CS_Phase2_S22 - FINAL

April 7, 2022

1 Phase 2 - Ingestion and Cleaning

Group members: - Mira Kasari, mkasari - Sairathan Rajuladevi, srajulad - Kelly McManus, kellymcm - Cole Thomas, nhthomas

In the Phase 2 of the Case Study, we will carry out the following steps: - Ingest raw downloaded data - Output a combined dataset ready for analysis and modeling

[1]:
```
import pandas as pd
import os
from sys import platform
import matplotlib.pyplot as plt
import datetime
import numpy as np
import pickle
import seaborn
```

[2]:
```
# A helper function that you'll be using while reading the raw files
def is_integer(x):
    '''
    This function returns True if x is an integer, and False otherwise
    '''
    try:
        return (int(x) == float(x))
    except:
        return False
```

[3]:
```
# Display all rows and columns in dataframes
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

1.1 Parameters

[4]:
```
# Define the directories that contain the files downloaded
dir_cs = 'zips'

# Define the output path for the pickle
pickle_file = 'clean_data.pickle' # path to save cleaned data
```

[5]:
```
# Identify the columns we'll be keeping from the dataset
cols_to_pick = ['id', 'loan_amnt',
                'funded_amnt', 'term', 'int_rate', 'grade', 'emp_length',␣
 ↪'home_ownership',
                'annual_inc', 'verification_status', 'issue_d', 'loan_status',␣
 ↪'purpose','dti',
                'delinq_2yrs', 'earliest_cr_line', 'open_acc', 'pub_rec',␣
 ↪'fico_range_high',
                'fico_range_low', 'revol_bal', 'revol_util', 'total_pymnt',␣
 ↪'recoveries', 'last_pymnt_d'] # list of features to use for this study as␣
 ↪indicated in the handout

# Identify the type of each of these column based on your CS-Phase 1 response
float_cols = ['loan_amnt', 'funded_amnt', 'annual_inc', 'dti', 'delinq_2yrs',␣
 ↪'open_acc', 'pub_rec'
              , 'fico_range_high', 'fico_range_low', 'revol_bal',␣
 ↪'total_pymnt', 'recoveries']
cat_cols = ['term', 'grade', 'emp_length', 'home_ownership',␣
 ↪'verification_status', 'loan_status', 'purpose'] # categorical features
perc_cols = ['int_rate', 'revol_util']
date_cols = ['issue_d', 'earliest_cr_line', 'last_pymnt_d']

# Ensure that we have types for every column
assert set(cols_to_pick) - set(float_cols) - set(cat_cols) - set(perc_cols) -␣
 ↪set(date_cols) == set(["id"])
```

[6]:
```
# Some of the columns selected will not be used directly in the model,
# but will be used to generate other features.
# Create variables specifying the features that will be used

# All categorical columns other than "loan_status" will be used as
# discrete features
discrete_features = list(set(cat_cols) - set(["loan_status"]))

# All numeric columns will be used as continuous features
continuous_features = list(float_cols + perc_cols)
```

1.2 Ingestion

Ingest the data files from both sets, perform consistency checks, and prepare one single file for each set

[7]:
```
import glob
```

[8]:
```
def ingest_files(directory):
    '''
    This function will ingest every file in the specified directory
```

```
    into a pandas dataframe. It will return a dictionary containing
    these dataframes, keyed by the file name.

    We assume the directory contains files directly downloaded from
    the link given in the handout, and *only* those files. Thus, we
    assume the files are zipped (pd.read_csv can read zipped files)
    and we assume the first line in each file needs to be skipped.

    Note that each file will be read *without* formatting
    '''

    # If the directory has no trailing slash, add one
    if directory[-1] != "/":
        directory +='/'

    all_files = glob.glob(directory+ '*') # get list of all files from the␣
↪directory
    output = {}

    print("Directory " + directory + " has " + str(len(all_files)) + " files:")
    for i in all_files:
        print("    Reading file " + i)
        output[i] = pd.read_csv(i, dtype='str', skiprows=1) # read each with␣
↪dtype='str' and skip_rows =1

        # Some of the files have "summary" lines that, for example
        # read "Total number of loans number in Policy 1: ....."
        # To remove those lines, find any lines with non-integer IDs
        # and remove them

        invalid_rows =  ~output[i]['id'].apply(is_integer) # mask rows that␣
↪have non-integer IDs. Use is_integer method
        invalid_rows_list = invalid_rows.index[invalid_rows == True].tolist()
        if len(invalid_rows_list) > 0:
            print("Found " + str(len(invalid_rows_list)) + " invalid rows which␣
↪were removed")
            output[i] = output[i].drop(index=invalid_rows_list) # remove␣
↪invalid rows

    return output # return dictionary of dataframe
```

[9]:
```
# Ingest the set of files we downloaded using the defined method "ingest_files"
files_cs = ingest_files(dir_cs) # dictionary of (filename, dataframe) as (key,␣
↪value)
```

```
Directory zips/ has 20 files:
    Reading file zips\LoanStats3a_securev1.csv.zip
```

```
Found 3 invalid rows which were removed
    Reading file zips\LoanStats3b_securev1.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats3c_securev1.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats3d_securev1.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2016Q1.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2016Q2.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2016Q3.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2016Q4.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2017Q1.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2017Q2.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2017Q3.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2017Q4.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2018Q1.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2018Q2.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2018Q3.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2018Q4.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2019Q1.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2019Q2.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2019Q3.csv.zip
Found 2 invalid rows which were removed
    Reading file zips\LoanStats_securev1_2019Q4.csv.zip
Found 2 invalid rows which were removed
```

1.2.1 Combine the files

[10]:
```
data_cs = pd.concat(files_cs.values()) # combine "files_cs" into a pandas␣
↪dataframe
data_cs.reset_index(drop=True, inplace=True) # reset index with drop = True
```

[11]:
```
data_cs.shape
```

[11]: (2777776, 151)

[12]:
```
data_cs.head()
```

[12]:

```
         id member_id loan_amnt funded_amnt funded_amnt_inv        term  \
0  1077501       NaN      5000        5000            4975   36 months
1  1077430       NaN      2500        2500            2500   60 months
2  1077175       NaN      2400        2400            2400   36 months
3  1076863       NaN     10000       10000           10000   36 months
4  1075358       NaN      3000        3000            3000   60 months

  int_rate installment grade sub_grade                 emp_title emp_length  \
0   10.65%      162.87     B        B2                       NaN  10+ years
1   15.27%       59.83     C        C4                     Ryder   < 1 year
2   15.96%       84.33     C        C5                       NaN  10+ years
3   13.49%      339.31     C        C1       AIR RESOURCES BOARD  10+ years
4   12.69%       67.79     B        B5  University Medical Group     1 year

  home_ownership annual_inc verification_status   issue_d  loan_status  \
0           RENT      24000            Verified  Dec-2011   Fully Paid
1           RENT      30000     Source Verified  Dec-2011  Charged Off
2           RENT      12252        Not Verified  Dec-2011   Fully Paid
3           RENT      49200     Source Verified  Dec-2011   Fully Paid
4           RENT      80000     Source Verified  Dec-2011   Fully Paid

  pymnt_plan                                               url  \
0          n  https://lendingclub.com/browse/loanDetail.acti…
1          n  https://lendingclub.com/browse/loanDetail.acti…
2          n  https://lendingclub.com/browse/loanDetail.acti…
3          n  https://lendingclub.com/browse/loanDetail.acti…
4          n  https://lendingclub.com/browse/loanDetail.acti…

                                               desc         purpose  \
0    Borrower added on 12/22/11 > I need to upgra…     credit_card
1    Borrower added on 12/22/11 > I plan to use t…             car
2                                               NaN  small_business
3    Borrower added on 12/21/11 > to pay for prop…           other
4    Borrower added on 12/21/11 > I plan on combi…           other

                 title zip_code addr_state    dti delinq_2yrs  \
0             Computer    860xx         AZ  27.65           0
1                 bike    309xx         GA      1           0
2  real estate business    606xx         IL   8.72           0
3             personel    917xx         CA     20           0
4             Personal    972xx         OR  17.94           0

  earliest_cr_line fico_range_low fico_range_high inq_last_6mths  \
```

```
0         Jan-1985           735             739             1
1         Apr-1999           740             744             5
2         Nov-2001           735             739             2
3         Feb-1996           690             694             1
4         Jan-1996           695             699             0

  mths_since_last_delinq mths_since_last_record open_acc pub_rec revol_bal  \
0                    NaN                    NaN        3       0     13648
1                    NaN                    NaN        3       0      1687
2                    NaN                    NaN        2       0      2956
3                     35                    NaN       10       0      5598
4                     38                    NaN       15       0     27783

  revol_util total_acc initial_list_status out_prncp out_prncp_inv  \
0      83.7%         9                   f      0.00          0.00
1       9.4%         4                   f      0.00          0.00
2      98.5%        10                   f      0.00          0.00
3        21%        37                   f      0.00          0.00
4      53.9%        38                   f      0.00          0.00

         total_pymnt total_pymnt_inv total_rec_prncp total_rec_int  \
0     5863.1551866952         5833.84         5000.00        863.16
1             1014.53         1014.53          456.46        435.17
2     3005.6668441393         3005.67         2400.00        605.67
3  12231.890000000902        12231.89        10000.00       2214.92
4     4066.9081610817         4066.91         3000.00       1066.91

  total_rec_late_fee recoveries collection_recovery_fee last_pymnt_d  \
0                0.0        0.0                    0.0     Jan-2015
1                0.0      122.9                   1.11     Apr-2013
2                0.0        0.0                    0.0     Jun-2014
3              16.97        0.0                    0.0     Jan-2015
4                0.0        0.0                    0.0     Jan-2017

  last_pymnt_amnt next_pymnt_d last_credit_pull_d last_fico_range_high  \
0          171.62          NaN           Apr-2018                  719
1          119.66          NaN           Oct-2016                  499
2          649.91          NaN           Jun-2017                  739
3          357.48          NaN           Apr-2016                  604
4            67.3          NaN           Apr-2018                  684

  last_fico_range_low collections_12_mths_ex_med mths_since_last_major_derog  \
0                 715                         0                         NaN
1                   0                         0                         NaN
2                 735                         0                         NaN
3                 600                         0                         NaN
4                 680                         0                         NaN
```

```
  policy_code application_type annual_inc_joint dti_joint  \
0           1       Individual              NaN       NaN
1           1       Individual              NaN       NaN
2           1       Individual              NaN       NaN
3           1       Individual              NaN       NaN
4           1       Individual              NaN       NaN

  verification_status_joint acc_now_delinq tot_coll_amt tot_cur_bal  \
0                       NaN              0          NaN         NaN
1                       NaN              0          NaN         NaN
2                       NaN              0          NaN         NaN
3                       NaN              0          NaN         NaN
4                       NaN              0          NaN         NaN

  open_acc_6m open_act_il open_il_12m open_il_24m mths_since_rcnt_il  \
0         NaN         NaN         NaN         NaN                NaN
1         NaN         NaN         NaN         NaN                NaN
2         NaN         NaN         NaN         NaN                NaN
3         NaN         NaN         NaN         NaN                NaN
4         NaN         NaN         NaN         NaN                NaN

  total_bal_il il_util open_rv_12m open_rv_24m max_bal_bc all_util  \
0          NaN     NaN         NaN         NaN        NaN      NaN
1          NaN     NaN         NaN         NaN        NaN      NaN
2          NaN     NaN         NaN         NaN        NaN      NaN
3          NaN     NaN         NaN         NaN        NaN      NaN
4          NaN     NaN         NaN         NaN        NaN      NaN

  total_rev_hi_lim inq_fi total_cu_tl inq_last_12m acc_open_past_24mths  \
0              NaN    NaN         NaN          NaN                  NaN
1              NaN    NaN         NaN          NaN                  NaN
2              NaN    NaN         NaN          NaN                  NaN
3              NaN    NaN         NaN          NaN                  NaN
4              NaN    NaN         NaN          NaN                  NaN

  avg_cur_bal bc_open_to_buy bc_util chargeoff_within_12_mths delinq_amnt  \
0         NaN            NaN     NaN                       0          0
1         NaN            NaN     NaN                       0          0
2         NaN            NaN     NaN                       0          0
3         NaN            NaN     NaN                       0          0
4         NaN            NaN     NaN                       0          0

  mo_sin_old_il_acct mo_sin_old_rev_tl_op mo_sin_rcnt_rev_tl_op  \
0                NaN                  NaN                   NaN
1                NaN                  NaN                   NaN
2                NaN                  NaN                   NaN
```

```
3                NaN                  NaN                  NaN
4                NaN                  NaN                  NaN

  mo_sin_rcnt_tl mort_acc mths_since_recent_bc mths_since_recent_bc_dlq  \
0            NaN      NaN                  NaN                      NaN
1            NaN      NaN                  NaN                      NaN
2            NaN      NaN                  NaN                      NaN
3            NaN      NaN                  NaN                      NaN
4            NaN      NaN                  NaN                      NaN

  mths_since_recent_inq mths_since_recent_revol_delinq num_accts_ever_120_pd  \
0                   NaN                           NaN                   NaN
1                   NaN                           NaN                   NaN
2                   NaN                           NaN                   NaN
3                   NaN                           NaN                   NaN
4                   NaN                           NaN                   NaN

  num_actv_bc_tl num_actv_rev_tl num_bc_sats num_bc_tl num_il_tl  \
0            NaN             NaN         NaN       NaN       NaN
1            NaN             NaN         NaN       NaN       NaN
2            NaN             NaN         NaN       NaN       NaN
3            NaN             NaN         NaN       NaN       NaN
4            NaN             NaN         NaN       NaN       NaN

  num_op_rev_tl num_rev_accts num_rev_tl_bal_gt_0 num_sats num_tl_120dpd_2m  \
0           NaN           NaN                 NaN      NaN              NaN
1           NaN           NaN                 NaN      NaN              NaN
2           NaN           NaN                 NaN      NaN              NaN
3           NaN           NaN                 NaN      NaN              NaN
4           NaN           NaN                 NaN      NaN              NaN

  num_tl_30dpd num_tl_90g_dpd_24m num_tl_op_past_12m pct_tl_nvr_dlq  \
0          NaN                NaN                NaN            NaN
1          NaN                NaN                NaN            NaN
2          NaN                NaN                NaN            NaN
3          NaN                NaN                NaN            NaN
4          NaN                NaN                NaN            NaN

  percent_bc_gt_75 pub_rec_bankruptcies tax_liens tot_hi_cred_lim  \
0              NaN                    0         0             NaN
1              NaN                    0         0             NaN
2              NaN                    0         0             NaN
3              NaN                    0         0             NaN
4              NaN                    0         0             NaN

  total_bal_ex_mort total_bc_limit total_il_high_credit_limit revol_bal_joint  \
0               NaN            NaN                        NaN             NaN
```

```
1               NaN            NaN                         NaN             NaN
2               NaN            NaN                         NaN             NaN
3               NaN            NaN                         NaN             NaN
4               NaN            NaN                         NaN             NaN

  sec_app_fico_range_low sec_app_fico_range_high sec_app_earliest_cr_line  \
0                    NaN                    NaN                      NaN
1                    NaN                    NaN                      NaN
2                    NaN                    NaN                      NaN
3                    NaN                    NaN                      NaN
4                    NaN                    NaN                      NaN

  sec_app_inq_last_6mths sec_app_mort_acc sec_app_open_acc sec_app_revol_util  \
0                    NaN             NaN             NaN                NaN
1                    NaN             NaN             NaN                NaN
2                    NaN             NaN             NaN                NaN
3                    NaN             NaN             NaN                NaN
4                    NaN             NaN             NaN                NaN

  sec_app_open_act_il sec_app_num_rev_accts sec_app_chargeoff_within_12_mths  \
0                 NaN                   NaN                              NaN
1                 NaN                   NaN                              NaN
2                 NaN                   NaN                              NaN
3                 NaN                   NaN                              NaN
4                 NaN                   NaN                              NaN

  sec_app_collections_12_mths_ex_med sec_app_mths_since_last_major_derog  \
0                                NaN                                 NaN
1                                NaN                                 NaN
2                                NaN                                 NaN
3                                NaN                                 NaN
4                                NaN                                 NaN

  hardship_flag hardship_type hardship_reason hardship_status deferral_term  \
0             N           NaN             NaN             NaN           NaN
1             N           NaN             NaN             NaN           NaN
2             N           NaN             NaN             NaN           NaN
3             N           NaN             NaN             NaN           NaN
4             N           NaN             NaN             NaN           NaN

  hardship_amount hardship_start_date hardship_end_date  \
0             NaN                 NaN               NaN
1             NaN                 NaN               NaN
2             NaN                 NaN               NaN
3             NaN                 NaN               NaN
4             NaN                 NaN               NaN
```

```
  payment_plan_start_date hardship_length hardship_dpd hardship_loan_status  \
0                     NaN             NaN          NaN                   NaN
1                     NaN             NaN          NaN                   NaN
2                     NaN             NaN          NaN                   NaN
3                     NaN             NaN          NaN                   NaN
4                     NaN             NaN          NaN                   NaN

  orig_projected_additional_accrued_interest hardship_payoff_balance_amount  \
0                                        NaN                           NaN
1                                        NaN                           NaN
2                                        NaN                           NaN
3                                        NaN                           NaN
4                                        NaN                           NaN

  hardship_last_payment_amount disbursement_method debt_settlement_flag  \
0                          NaN               Cash                    N
1                          NaN               Cash                    N
2                          NaN               Cash                    N
3                          NaN               Cash                    N
4                          NaN               Cash                    N

  debt_settlement_flag_date settlement_status settlement_date  \
0                       NaN               NaN             NaN
1                       NaN               NaN             NaN
2                       NaN               NaN             NaN
3                       NaN               NaN             NaN
4                       NaN               NaN             NaN

  settlement_amount settlement_percentage settlement_term
0               NaN                   NaN             NaN
1               NaN                   NaN             NaN
2               NaN                   NaN             NaN
3               NaN                   NaN             NaN
4               NaN                   NaN             NaN
```

[13]:
```
data_cs.tail()
```

[13]:
```
                id member_id loan_amnt funded_amnt funded_amnt_inv  \
2777771  158872331       NaN      3000        3000            3000
2777772  158833440       NaN     10000       10000           10000
2777773  158748525       NaN     19000       19000           19000
2777774  158298751       NaN     10000       10000           10000
2777775  158206429       NaN     14875       14875           14875

                term int_rate installment grade sub_grade          emp_title  \
2777771   36 months   17.74%      108.07     C        C5  Machine operator
2777772   36 months    6.46%      306.31     A        A1      IT SUPERVISOR
```

```
2777773   36 months    6.46%      581.99     A        A1          Professor
2777774   60 months   28.80%      316.21     D        D5                NaN
2777775   36 months   16.95%      529.97     C        C4                NaN

        emp_length home_ownership annual_inc verification_status   issue_d  \
2777771  10+ years            OWN      44000        Not Verified  Oct-2019
2777772   < 1 year           RENT      60000        Not Verified  Oct-2019
2777773    4 years       MORTGAGE      67350     Source Verified  Oct-2019
2777774        NaN       MORTGAGE      40000        Not Verified  Oct-2019
2777775        NaN       MORTGAGE     150000     Source Verified  Oct-2019

        loan_status pymnt_plan  \
2777771  Fully Paid          n
2777772     Current          n
2777773     Current          n
2777774     Current          n
2777775     Current          n

                                                    url desc  \
2777771  https://lendingclub.com/browse/loanDetail.acti…  NaN
2777772  https://lendingclub.com/browse/loanDetail.acti…  NaN
2777773  https://lendingclub.com/browse/loanDetail.acti…  NaN
2777774  https://lendingclub.com/browse/loanDetail.acti…  NaN
2777775  https://lendingclub.com/browse/loanDetail.acti…  NaN

                 purpose                     title zip_code addr_state  \
2777771        credit_card  Credit card refinancing    136xx         NY
2777772        credit_card  Credit card refinancing    928xx         CA
2777773  debt_consolidation       Debt consolidation    200xx         DC
2777774    home_improvement         Home improvement    985xx         WA
2777775        credit_card  Credit card refinancing    773xx         TX

          dti delinq_2yrs earliest_cr_line fico_range_low fico_range_high  \
2777771  30.01          0         Apr-2009            705             709
2777772  14.18          0         Dec-2008            750             754
2777773      6          0         Feb-2007            680             684
2777774    2.1          0         Oct-1991            660             664
2777775   8.76          0         Jul-2000            660             664

        inq_last_6mths mths_since_last_delinq mths_since_last_record open_acc  \
2777771              1                     39                    NaN       12
2777772              0                    NaN                    NaN        7
2777773              0                     36                    NaN        6
2777774              0                    NaN                     16        6
2777775              0                    NaN                    NaN       16

        pub_rec revol_bal revol_util total_acc initial_list_status out_prncp  \
```

2777771	0	4599	35.9%	16	w	0.00
2777772	0	18259	27.3%	10	w	8981.91
2777773	0	21809	45.7%	10	w	17065.62
2777774	1	1947	26.3%	27	w	9684.01
2777775	0	43530	71.8%	25	w	13568.19

	out_prncp_inv	total_pymnt	total_pymnt_inv	total_rec_prncp
2777771	0.00	3032.5199999992	3032.52	3000.00
2777772	8981.91	1221.65	1221.65	1018.09
2777773	17065.62	2310.91	2310.91	1934.38
2777774	9684.01	1248.84	1248.84	315.99
2777775	13568.19	2105.87	2105.87	1306.81

	total_rec_int	total_rec_late_fee	recoveries	collection_recovery_fee
2777771	32.52	0.0	0.0	0.0
2777772	203.56	0.0	0.0	0.0
2777773	376.53	0.0	0.0	0.0
2777774	932.85	0.0	0.0	0.0
2777775	799.06	0.0	0.0	0.0

	last_pymnt_d	last_pymnt_amnt	next_pymnt_d	last_credit_pull_d
2777771	Nov-2019	35.48	NaN	Sep-2019
2777772	Feb-2020	306.31	Mar-2020	Jan-2020
2777773	Feb-2020	581.99	Mar-2020	Jan-2020
2777774	Feb-2020	316.21	Mar-2020	Jan-2020
2777775	Feb-2020	529.97	Mar-2020	Jan-2020

	last_fico_range_high	last_fico_range_low	collections_12_mths_ex_med
2777771	709	705	0
2777772	759	755	0
2777773	659	655	0
2777774	664	660	0
2777775	689	685	0

	mths_since_last_major_derog	policy_code	application_type
2777771	NaN	1	Individual
2777772	NaN	1	Individual
2777773	36	1	Individual
2777774	NaN	1	Individual
2777775	NaN	1	Individual

	annual_inc_joint	dti_joint	verification_status_joint	acc_now_delinq
2777771	NaN	NaN	NaN	0
2777772	NaN	NaN	NaN	0
2777773	NaN	NaN	NaN	0
2777774	NaN	NaN	NaN	0
2777775	NaN	NaN	NaN	0

```
        tot_coll_amt tot_cur_bal open_acc_6m open_act_il open_il_12m  \
2777771            0       51462           3           3           2
2777772            0       23872           0           1           0
2777773            0       21809           1           0           0
2777774            0        1947           0           0           0
2777775         1382      136511           1           0           0

        open_il_24m mths_since_rcnt_il total_bal_il il_util open_rv_12m  \
2777771           4                  5        46863      93           2
2777772           0                 35         5613      50           1
2777773           0                 80            0     NaN           2
2777774           0                 41            0     NaN           1
2777775           0                117            0     NaN           2

        open_rv_24m max_bal_bc all_util total_rev_hi_lim inq_fi total_cu_tl  \
2777771           5       2312       81            12800      1           1
2777772           1       9261       31            67000      0           0
2777773           2      12077       46            47700      1           0
2777774           2       1494       26             7400      0           0
2777775           3       7826       72            60600      0           0

        inq_last_12m acc_open_past_24mths avg_cur_bal bc_open_to_buy bc_util  \
2777771            4                    9        4289           2488    48.2
2777772            0                    1        3410          48441    27.4
2777773            0                    2        3635          25891    45.7
2777774            1                    2         389           2653    42.3
2777775            0                    3        8532          10246    77.9

        chargeoff_within_12_mths delinq_amnt mo_sin_old_il_acct  \
2777771                        0           0                36
2777772                        0           0               129
2777773                        0           0                80
2777774                        0           0               335
2777775                        0           0               147

        mo_sin_old_rev_tl_op mo_sin_rcnt_rev_tl_op mo_sin_rcnt_tl mort_acc  \
2777771                  125                     2              2        0
2777772                   90                    10             10        0
2777773                  151                     4              4        0
2777774                  258                    10             10        1
2777775                  230                     6              6        3

        mths_since_recent_bc mths_since_recent_bc_dlq mths_since_recent_inq  \
2777771                   21                      NaN                     2
2777772                   27                      NaN                   NaN
2777773                    4                       36                    19
```

```
2777774                  10                      NaN                    10
2777775                   7                      NaN                   NaN

        mths_since_recent_revol_delinq num_accts_ever_120_pd num_actv_bc_tl  \
2777771                             39                    0             1
2777772                            NaN                    0             4
2777773                             36                    1             4
2777774                            NaN                    0             2
2777775                            NaN                    0             8

        num_actv_rev_tl num_bc_sats num_bc_tl num_il_tl num_op_rev_tl  \
2777771               5           1         1         5             9
2777772               4           5         6         3             6
2777773               4           6         8         1             6
2777774               2           4        19         2             6
2777775              11          11        14         2            15

        num_rev_accts num_rev_tl_bal_gt_0 num_sats num_tl_120dpd_2m  \
2777771            11                   5       12               0
2777772             7                   4        7               0
2777773             9                   4        6               0
2777774            24                   2        6               0
2777775            20                  11       16               0

        num_tl_30dpd num_tl_90g_dpd_24m num_tl_op_past_12m pct_tl_nvr_dlq  \
2777771            0                  0                  4           93.8
2777772            0                  0                  1            100
2777773            0                  0                  2             90
2777774            0                  0                  1            100
2777775            0                  0                  2            100

        percent_bc_gt_75 pub_rec_bankruptcies tax_liens tot_hi_cred_lim  \
2777771                0                    0         0           63484
2777772                0                    0         0           78282
2777773             33.3                    0         0           47700
2777774             33.3                    1         0            7400
2777775             72.7                    0         0          180600

        total_bal_ex_mort total_bc_limit total_il_high_credit_limit  \
2777771             51462           4800                      50684
2777772             23872          66700                      11282
2777773             21809          47700                          0
2777774              1947           4600                          0
2777775             43530          46300                          0

        revol_bal_joint sec_app_fico_range_low sec_app_fico_range_high  \
2777771             NaN                    NaN                     NaN
```

