concat([df_customers, df_pre2018], axis = 0)
df_loandata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 37142 entries, 0 to 18879
Data columns (total 42 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        37138 non-null   float64
 1   loan_amnt       37136 non-null   float64
 2   orig_date        37136 non-null   object 
 3   term             37136 non-null   float64
 4   int_rate          37136 non-null   float64
 5   installment       37136 non-null   float64
 6   annual_inc       37136 non-null   float64
 7   delinq_2yrs       5815 non-null   float64
 8   inq_last_6mths    37132 non-null   float64
 9   mths_since_last_delinq  16746 non-null   float64
 10  mths_since_last_record 37132 non-null   float64
 11  open_acc          37118 non-null   float64
 12  pub_rec            37132 non-null   float64
 13  revol_bal         37132 non-null   float64
 14  total_acc          37101 non-null   float64
 15  out_prncp         37132 non-null   float64
 16  total_pymnt        37132 non-null   float64
 17  total_rec_prncp    37132 non-null   float64
 18  total_debt_paid    37132 non-null   float64
 19  total_rec_int       37132 non-null   float64
 20  princ_int_ratio     37132 non-null   float64
 21  total_rec_late_fee  37132 non-null   float64
 22  recoveries          37132 non-null   float64
 23  collection_recovery_fee 37132 non-null   float64
 24  last_pymnt_amnt     37132 non-null   float64
 25  collections_12_mths_ex_med 37132 non-null   float64
 26  mths_since_last_major_derog 37132 non-null   float64
 27  acc_now_delinq      37132 non-null   float64
 28  tot_coll_amt        37132 non-null   float64
 29  tot_cur_bal          37132 non-null   float64
 30  total_credit_rv      37132 non-null   float64
 31  revol_util          37132 non-null   float64
 32  sub_grade            37132 non-null   object 
 33  emp_length           37132 non-null   float64
 34  home_ownership        37132 non-null   object 
 35  loan_status           37132 non-null   object 
 36  initial_list_status    37132 non-null   object 
 37  months_since_issue     37132 non-null   float64
 38  months_since_payment    37132 non-null   float64
 39  months_since_last_credit_pull 37132 non-null   float64
 40  months_since_earliest_credit 37132 non-null   float64
 41  reason               37132 non-null   object 

dtypes: float64(36), object(6)
memory usage: 12.2+ MB
```

```
In [15]: df_loandata.loc[[55, 66], :]
```

Out[15]:

	member_id	loan_amnt	orig_date	term	int_rate	installment	annual_inc	delinq_2yrs	inq_last_6mths	mths_since_last_delinq
55	1719288.0	12000.0	1/13/18	36.0	12.12	399.26	70000.0	NaN	0.0	NaN
55	1479194.0	10000.0	12/28/16	36.0	11.14	328.06	95000.0	NaN	1.0	70.1
66	1696986.0	12000.0	7/5/18	36.0	7.62	373.94	120000.0	NaN	1.0	NaN
66	1531387.0	14400.0	10/27/17	36.0	12.12	479.12	100000.0	NaN	0.0	NaN

```
In [16]: df_loandata = df_loandata.reset_index(drop=True)
df_loandata.head()
```

Out[16]:

	member_id	loan_amnt	orig_date	term	int_rate	installment	annual_inc	delinq_2yrs	inq_last_6mths	mths_since_last_delinq
0	1581986.0	9000.0	12/22/18	36.0	12.12	299.45	45000.0	NaN	3.0	NaN
1	1751708.0	6625.0	4/14/18	36.0	11.14	217.34	28000.0	1.0	0.0	23.0
2	1666916.0	9800.0	8/25/18	36.0	12.12	326.07	50000.0	NaN	0.0	NaN
3	1758003.0	4250.0	3/7/18	36.0	8.90	134.96	38000.0	2.0	3.0	21.0
4	1730191.0	16000.0	4/22/18	36.0	7.90	500.65	60000.0	NaN	0.0	28.0

```
In [17]: df_loandata.loc[[55, 66], :]
```

Out[17]:

	member_id	loan_amnt	orig_date	term	int_rate	installment	annual_inc	delinq_2yrs	inq_last_6mths	mths_since_last_delinq
55	1719288.0	12000.0	1/13/18	36.0	12.12	399.26	70000.0	NaN	0.0	NaN
66	1696986.0	12000.0	7/5/18	36.0	7.62	373.94	120000.0	NaN	1.0	NaN

In [18]: #Code Block 15

```
df_loandata = df_loandata.convert_dtypes()
df_loandata.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37142 entries, 0 to 37141
Data columns (total 42 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        37138 non-null   Int64  
 1   loan_amnt       37136 non-null   Int64  
 2   orig_date       37136 non-null   string  
 3   term            37136 non-null   Int64  
 4   int_rate         37136 non-null   float64 
 5   installment      37136 non-null   float64 
 6   annual_inc      37136 non-null   float64 
 7   delinq_2yrs      5815 non-null   Int64  
 8   inq_last_6mths   37132 non-null   Int64  
 9   mths_since_last_delinq  16746 non-null   Int64  
 10  mths_since_last_record 37132 non-null   Int64  
 11  open_acc         37118 non-null   Int64  
 12  pub_rec          37132 non-null   Int64  
 13  revol_bal        37132 non-null   Int64  
 14  total_acc        37101 non-null   Int64  
 15  out_prncp        37132 non-null   float64 
 16  total_pymnt      37132 non-null   float64 
 17  total_rec_prncp  37132 non-null   float64 
 18  total_debt_paid  37132 non-null   float64 
 19  total_rec_int    37132 non-null   float64 
 20  princ_int_ratio  37132 non-null   float64 
 21  total_rec_late_fee 37132 non-null   float64 
 22  recoveries       37132 non-null   float64 
 23  collection_recovery_fee 37132 non-null   float64 
 24  last_pymnt_amnt 37132 non-null   float64 
 25  collections_12_mths_ex_med 37132 non-null   Int64  
 26  mths_since_last_major_derog 37132 non-null   Int64  
 27  acc_now_delinq   37132 non-null   Int64  
 28  tot_coll_amt     37132 non-null   Int64  
 29  tot_cur_bal      37132 non-null   Int64  
 30  total_credit_rv  37132 non-null   Int64  
 31  revol_util       37132 non-null   float64 
 32  sub_grade        37132 non-null   string  
 33  emp_length       37132 non-null   Int64  
 34  home_ownership   37132 non-null   string  
 35  loan_status       37132 non-null   string  
 36  initial_list_status 37132 non-null   string  
 37  months_since_issue 37132 non-null   Int64  
 38  months_since_payment 37132 non-null   Int64  
 39  months_since_last_credit_pull 37132 non-null   Int64  
 40  months_since_earliest_credit 37132 non-null   Int64  
 41  reason           37132 non-null   string  
dtypes: Int64(22), float64(14), string(6)
memory usage: 12.7 MB
```

In [19]: #Code Block 16

```
df_loandata.head()
```

Out[19]:

	member_id	loan_amnt	orig_date	term	int_rate	installment	annual_inc	delinq_2yrs	inq_last_6mths	mths_since_last_delinq
0	1581986	9000	12/22/18	36	12.12	299.45	45000.0	<NA>	3	<NA>
1	1751708	6625	4/14/18	36	11.14	217.34	28000.0	1	0	23
2	1666916	9800	8/25/18	36	12.12	326.07	50000.0	<NA>	0	<NA>
3	1758003	4250	3/7/18	36	8.90	134.96	38000.0	2	3	21
4	1730191	16000	4/22/18	36	7.90	500.65	60000.0	<NA>	0	28