```
2777772             NaN                    NaN                    NaN
2777773             NaN                    NaN                    NaN
2777774             NaN                    NaN                    NaN
2777775             NaN                    NaN                    NaN

        sec_app_earliest_cr_line sec_app_inq_last_6mths sec_app_mort_acc  \
2777771                      NaN                   NaN              NaN
2777772                      NaN                   NaN              NaN
2777773                      NaN                   NaN              NaN
2777774                      NaN                   NaN              NaN
2777775                      NaN                   NaN              NaN

        sec_app_open_acc sec_app_revol_util sec_app_open_act_il  \
2777771              NaN                NaN                 NaN
2777772              NaN                NaN                 NaN
2777773              NaN                NaN                 NaN
2777774              NaN                NaN                 NaN
2777775              NaN                NaN                 NaN

        sec_app_num_rev_accts sec_app_chargeoff_within_12_mths  \
2777771                   NaN                             NaN
2777772                   NaN                             NaN
2777773                   NaN                             NaN
2777774                   NaN                             NaN
2777775                   NaN                             NaN

        sec_app_collections_12_mths_ex_med  \
2777771                                NaN
2777772                                NaN
2777773                                NaN
2777774                                NaN
2777775                                NaN

        sec_app_mths_since_last_major_derog hardship_flag hardship_type  \
2777771                                 NaN             N           NaN
2777772                                 NaN             N           NaN
2777773                                 NaN             N           NaN
2777774                                 NaN             N           NaN
2777775                                 NaN             N           NaN

        hardship_reason hardship_status deferral_term hardship_amount  \
2777771             NaN             NaN           NaN             NaN
2777772             NaN             NaN           NaN             NaN
2777773             NaN             NaN           NaN             NaN
2777774             NaN             NaN           NaN             NaN
2777775             NaN             NaN           NaN             NaN
```

```
        hardship_start_date hardship_end_date payment_plan_start_date  \
2777771                 NaN               NaN                     NaN
2777772                 NaN               NaN                     NaN
2777773                 NaN               NaN                     NaN
2777774                 NaN               NaN                     NaN
2777775                 NaN               NaN                     NaN

        hardship_length hardship_dpd hardship_loan_status  \
2777771             NaN          NaN                  NaN
2777772             NaN          NaN                  NaN
2777773             NaN          NaN                  NaN
2777774             NaN          NaN                  NaN
2777775             NaN          NaN                  NaN

        orig_projected_additional_accrued_interest  \
2777771                                        NaN
2777772                                        NaN
2777773                                        NaN
2777774                                        NaN
2777775                                        NaN

        hardship_payoff_balance_amount hardship_last_payment_amount  \
2777771                            NaN                          NaN
2777772                            NaN                          NaN
2777773                            NaN                          NaN
2777774                            NaN                          NaN
2777775                            NaN                          NaN

        disbursement_method debt_settlement_flag debt_settlement_flag_date  \
2777771                 NaN                    N                        NaN
2777772                 NaN                    N                        NaN
2777773                 NaN                    N                        NaN
2777774                 NaN                    N                        NaN
2777775                 NaN                    N                        NaN

        settlement_status settlement_date settlement_amount  \
2777771               NaN             NaN               NaN
2777772               NaN             NaN               NaN
2777773               NaN             NaN               NaN
2777774               NaN             NaN               NaN
2777775               NaN             NaN               NaN

        settlement_percentage settlement_term
2777771                   NaN             NaN
2777772                   NaN             NaN
2777773                   NaN             NaN
2777774                   NaN             NaN
```

```
2777775                    NaN                NaN
```

1.3 Prepare Final Dataset

[14]:
```
# Keep only the columns of interest from 'data_cs'
final_data = data_cs[cols_to_pick]
```

[15]:
```
print("Starting with", str(len(final_data)), "rows and", str(len(final_data.
 ↪columns)), "columns")
```

```
Starting with 2777776 rows and 25 columns
```

[16]:
```
final_original = final_data # Save original combined data
```

1.3.1 Typecast the columns

[17]:
```
# Remember that we read the data as string (without any formatting).
# Now we would typecast the columns based on feature types which you found out␣
 ↪in CS Phase 1

final_data = final_data.copy() # Create copy to typecast on copy rather than␣
 ↪original dataset

# FLOATS #

for i in float_cols:
    final_data[i] = pd.to_numeric(final_data[i], downcast='float') # typecast␣
 ↪float columns
```

[18]:
```
# PERCENT #

def clean_perc(x):
    if pd.isnull(x):
        return np.nan
    else:
        return float(x.strip()[:-1])

for i in perc_cols:
    final_data[i] = [clean_perc(x) for x in final_data[i]] # apply clean_perc␣
 ↪to percentage columns
```

[19]:
```
# DATES #

def clean_date(x):
    if pd.isnull(x):
        return None
    else:
```

```
        return datetime.datetime.strptime(x, "%b-%Y").date()

for i in date_cols:
    final_data[i] = [clean_date(x) for x in final_data[i]] # typecast date␣
 ↪columns to datatime using clean_date
```

[20]:
```
# CATEGORICAL #
# for categorical features if the value is null/empty set it to None

for i in cat_cols:
    final_data[i] = final_data[i].fillna("None")
```

1.4 Calculate returns for each loan

[21]:
```
# Define the names of the four returns we'll be calculating as described in Q.6
# ret_PESS: Pessimistic return
# ret_OPT: Optimistic return
# ret_INTa, ret_INTb: Method3 at two differnt values of "i"
ret_cols = ["ret_PESS", "ret_OPT", "ret_INTa", "ret_INTb"]
```

[22]:
```
# Remove all rows for loans that were paid back on the days they were issued
final_data['loan_length'] = (final_data.last_pymnt_d - final_data.issue_d) / np.
 ↪timedelta64(1, 'M')
n_rows = len(final_data)

final_data = final_data[final_data['loan_length'] > 0] # select rows where␣
 ↪loan_length is not 0.

print("Removed " + str(n_rows - len(final_data)) + " rows")
```

```
Removed 15658 rows
```

1.4.1 M1-Pessimistic Method

[23]:
```
# Calculate the return using a simple annualized profit margin
# Pessimistic definition (Handout 6a.) (M1)

final_data['term_num'] = final_data.term.str.extract('(\d+)',expand=False).
 ↪astype(int) # length of loan in months

def calculateM1Return(df, totalAmountInvested, totalAmountRepaid,␣
 ↪TermLengthInMonths):
    m1Return = ((df[totalAmountRepaid] - df[totalAmountInvested]) /␣
 ↪df[totalAmountInvested]) * (12 / df[TermLengthInMonths])
    return m1Return
```

```
final_data['ret_PESS'] = calculateM1Return(final_data, 'funded_amnt',␣
↪'total_pymnt', 'term_num')
```

1.4.2 M2-Optimistic Method

[24]:
```
# Assuming that if a loan gives a positive return, we can
# immediately find a similar loan to invest in; if the loan
# takes a loss, we use M1-pessimistic to compute the return

def calculateM2Return(df, totalAmountInvested, totalAmountRepaid,␣
↪ActualTermLength):
    m2Return = ((df[totalAmountRepaid] - df[totalAmountInvested]) /␣
↪df[totalAmountInvested]) * (12 / df[ActualTermLength])
    return m2Return

final_data['ret_OPT'] = calculateM2Return(final_data, 'funded_amnt',␣
↪'total_pymnt', 'loan_length')

# Two-piece formula
final_data.loc[final_data.total_pymnt - final_data.funded_amnt <= 0,'ret_OPT']␣
↪= final_data['ret_PESS']
```

1.4.3 Method 3

[25]:
```
def ret_method_3(T, i):
    '''
    Given an investment time horizon (in months) and re-investment
    interest rate, calculate the return of each loan
    '''

    # Assuming that the total amount paid back was paid at equal
    # intervals during the duration of the loan, calculate the
    # size of each of these installment
    # p/m
    actual_installment = (final_data.total_pymnt - final_data.recoveries) /␣
↪final_data.loan_length

    # Assuming the amount is immediately re-invested at the prime
    # rate, find the total amount of money we'll have by the end
    # of the loan
    # compute the quantity given in [] in eq.2.3 of handout
    cash_by_end_of_loan = actual_installment * ((1 - ((1 + i)**final_data.
↪loan_length)) / (1 - (1 + i)))
    cash_by_end_of_loan = cash_by_end_of_loan + final_data.recoveries

    # Assuming that cash is then re-invested at the prime rate,
    # with monthly re-investment, until T months from the start
```

```
    # of the loan
    remaining_months = T - final_data['loan_length']
    final_return = (cash_by_end_of_loan * ((1 + i)**remaining_months)) -␣
↪final_data.funded_amnt

    # Find the percentage return
    ret_val = final_return * (1/final_data.funded_amnt) * (12/T)
    return ret_val
```

[26]:
```
final_data['ret_INTa'] = ret_method_3(60, 0.023) # call ret_method_3 with T=60,␣
↪i=0.023
final_data['ret_INTb'] = ret_method_3(60, 0.04) # call ret_method_3 with T=60,␣
↪i=0.04
```

1.4.4 Visualize the variables

[27]:
```
def visualize_float_columns():
    '''
    This function visualizes Box-and-whisker plots for continuous variables
    '''

    # Float columns
    for i in float_cols + perc_cols + ret_cols:
        try:
            seaborn.boxplot(final_data[i])
        except:
            print(i)

        # Print the three highest values
        highest_vals = sorted(final_data[i].unique(), reverse=True)[0:3] # get␣
↪3 highest values

        smallest_val = min(final_data[i])

        plt.text(smallest_val, -0.3, highest_vals[0])
        plt.text(smallest_val, -0.2, highest_vals[1])
        plt.text(smallest_val, -0.1, highest_vals[2])

        plt.show()
```

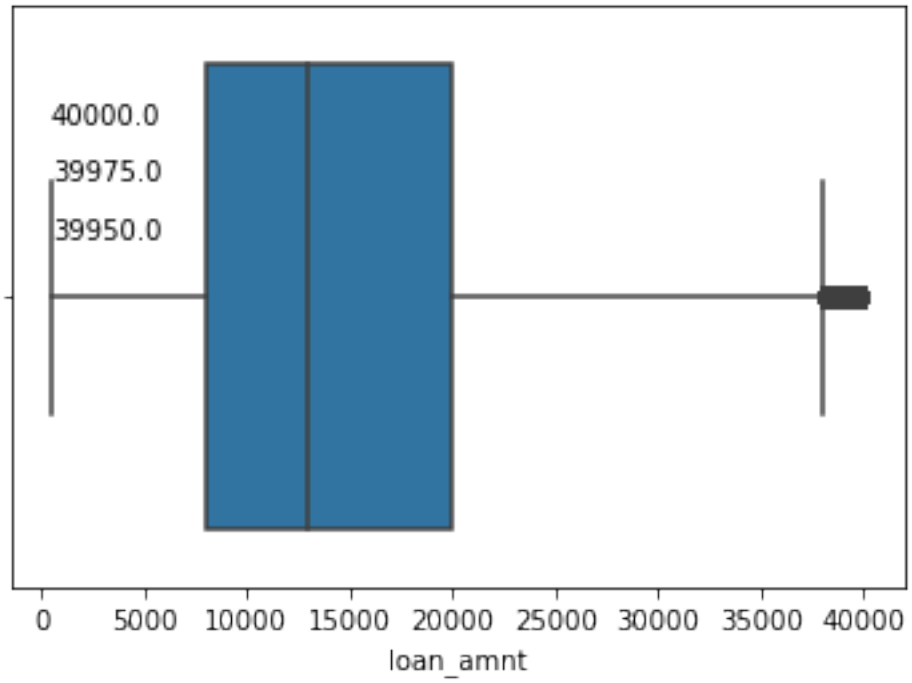
[28]:
```
def visualize_cat_columns():
    '''
    Lists the distinct values for categorical columns
    '''
    # Categorical columns
    for i in cat_cols:
```

```
    print('Column', i, 'has', len(final_data[i].unique()), 'unique values') #␣
↪print field name, print number of unique values
    print(final_data[i].value_counts()) # print number of occurrences for␣
↪each unique value
    print(" ")
    print(" ")
```

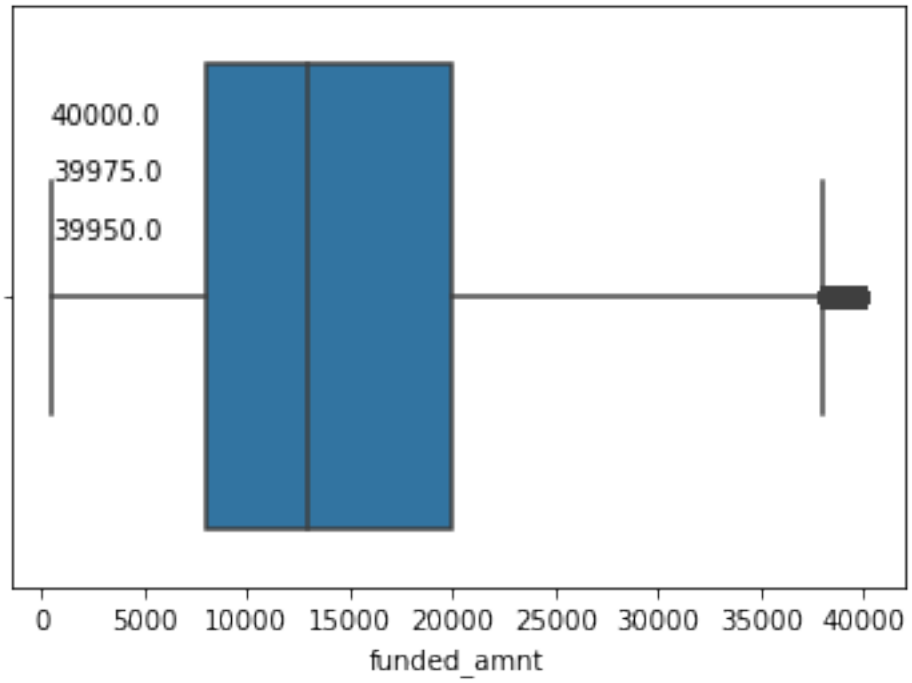
[29]:
```
def visualize_date_columns():
    '''
    This function visualizes a timeline density for dates
    '''

    # Date columns
    for i in date_cols:
        final_data[final_data[i].isnull() == False][i].apply(lambda x : str(x.
↪year) +
                                                "-" + str(x.month)).
↪value_counts(ascending = True).plot()
        plt.title(i + " (" + str(final_data[i].isnull().sum()) + " null␣
↪values)")
        plt.show()
```

[30]:
```
# visualize continuous features
visualize_float_columns()

# visulaize categorical features
visualize_cat_columns()

# visualize date columns
visualize_date_columns()
```

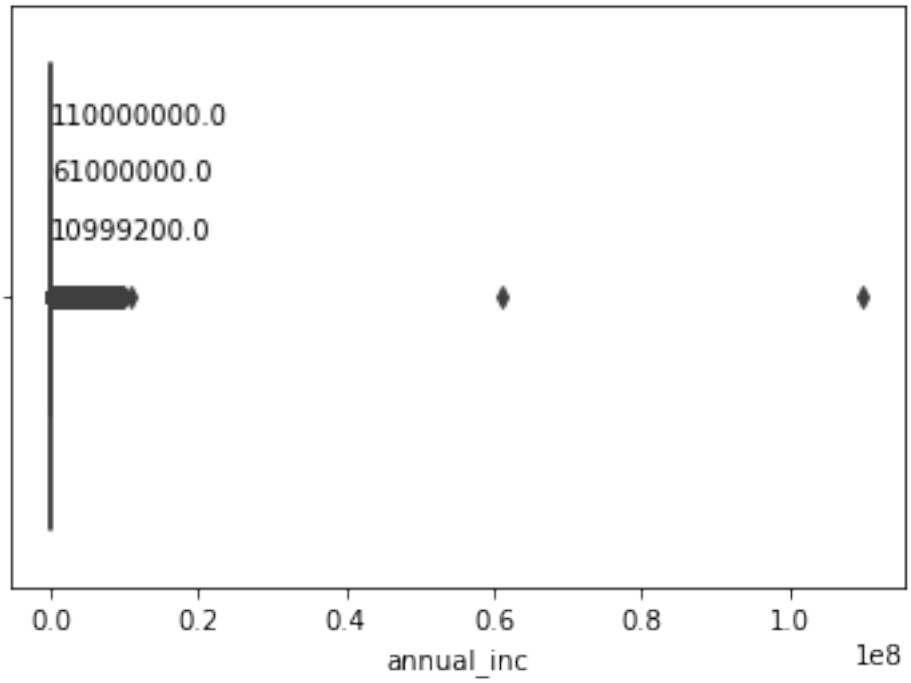
```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