```
In [20]: #Code Block 17
```

```
df_loandata['orig_date'] = pd.to_datetime(df_loandata['orig_date'])
df_loandata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37142 entries, 0 to 37141
Data columns (total 42 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   member_id        37138 non-null   Int64  
 1   loan_amnt        37136 non-null   Int64  
 2   orig_date        37136 non-null   datetime64[ns] 
 3   term             37136 non-null   Int64  
 4   int_rate          37136 non-null   float64 
 5   installment       37136 non-null   float64 
 6   annual_inc        37136 non-null   float64 
 7   delinq_2yrs       5815 non-null   Int64  
 8   inq_last_6mths    37132 non-null   Int64  
 9   mths_since_last_delin 16746 non-null   Int64  
 10  mths_since_last_record 37132 non-null   Int64  
 11  open_acc          37118 non-null   Int64  
 12  pub_rec           37132 non-null   Int64  
 13  revol_bal         37132 non-null   Int64  
 14  total_acc          37101 non-null   Int64  
 15  out_prncp         37132 non-null   float64 
 16  total_pymnt        37132 non-null   float64 
 17  total_rec_prncp    37132 non-null   float64 
 18  total_debt_paid    37132 non-null   float64 
 19  total_rec_int        37132 non-null   float64 
 20  princ_int_ratio     37132 non-null   float64 
 21  total_rec_late_fee   37132 non-null   float64 
 22  recoveries          37132 non-null   float64 
 23  collection_recovery_fee 37132 non-null   float64 
 24  last_pymnt_amnt     37132 non-null   float64 
 25  collections_12_mths_ex_med 37132 non-null   Int64  
 26  mths_since_last_major_derog 37132 non-null   Int64  
 27  acc_now_delinq      37132 non-null   Int64  
 28  tot_coll_amt        37132 non-null   Int64  
 29  tot_cur_bal          37132 non-null   Int64  
 30  total_credit_rv      37132 non-null   Int64  
 31  revol_util          37132 non-null   float64 
 32  sub_grade           37132 non-null   string  
 33  emp_length          37132 non-null   Int64  
 34  home_ownership       37132 non-null   string  
 35  loan_status          37132 non-null   string  
 36  initial_list_status    37132 non-null   string  
 37  months_since_issue     37132 non-null   Int64  
 38  months_since_payment    37132 non-null   Int64  
 39  months_since_last_credit_pull 37132 non-null   Int64  
 40  months_since_earliest_credit 37132 non-null   Int64  
 41  reason              37132 non-null   string  
dtypes: Int64(22), datetime64[ns](1), float64(14), string(5)
memory usage: 12.7 MB
```

[Top](#)

Explore the Data

Why explore before cleanse? Exploring and cleansing may be conducted simultaneously, but you should look at the data before you start manipulating the data.

<https://seaborn.pydata.org/examples/index.html> (<https://seaborn.pydata.org/examples/index.html>)

Describe the data

Shows a list of summary statistics - what to look for?

- Min and max for outliers
- Count to see how many columns have NaN values

<https://pandas.pydata.org/pandas-docs/stable/basics.html> (<https://pandas.pydata.org/pandas-docs/stable/basics.html>)

```
#Code Block 19 df_loandata[['loan_amnt', 'annual_inc', 'revol_bal', 'total_acc', 'tot_coll_amt']].corr()
```

In [21]: #Code Block 20

```
df_loandata_explore = df_loandata[['loan_amnt', 'emp_length', 'annual_inc', 'revol_bal', 'tot_c  
ur_bal', 'total_credit_rv', 'reason']]  
df_loandata_explore.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 37142 entries, 0 to 37141  
Data columns (total 7 columns):  
 #   Column           Non-Null Count  Dtype    
---  --    
 0   loan_amnt        37136 non-null   Int64  
 1   emp_length       37132 non-null   Int64  
 2   annual_inc       37136 non-null   float64  
 3   revol_bal        37132 non-null   Int64  
 4   tot_cur_bal      37132 non-null   Int64  
 5   total_credit_rv  37132 non-null   Int64  
 6   reason           37132 non-null   string  
dtypes: Int64(5), float64(1), string(1)  
memory usage: 2.2 MB
```

In [22]: #Code Block 21

```
df_loandata_explore = df_loandata_explore.dropna(how='any')  
df_loandata_explore.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 37132 entries, 0 to 37141  
Data columns (total 7 columns):  
 #   Column           Non-Null Count  Dtype    
---  --    
 0   loan_amnt        37132 non-null   Int64  
 1   emp_length       37132 non-null   Int64  
 2   annual_inc       37132 non-null   float64  
 3   revol_bal        37132 non-null   Int64  
 4   tot_cur_bal      37132 non-null   Int64  
 5   total_credit_rv  37132 non-null   Int64  
 6   reason           37132 non-null   string  
dtypes: Int64(5), float64(1), string(1)  
memory usage: 2.4 MB
```

```
In [23]: #Code Block 22
```

```
df_loandata_explore.corr()
```

```
Out[23]:
```

	loan_amnt	emp_length	annual_inc	revol_bal	tot_cur_bal	total_credit_rv
loan_amnt	1.000000	0.142102	0.294602	0.359271	0.238812	0.271002
emp_length	0.142102	1.000000	0.086283	0.120366	0.096401	0.084107
annual_inc	0.294602	0.086283	1.000000	0.375441	0.404928	0.275902
revol_bal	0.359271	0.120366	0.375441	1.000000	0.413173	0.614834
tot_cur_bal	0.238812	0.096401	0.404928	0.413173	1.000000	0.558259
total_credit_rv	0.271002	0.084107	0.275902	0.614834	0.558259	1.000000

```
In [24]: #Code Block 23
```

```
sns.pairplot(df_loandata_explore, hue = 'reason')
```

```
Out[24]: <seaborn.axisgrid.PairGrid at 0x7f8ddaa3f3790>
```



```
In [25]: #Code Block 24
```

```
df_loandata_explore_num = df_loandata_explore.drop('reason', axis = 1)
```

In [26]: #Code Block 25

```
colormap = plt.cm.viridis
plt.figure(figsize=(14,12))
plt.title('Pearson Correlation of Features', y=1.05, size=15)
sns.heatmap(df_loandata_explore_num.astype(float).corr(), linewidths=0.1,vmax=1.0, square=True,
cmap=colormap, linecolor='white', annot=True)
```

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8ddb1ab310>



```
In [27]: #Code Block 18
```

```
df_loandata.describe()
```

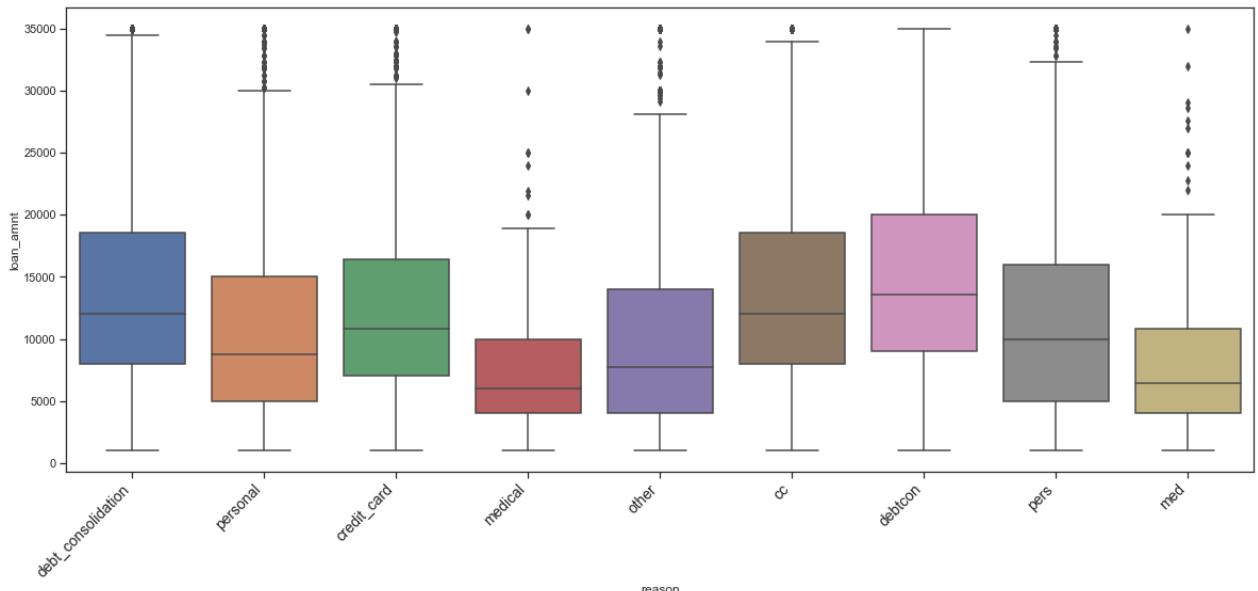
```
Out[27]:
```

	member_id	loan_amnt	term	int_rate	installment	annual_inc	delinq_2yrs	inq_last_6mths	mths
count	3.713800e+04	37136.000000	37136.000000	37136.000000	37136.000000	3.713600e+04	5815.000000	37132.000000	
mean	1.974723e+06	13479.738663	40.391426	14.034174	425.204761	6.986643e+04	1.490800	0.847786	
std	4.209572e+05	8019.532700	9.279651	4.338130	245.057149	6.390174e+04	1.072153	1.016835	
min	1.495120e+05	1000.000000	36.000000	6.000000	25.810000	5.000000e+03	1.000000	0.000000	
25%	1.692587e+06	7200.000000	36.000000	11.140000	239.560000	4.338450e+04	1.000000	0.000000	
50%	1.814895e+06	12000.000000	36.000000	14.090000	383.895000	6.000000e+04	1.000000	1.000000	
75%	2.036799e+06	18225.000000	36.000000	17.270000	553.110000	8.400000e+04	2.000000	1.000000	
max	3.426825e+06	35000.000000	60.000000	24.890000	1388.450000	7.141778e+06	18.000000	8.000000	

```
In [28]: #Code Block 27
```

```
plt.figure(figsize=(20,8))
sns.set(style="ticks")
chart = sns.boxplot(y='loan_amnt', x = 'reason', data=df_loandata_explore)
chart.set_xticklabels(chart.get_xticklabels(), rotation=45, horizontalalignment='right', fontsize=14)
```

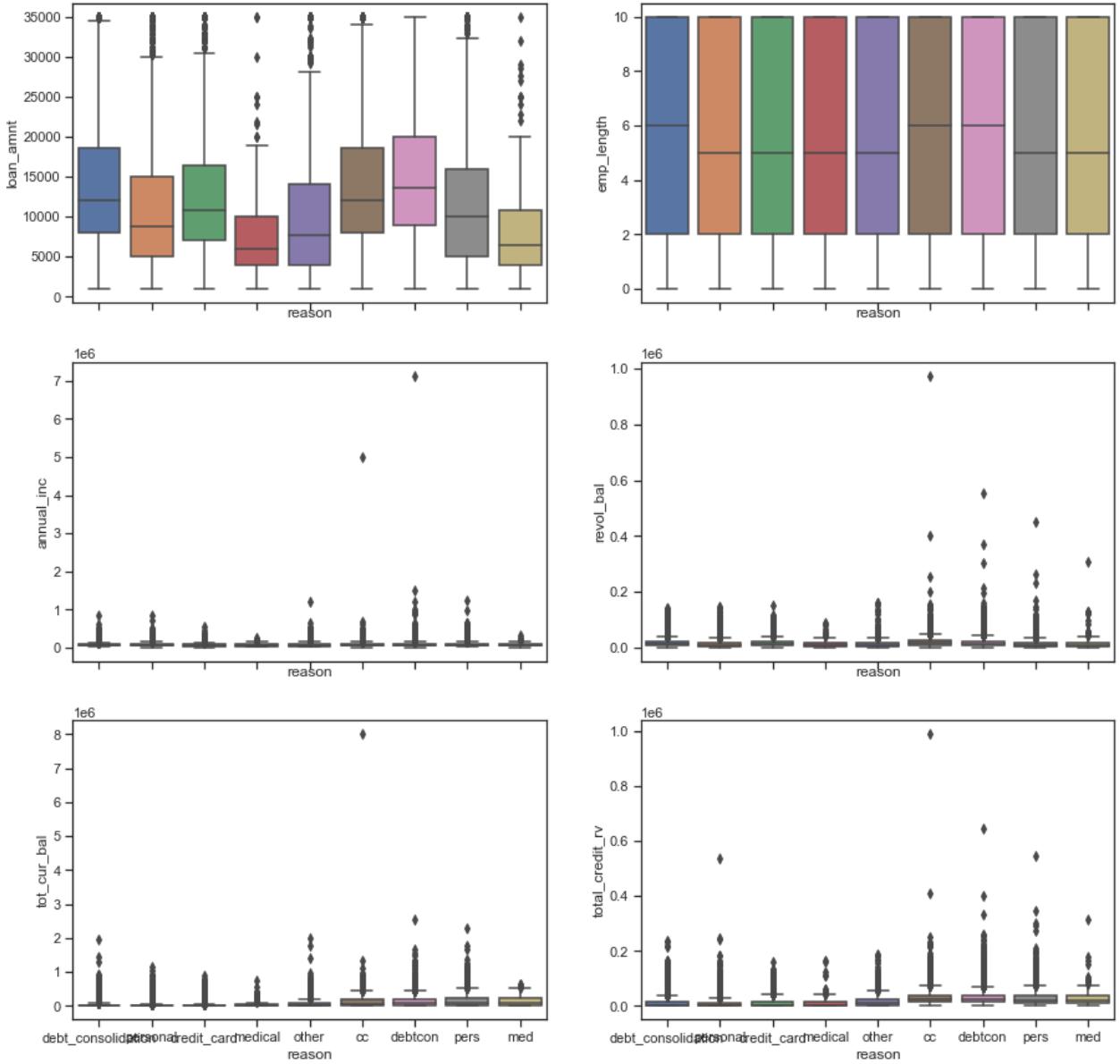
```
Out[28]: [Text(0, 0, 'debt_consolidation'),
Text(0, 0, 'personal'),
Text(0, 0, 'credit_card'),
Text(0, 0, 'medical'),
Text(0, 0, 'other'),
Text(0, 0, 'cc'),
Text(0, 0, 'debtcon'),
Text(0, 0, 'pers'),
Text(0, 0, 'med')]
```



In [29]: #Code Block 28

```
f, axes = plt.subplots(3, 2, figsize=(15, 15), sharex=True)
sns.boxplot(y='loan_amnt', x = 'reason', data=df_loandata_explore, ax=axes[0,0])
sns.boxplot(y='emp_length', x = 'reason', data=df_loandata_explore, ax=axes[0,1])
sns.boxplot(y='annual_inc', x = 'reason', data=df_loandata_explore, ax=axes[1,0])
sns.boxplot(y='revol_bal', x = 'reason', data=df_loandata_explore, ax=axes[1,1])
sns.boxplot(y='tot_cur_bal', x = 'reason', data=df_loandata_explore, ax=axes[2,0])
sns.boxplot(y='total_credit_rv', x = 'reason', data=df_loandata_explore, ax=axes[2,1])
```

Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8dc1865d90>



DATA PREPARATION

Drop columns that are not needed

Delete before data cleansing

Features that are not known when prediction is made:

- installment
- total_pymnt
- last_pymnt_amnt
- months_since_issue
- months_since_payment
- months_since_last_credit_pull
- months_since_earliest_credit
- total_rec_late_fee
- recoveries
- collection_recovery_fee
- total_rec_prncp
- total_rec_int

Note: we are keeping 'term' because that will be determined by member when they apply

Features that are not helpful or linear

- initial_list_status
- orig_date
- member_id (only helpful to trace back to member - but not linear)

Which features need to converted (dummy or label encoder)

- sub_grade (label encoder)
 - home_ownership (dummy)
 - reason (dummy)
 - term (change from 36 and 60 to 0 or 1)
-

Delete after data cleansing

What is the expected target variables?

- int_rate
 - loan_amount
 - loan_status
-

Cleanse the Data

Remove columns not needed

```
In [30]: #Code Block 29
```

```
df_loandata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37142 entries, 0 to 37141
Data columns (total 42 columns):
 #   Column           Non-Null Count  Dtype  
 --- 
 0   member_id        37138 non-null   Int64  
 1   loan_amnt        37136 non-null   Int64  
 2   orig_date        37136 non-null   datetime64[ns] 
 3   term              37136 non-null   Int64  
 4   int_rate          37136 non-null   float64 
 5   installment       37136 non-null   float64 
 6   annual_inc        37136 non-null   float64 
 7   delinq_2yrs       5815 non-null   Int64  
 8   inq_last_6mths    37132 non-null   Int64  
 9   mths_since_last_delinq 16746 non-null   Int64  
 10  mths_since_last_record 37132 non-null   Int64  
 11  open_acc          37118 non-null   Int64  
 12  pub_rec            37132 non-null   Int64  
 13  revol_bal          37132 non-null   Int64  
 14  total_acc          37101 non-null   Int64  
 15  out_prncp         37132 non-null   float64 
 16  total_pymnt        37132 non-null   float64 
 17  total_rec_prncp    37132 non-null   float64 
 18  total_debt_paid    37132 non-null   float64 
 19  total_rec_int       37132 non-null   float64 
 20  princ_int_ratio     37132 non-null   float64 
 21  total_rec_late_fee  37132 non-null   float64 
 22  recoveries          37132 non-null   float64 
 23  collection_recovery_fee 37132 non-null   float64 
 24  last_pymnt_amnt    37132 non-null   float64 
 25  collections_12_mths_ex_med 37132 non-null   Int64  
 26  mths_since_last_major_derog 37132 non-null   Int64  
 27  acc_now_delinq      37132 non-null   Int64  
 28  tot_coll_amt        37132 non-null   Int64  
 29  tot_cur_bal          37132 non-null   Int64  
 30  total_credit_rv      37132 non-null   Int64  
 31  revol_util          37132 non-null   float64 
 32  sub_grade            37132 non-null   string  
 33  emp_length           37132 non-null   Int64  
 34  home_ownership        37132 non-null   string  
 35  loan_status           37132 non-null   string  
 36  initial_list_status    37132 non-null   string  
 37  months_since_issue     37132 non-null   Int64  
 38  months_since_payment    37132 non-null   Int64  
 39  months_since_last_credit_pull 37132 non-null   Int64  
 40  months_since_earliest_credit 37132 non-null   Int64  
 41  reason                37132 non-null   string  
dtypes: Int64(22), datetime64[ns](1), float64(14), string(5)
memory usage: 12.7 MB
```