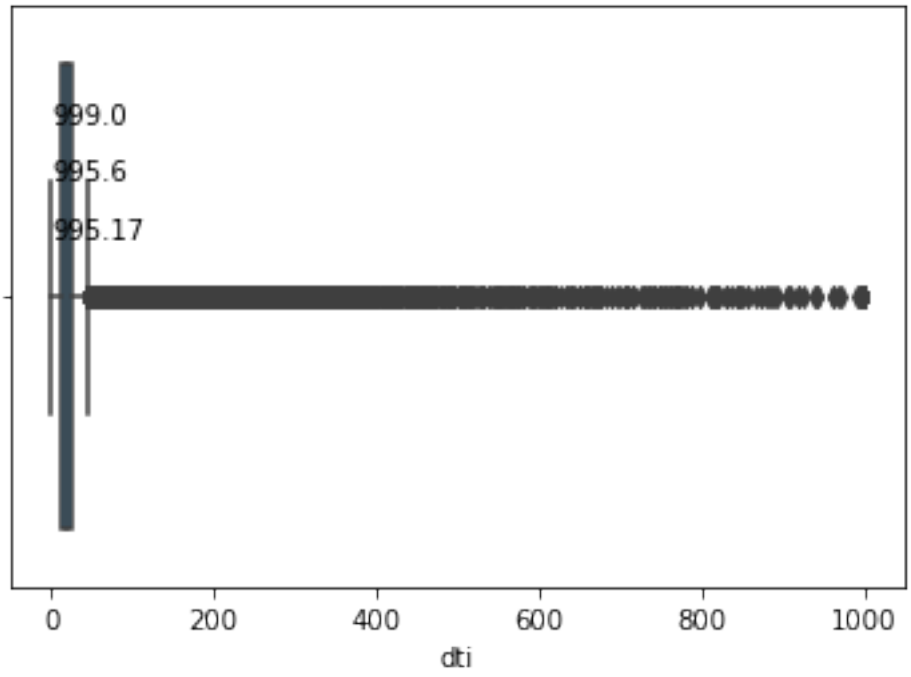

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



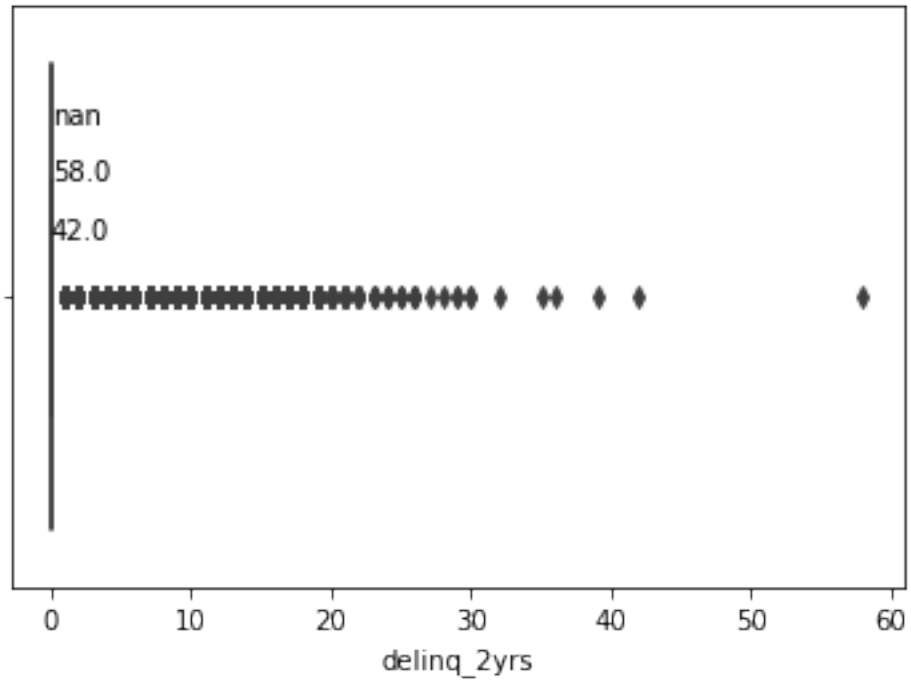

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



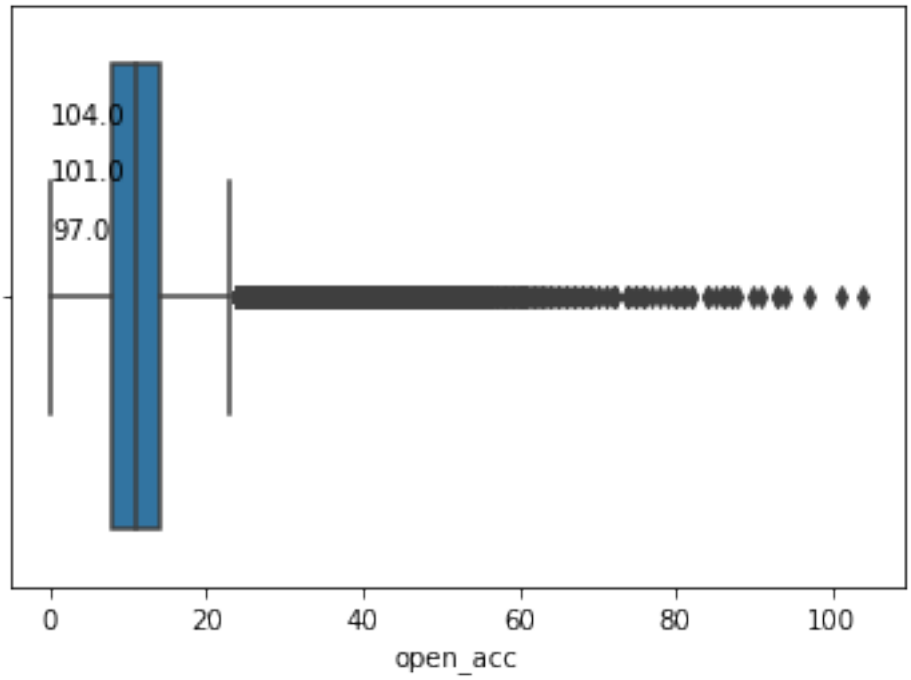

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



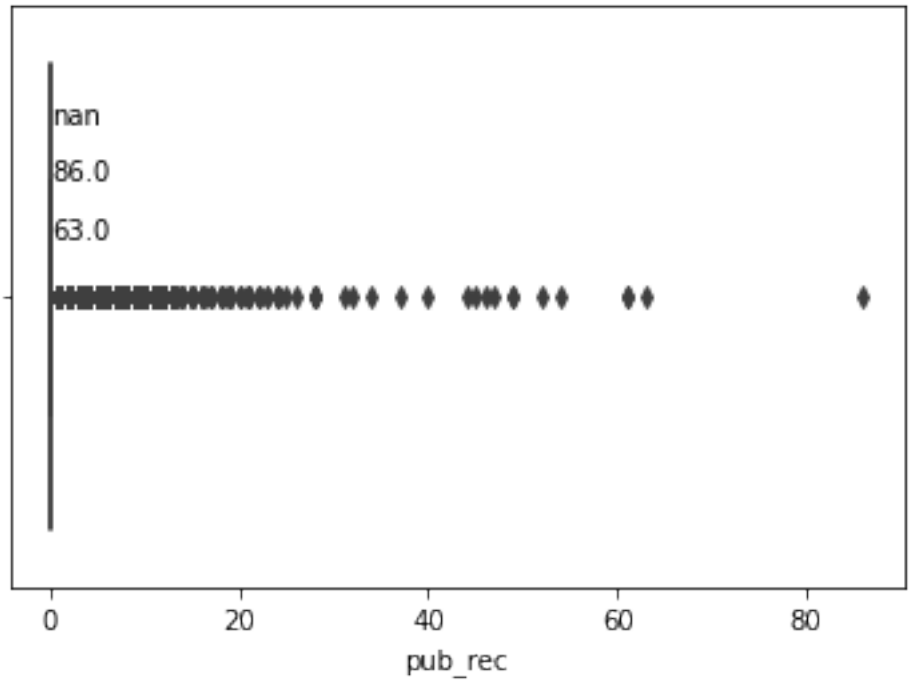

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



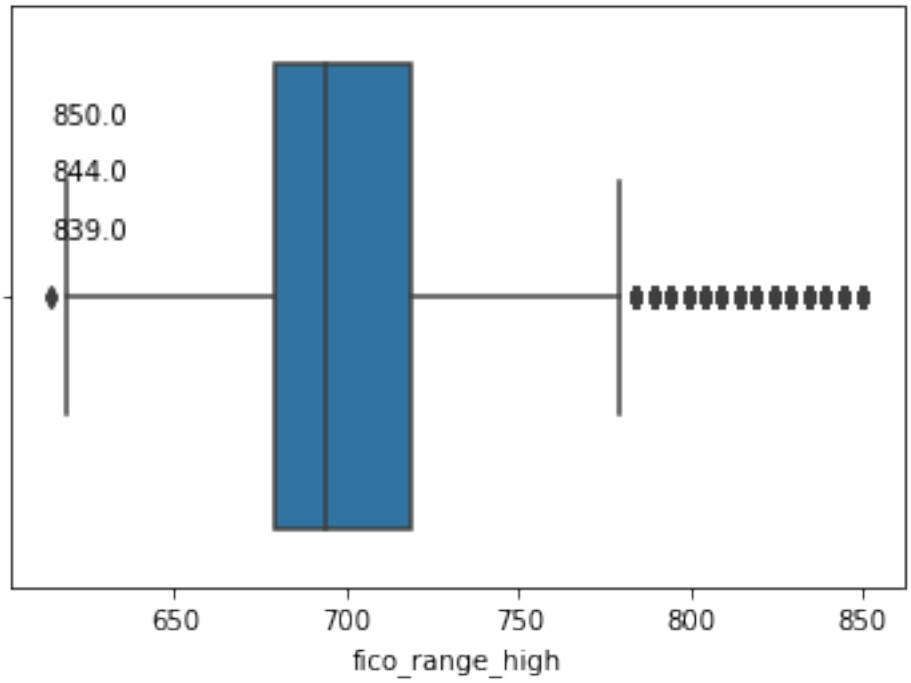

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



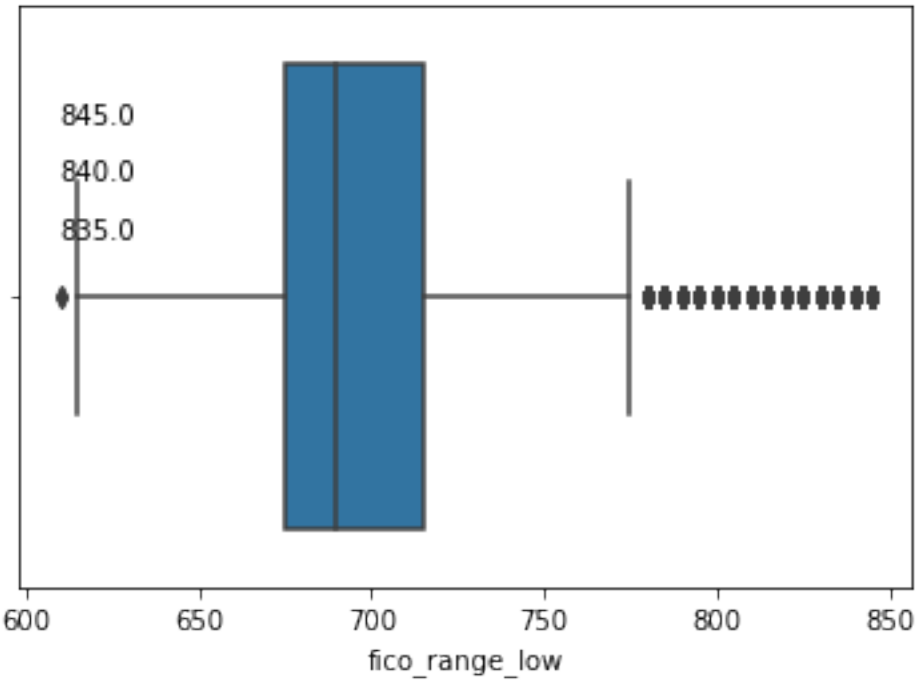

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



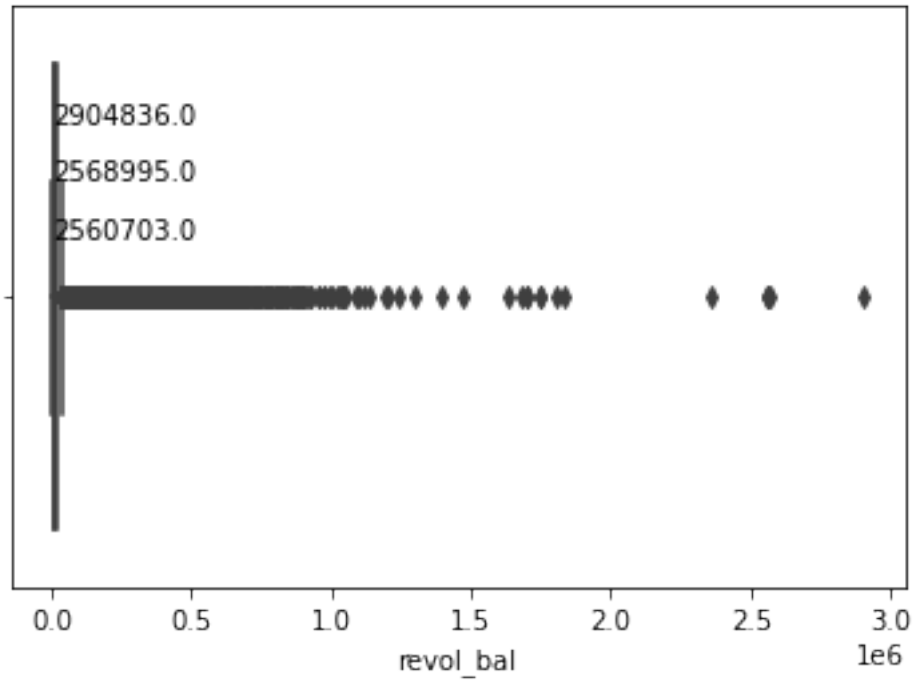

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



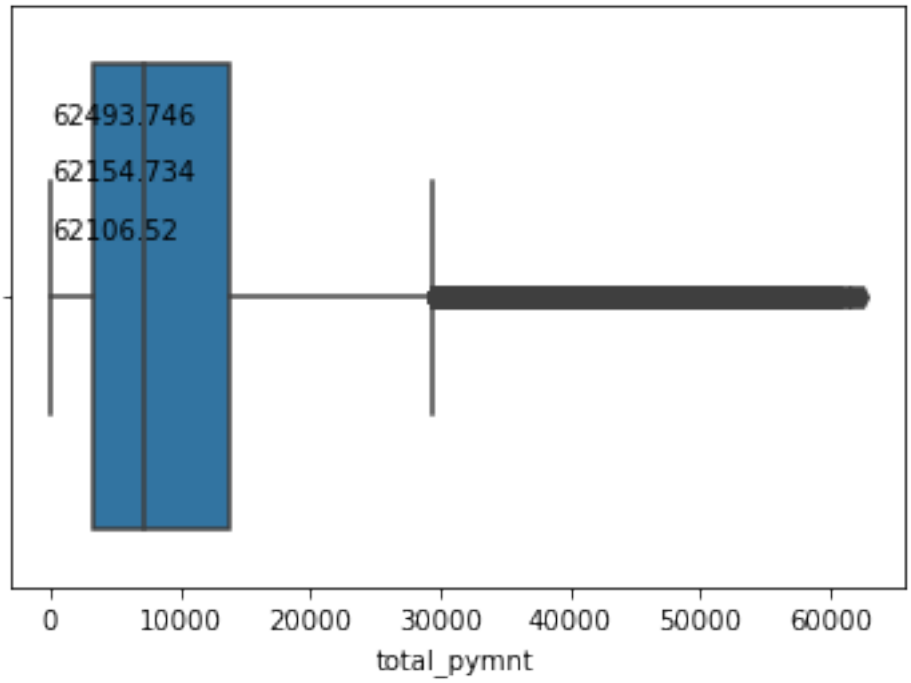

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



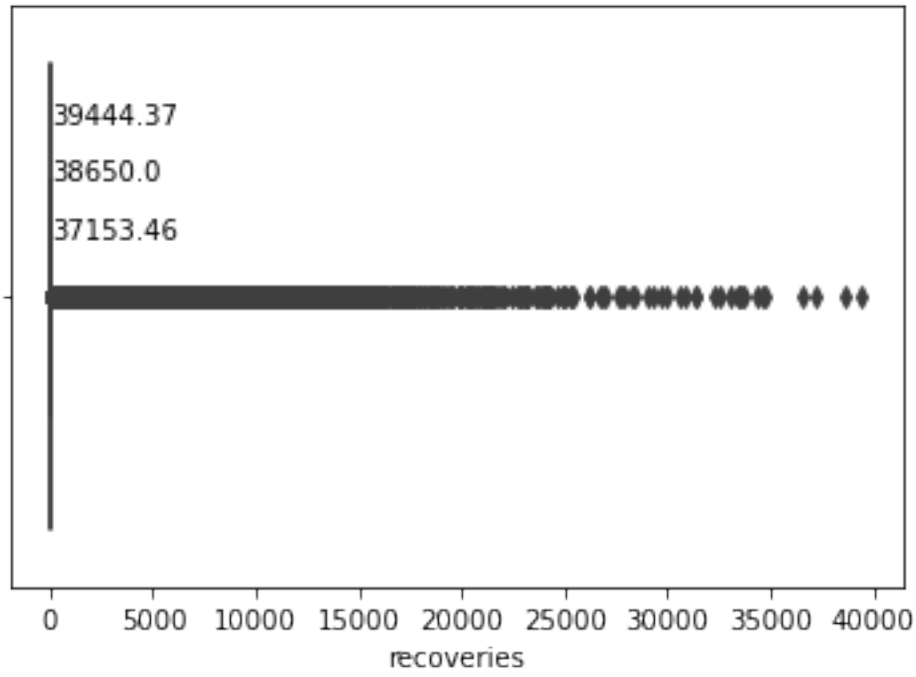

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



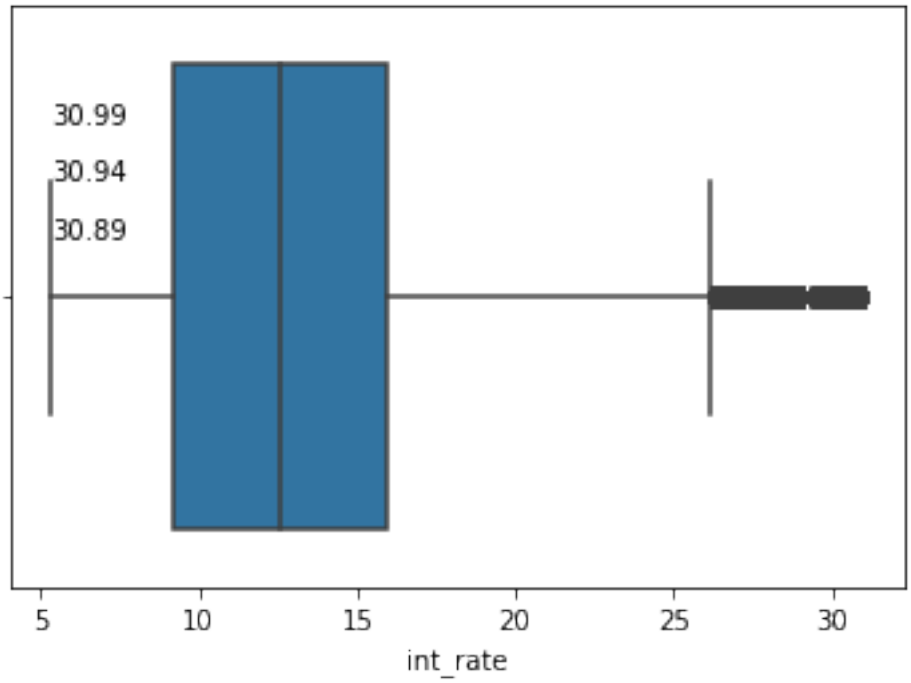

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



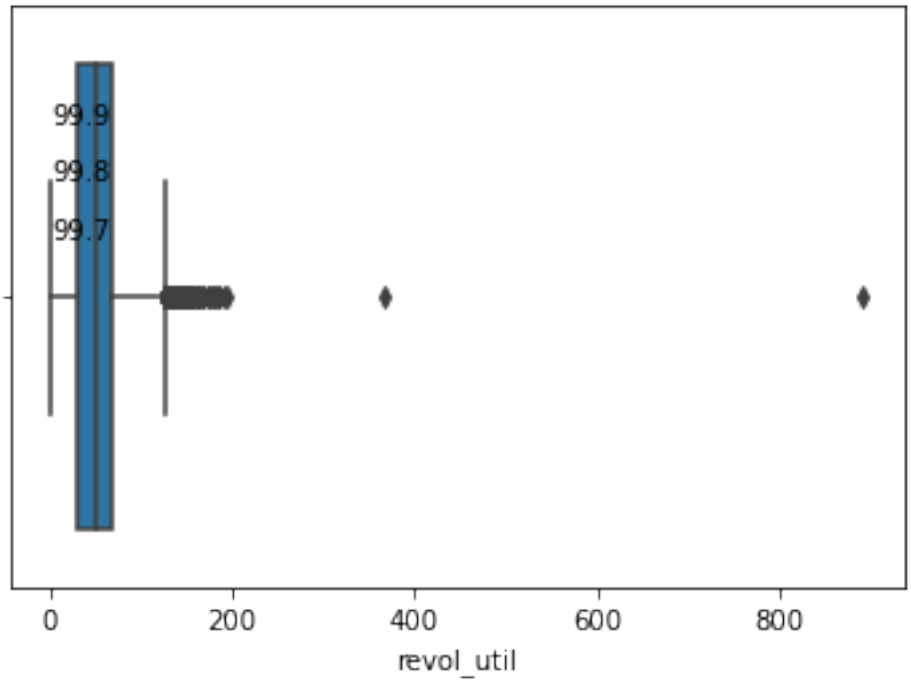

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



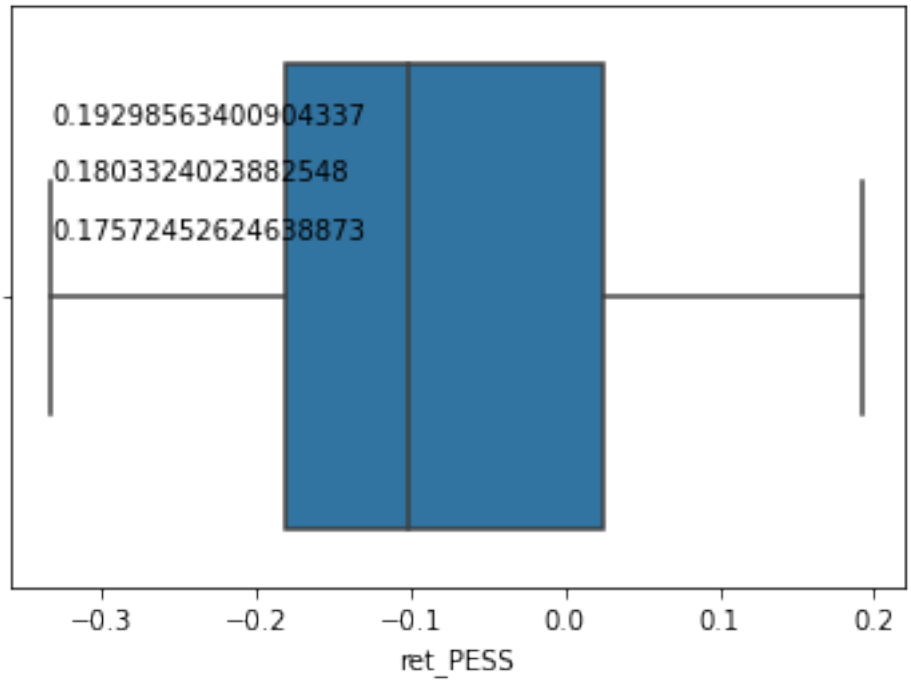

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



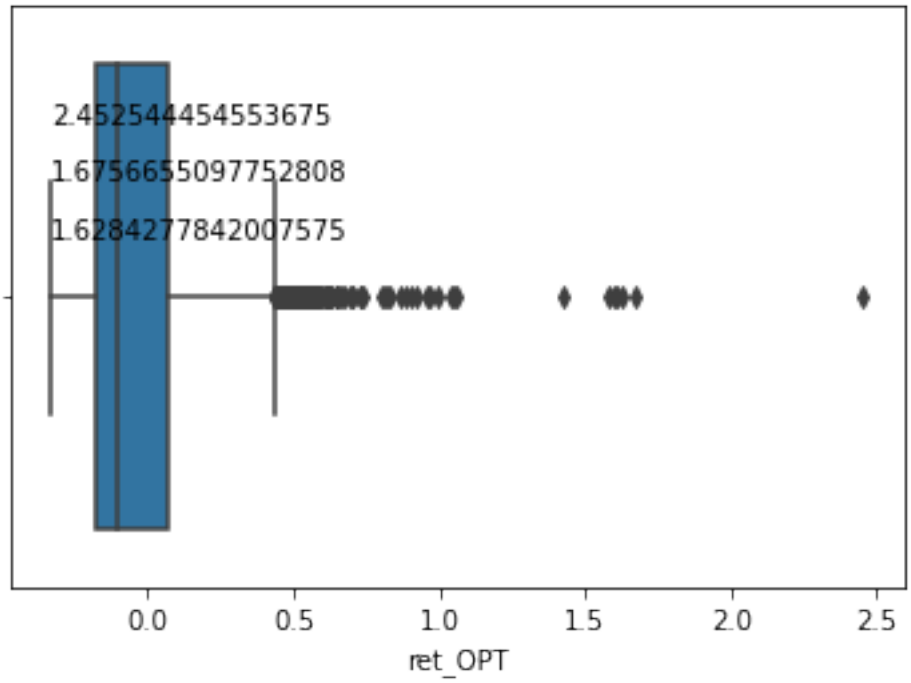

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



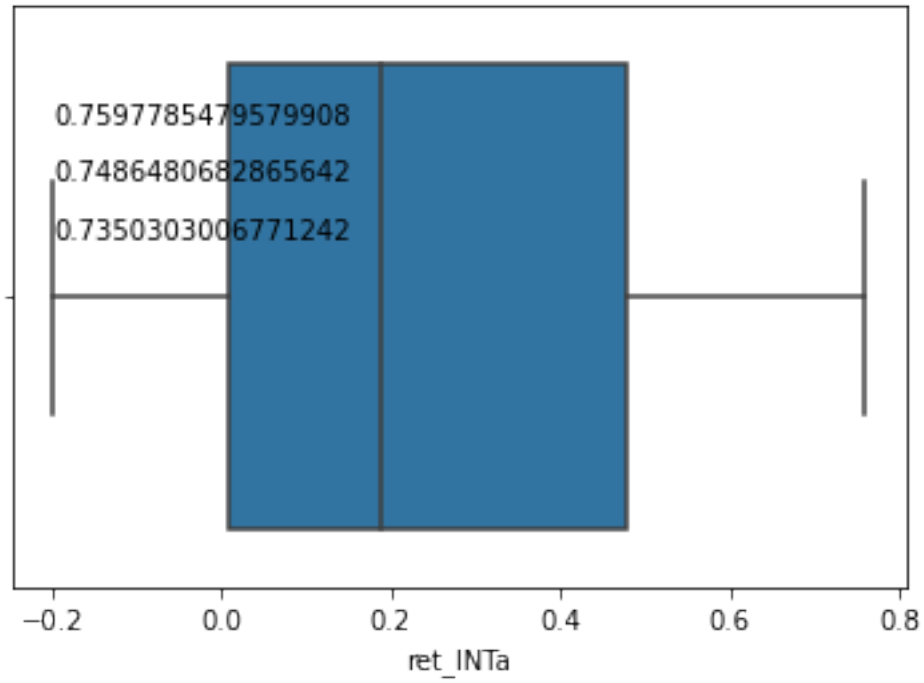

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



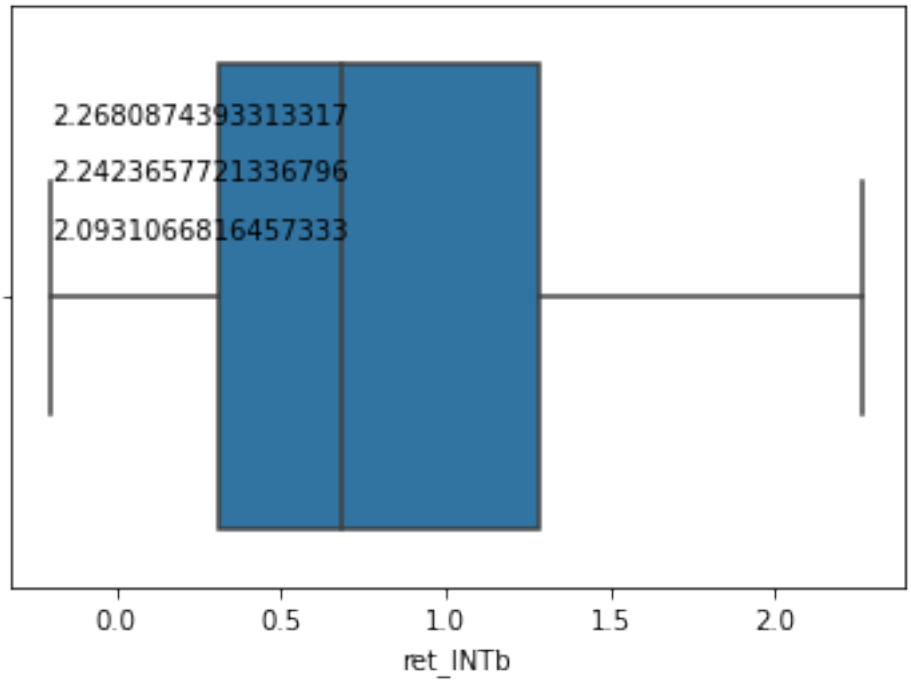

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```




```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



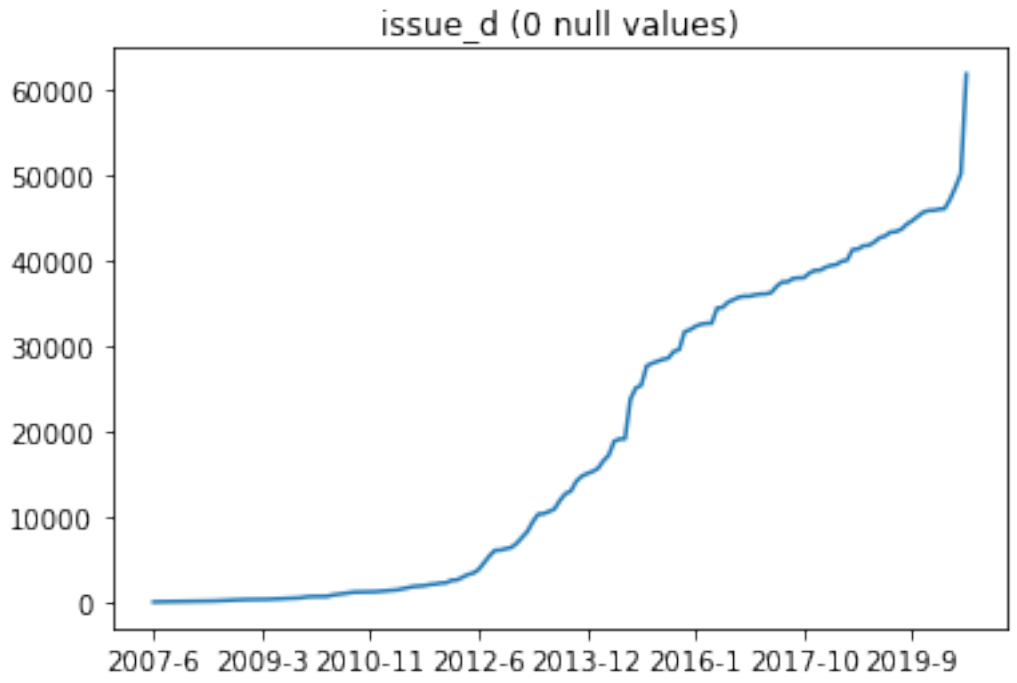

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

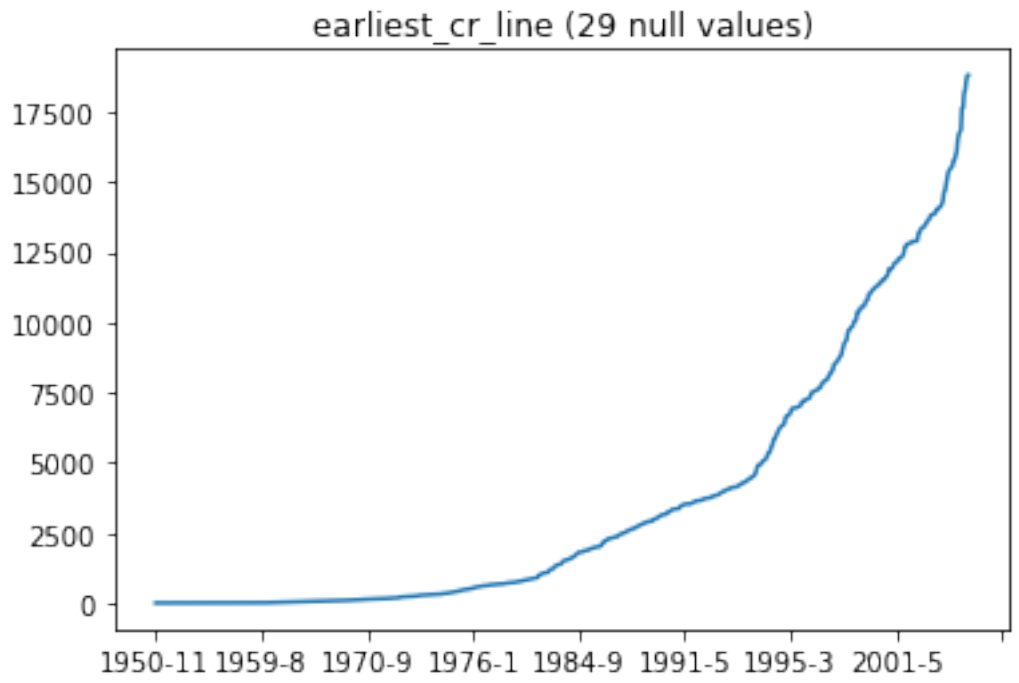


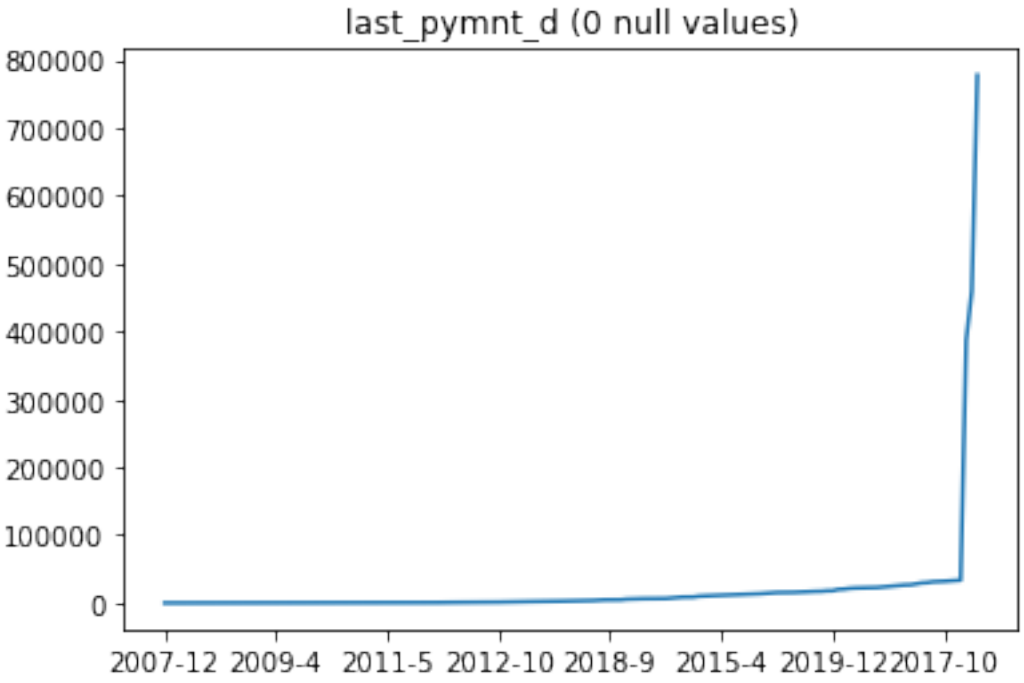