```
In [31]: #Code Block 30
```

```
df_loandata.columns
```

```
Out[31]: Index(['member_id', 'loan_amnt', 'orig_date', 'term', 'int_rate',
       'installment', 'annual_inc', 'delinq_2yrs', 'inq_last_6mths',
       'mths_since_last_delinq', 'mths_since_last_record', 'open_acc',
       'pub_rec', 'revol_bal', 'total_acc', 'out_prncp', 'total_pymnt',
       'total_rec_prncp', 'total_debt_paid', 'total_rec_int',
       'princ_int_ratio', 'total_rec_late_fee', 'recoveries',
       'collection_recovery_fee', 'last_pymnt_amnt',
       'collections_12_mths_ex_med', 'mths_since_last_major_derog',
       'acc_now_delinq', 'tot_coll_amt', 'tot_cur_bal', 'total_credit_rv',
       'revol_util', 'sub_grade', 'emp_length', 'home_ownership',
       'loan_status', 'initial_list_status', 'months_since_issue',
       'months_since_payment', 'months_since_last_credit_pull',
       'months_since_earliest_credit', 'reason'],
      dtype='object')
```

Create clean list of features

Drop columns that are not relevant

```
In [32]: #Code Block 31

df_loandata_clean = df_loandata.drop(['installment', 'initial_list_status', 'months_since_issue',
                                     'months_since_payment', 'months_since_last_credit_pull',
                                     'months_since_earliest_credit', 'total_pymnt', 'last_pymnt',
                                     '_amnt',
                                     'orig_date', 'total_rec_late_fee', 'recoveries', 'collection_recovery_fee',
                                     'total_rec_prncp', 'total_rec_int'], axis = 1)
df_loandata_clean.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37142 entries, 0 to 37141
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   member_id        37138 non-null   Int64  
 1   loan_amnt        37136 non-null   Int64  
 2   term              37136 non-null   Int64  
 3   int_rate          37136 non-null   float64 
 4   annual_inc        37136 non-null   float64 
 5   delinq_2yrs       5815 non-null    Int64  
 6   inq_last_6mths    37132 non-null   Int64  
 7   mths_since_last_delinq 16746 non-null   Int64  
 8   mths_since_last_record 37132 non-null   Int64  
 9   open_acc          37118 non-null   Int64  
 10  pub_rec            37132 non-null   Int64  
 11  revol_bal         37132 non-null   Int64  
 12  total_acc          37101 non-null   Int64  
 13  total_debt_paid    37132 non-null   float64 
 14  princ_int_ratio    37132 non-null   float64 
 15  collections_12_mths_ex_med 37132 non-null   Int64  
 16  mths_since_last_major_derog 37132 non-null   Int64  
 17  acc_now_delinq     37132 non-null   Int64  
 18  tot_coll_amt       37132 non-null   Int64  
 19  tot_cur_bal        37132 non-null   Int64  
 20  total_credit_rv    37132 non-null   Int64  
 21  revol_util          37132 non-null   float64 
 22  sub_grade          37132 non-null   string  
 23  emp_length          37132 non-null   Int64  
 24  home_ownership      37132 non-null   string  
 25  loan_status          37132 non-null   string  
 26  reason              37132 non-null   string  
dtypes: Int64(18), float64(5), string(4)
memory usage: 8.3 MB
```

Create target variables

```
In [33]: #Code Block 32

df_loandata_target = df_loandata[['member_id', 'loan_status', 'int_rate', 'loan_amnt']]
df_loandata_target.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37142 entries, 0 to 37141
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   member_id        37138 non-null   Int64  
 1   loan_status       37132 non-null   string  
 2   int_rate          37136 non-null   float64 
 3   loan_amnt         37136 non-null   Int64  
dtypes: Int64(2), float64(1), string(1)
memory usage: 1.2 MB
```

Remove rows with null values

- If a row has too many null values and is not useful, you should remove the row

How to DropNa

- how = 'any' - If any NA values are present, drop that row or column. (default)
- how = 'all' - If all values are NA, drop that row or column.
- thresh = int - Require that many non-NA values to drop the row or column.
- subset = ['column or row'] - If a value in that column or row is NA then drop.
- axis = 0 or 'index', 1 or 'columns' Determine if rows or columns which contain missing values are removed. (default = 0)
 - 0, or 'index' : Drop rows which contain missing values.
 - 1, or 'columns' : Drop columns which contain missing value. <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.dropna.html> (<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.dropna.html>)

```
In [34]: #Code Block 33
```

```
df_loandata_clean.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37142 entries, 0 to 37141
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        37138 non-null   Int64  
 1   loan_amnt        37136 non-null   Int64  
 2   term              37136 non-null   Int64  
 3   int_rate          37136 non-null   float64 
 4   annual_inc        37136 non-null   float64 
 5   delinq_2yrs       5815 non-null    Int64  
 6   inq_last_6mths   37132 non-null   Int64  
 7   mths_since_last_delinq 16746 non-null   Int64  
 8   mths_since_last_record 37132 non-null   Int64  
 9   open_acc          37118 non-null   Int64  
 10  pub_rec           37132 non-null   Int64  
 11  revol_bal         37132 non-null   Int64  
 12  total_acc         37101 non-null   Int64  
 13  total_debt_paid  37132 non-null   float64 
 14  princ_int_ratio  37132 non-null   float64 
 15  collections_12_mths_ex_med 37132 non-null   Int64  
 16  mths_since_last_major_derog 37132 non-null   Int64  
 17  acc_now_delinq   37132 non-null   Int64  
 18  tot_coll_amt     37132 non-null   Int64  
 19  tot_cur_bal      37132 non-null   Int64  
 20  total_credit_rv 37132 non-null   Int64  
 21  revol_util       37132 non-null   float64 
 22  sub_grade         37132 non-null   string  
 23  emp_length        37132 non-null   Int64  
 24  home_ownership   37132 non-null   string  
 25  loan_status       37132 non-null   string  
 26  reason            37132 non-null   string  
dtypes: Int64(18), float64(5), string(4)
memory usage: 8.3 MB
```

DropNA if any values in a row are null

```
In [35]: #Code Block 34
```

```
df_loandata_clean_any = df_loandata_clean.dropna(how='any')
df_loandata_clean_any.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 5796 entries, 1 to 37139
Data columns (total 27 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   member_id          5796 non-null   Int64  
 1   loan_amnt          5796 non-null   Int64  
 2   term                5796 non-null   Int64  
 3   int_rate            5796 non-null   float64 
 4   annual_inc          5796 non-null   float64 
 5   delinq_2yrs          5796 non-null   Int64  
 6   inq_last_6mths      5796 non-null   Int64  
 7   mths_since_last_delinq  5796 non-null   Int64  
 8   mths_since_last_record  5796 non-null   Int64  
 9   open_acc             5796 non-null   Int64  
 10  pub_rec              5796 non-null   Int64  
 11  revol_bal            5796 non-null   Int64  
 12  total_acc            5796 non-null   Int64  
 13  total_debt_paid      5796 non-null   float64 
 14  princ_int_ratio      5796 non-null   float64 
 15  collections_12_mths_ex_med 5796 non-null   Int64  
 16  mths_since_last_major_derog  5796 non-null   Int64  
 17  acc_now_delinq        5796 non-null   Int64  
 18  tot_coll_amt          5796 non-null   Int64  
 19  tot_cur_bal            5796 non-null   Int64  
 20  total_credit_rv        5796 non-null   Int64  
 21  revol_util             5796 non-null   float64 
 22  sub_grade              5796 non-null   string  
 23  emp_length             5796 non-null   Int64  
 24  home_ownership          5796 non-null   string  
 25  loan_status             5796 non-null   string  
 26  reason                 5796 non-null   string  
dtypes: Int64(18), float64(5), string(4)
memory usage: 1.3 MB
```

DropNA if all values in a row are null

```
In [36]: #Code Block 35
```

```
df_loandata_clean_all = df_loandata_clean.dropna(how='all')
df_loandata_clean_all.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 37138 entries, 0 to 37141
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        37138 non-null   Int64  
 1   loan_amnt       37136 non-null   Int64  
 2   term             37136 non-null   Int64  
 3   int_rate         37136 non-null   float64 
 4   annual_inc      37136 non-null   float64 
 5   delinq_2yrs     5815  non-null   Int64  
 6   inq_last_6mths  37132 non-null   Int64  
 7   mths_since_last_delinq  16746 non-null   Int64  
 8   mths_since_last_record 37132 non-null   Int64  
 9   open_acc         37118 non-null   Int64  
 10  pub_rec          37132 non-null   Int64  
 11  revol_bal       37132 non-null   Int64  
 12  total_acc        37101 non-null   Int64  
 13  total_debt_paid 37132 non-null   float64 
 14  princ_int_ratio 37132 non-null   float64 
 15  collections_12_mths_ex_med 37132 non-null   Int64  
 16  mths_since_last_major_derog 37132 non-null   Int64  
 17  acc_now_delinq   37132 non-null   Int64  
 18  tot_coll_amt    37132 non-null   Int64  
 19  tot_cur_bal     37132 non-null   Int64  
 20  total_credit_rv 37132 non-null   Int64  
 21  revol_util      37132 non-null   float64 
 22  sub_grade        37132 non-null   string  
 23  emp_length       37132 non-null   Int64  
 24  home_ownership   37132 non-null   string  
 25  loan_status      37132 non-null   string  
 26  reason           37132 non-null   string  
dtypes: Int64(18), float64(5), string(4)
memory usage: 8.6 MB
```

DropNa if any value for the column 'total_acc' is null (subset)

```
In [37]: #Code Block 36
```

```
df_loandata_clean_subset = df_loandata_clean.dropna(subset=['total_acc'])
df_loandata_clean_subset.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 37101 entries, 0 to 37141
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        37101 non-null   Int64  
 1   loan_amnt       37101 non-null   Int64  
 2   term             37101 non-null   Int64  
 3   int_rate         37101 non-null   float64 
 4   annual_inc      37101 non-null   float64 
 5   delinq_2yrs     5801 non-null    Int64  
 6   inq_last_6mths  37101 non-null   Int64  
 7   mths_since_last_delinq  16723 non-null   Int64  
 8   mths_since_last_record 37101 non-null   Int64  
 9   open_acc         37088 non-null   Int64  
 10  pub_rec          37101 non-null   Int64  
 11  revol_bal        37101 non-null   Int64  
 12  total_acc        37101 non-null   Int64  
 13  total_debt_paid 37101 non-null   float64 
 14  princ_int_ratio 37101 non-null   float64 
 15  collections_12_mths_ex_med 37101 non-null   Int64  
 16  mths_since_last_major_derog 37101 non-null   Int64  
 17  acc_now_delinq   37101 non-null   Int64  
 18  tot_coll_amt    37101 non-null   Int64  
 19  tot_cur_bal     37101 non-null   Int64  
 20  total_credit_rv 37101 non-null   Int64  
 21  revol_util      37101 non-null   float64 
 22  sub_grade        37101 non-null   string  
 23  emp_length       37101 non-null   Int64  
 24  home_ownership   37101 non-null   string  
 25  loan_status      37101 non-null   string  
 26  reason           37101 non-null   string  
dtypes: Int64(18), float64(5), string(4)
memory usage: 8.6 MB
```

DropNa if at least 6 or more columns for a row are null (thresh)

- If thresh = 1, then any rows with only 1 or less non-null values in a row will be dropped
- If thresh = 6, then any rows with only 6 or less non-null values in a row will be dropped

thresh = 1

- 1 or less with non-null values
- There are 2 rows with no non-null and 2 with 1 non-null

In [38]: #Code Block 37

```
df_loandata_clean_thresh = df_loandata_clean.dropna(thresh=1)
df_loandata_clean_thresh.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 37138 entries, 0 to 37141
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        37138 non-null   Int64  
 1   loan_amnt       37136 non-null   Int64  
 2   term             37136 non-null   Int64  
 3   int_rate         37136 non-null   float64 
 4   annual_inc      37136 non-null   float64 
 5   delinq_2yrs     5815  non-null   Int64  
 6   inq_last_6mths  37132 non-null   Int64  
 7   mths_since_last_delinq  16746 non-null   Int64  
 8   mths_since_last_record 37132 non-null   Int64  
 9   open_acc         37118 non-null   Int64  
 10  pub_rec          37132 non-null   Int64  
 11  revol_bal        37132 non-null   Int64  
 12  total_acc        37101 non-null   Int64  
 13  total_debt_paid 37132 non-null   float64 
 14  princ_int_ratio 37132 non-null   float64 
 15  collections_12_mths_ex_med 37132 non-null   Int64  
 16  mths_since_last_major_derog 37132 non-null   Int64  
 17  acc_now_delinq   37132 non-null   Int64  
 18  tot_coll_amt    37132 non-null   Int64  
 19  tot_cur_bal     37132 non-null   Int64  
 20  total_credit_rv 37132 non-null   Int64  
 21  revol_util      37132 non-null   float64 
 22  sub_grade        37132 non-null   string  
 23  emp_length       37132 non-null   Int64  
 24  home_ownership   37132 non-null   string  
 25  loan_status      37132 non-null   string  
 26  reason           37132 non-null   string  
dtypes: Int64(18), float64(5), string(4)
memory usage: 8.6 MB
```

thresh = 5

- 5 or less with non-null values
- There are 2 rows with no non-null and 2 with 1 non-null

```
In [39]: #Code Block 38
```

```
df_loandata_clean_thresh = df_loandata_clean.dropna(thresh=5)
df_loandata_clean_thresh.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 37136 entries, 0 to 37141
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        37136 non-null   Int64  
 1   loan_amnt       37136 non-null   Int64  
 2   term             37136 non-null   Int64  
 3   int_rate         37136 non-null   float64 
 4   annual_inc      37136 non-null   float64 
 5   delinq_2yrs     5815  non-null   Int64  
 6   inq_last_6mths  37132 non-null   Int64  
 7   mths_since_last_delinq  16746 non-null   Int64  
 8   mths_since_last_record 37132 non-null   Int64  
 9   open_acc         37118 non-null   Int64  
 10  pub_rec          37132 non-null   Int64  
 11  revol_bal       37132 non-null   Int64  
 12  total_acc       37101 non-null   Int64  
 13  total_debt_paid 37132 non-null   float64 
 14  princ_int_ratio 37132 non-null   float64 
 15  collections_12_mths_ex_med 37132 non-null   Int64  
 16  mths_since_last_major_derog 37132 non-null   Int64  
 17  acc_now_delinq  37132 non-null   Int64  
 18  tot_coll_amt    37132 non-null   Int64  
 19  tot_cur_bal     37132 non-null   Int64  
 20  total_credit_rv 37132 non-null   Int64  
 21  revol_util     37132 non-null   float64 
 22  sub_grade       37132 non-null   string  
 23  emp_length      37132 non-null   Int64  
 24  home_ownership  37132 non-null   string  
 25  loan_status     37132 non-null   string  
 26  reason          37132 non-null   string  
dtypes: Int64(18), float64(5), string(4)
memory usage: 8.6 MB
```

thresh = 6

- 6 or less with non-null values
- There are 2 rows with no non-null, 2 with 1 non-null and 4 with 6 non-null values

```
In [40]: #Code Block 39
```

```
df_loandata_clean_thresh = df_loandata_clean.dropna(thresh=6)
df_loandata_clean_thresh.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 37132 entries, 0 to 37141
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        37132 non-null   Int64  
 1   loan_amnt       37132 non-null   Int64  
 2   term             37132 non-null   Int64  
 3   int_rate         37132 non-null   float64 
 4   annual_inc      37132 non-null   float64 
 5   delinq_2yrs     5815  non-null    Int64  
 6   inq_last_6mths  37132 non-null   Int64  
 7   mths_since_last_delinq  16746 non-null   Int64  
 8   mths_since_last_record 37132 non-null   Int64  
 9   open_acc         37118 non-null   Int64  
 10  pub_rec          37132 non-null   Int64  
 11  revol_bal       37132 non-null   Int64  
 12  total_acc       37101 non-null   Int64  
 13  total_debt_paid 37132 non-null   float64 
 14  princ_int_ratio 37132 non-null   float64 
 15  collections_12_mths_ex_med 37132 non-null   Int64  
 16  mths_since_last_major_derog 37132 non-null   Int64  
 17  acc_now_delinq  37132 non-null   Int64  
 18  tot_coll_amt   37132 non-null   Int64  
 19  tot_cur_bal    37132 non-null   Int64  
 20  total_credit_rv 37132 non-null   Int64  
 21  revol_util     37132 non-null   float64 
 22  sub_grade       37132 non-null   string  
 23  emp_length      37132 non-null   Int64  
 24  home_ownership  37132 non-null   string  
 25  loan_status     37132 non-null   string  
 26  reason          37132 non-null   string  
dtypes: Int64(18), float64(5), string(4)
memory usage: 8.6 MB
```

In [41]: #Code Block 40

```
print('-----thresh = 5 -----')
df_loandata_clean_thresh5 = df_loandata_clean.dropna(thresh=5)
display(df_loandata_clean_thresh5.shape)
print('-----')
print('')

print('-----thresh = 6 -----')
df_loandata_clean_thresh6 = df_loandata_clean.dropna(thresh=6)
display(df_loandata_clean_thresh6.shape)
print('-----')
print('')

print('-----thresh = 10 -----')
df_loandata_clean_thresh10 = df_loandata_clean.dropna(thresh=10)
display(df_loandata_clean_thresh10.shape)
print('-----')
print('')

print('-----thresh = 25 -----')
df_loandata_clean_thresh25 = df_loandata_clean.dropna(thresh=25)
display(df_loandata_clean_thresh25.shape)
print('-----')
print('')

print('-----thresh = 25 -----')
print('---2 null values in one row ---')
df_loandata_clean_thresh25 = df_loandata_clean.dropna(thresh=25)
display(df_loandata_clean_thresh25.shape)
print('-----')
print('')

print('-----thresh = 26 -----')
print('---only 1 null value in one row ---')
df_loandata_clean_thresh26 = df_loandata_clean.dropna(thresh=26)
display(df_loandata_clean_thresh26.shape)
print('-----')
```