```
Column term has 2 unique values
 36 months    1947319
 60 months     814799
Name: term, dtype: int64


Column grade has 7 unique values
B    810672
C    767781
A    595897
D    396411
E    137921
F     41450
G     11986
Name: grade, dtype: int64


Column emp_length has 12 unique values
10+ years    897562
< 1 year     253308
2 years      247089
3 years      219473
None         191759
1 year       182798
5 years      171573
```

```
4 years      166329
6 years      123540
7 years      109983
8 years      107362
9 years       91342
Name: emp_length, dtype: int64


Column home_ownership has 6 unique values
MORTGAGE    1357826
RENT        1089070
OWN          311595
ANY            3390
OTHER           182
NONE             55
Name: home_ownership, dtype: int64


Column verification_status has 3 unique values
Source Verified    1065494
Not Verified        996591
Verified            700033
Name: verification_status, dtype: int64


Column loan_status has 9 unique values
Current                                                1559755
Fully Paid                                              901142
Charged Off                                             235079
Late (31-120 days)                                       33803
In Grace Period                                          19614
Late (16-30 days)                                         9049
Does not meet the credit policy. Status:Fully Paid        1988
Default                                                    939
Does not meet the credit policy. Status:Charged Off        749
Name: loan_status, dtype: int64


Column purpose has 14 unique values
debt_consolidation    1551563
credit_card            653450
home_improvement       181170
other                  167299
major_purchase          58928
medical                 33093
small_business          28520
car                     28144
vacation                19150
```

```
house                   18308
moving                  18056
wedding                  2342
renewable_energy         1674
educational               421
Name: purpose, dtype: int64
```




earliest_cr_line (29 null values)
17500
15000
12500
10000
7500
5000
2500
0
1950-11
1959-8
1970-9
1976-1
1984-9
1991-5
1995-3
2001-5


last_pymnt_d (0 null values)
800000
700000
600000
500000
400000
300000
200000
100000
0
2007-12
2009-4
2011-5
2012-10
2018-9
2015-4
2019-12
2017-10

1.4.5 Drop current and old loans

[31]:
```
# Remove all loans that are still CURRENT
n_rows = len(final_data)
non_current_statuses = ['Fully Paid', 'Charged Off', 'Default']

final_data = final_data[final_data['loan_status'].isin(non_current_statuses)]

print("Removed " + str(n_rows - len(final_data)) + " rows")
```

```
Removed 1624958 rows
```

[32]:
```
# Only include loans isssued since 2010
n_rows = len(final_data)

final_data = final_data[final_data['issue_d'] >= datetime.date(2010, 1, 1)]

print("Removed " + str(n_rows - len(final_data)) + " rows")
```

```
Removed 6516 rows
```

1.4.6 Drop null values

[33]:
```
# Deal with null values. We allow cateogrical variables to be null
# OTHER than grade, which is a particularly important categorical.
# All non-categorical variables must be non-null, and we drop
# rows that do not meet this requirement

required_cols = set(cols_to_pick) - set(cat_cols) - set(["id"])
required_cols.add("grade")

n_rows = len(final_data)

final_data = final_data.dropna(subset=required_cols) # drop rows that contain␣
 ↪null based only on "required_cols"

print("Removed " + str(n_rows - len(final_data)) + " rows")
```

```
Removed 1434 rows
```

1.4.7 Imput annual_inc, dti of 0 with median

[34]:
```
final_data.loc[final_data.annual_inc == 0,'annual_inc'] =␣
 ↪final_data['annual_inc'].median()
final_data.loc[final_data.dti == 0,'dti'] = final_data['dti'].median()
```

1.4.8 Handle outliers

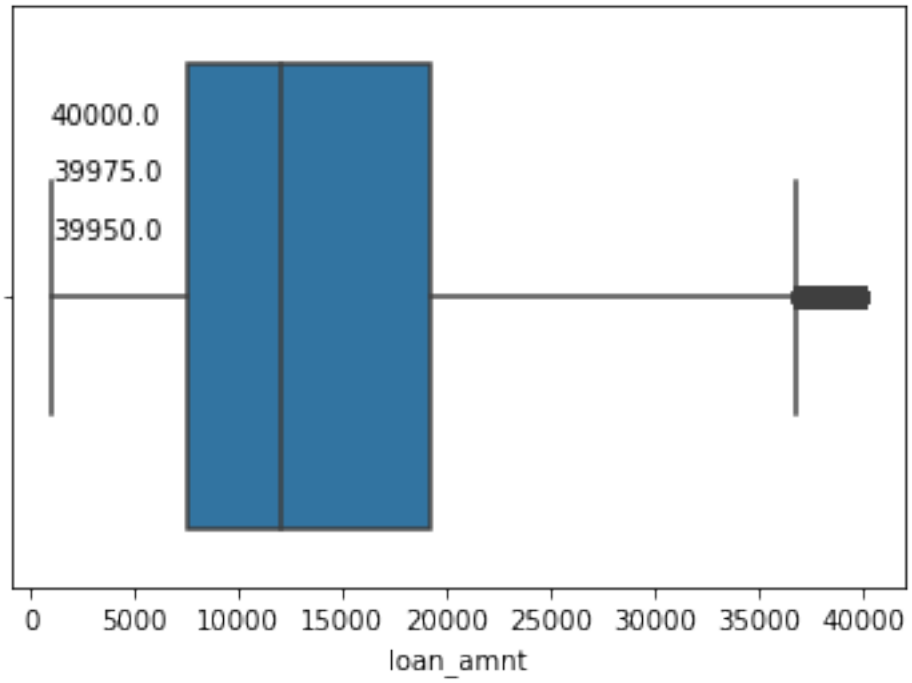
[35]:
```
# There are quite a few outliers.
# Please identify top-k (decide this based on the visualization) features where␣
 ↪outliers are most obvious

n_rows = len(final_data)

outliers = ['annual_inc', 'dti', 'open_acc', 'total_pymnt', 'recoveries',␣
 ↪'revol_bal', 'revol_util']
for i in outliers:
    final_data = final_data[np.abs(final_data[i] - final_data[i].mean()) <= (3␣
 ↪* final_data[i].std())] # remove outliers based kth obvious feature

print("Removed " + str(n_rows - len(final_data)) + " rows")
```

```
Removed 73750 rows
```

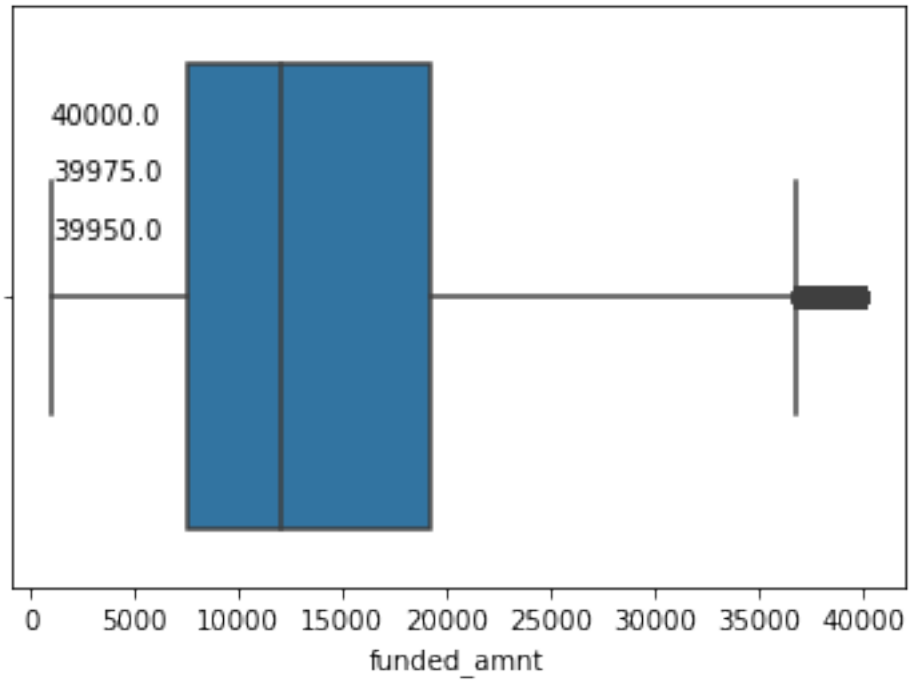
1.4.9 Visualize clean data

[36]:
```
# Visualize the data again after cleaning
visualize_float_columns()
visualize_cat_columns()
visualize_date_columns()
```

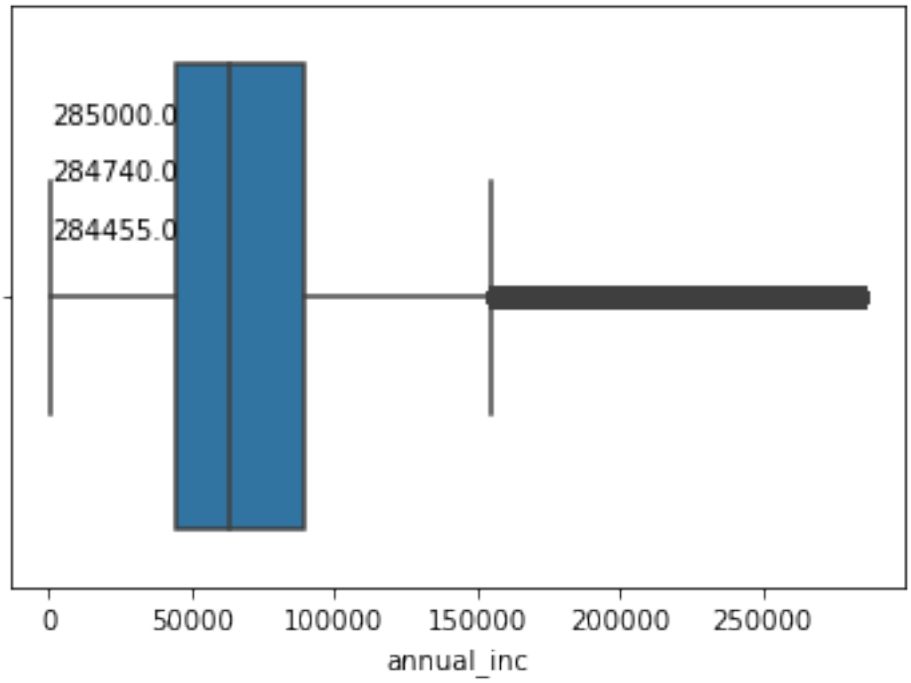
```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



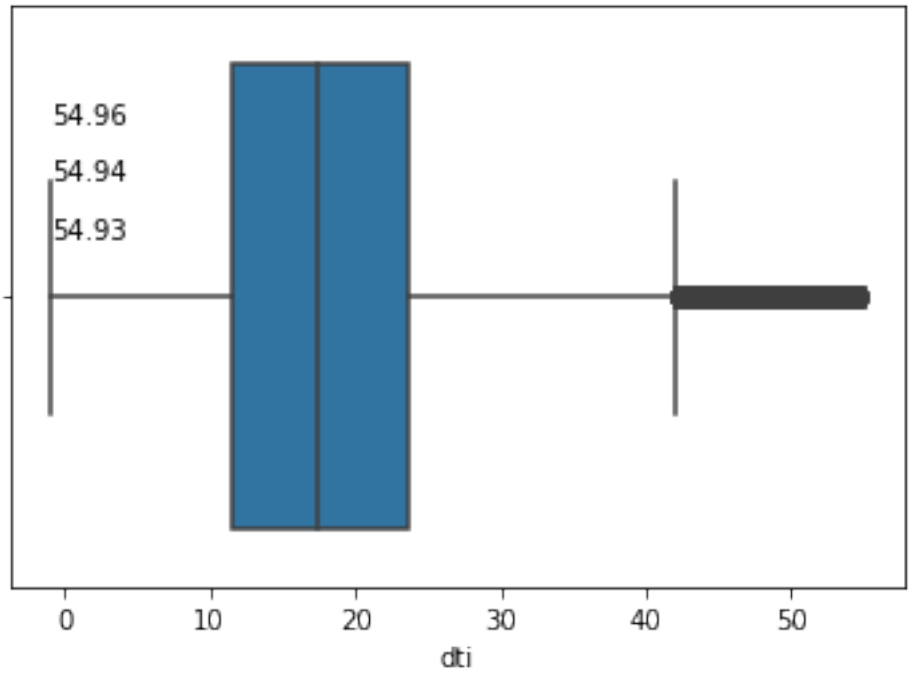

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



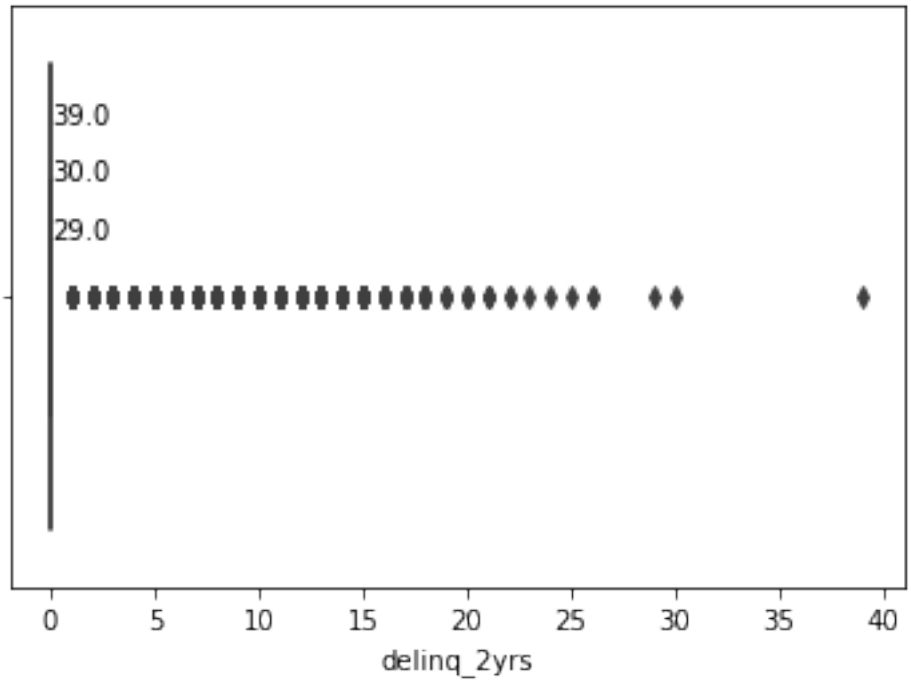

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



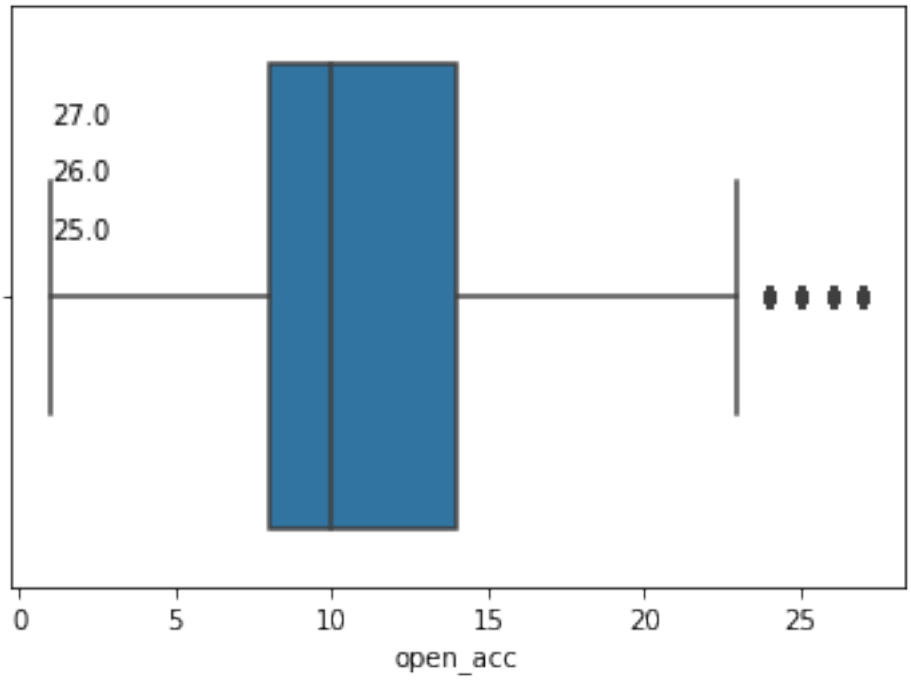

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



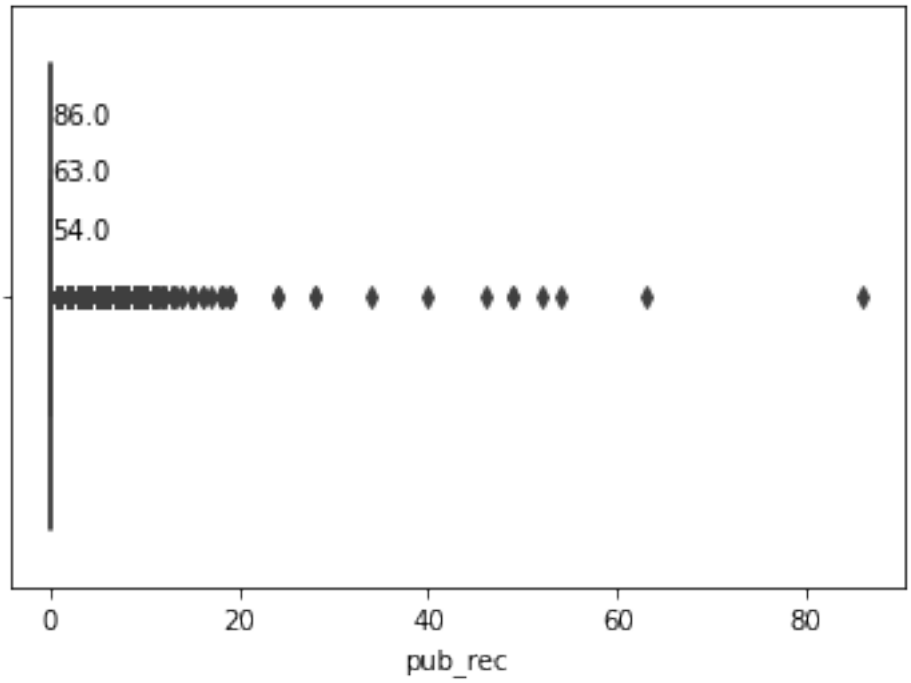

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



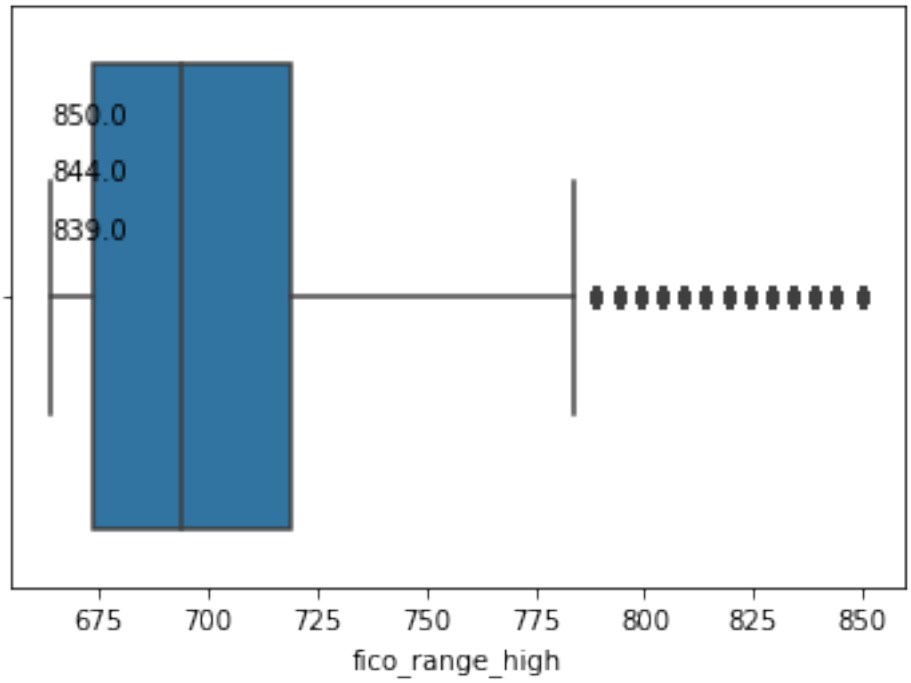

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



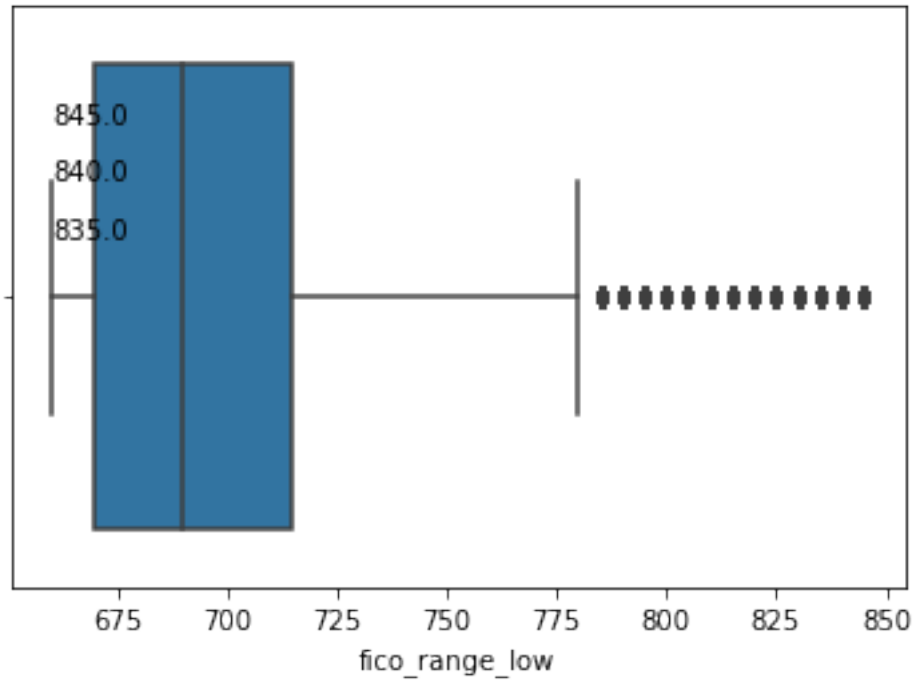

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



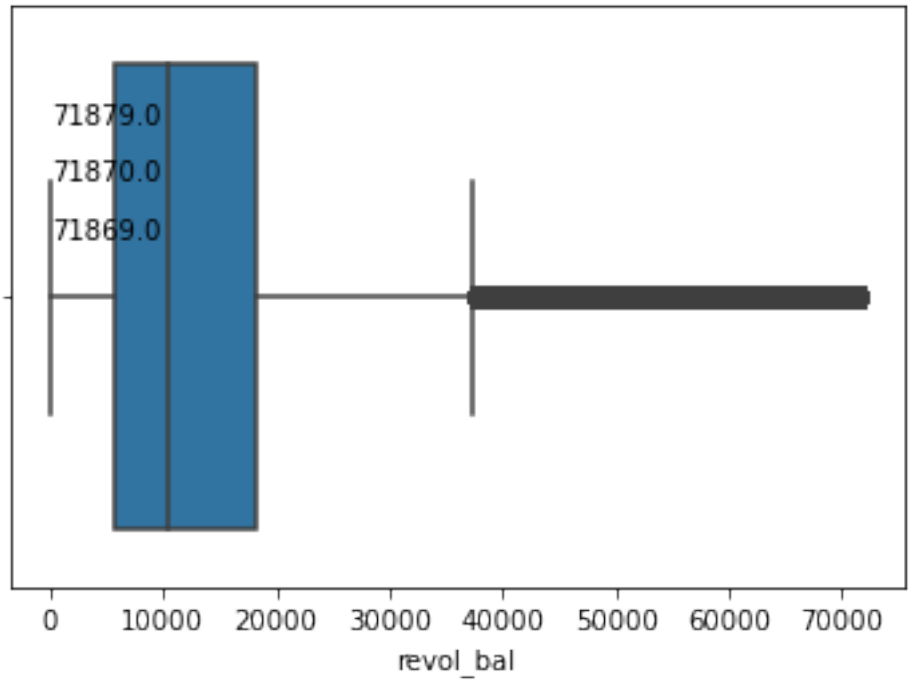

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



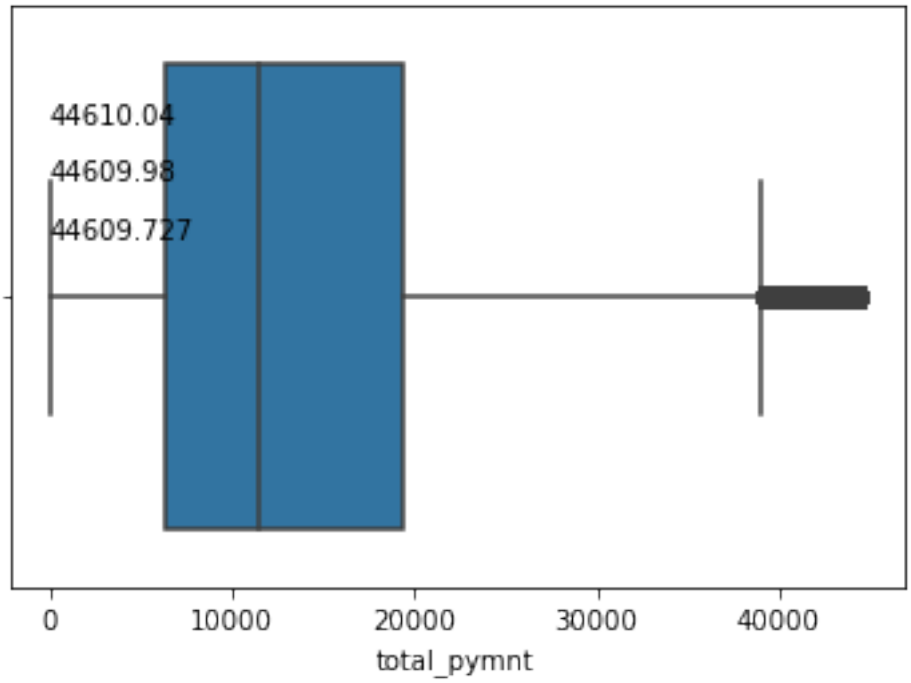

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



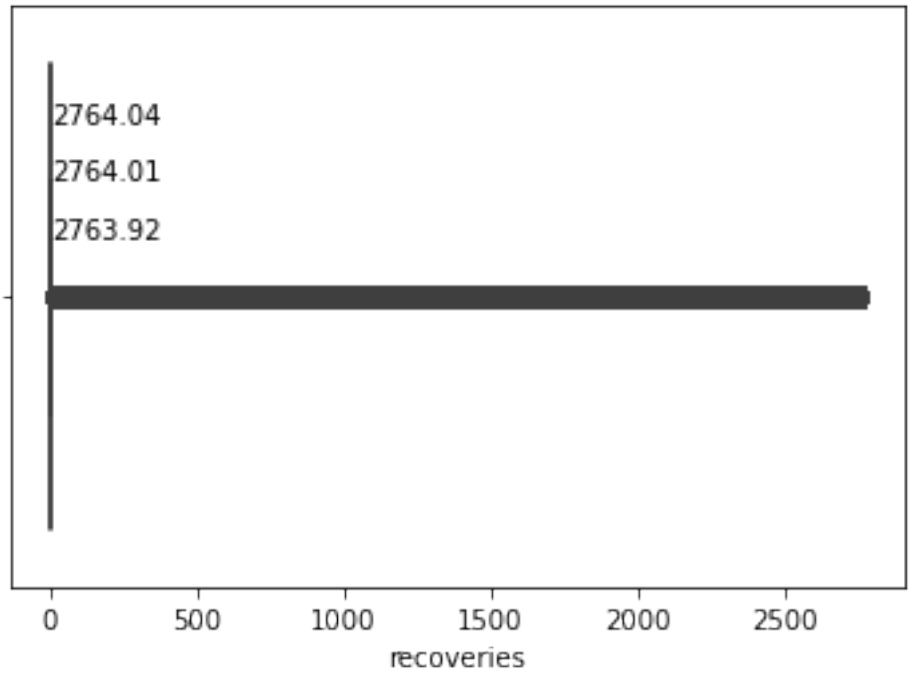

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



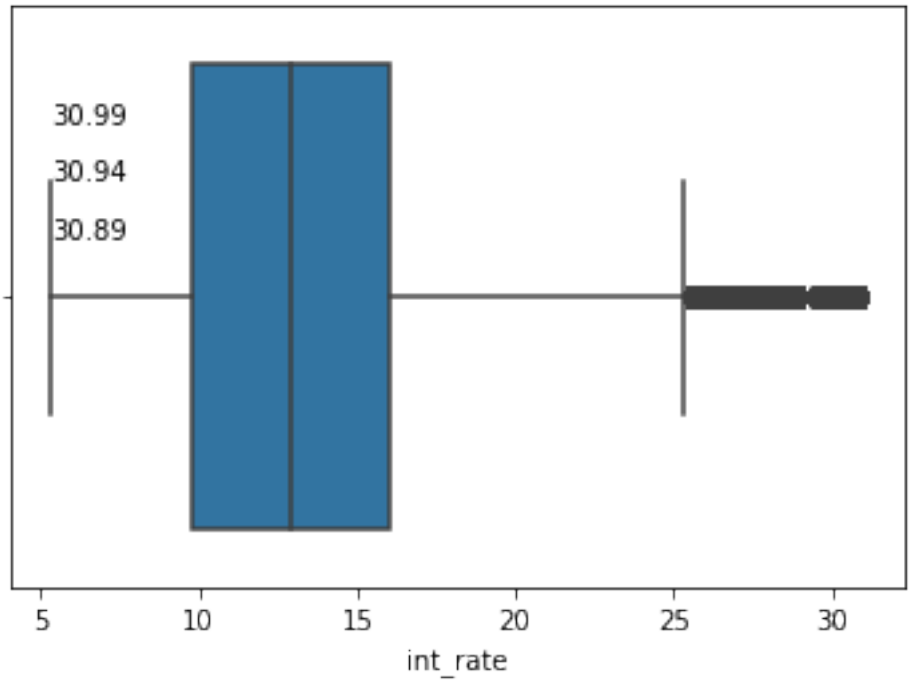

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



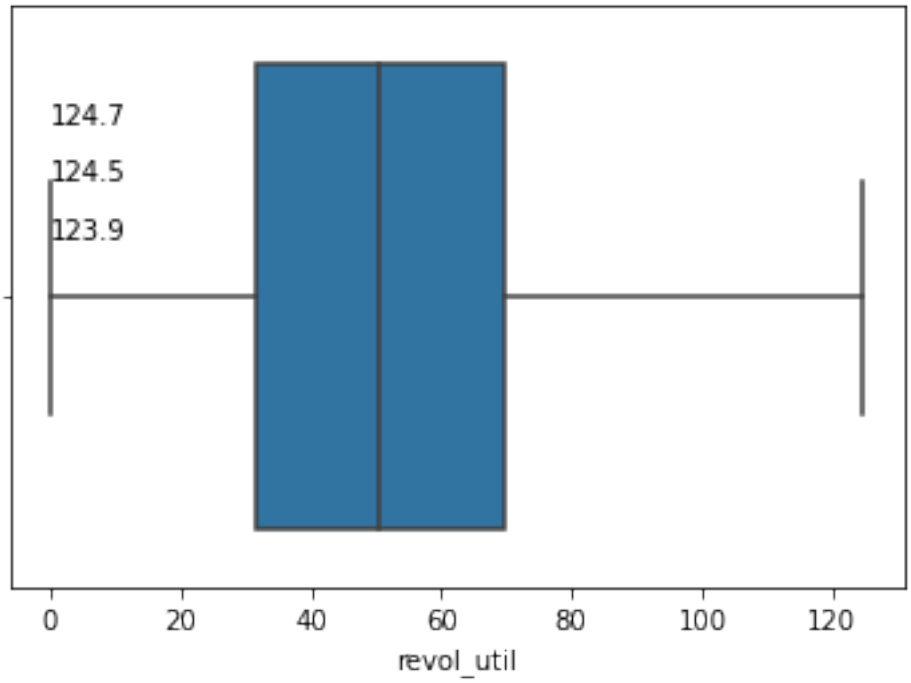

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



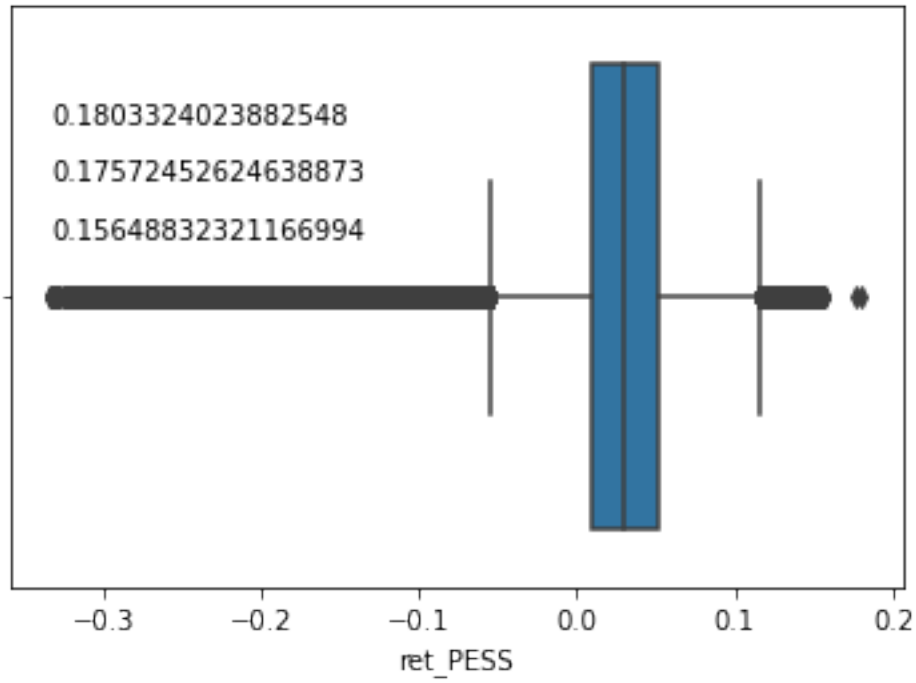

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



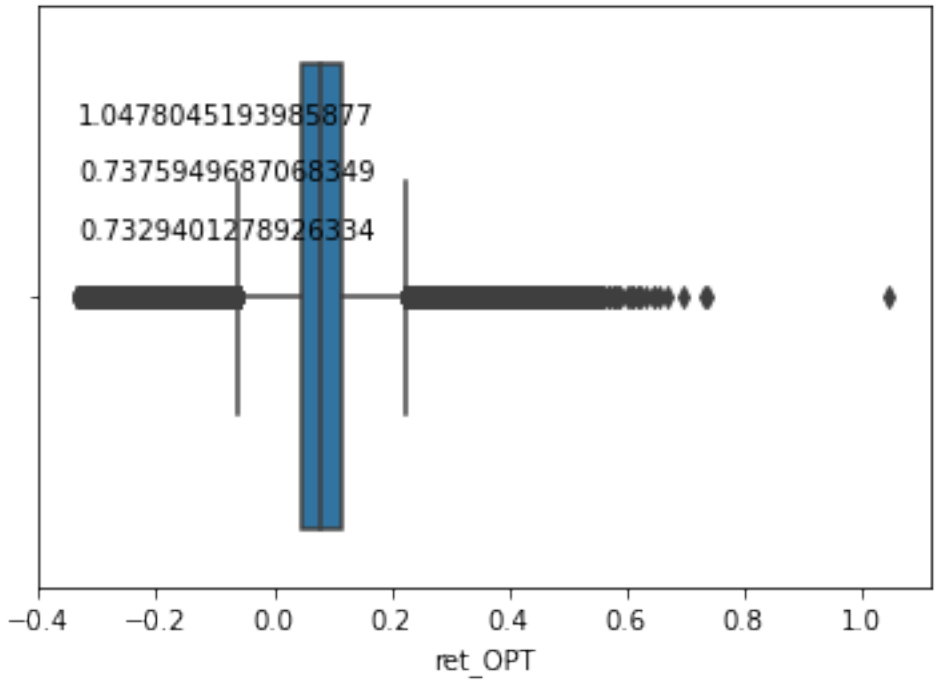

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



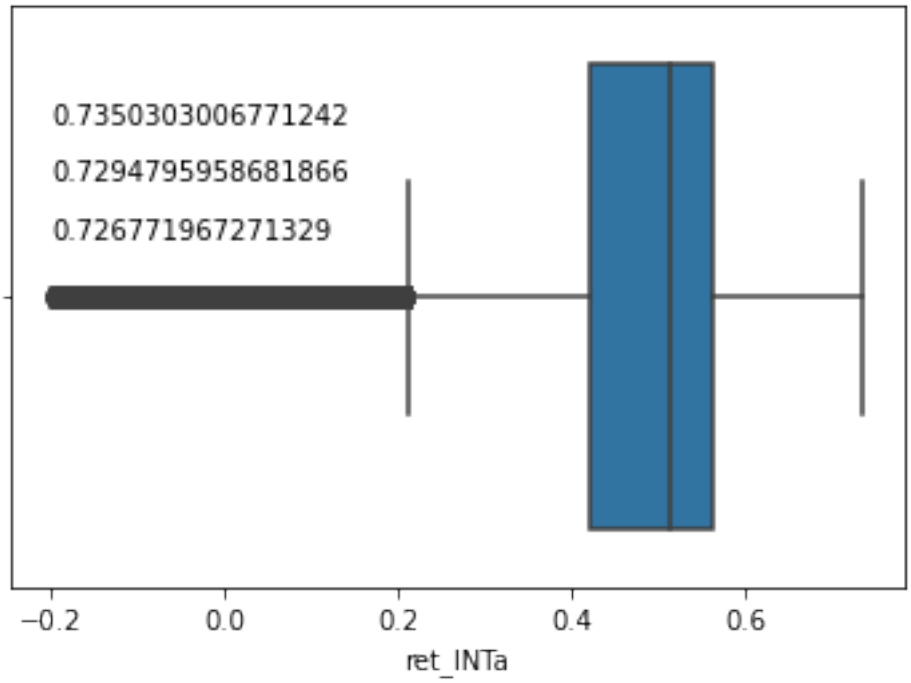

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```



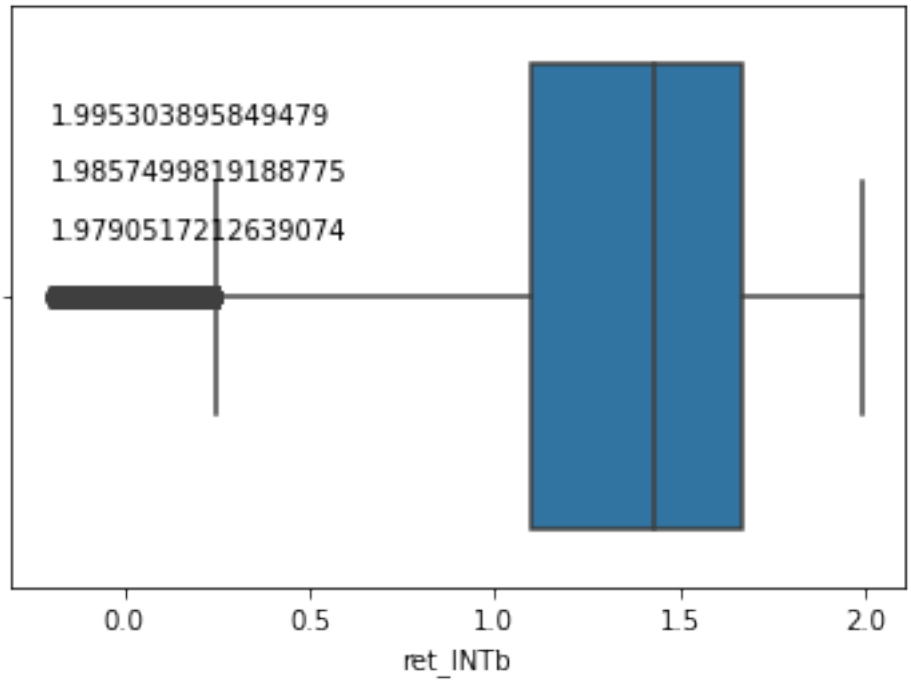

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```




```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

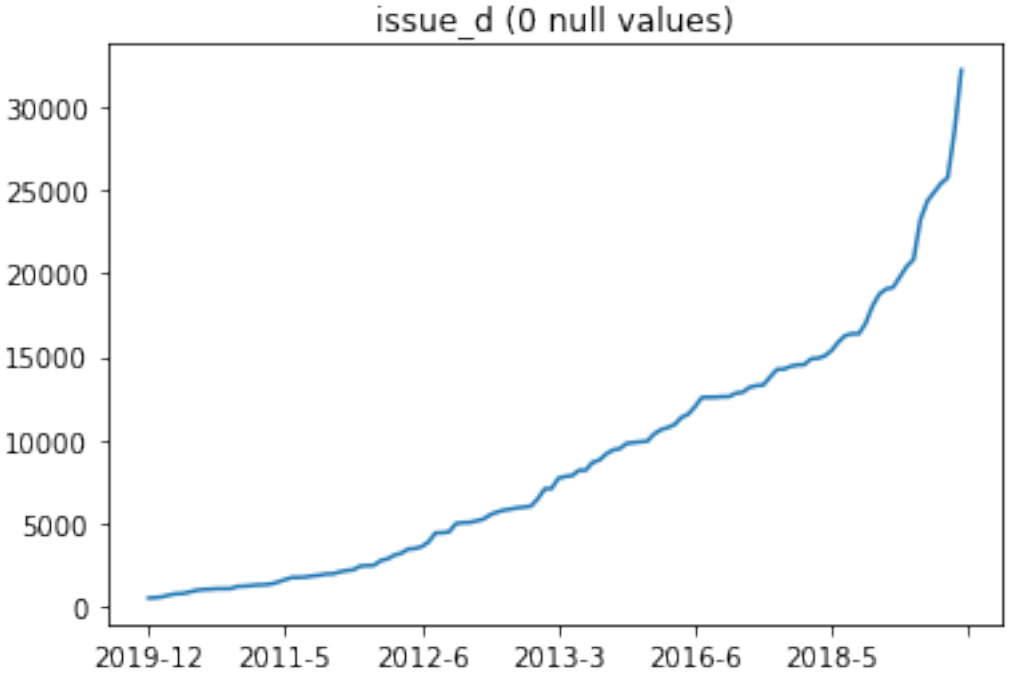


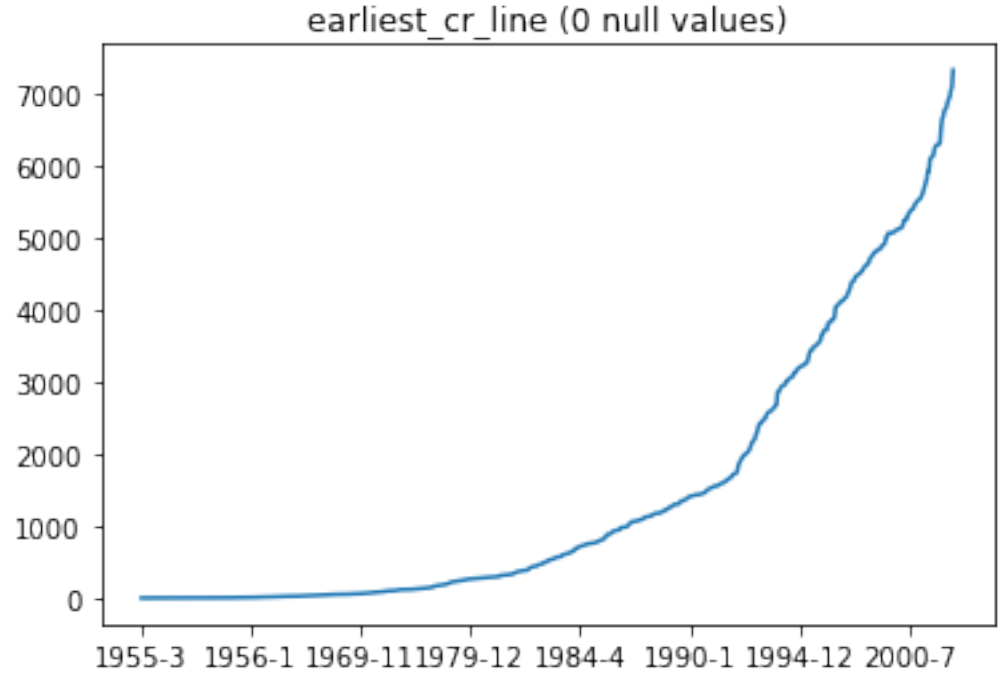

```
C:\Users\Kelly\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

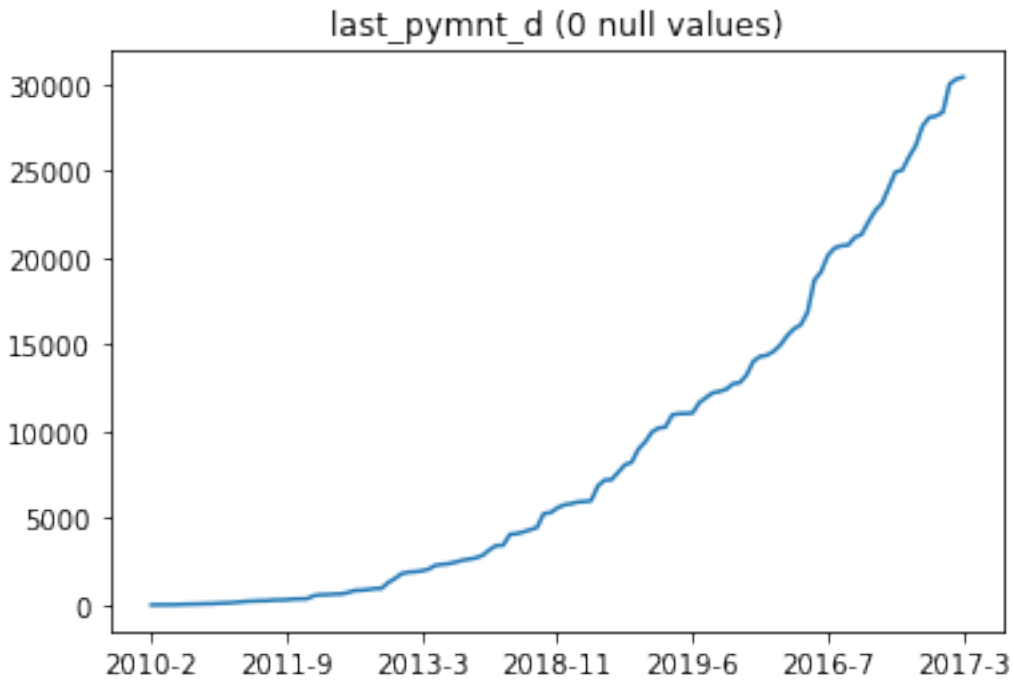



```
Column term has 2 unique values
 36 months    812568
 60 months    242892
Name: term, dtype: int64


Column grade has 7 unique values
B    308254
C    293037
A    195679
D    161877
E     68942
F     22119
G      5552
Name: grade, dtype: int64


Column emp_length has 12 unique values
10+ years    339529
2 years       95994
< 1 year      85249
3 years       85117
1 year        69892
5 years       67290
4 years       64166
```

```
None          63655
6 years       50952
7 years       47959
8 years       47221
9 years       38436
Name: emp_length, dtype: int64


Column home_ownership has 6 unique values
MORTGAGE    517804
RENT        425275
OWN         111706
ANY            588
OTHER           48
NONE            39
Name: home_ownership, dtype: int64


Column verification_status has 3 unique values
Source Verified    394420
Not Verified       351564
Verified           309476
Name: verification_status, dtype: int64


Column loan_status has 3 unique values
Fully Paid     853227
Charged Off    201333
Default           900
Name: loan_status, dtype: int64


Column purpose has 14 unique values
debt_consolidation    612369
credit_card           231756
home_improvement       66633
other                  61947
major_purchase         23220
medical                12239
car                    11576
small_business         11320
moving                  7610
vacation                7367
house                   6545
wedding                 2030
renewable_energy         752
educational               96
Name: purpose, dtype: int64
```


issue_d (0 null values)
30000
25000
20000
15000
10000
5000
0
2019-12
2011-5
2012-6
2013-3
2016-6
2018-5

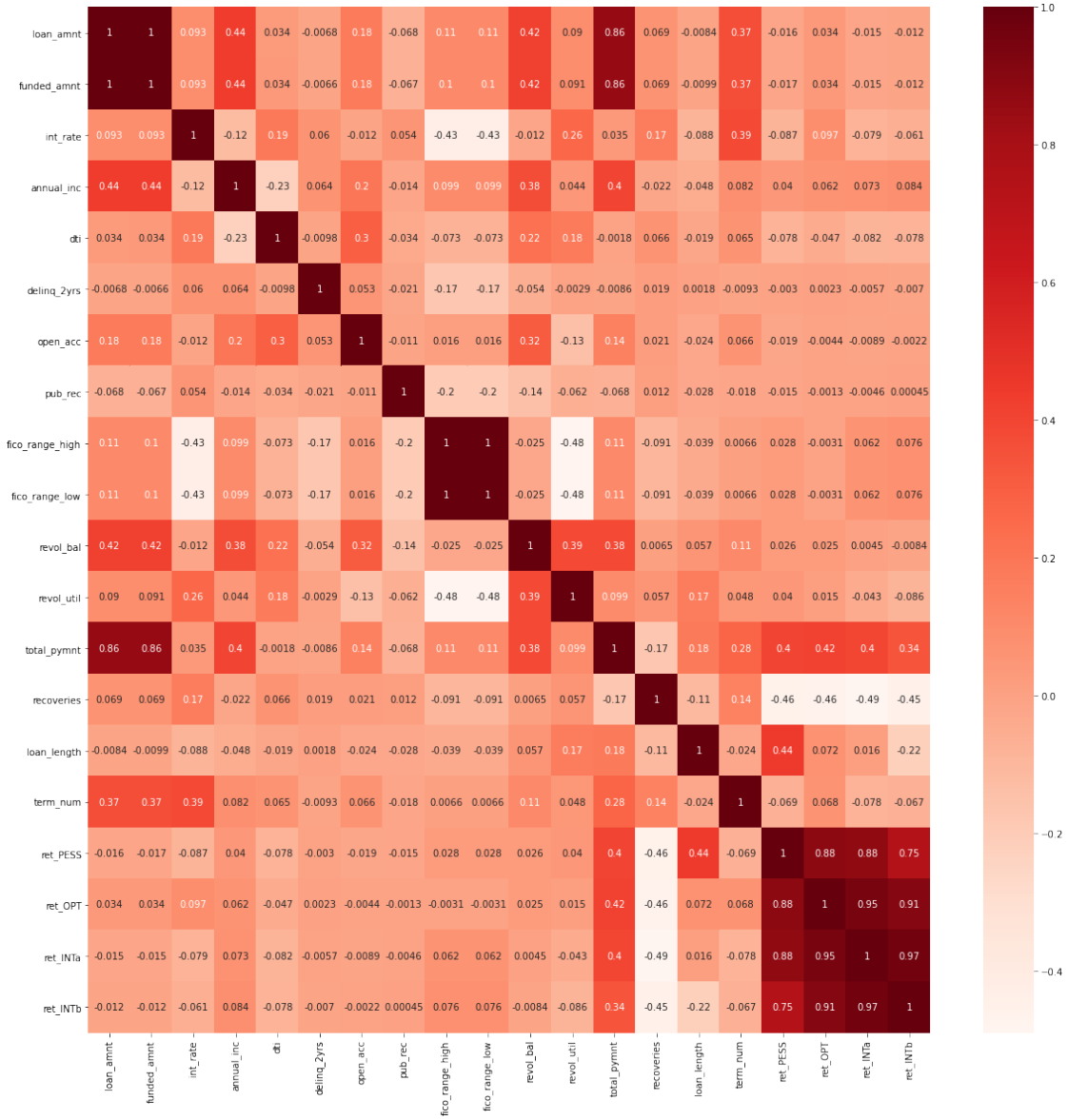

earliest_cr_line (0 null values)
7000
6000
5000
4000
3000
2000
1000
0
1955-3
1956-1
1969-11
1979-12
1984-4
1990-1
1994-12
2000-7



What do you observe after removing the outliers?