```
-----thresh = 5 -----
(37136, 27)

-----
-----thresh = 6 -----
(37132, 27)

-----
-----thresh = 10 -----
(37132, 27)

-----
-----thresh = 25 -----
(37119, 27)

-----
-----thresh = 25 -----
---2 null values in one row ---
(37119, 27)

-----
-----thresh = 26 -----
---only 1 null value in one row ---
(16733, 27)
```

Drop all rows that have 6 or less non-null values

- Also 20 or more null values

```
In [42]: #Code Block 41
```

```
df_loandata_clean = df_loandata_clean.dropna(thresh=6)
df_loandata_clean.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 37132 entries, 0 to 37141
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        37132 non-null   Int64  
 1   loan_amnt       37132 non-null   Int64  
 2   term             37132 non-null   Int64  
 3   int_rate         37132 non-null   float64 
 4   annual_inc      37132 non-null   float64 
 5   delinq_2yrs     5815  non-null    Int64  
 6   inq_last_6mths  37132 non-null   Int64  
 7   mths_since_last_delinq  16746 non-null   Int64  
 8   mths_since_last_record 37132 non-null   Int64  
 9   open_acc         37118 non-null   Int64  
 10  pub_rec          37132 non-null   Int64  
 11  revol_bal        37132 non-null   Int64  
 12  total_acc        37101 non-null   Int64  
 13  total_debt_paid 37132 non-null   float64 
 14  princ_int_ratio 37132 non-null   float64 
 15  collections_12_mths_ex_med 37132 non-null   Int64  
 16  mths_since_last_major_derog 37132 non-null   Int64  
 17  acc_now_delinq   37132 non-null   Int64  
 18  tot_coll_amt    37132 non-null   Int64  
 19  tot_cur_bal     37132 non-null   Int64  
 20  total_credit_rv 37132 non-null   Int64  
 21  revol_util      37132 non-null   float64 
 22  sub_grade        37132 non-null   string  
 23  emp_length       37132 non-null   Int64  
 24  home_ownership   37132 non-null   string  
 25  loan_status      37132 non-null   string  
 26  reason           37132 non-null   string  
dtypes: Int64(18), float64(5), string(4)
memory usage: 8.6 MB
```

[Top](#)

Clean observations - Fill NaNs with a specific value

Fill with a specific value

- Change all NaN values to 0.

```
In [43]: #Code Block 42
```

```
df_loandata.mths_since_last_delinq.isnull().sum()
```

```
Out[43]: 20396
```

Two different ways to fill in NaNs

- use `fillna()` from pandas
 - <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html> (<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html>)
- use `SimpleImputer` from sklearn
 - <https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html> (<https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html>)

Use `fillna()` from pandas

```
In [44]: #Code Block 43

df_fillna = df_loandata[['member_id', 'mths_since_last_delinq']]
df_fillna.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37142 entries, 0 to 37141
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   member_id        37138 non-null   Int64  
 1   mths_since_last_delinq  16746 non-null   Int64  
dtypes: Int64(2)
memory usage: 653.0 KB
```

```
In [45]: #Code Block 44

df_fillna['mths_since_last_delinq'].value_counts().head()

Out[45]: 8      314
          20     310
          18     310
          21     308
          10     307
Name: mths_since_last_delinq, dtype: Int64
```

```
In [46]: #Code Block 45

df_fillna['mths_since_last_delinq'] = df_fillna['mths_since_last_delinq'].fillna(0)
df_fillna['mths_since_last_delinq'].value_counts().head()

<ipython-input-46-091ff1c027d5>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
    df_fillna['mths_since_last_delinq'] = df_fillna['mths_since_last_delinq'].fillna(0)

Out[46]: 0      20396
          8      314
          18     310
          20     310
          21     308
Name: mths_since_last_delinq, dtype: Int64
```

```
In [47]: #Code Block 46

df_loandata_clean[['mths_since_last_delinq', 'delinq_2yrs']] = df_loandata_clean[['mths_since_last_delinq', 'delinq_2yrs']].fillna(0)

#same as:
#df_loandata_clean['mths_since_last_delinq'] = df_loandata_clean['mths_since_last_delinq'].fillna(0)
#df_loandata_clean['delinq_2yrs'] = df_loandata_clean['delinq_2yrs'].fillna(0)

df_loandata_clean.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 37132 entries, 0 to 37141
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   member_id        37132 non-null   Int64  
 1   loan_amnt       37132 non-null   Int64  
 2   term             37132 non-null   Int64  
 3   int_rate         37132 non-null   float64 
 4   annual_inc      37132 non-null   float64 
 5   delinq_2yrs     37132 non-null   Int64  
 6   inq_last_6mths  37132 non-null   Int64  
 7   mths_since_last_delinq  37132 non-null   Int64  
 8   mths_since_last_record 37132 non-null   Int64  
 9   open_acc         37118 non-null   Int64  
 10  pub_rec          37132 non-null   Int64  
 11  revol_bal       37132 non-null   Int64  
 12  total_acc        37101 non-null   Int64  
 13  total_debt_paid 37132 non-null   float64 
 14  princ_int_ratio 37132 non-null   float64 
 15  collections_12_mths_ex_med 37132 non-null   Int64  
 16  mths_since_last_major_derog 37132 non-null   Int64  
 17  acc_now_delinq   37132 non-null   Int64  
 18  tot_coll_amt    37132 non-null   Int64  
 19  tot_cur_bal     37132 non-null   Int64  
 20  total_credit_rv 37132 non-null   Int64  
 21  revol_util      37132 non-null   float64 
 22  sub_grade        37132 non-null   string  
 23  emp_length       37132 non-null   Int64  
 24  home_ownership   37132 non-null   string  
 25  loan_status      37132 non-null   string  
 26  reason           37132 non-null   string  
dtypes: Int64(18), float64(5), string(4)
memory usage: 8.6 MB
```

[Top](#)

Clean Observations - Fill in NaNs with mean or median

Fill in the mean for open_acc and total_acc

Groupby for open_acc for mean and median

```
In [48]: #Code Block 47
```

```
print('Mean for open_acc')
display(round(df_loandata_clean['open_acc'].mean(), 2))
print(' ')
print('-----')
print('Median for open_acc')
display(round(df_loandata_clean['open_acc'].median(), 2))
```

```
Mean for open_acc
```

```
10.96
```

```
-----
```

```
Median for open_acc
```

```
10.0
```

```
In [49]: df_loandata_clean['home_ownership'].value_counts()
```

```
Out[49]: MORTGAGE    18085
RENT        16096
OWN         2940
MORTGAGE     11
Name: home_ownership, dtype: Int64
```

```
In [50]: #Code Block 48
```

```
pd.pivot_table(df_loandata_clean, index=["home_ownership"], values=[ "open_acc" ])
```

```
Out[50]:
```

```
open_acc
```

```
home_ownership
```

home_ownership	open_acc
MORTGAGE	11.727273
MORTGAGE	11.638953
OWN	10.600340
RENT	10.269501

```
In [51]: #Code Block 49
```

```
#use the aggfunc to specify median
pd.pivot_table(df_loandata_clean, index=["home_ownership"], values=[ "open_acc" ], aggfunc='median')
```

```
Out[51]:
```

```
open_acc
```

```
home_ownership
```

home_ownership	open_acc
MORTGAGE	10
MORTGAGE	11
OWN	10
RENT	10

```
In [52]: #Code Block 50
```

```
df_loandata_clean["open_acc"].fillna(df_loandata_clean["open_acc"].median(), inplace=True)
df_loandata_clean.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 37132 entries, 0 to 37141
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        37132 non-null   Int64  
 1   loan_amnt       37132 non-null   Int64  
 2   term             37132 non-null   Int64  
 3   int_rate         37132 non-null   float64 
 4   annual_inc      37132 non-null   float64 
 5   delinq_2yrs     37132 non-null   Int64  
 6   inq_last_6mths  37132 non-null   Int64  
 7   mths_since_last_delinq  37132 non-null   Int64  
 8   mths_since_last_record 37132 non-null   Int64  
 9   open_acc         37132 non-null   Int64  
 10  pub_rec          37132 non-null   Int64  
 11  revol_bal       37132 non-null   Int64  
 12  total_acc       37101 non-null   Int64  
 13  total_debt_paid 37132 non-null   float64 
 14  princ_int_ratio 37132 non-null   float64 
 15  collections_12_mths_ex_med 37132 non-null   Int64  
 16  mths_since_last_major_derog 37132 non-null   Int64  
 17  acc_now_delinq  37132 non-null   Int64  
 18  tot_coll_amt   37132 non-null   Int64  
 19  tot_cur_bal    37132 non-null   Int64  
 20  total_credit_rv 37132 non-null   Int64  
 21  revol_util     37132 non-null   float64 
 22  sub_grade      37132 non-null   string  
 23  emp_length     37132 non-null   Int64  
 24  home_ownership 37132 non-null   string  
 25  loan_status    37132 non-null   string  
 26  reason          37132 non-null   string  
dtypes: Int64(18), float64(5), string(4)
memory usage: 8.6 MB
```

[Top](#)

Recode Features with Replacement

Use PandasProfiling to look at shape of data and labeled features