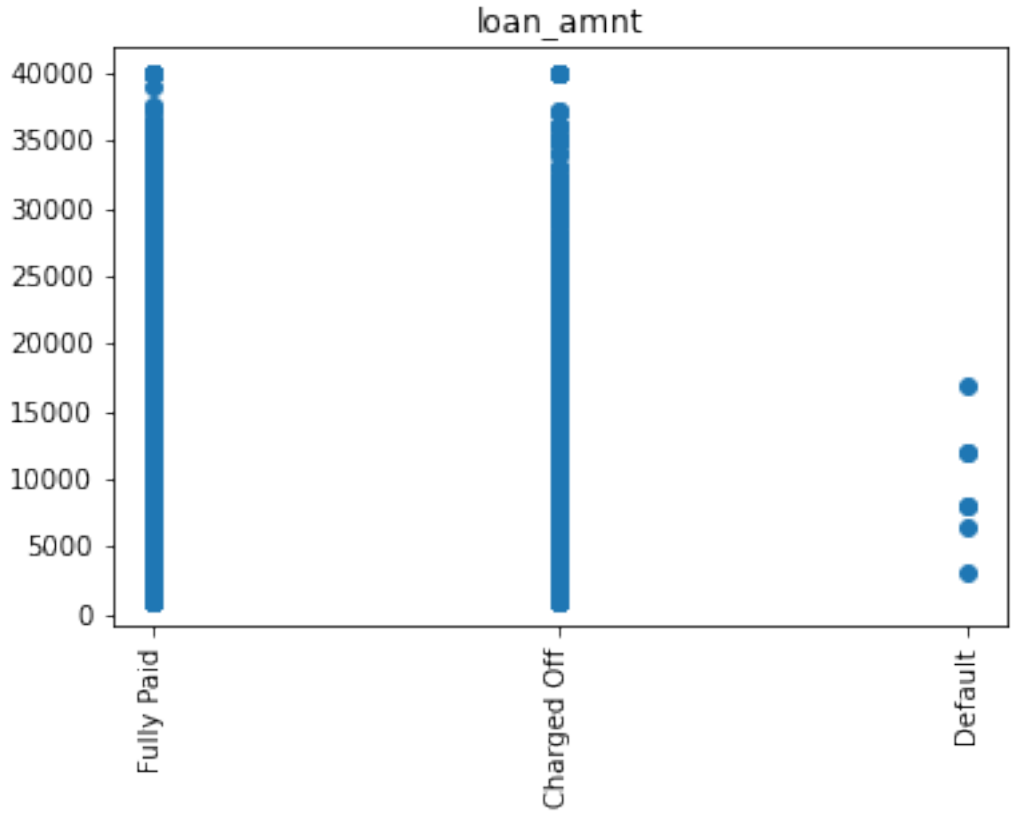
After removing outliers, we can see there was a significant decrease in variability in each of the features that we cleaned from outliers. The data is now more uniform and allows us to create return values that are statistically significant with skewed data. This also has an impact on the data as the mean and median will now be closer together because they are no longer skewed by the outliers. We also see that the data timelines are smoothed out with the removal of outliers.

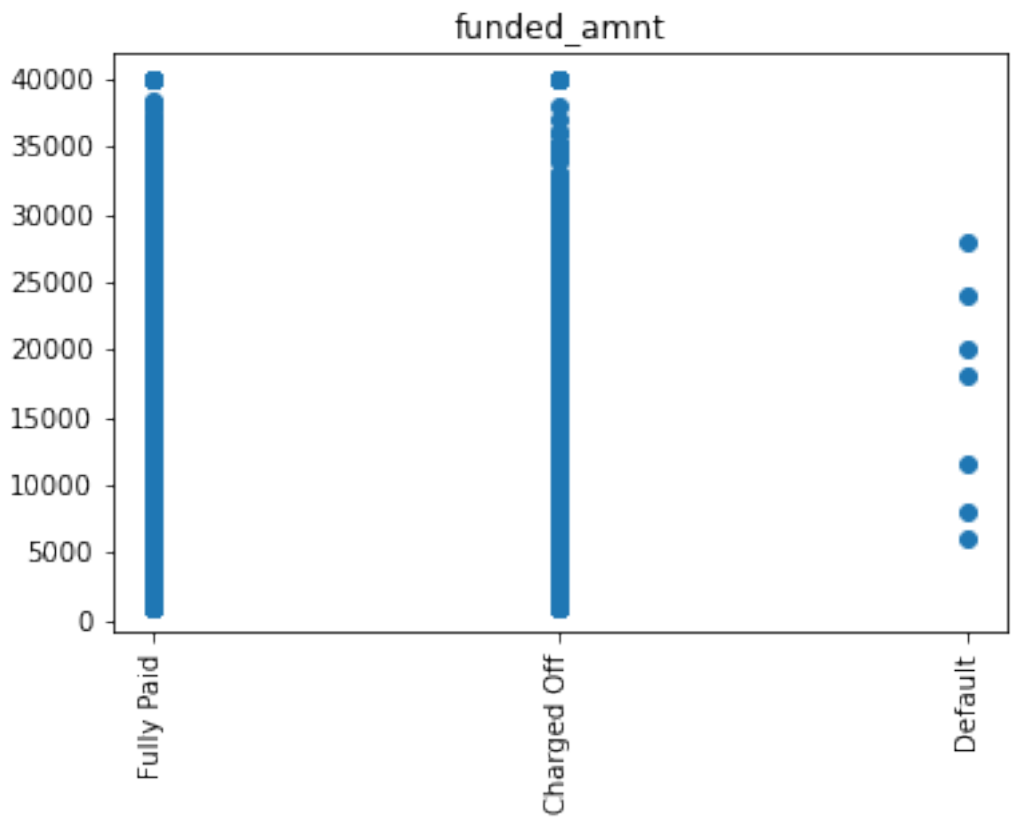
[37]:
```
# Visualize the feature correlations
    # You can compute the correlation among features and display a heat-map of␣
 ↪the matrix
    # OR use sns scatter or pairplot
plt.figure(figsize=(20,20))
cor = final_data.corr()
seaborn.heatmap(cor, annot=True, cmap=plt.cm.Reds)
plt.show()
```

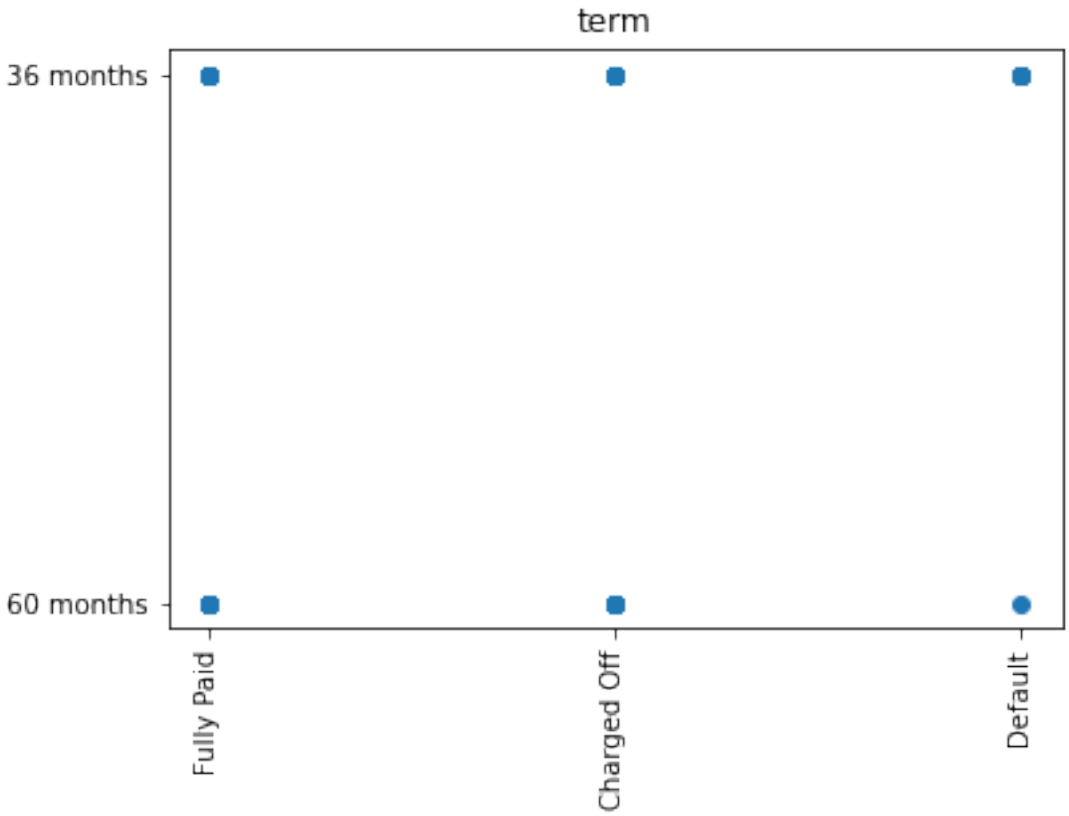


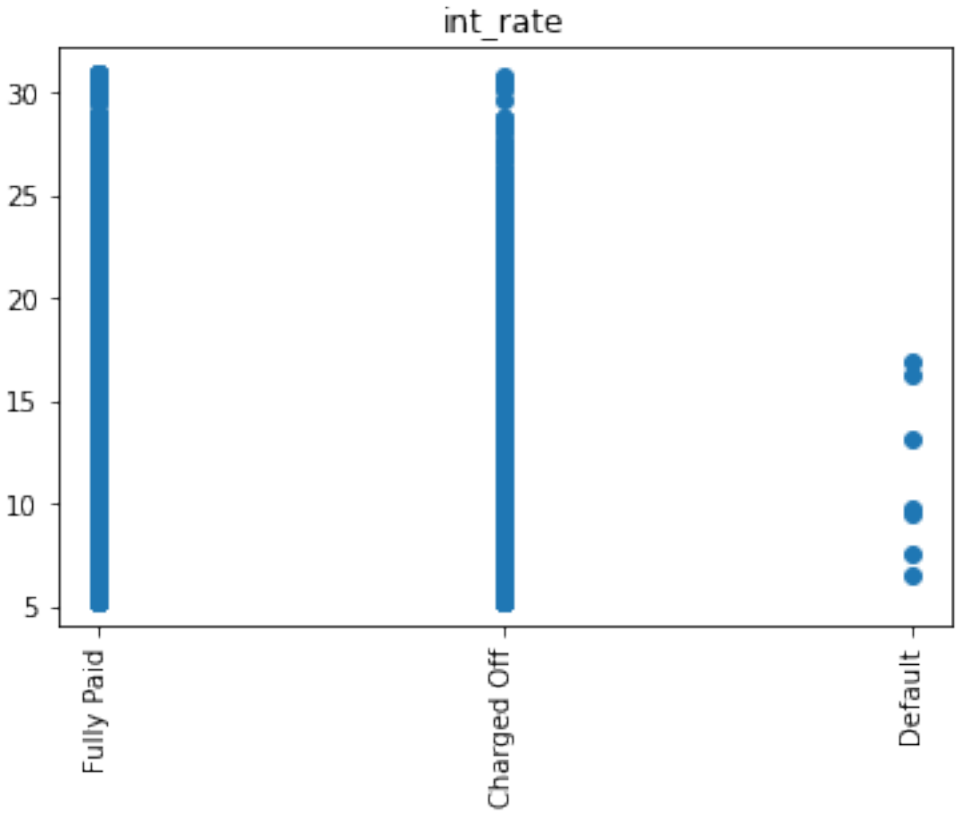
[38]:
```
# Visualize relation between loan status and features
# sns pairplot or scatter plot. Refer to recitations

status_samples = np.random.choice(final_data['loan_status'], 10000, replace =␣
↪False)
for i in final_data:
    if i == 'id':
        continue
    feature_samples = np.random.choice(final_data[i], 10000, replace=False)
    plt.scatter(status_samples, feature_samples)
    plt.xticks(rotation=90)
```

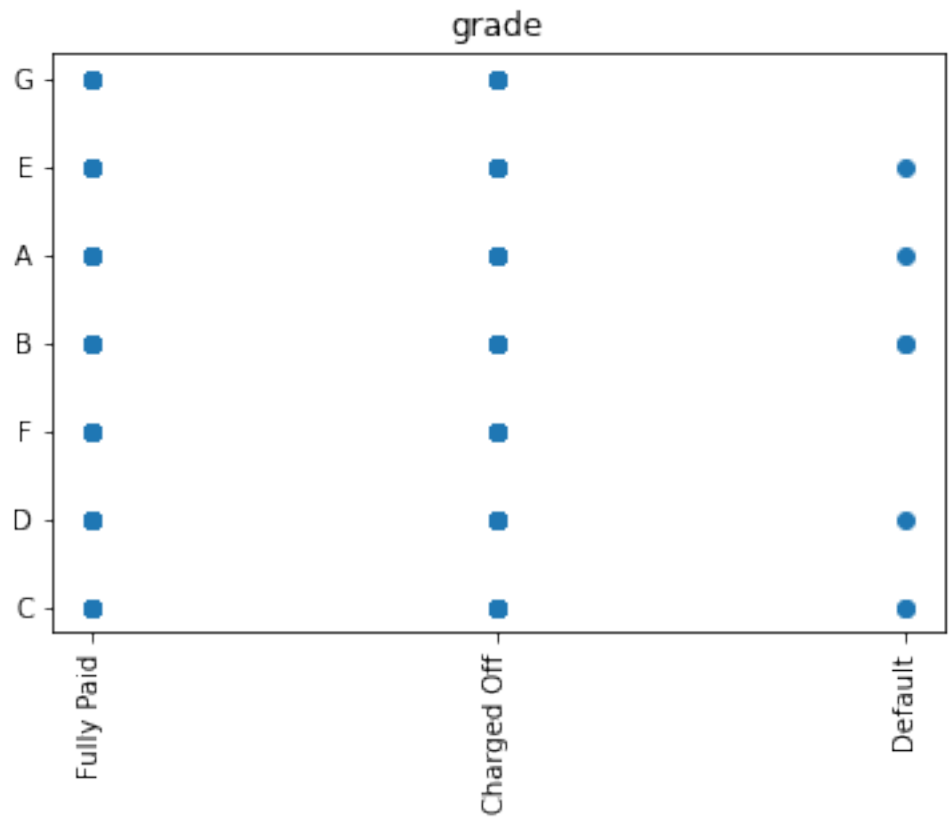
```
    plt.title(i)
    plt.show()
```

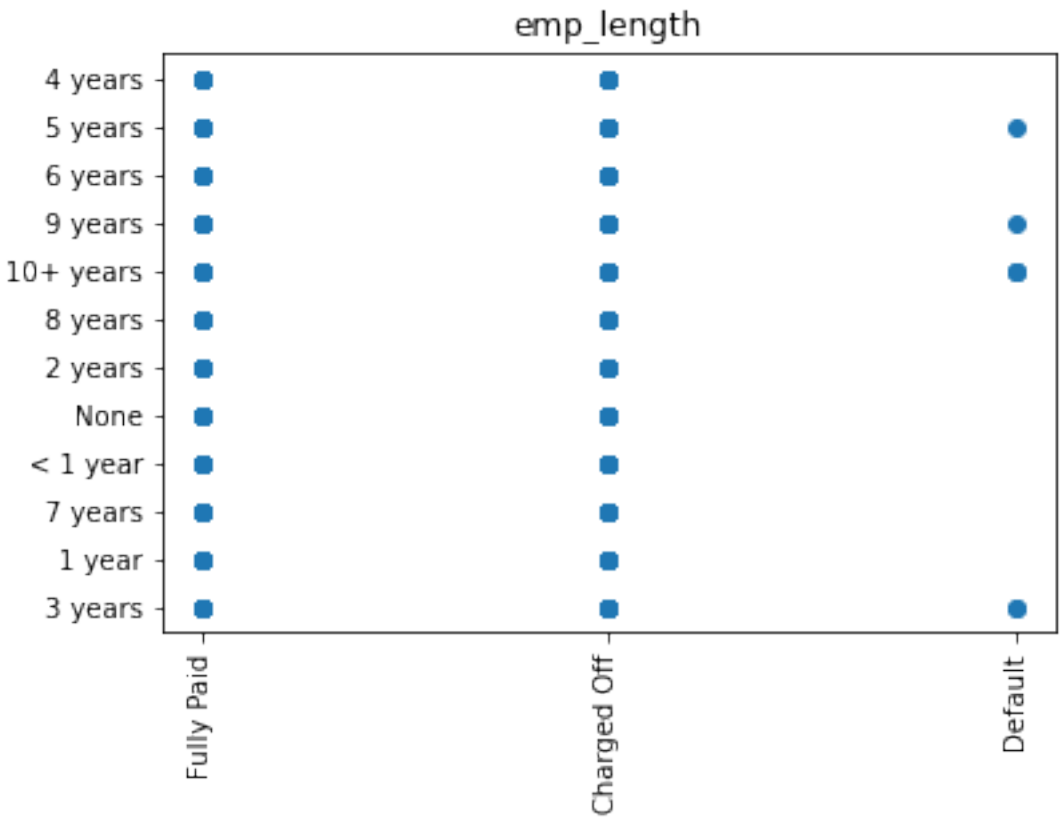


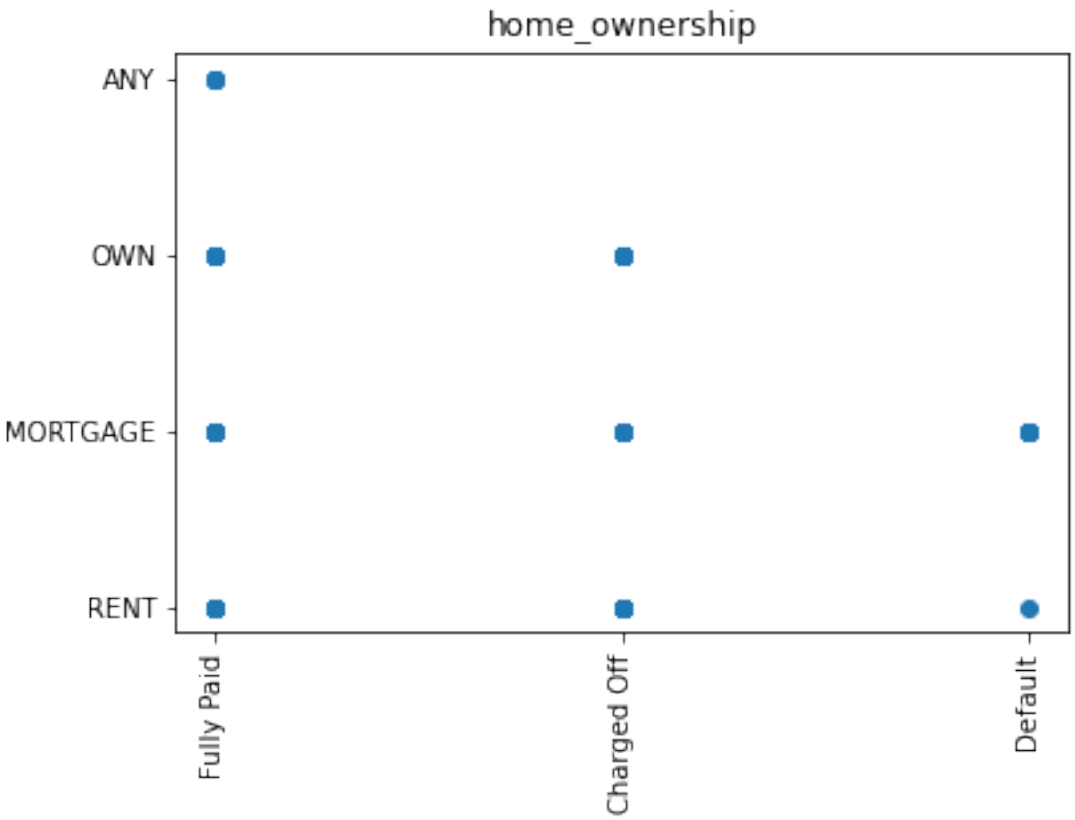

funded_amnt
40000
35000
30000
25000
20000
15000
10000
5000
0
Fully Paid
Charged Off
Default

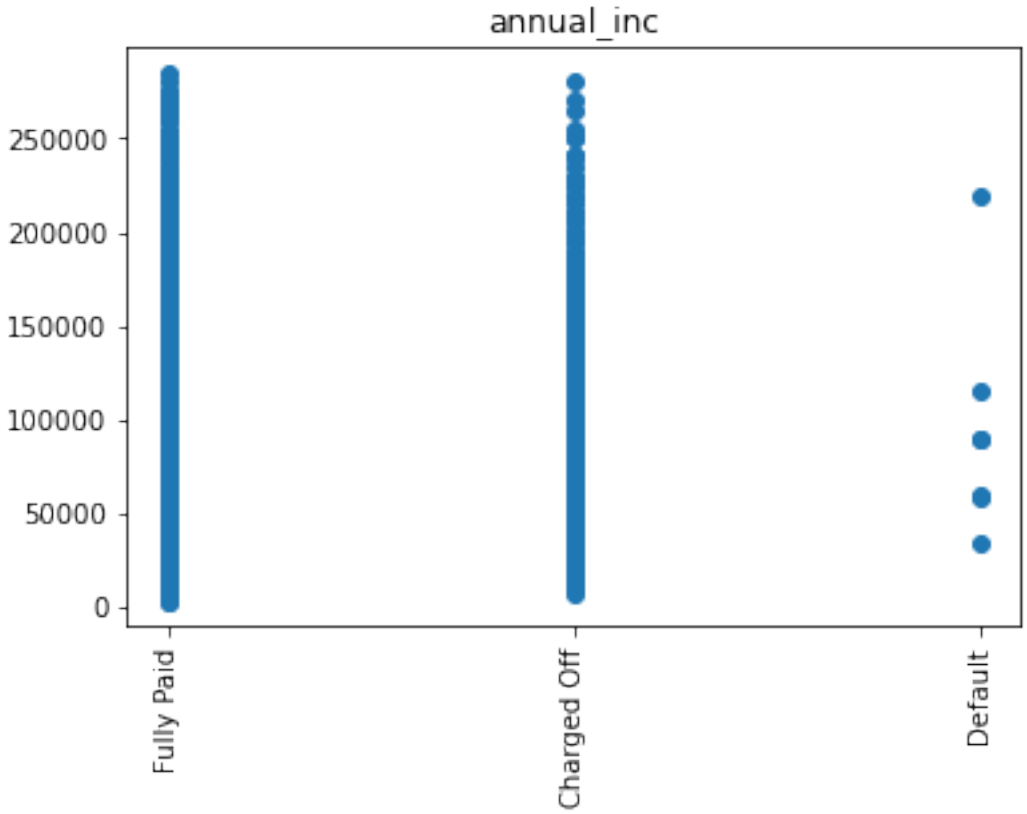

term
36 months
60 months
Fully Paid
Charged Off
Default

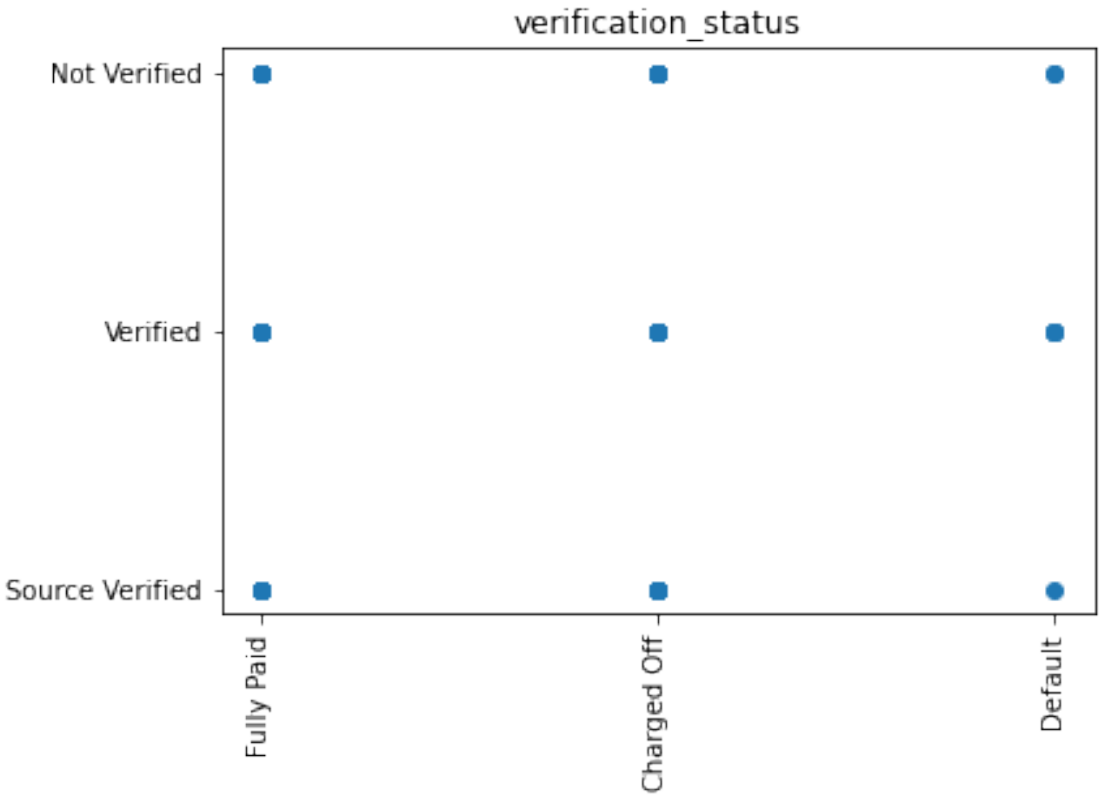

int_rate
30
25
20
15
10
5
Fully Paid
Charged Off
Default

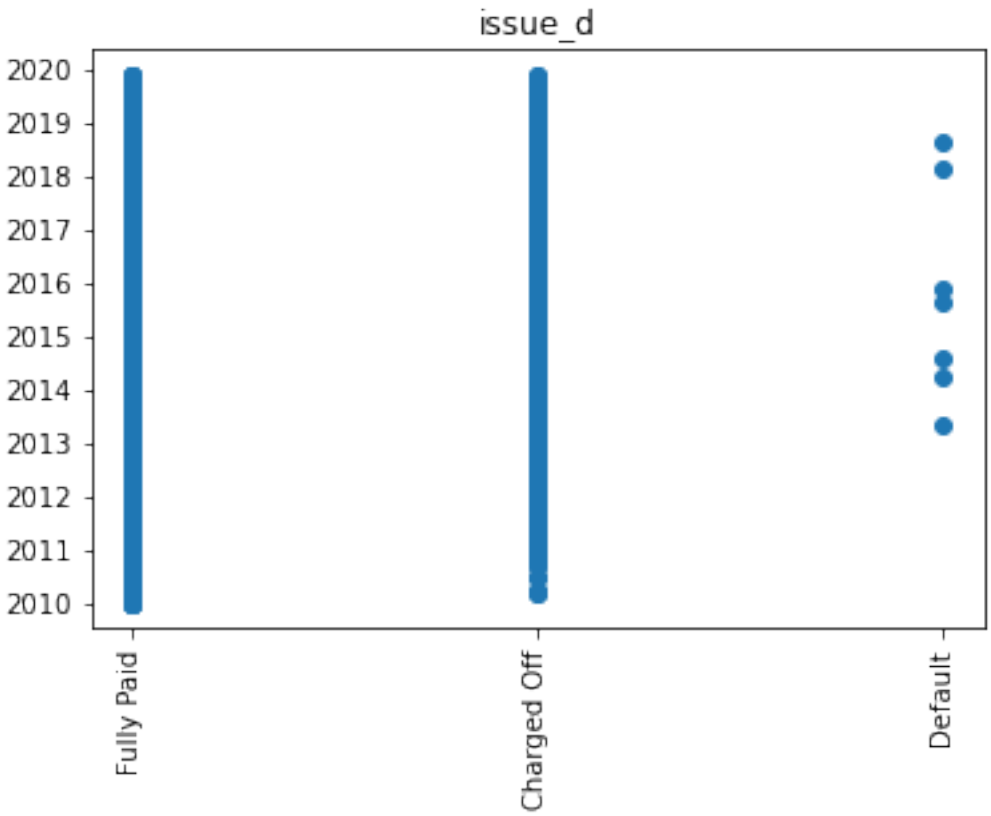

grade
G
E
A
B
F
D
C
Fully Paid
Charged Off
Default

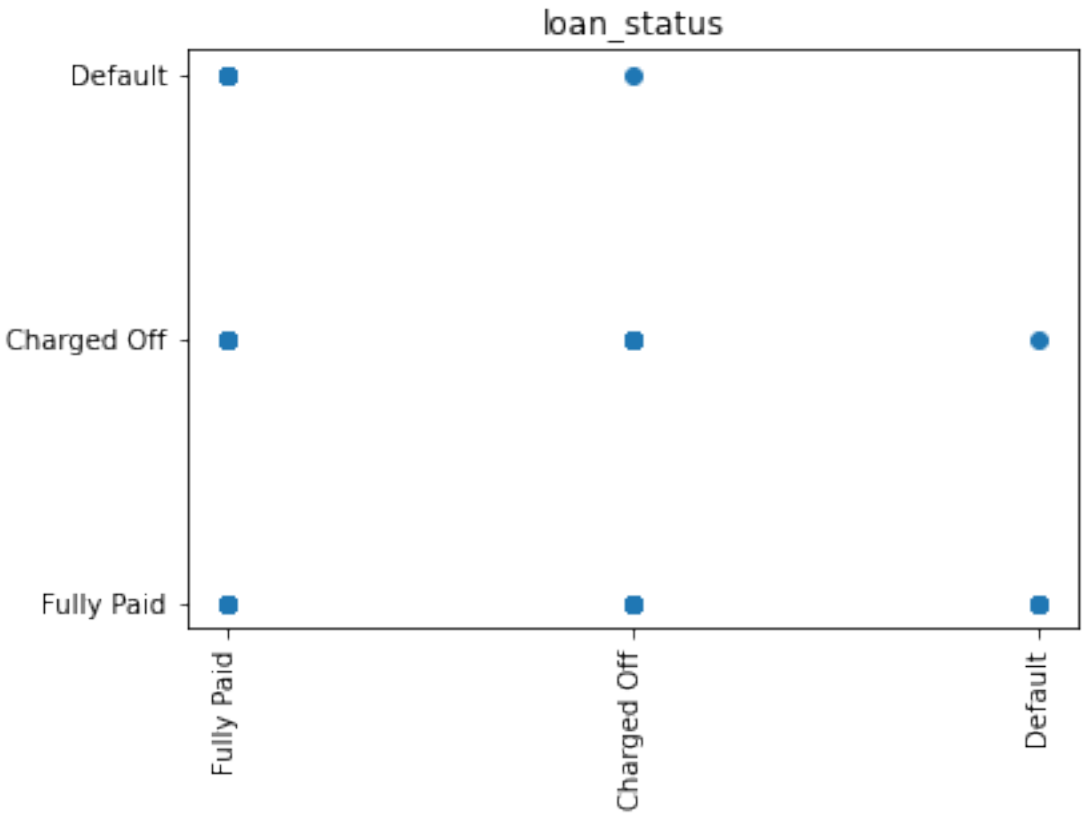

emp_length
4 years
5 years
6 years
9 years
10+ years
8 years
2 years
None
< 1 year
7 years
1 year
3 years
Fully Paid
Charged Off
Default

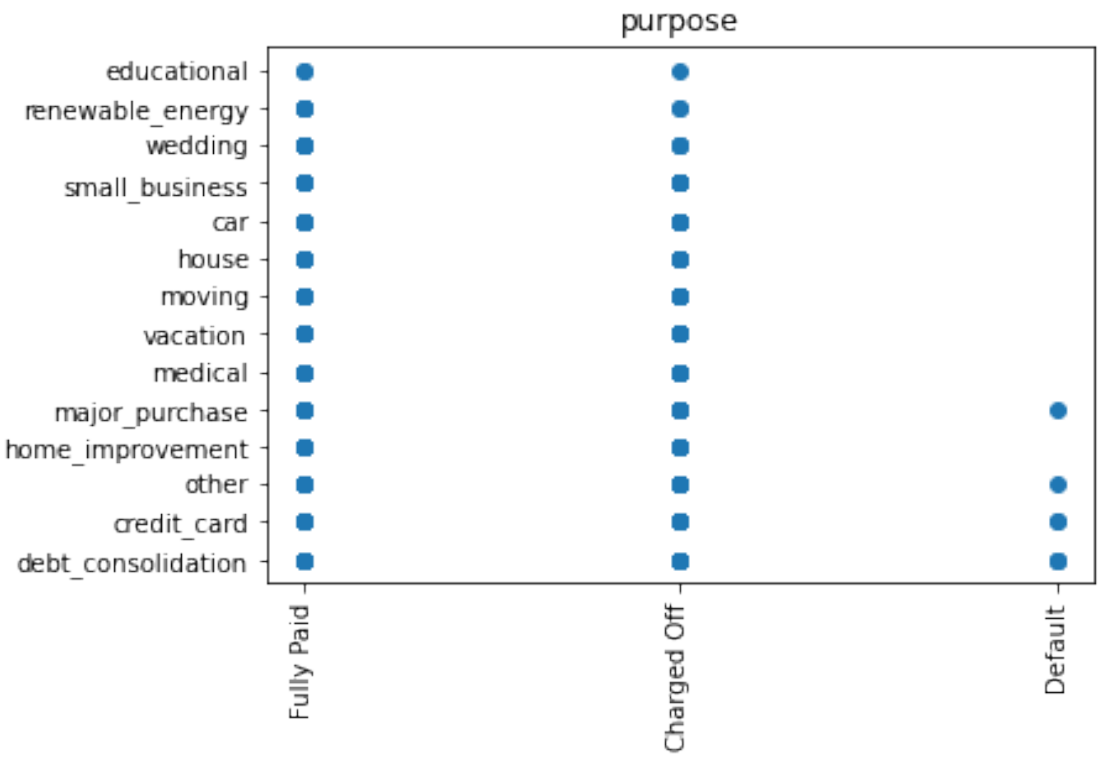

home_ownership
ANY
OWN
MORTGAGE
RENT
Fully Paid
Charged Off
Default

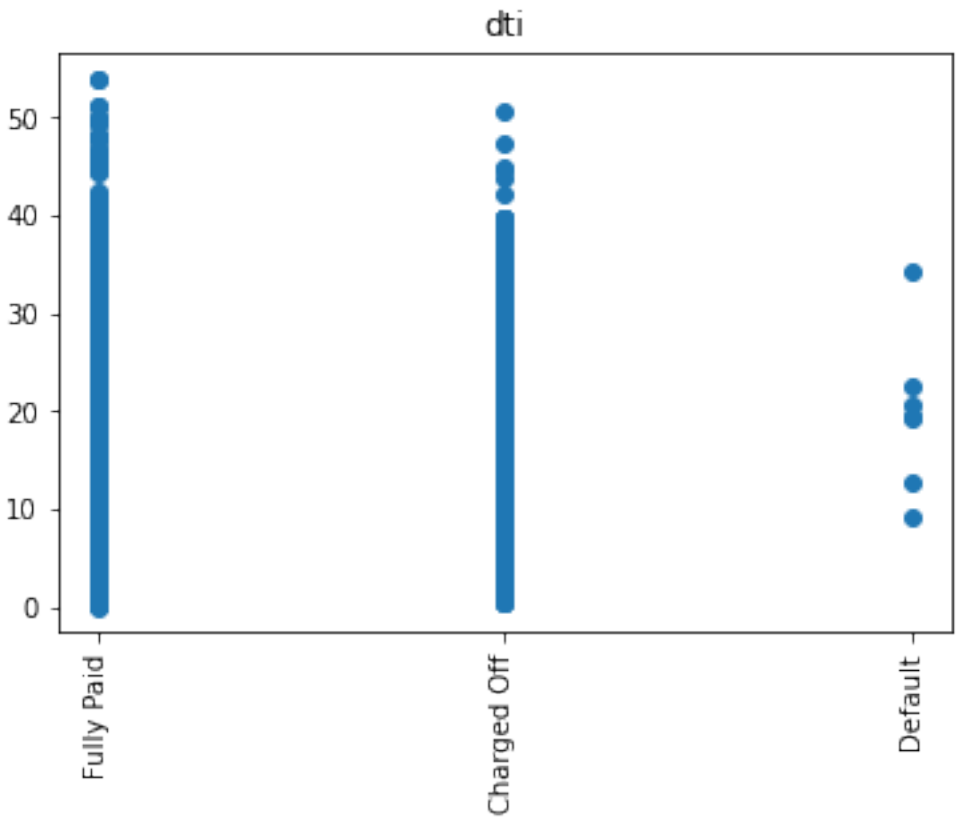

annual_inc
250000
200000
150000
100000
50000
0
Fully Paid
Charged Off
Default

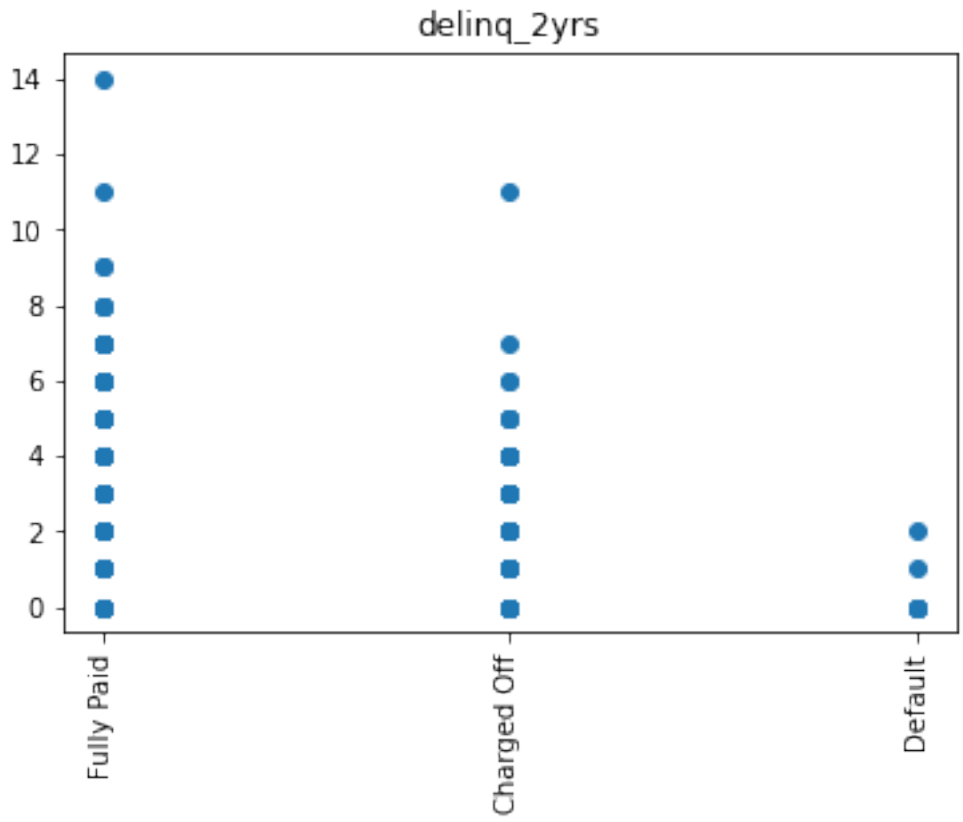

verification_status
Not Verified
Verified
Source Verified
Fully Paid
Charged Off
Default

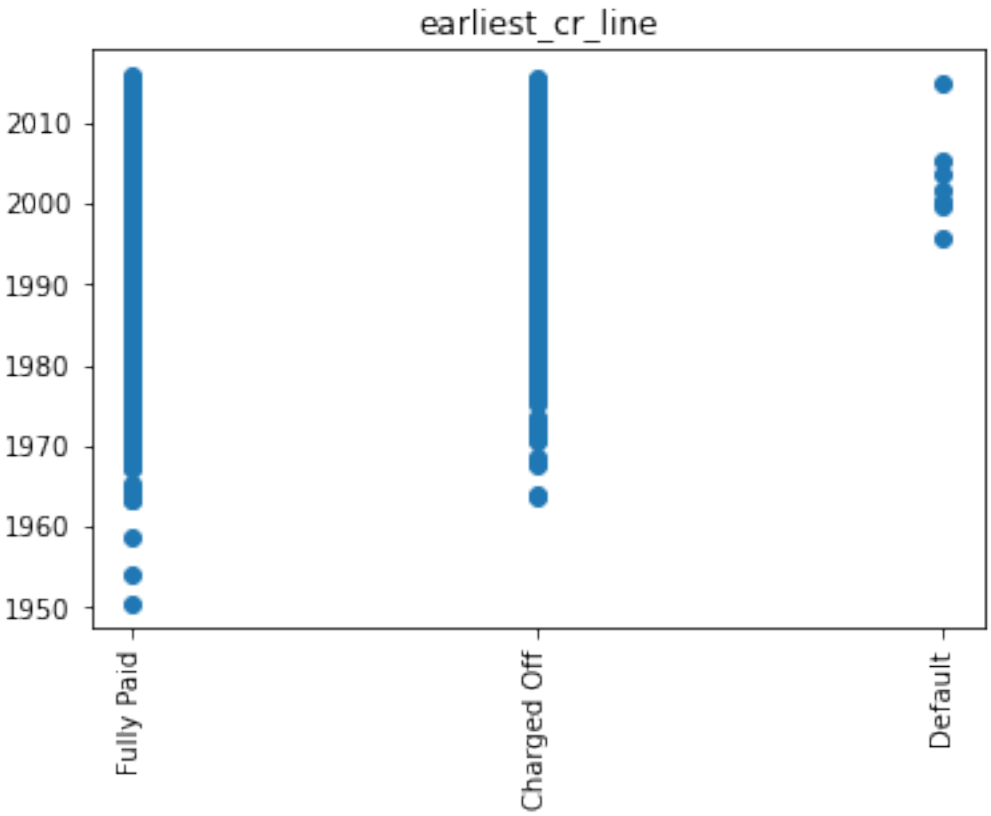

issue_d
2020
2019
2018
2017
2016
2015
2014
2013
2012
2011
2010
Fully Paid
Charged Off
Default

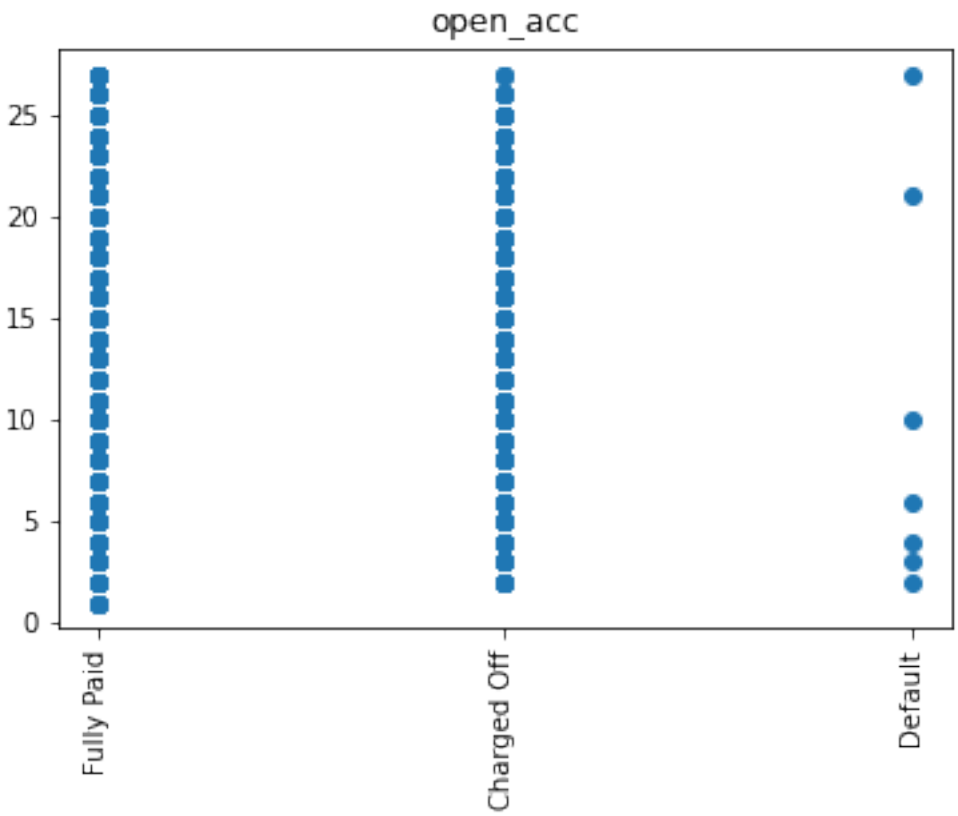

loan_status
Default
Charged Off
Fully Paid
Fully Paid
Charged Off
Default

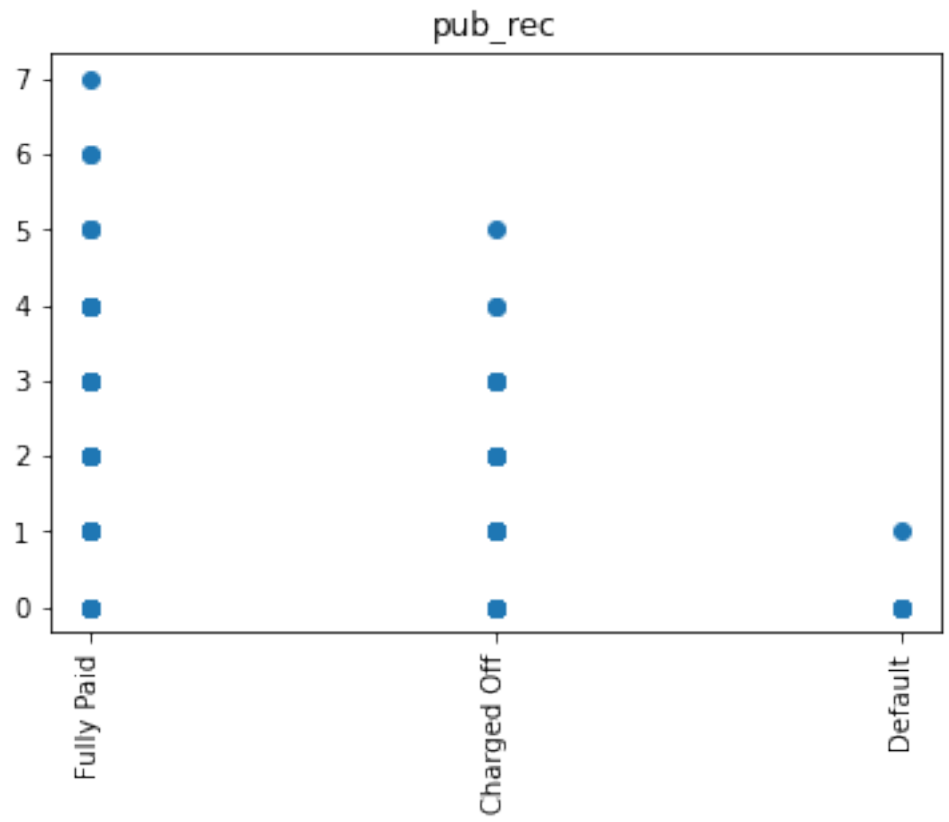

purpose
educational
renewable_energy
wedding
small_business
car
house
moving
vacation
medical
major_purchase
home_improvement
other
credit_card
debt_consolidation
Fully Paid
Charged Off
Default

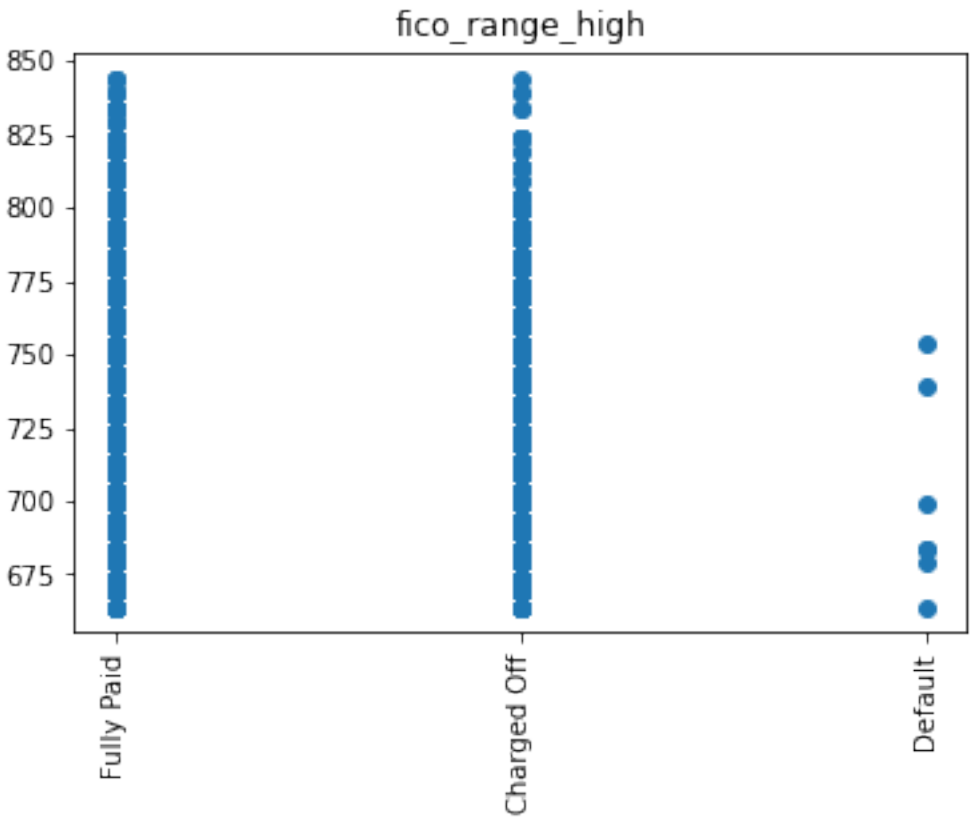

dti
50
40
30
20
10
0
Fully Paid
Charged Off
Default

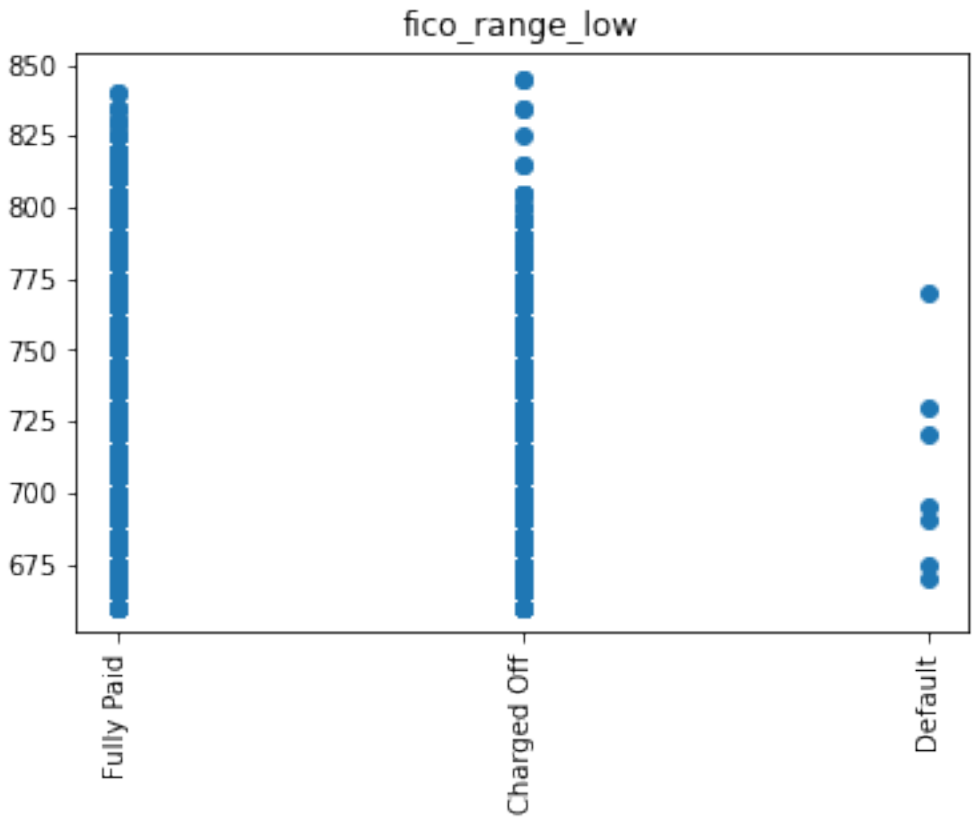

delinq_2yrs
14
12
10
8
6
4
2
0
Fully Paid
Charged Off
Default

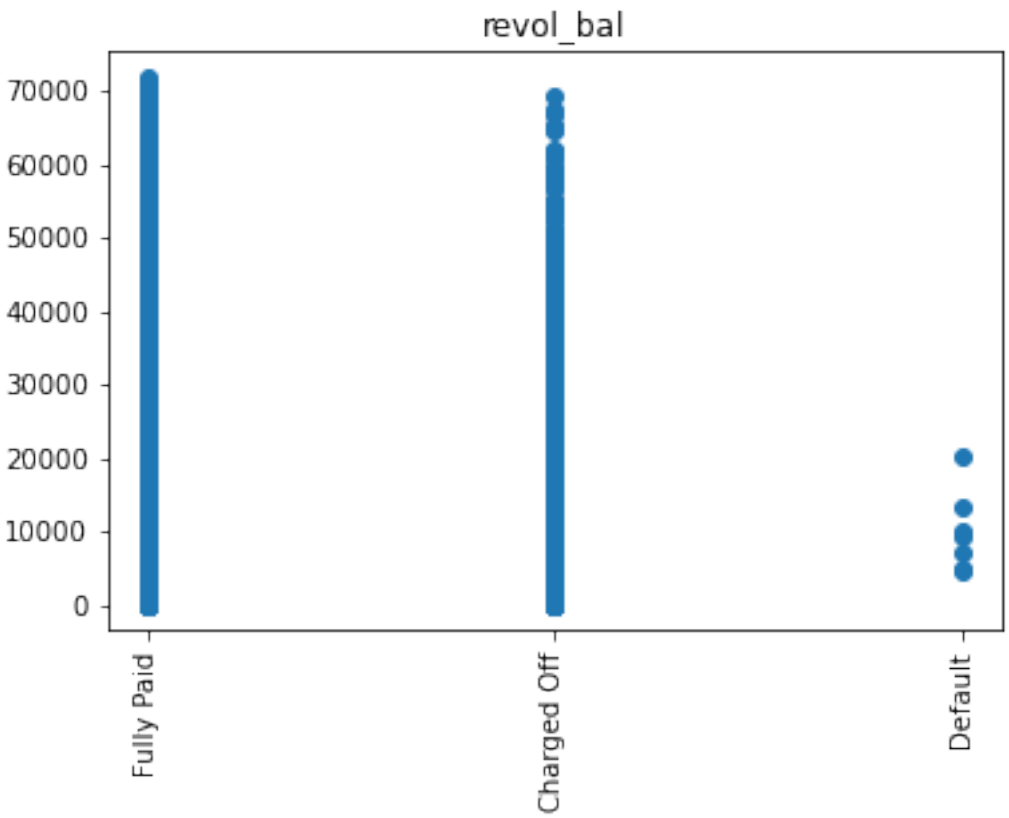

earliest_cr_line
2010
2000
1990
1980
1970
1960
1950
Fully Paid
Charged Off
Default