```
#Code Block 53 from pandas_profiling import ProfileReport
profile = ProfileReport(df_loandata_clean, title="Loan Data")
profile
```

Recode labeled categories

Replace values

- Can replace using .replace individually

Use a translation table

- If you have a lot of mislabeled observations, creating a translation table with two columns:
 - Column1: All possible labels
 - Column2: Correct label

```
In [54]: #Code Block 54
```

```
df_loandata_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 37132 entries, 0 to 37141
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   member_id        37132 non-null   Int64  
 1   loan_amnt       37132 non-null   Int64  
 2   term             37132 non-null   Int64  
 3   int_rate         37132 non-null   float64 
 4   annual_inc      37132 non-null   float64 
 5   delinq_2yrs     37132 non-null   Int64  
 6   inq_last_6mths  37132 non-null   Int64  
 7   mths_since_last_delinq  37132 non-null   Int64  
 8   mths_since_last_record 37132 non-null   Int64  
 9   open_acc         37132 non-null   Int64  
 10  pub_rec          37132 non-null   Int64  
 11  revol_bal       37132 non-null   Int64  
 12  total_acc       37101 non-null   Int64  
 13  total_debt_paid 37132 non-null   float64 
 14  princ_int_ratio 37132 non-null   float64 
 15  collections_12_mths_ex_med 37132 non-null   Int64  
 16  mths_since_last_major_derog 37132 non-null   Int64  
 17  acc_now_delinq  37132 non-null   Int64  
 18  tot_coll_amt    37132 non-null   Int64  
 19  tot_cur_bal     37132 non-null   Int64  
 20  total_credit_rv 37132 non-null   Int64  
 21  revol_util     37132 non-null   float64 
 22  sub_grade       37132 non-null   string  
 23  emp_length      37132 non-null   Int64  
 24  home_ownership  37132 non-null   string  
 25  loan_status     37132 non-null   string  
 26  reason          37132 non-null   string  
dtypes: Int64(18), float64(5), string(4)
memory usage: 8.6 MB
```

```
In [55]: #Code Block 55
```

```
df_loandata_clean['home_ownership'].value_counts()
```

```
Out[55]: MORTGAGE    18085
RENT        16096
OWN         2940
MORGTAGE     11
Name: home_ownership, dtype: Int64
```

```
In [56]: #Code Block 56
```

```
df_loandata_clean['home_ownership'] = df_loandata_clean['home_ownership'].replace('MORGTAGE',
'MORTGAGE')
df_loandata_clean['home_ownership'].value_counts()
```

```
Out[56]: MORTGAGE    18096
RENT        16096
OWN         2940
Name: home_ownership, dtype: Int64
```

[Top](#)

Clean Observations - Fill in NaNs with grouped mean

Groupby for total_acc for mean and median

In [57]: #Code Block 57

```
print('Mean for total_acc')
display(round(df_loandata_clean['total_acc'].mean(), 2))
print(' ')
print('-----')
print('Median for total_acc')
display(round(df_loandata_clean['total_acc'].median(), 2))
```

Mean for total_acc

24.2

Median for total_acc

23.0

In [58]: #Code Block 58

```
display(pd.pivot_table(df_loandata_clean, index=["home_ownership"], values=["total_acc"]))

#use the aggfunc to specify median
pd.pivot_table(df_loandata_clean, index=["home_ownership"], values=["total_acc"], aggfunc='median')
```

total_acc

home_ownership	total_acc
MORTGAGE	27.216886
OWN	22.638870
RENT	21.090559

Out[58]:

total_acc

home_ownership	total_acc
MORTGAGE	26
OWN	21
RENT	20

```
In [59]: #Code Block 59
```

```
#df_loandata_clean[df_loandata_clean['total_acc'].isnull()]['home_ownership', 'total_acc']].sample(15, random_state=42)
df_loandata_clean[df_loandata_clean['total_acc'].isnull()].sample(15, random_state=42)
```

```
Out[59]:
```

member_id	loan_amnt	term	int_rate	annual_inc	delinq_2yrs	inq_last_6mths	mths_since_last_delinq	mths_since_last_rec
36935	3256914	35000	60	19.72	100000.0	5	0	15
36157	2967409	18000	60	17.77	60000.0	5	1	16
36783	3176905	4500	36	19.05	55000.0	0	0	0
36558	3017133	4500	36	13.11	33000.0	0	1	40
35748	2845279	22000	36	11.14	135000.0	2	3	22
35885	2896943	4000	36	16.29	100000.0	2	0	12
37079	3410838	7750	36	13.11	39500.0	1	2	6
36784	3176962	7200	36	18.75	71000.0	4	0	1
36041	2927285	9000	36	18.75	40000.0	0	0	29
29072	2276826	9000	36	11.14	50000.0	0	0	42
34874	2834296	8325	36	13.11	45000.0	1	0	12
36339	2977028	10575	36	13.11	31530.0	1	0	11
34876	2834330	18000	36	7.62	90000.0	0	1	0
36131	2967136	8000	36	15.80	60000.0	1	1	11
36021	2926898	30000	60	15.80	140000.0	0	2	0

```
In [60]: #Code Block 60
```

```
df_loandata_clean_index = df_loandata_clean[df_loandata_clean['total_acc'].isnull()]['home_ownership', 'total_acc']].sample(15, random_state=42).index
df_loandata_clean_index
```

```
Out[60]: Int64Index([36935, 36157, 36783, 36558, 35748, 35885, 37079, 36784, 36041,
                     29072, 34874, 36339, 34876, 36131, 36021],
                     dtype='int64')
```

```
In [61]: #Code Block 61
```

```
df_loandata_clean.loc[df_loandata_clean_index, :]
```

Out[61]:

	member_id	loan_amnt	term	int_rate	annual_inc	delinq_2yrs	inq_last_6mths	mths_since_last_delinq	mths_since_last_rec
36935	3256914	35000	60	19.72	100000.0	5	0	15	
36157	2967409	18000	60	17.77	60000.0	5	1	16	
36783	3176905	4500	36	19.05	55000.0	0	0	0	
36558	3017133	4500	36	13.11	33000.0	0	1	40	
35748	2845279	22000	36	11.14	135000.0	2	3	22	
35885	2896943	4000	36	16.29	100000.0	2	0	12	
37079	3410838	7750	36	13.11	39500.0	1	2	6	
36784	3176962	7200	36	18.75	71000.0	4	0	1	
36041	2927285	9000	36	18.75	40000.0	0	0	29	
29072	2276826	9000	36	11.14	50000.0	0	0	42	
34874	2834296	8325	36	13.11	45000.0	1	0	12	
36339	2977028	10575	36	13.11	31530.0	1	0	11	
34876	2834330	18000	36	7.62	90000.0	0	1	0	
36131	2967136	8000	36	15.80	60000.0	1	1	11	
36021	2926898	30000	60	15.80	140000.0	0	2	0	

```
In [62]: #Code Block 62
```

```
df_loandata_clean["total_acc"].fillna(df_loandata_clean.groupby("home_ownership")["total_acc"].
transform("median"), inplace=True)
df_loandata_clean.head()
```

Out[62]:

	member_id	loan_amnt	term	int_rate	annual_inc	delinq_2yrs	inq_last_6mths	mths_since_last_delinq	mths_since_last_rec
0	1581986	9000	36	12.12	45000.0	0	3	0	0
1	1751708	6625	36	11.14	28000.0	1	0	23	0
2	1666916	9800	36	12.12	50000.0	0	0	0	0
3	1758003	4250	36	8.90	38000.0	2	3	21	0
4	1730191	16000	36	7.90	60000.0	0	0	28	0

In [63]:

```
#Code Block 63
display(df_loandata_clean.info())
df_loandata_clean.loc[df_loandata_clean_index, :]

<class 'pandas.core.frame.DataFrame'>
Int64Index: 37132 entries, 0 to 37141
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        37132 non-null   Int64  
 1   loan_amnt       37132 non-null   Int64  
 2   term             37132 non-null   Int64  
 3   int_rate         37132 non-null   float64 
 4   annual_inc      37132 non-null   float64 
 5   delinq_2yrs     37132 non-null   Int64  
 6   inq_last_6mths  37132 non-null   Int64  
 7   mths_since_last_delinq  37132 non-null   Int64  
 8   mths_since_last_record  37132 non-null   Int64  
 9   open_acc         37132 non-null   Int64  
 10  pub_rec          37132 non-null   Int64  
 11  revol_bal       37132 non-null   Int64  
 12  total_acc       37132 non-null   Int64  
 13  total_debt_paid 37132 non-null   float64 
 14  princ_int_ratio 37132 non-null   float64 
 15  collections_12_mths_ex_med 37132 non-null   Int64  
 16  mths_since_last_major_derog 37132 non-null   Int64  
 17  acc_now_delinq  37132 non-null   Int64  
 18  tot_coll_amt   37132 non-null   Int64  
 19  tot_cur_bal    37132 non-null   Int64  
 20  total_credit_rv 37132 non-null   Int64  
 21  revol_util     37132 non-null   float64 
 22  sub_grade       37132 non-null   string  
 23  emp_length      37132 non-null   Int64  
 24  home_ownership  37132 non-null   string  
 25  loan_status     37132 non-null   string  
 26  reason          37132 non-null   string  
dtypes: Int64(18), float64(5), string(4)
memory usage: 9.8 MB
```

None

Out[63]:

	member_id	loan_amnt	term	int_rate	annual_inc	delinq_2yrs	inq_last_6mths	mths_since_last_delinq	mths_since_last_record
36935	3256914	35000	60	19.72	100000.0	5	0		15
36157	2967409	18000	60	17.77	60000.0	5	1		16
36783	3176905	4500	36	19.05	55000.0	0	0		0
36558	3017133	4500	36	13.11	33000.0	0	1		40
35748	2845279	22000	36	11.14	135000.0	2	3		22
35885	2896943	4000	36	16.29	100000.0	2	0		12
37079	3410838	7750	36	13.11	39500.0	1	2		6
36784	3176962	7200	36	18.75	71000.0	4	0		1
36041	2927285	9000	36	18.75	40000.0	0	0		29
29072	2276826	9000	36	11.14	50000.0	0	0		42
34874	2834296	8325	36	13.11	45000.0	1	0		12
36339	2977028	10575	36	13.11	31530.0	1	0		11
34876	2834330	18000	36	7.62	90000.0	0	1		0
36131	2967136	8000	36	15.80	60000.0	1	1		11
36021	2926898	30000	60	15.80	140000.0	0	2		0

Recode with a translation table

Recode labeled categories

Replace values

- Can replace using .replace individually

Use a translation table

- If you have a lot of mislabeled observations, creating a translation table with two columns:
 - Column1: All possible labels
 - Column2: Correct label

```
In [64]: #Code Block 64
```

```
df_reason
```

```
Out[64]:
```

	reason_old	reason_recode
0	cc	credit_card
1	debtcon	debt_consolidation
2	other	other
3	pers	personal
4	med	medical
5	credit_card	credit_card
6	debt_consolidation	debt_consolidation
7	medical	medical
8	personal	personal

```
In [65]: #Code Block 65
```

```
df_loandata_clean['reason'].value_counts()
```

```
Out[65]: debtcon          11618
debt_consolidation    10701
cc                  3865
credit_card          3372
personal            2907
pers                2275
other               1982
med                 215
medical             197
Name: reason, dtype: Int64
```

```
In [66]: #Code Block 66

df_loandata_clean = pd.merge(df_loandata_clean, df_reason, left_on='reason', right_on='reason_ol
d', how='left')
df_loandata_clean.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 37132 entries, 0 to 37131
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        37132 non-null   Int64  
 1   loan_amnt       37132 non-null   Int64  
 2   term             37132 non-null   Int64  
 3   int_rate         37132 non-null   float64 
 4   annual_inc      37132 non-null   float64 
 5   delinq_2yrs     37132 non-null   Int64  
 6   inq_last_6mths  37132 non-null   Int64  
 7   mths_since_last_delinq  37132 non-null   Int64  
 8   mths_since_last_record  37132 non-null   Int64  
 9   open_acc         37132 non-null   Int64  
 10  pub_rec          37132 non-null   Int64  
 11  revol_bal       37132 non-null   Int64  
 12  total_acc       37132 non-null   Int64  
 13  total_debt_paid 37132 non-null   float64 
 14  princ_int_ratio 37132 non-null   float64 
 15  collections_12_mths_ex_med 37132 non-null   Int64  
 16  mths_since_last_major_derog 37132 non-null   Int64  
 17  acc_now_delinq  37132 non-null   Int64  
 18  tot_coll_amt   37132 non-null   Int64  
 19  tot_cur_bal    37132 non-null   Int64  
 20  total_credit_rv 37132 non-null   Int64  
 21  revol_util     37132 non-null   float64 
 22  sub_grade      37132 non-null   string  
 23  emp_length     37132 non-null   Int64  
 24  home_ownership 37132 non-null   string  
 25  loan_status    37132 non-null   string  
 26  reason          37132 non-null   object  
 27  reason_old     37132 non-null   object  
 28  reason_recode  37132 non-null   object  
dtypes: Int64(18), float64(5), object(3), string(3)
memory usage: 9.1+ MB
```

```
In [67]: #Code Block 67

df_loandata_clean[['reason', 'reason_old', 'reason_recode']].sample(10, random_state=42)
```

```
Out[67]:
```

	reason	reason_old	reason_recode
8248	personal	personal	personal
19864	cc	cc	credit_card
8829	debt_consolidation	debt_consolidation	debt_consolidation
26912	debtcon	debtcon	debt_consolidation
3488	personal	personal	personal
35798	debtcon	debtcon	debt_consolidation
6231	debt_consolidation	debt_consolidation	debt_consolidation
35045	cc	cc	credit_card
15747	personal	personal	personal
465	debt_consolidation	debt_consolidation	debt_consolidation

```
In [68]: #Code Block 68
```

```
df_loandata_clean = df_loandata_clean.drop(['reason', 'reason_old'], axis = 1)
df_loandata_clean = df_loandata_clean.rename(columns={'reason_recode': 'reason'})
df_loandata_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 37132 entries, 0 to 37131
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   member_id        37132 non-null   Int64  
 1   loan_amnt       37132 non-null   Int64  
 2   term             37132 non-null   Int64  
 3   int_rate         37132 non-null   float64 
 4   annual_inc      37132 non-null   float64 
 5   delinq_2yrs     37132 non-null   Int64  
 6   inq_last_6mths  37132 non-null   Int64  
 7   mths_since_last_delinq  37132 non-null   Int64  
 8   mths_since_last_record  37132 non-null   Int64  
 9   open_acc         37132 non-null   Int64  
 10  pub_rec          37132 non-null   Int64  
 11  revol_bal       37132 non-null   Int64  
 12  total_acc       37132 non-null   Int64  
 13  total_debt_paid 37132 non-null   float64 
 14  princ_int_ratio 37132 non-null   float64 
 15  collections_12_mths_ex_med 37132 non-null   Int64  
 16  mths_since_last_major_derog 37132 non-null   Int64  
 17  acc_now_delinq  37132 non-null   Int64  
 18  tot_coll_amt   37132 non-null   Int64  
 19  tot_cur_bal    37132 non-null   Int64  
 20  total_credit_rv 37132 non-null   Int64  
 21  revol_util     37132 non-null   float64 
 22  sub_grade       37132 non-null   string  
 23  emp_length      37132 non-null   Int64  
 24  home_ownership  37132 non-null   string  
 25  loan_status     37132 non-null   string  
 26  reason          37132 non-null   object  
dtypes: Int64(18), float64(5), object(1), string(3)
memory usage: 8.6+ MB
```

```
In [69]: df_loandata_clean.sample(10, random_state=42)
```

Out[69]:

	member_id	loan_amnt	term	int_rate	annual_inc	delinq_2yrs	inq_last_6mths	mths_since_last_delinq	mths_since_last_r
8248	1720463	19000	36	7.62	80000.0	0	3	0	
19864	1830624	9000	36	13.11	40000.0	1	0	23	
8829	1595309	19000	60	17.99	48000.0	0	0	0	
26912	1972325	3300	36	14.09	50000.0	0	2	0	
3488	1742073	11875	36	8.90	36000.0	0	2	0	
35798	2845903	35000	60	17.77	120000.0	0	1	0	
6231	1595537	10125	36	18.55	50000.0	1	0	11	
35045	2836909	6750	36	12.12	62000.0	1	2	22	
15747	1739744	13500	36	10.16	115000.0	0	0	0	
465	1663947	25000	60	20.49	68000.0	1	2	20	

```
In [70]: #df_loandata_clean.to_csv('data/DATA6320_Scenario3.csv')
```