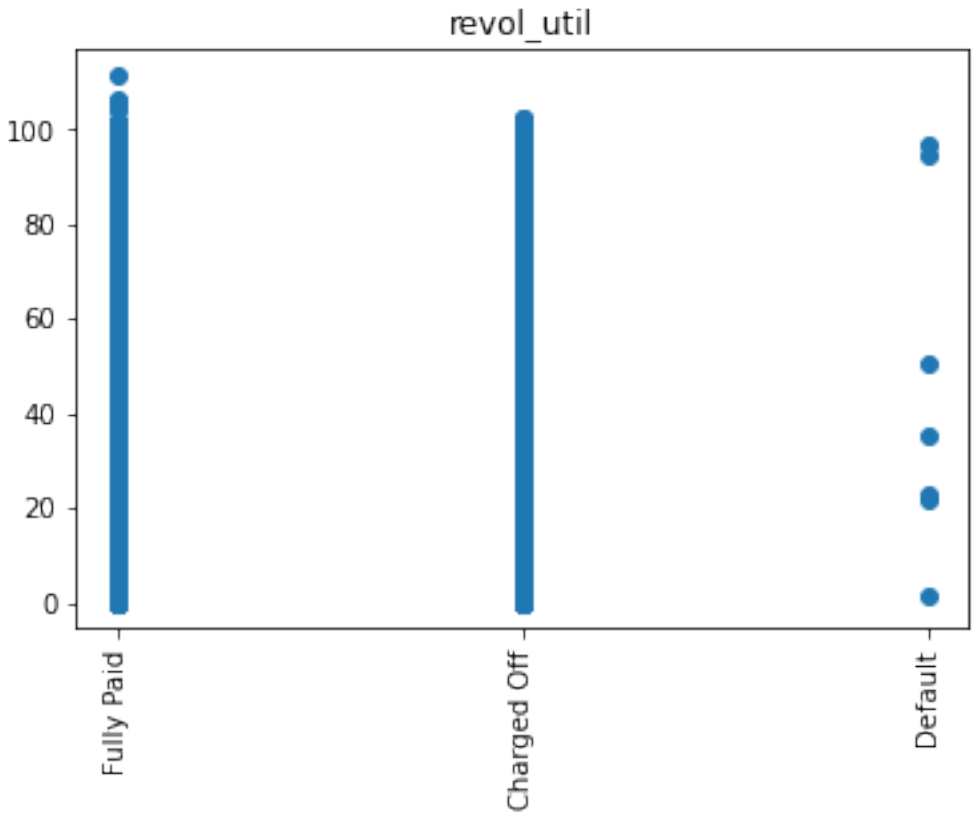

open_acc
25
20
15
10
5
0
Fully Paid
Charged Off
Default

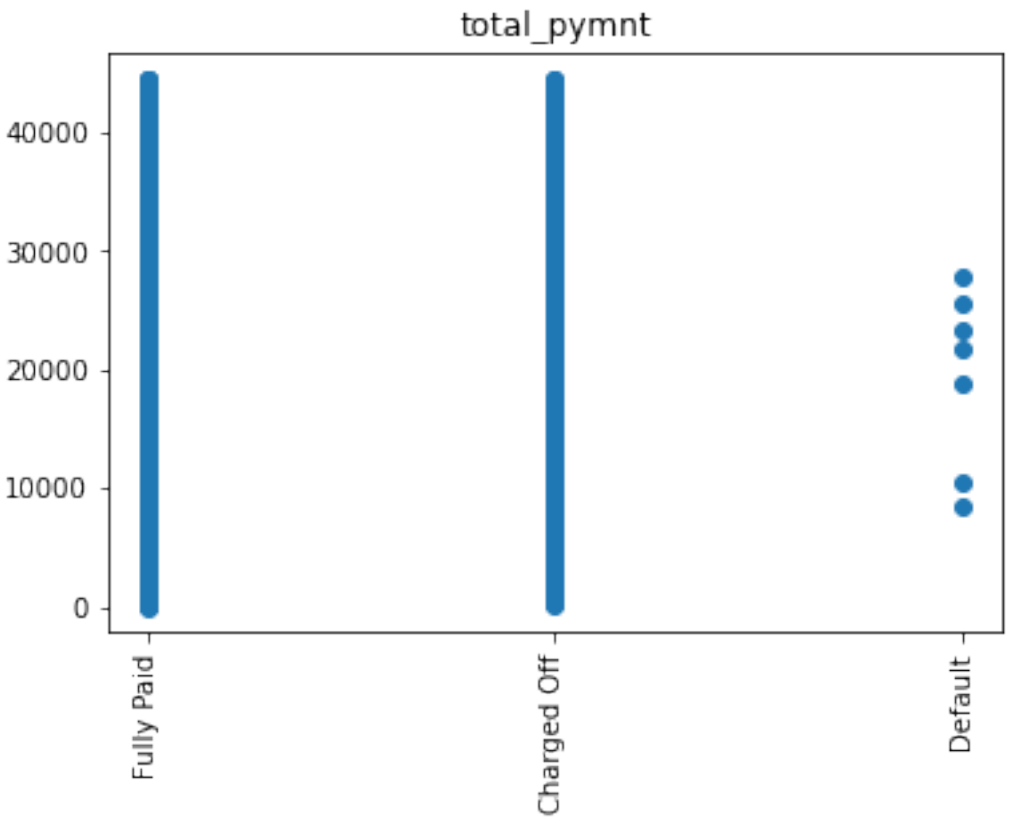

pub_rec
7
6
5
4
3
2
1
0
Fully Paid
Charged Off
Default

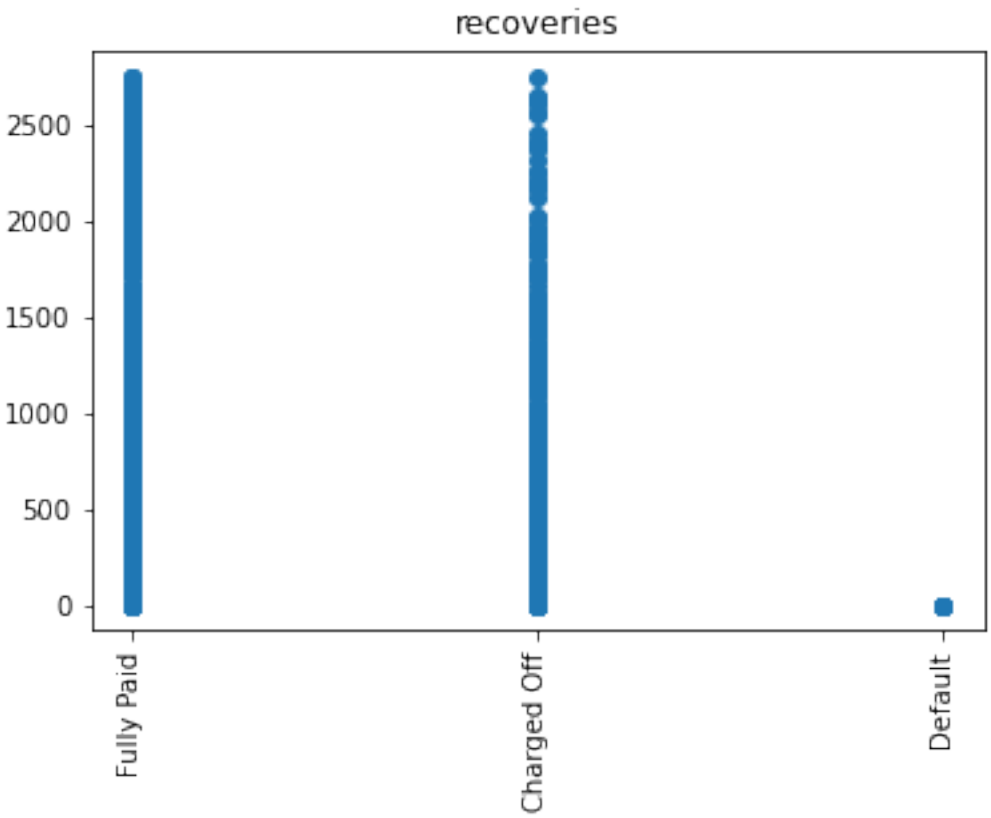

fico_range_high
850
825
800
775
750
725
700
675
Fully Paid
Charged Off
Default

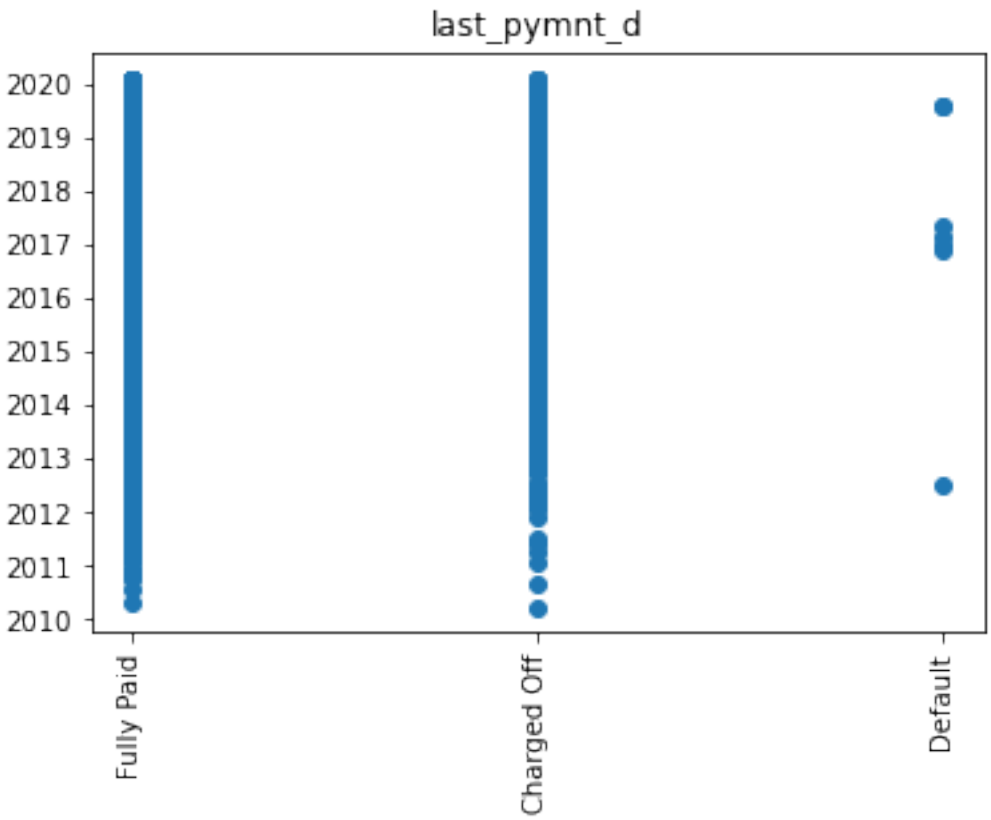

fico_range_low
850
825
800
775
750
725
700
675
Fully Paid
Charged Off
Default

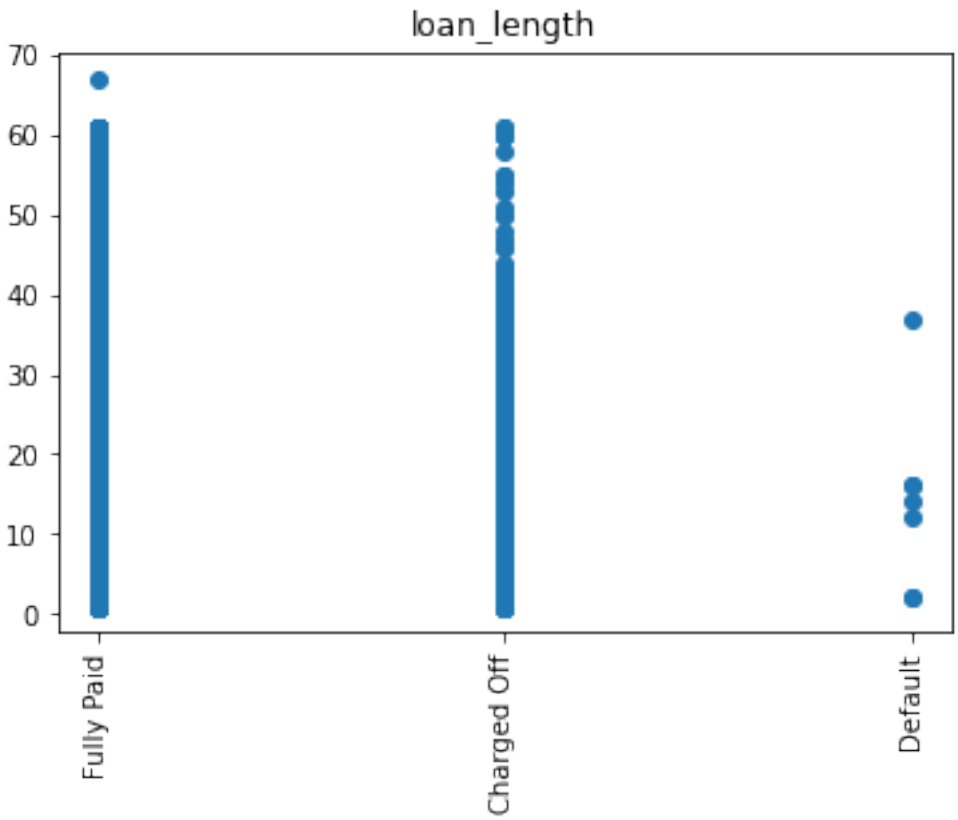

revol_bal
70000
60000
50000
40000
30000
20000
10000
0
Fully Paid
Charged Off
Default

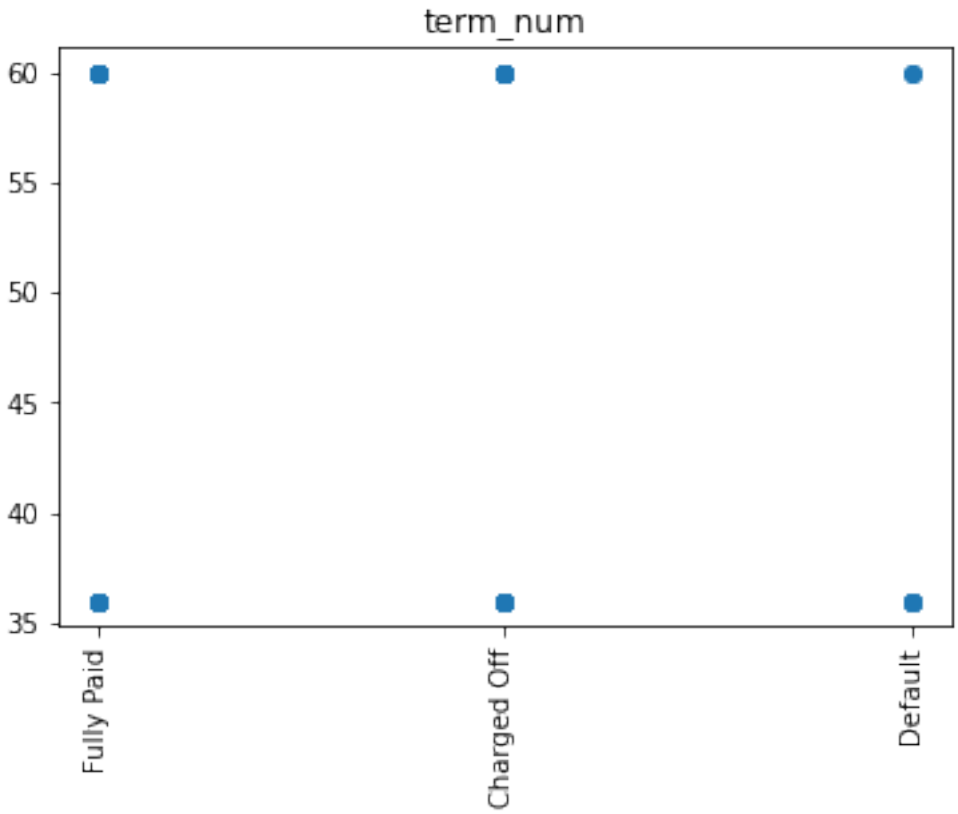

revol_util
100
80
60
40
20
0
Fully Paid
Charged Off
Default

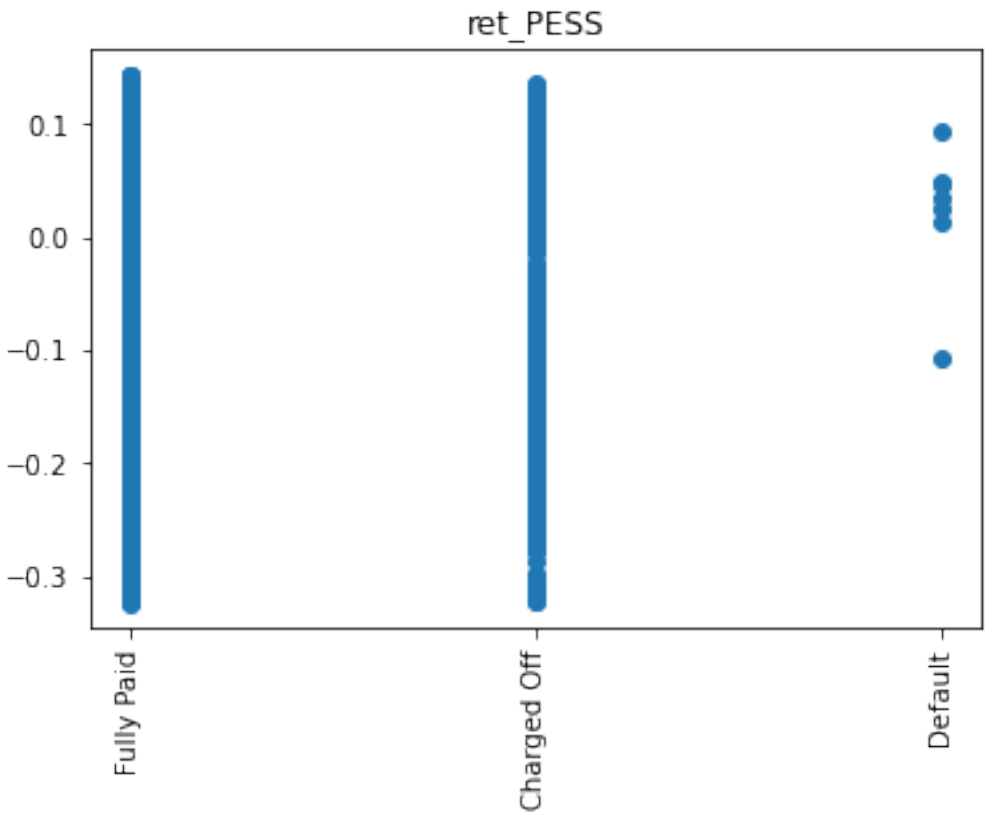

total_pymnt
40000
30000
20000
10000
0
Fully Paid
Charged Off
Default

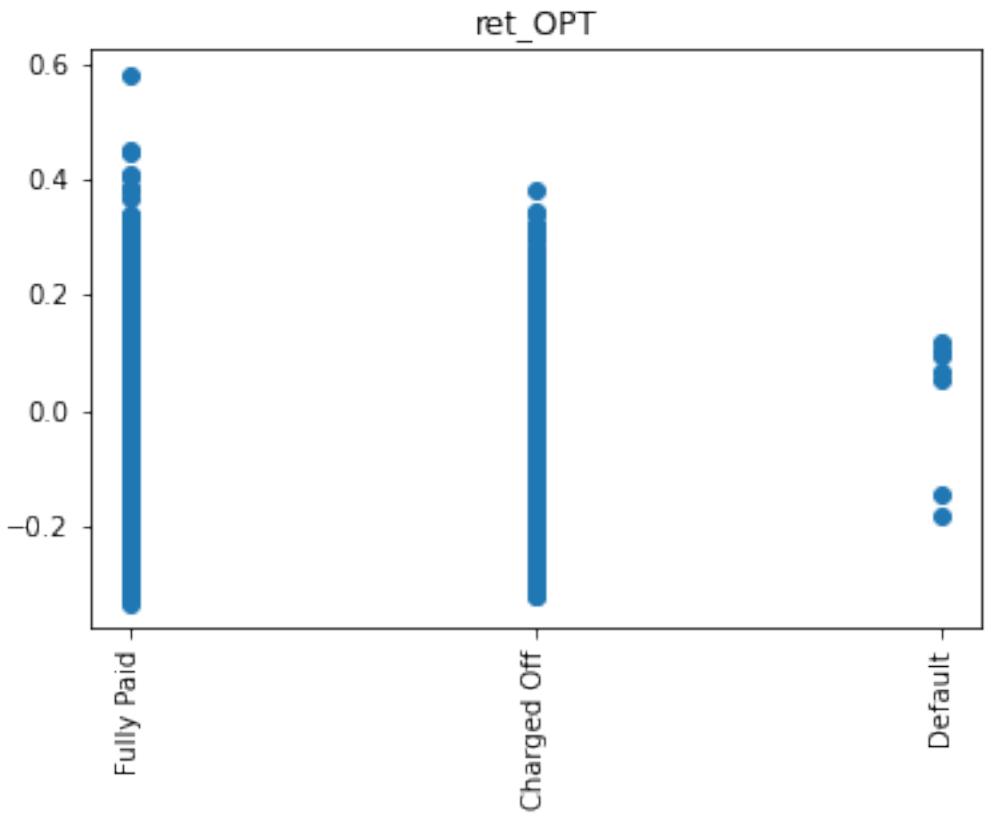

recoveries
2500
2000
1500
1000
500
0
Fully Paid
Charged Off
Default

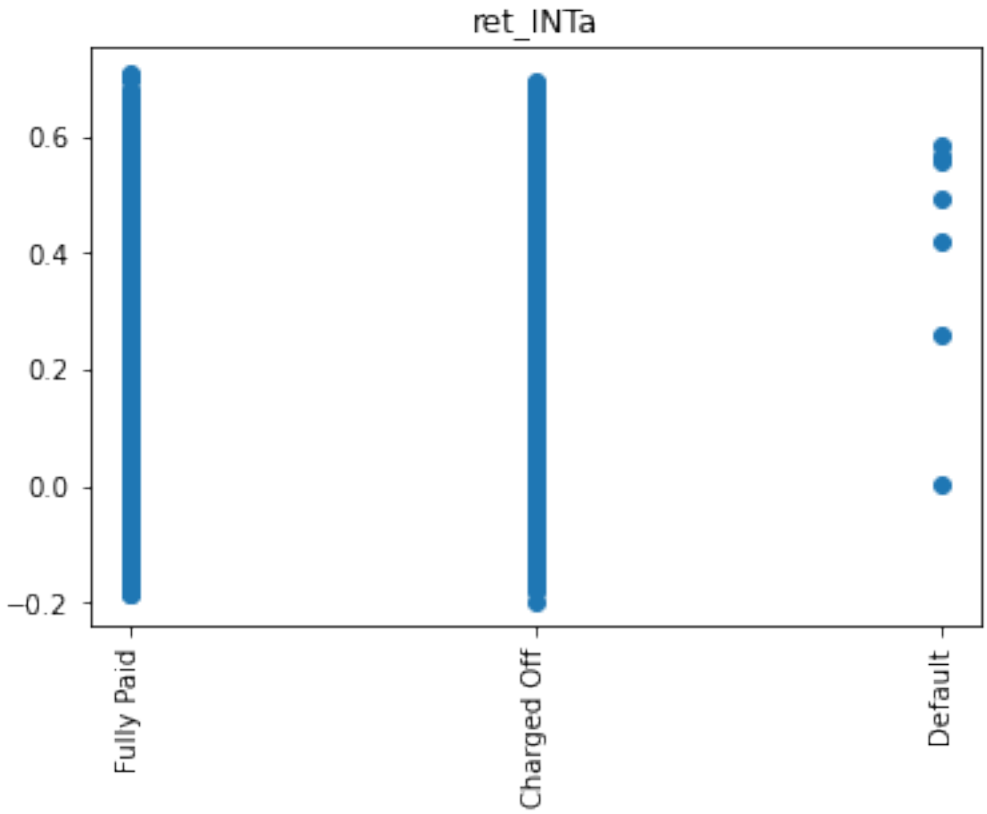

last_pymnt_d
2020
2019
2018
2017
2016
2015
2014
2013
2012
2011
2010
Fully Paid
Charged Off
Default

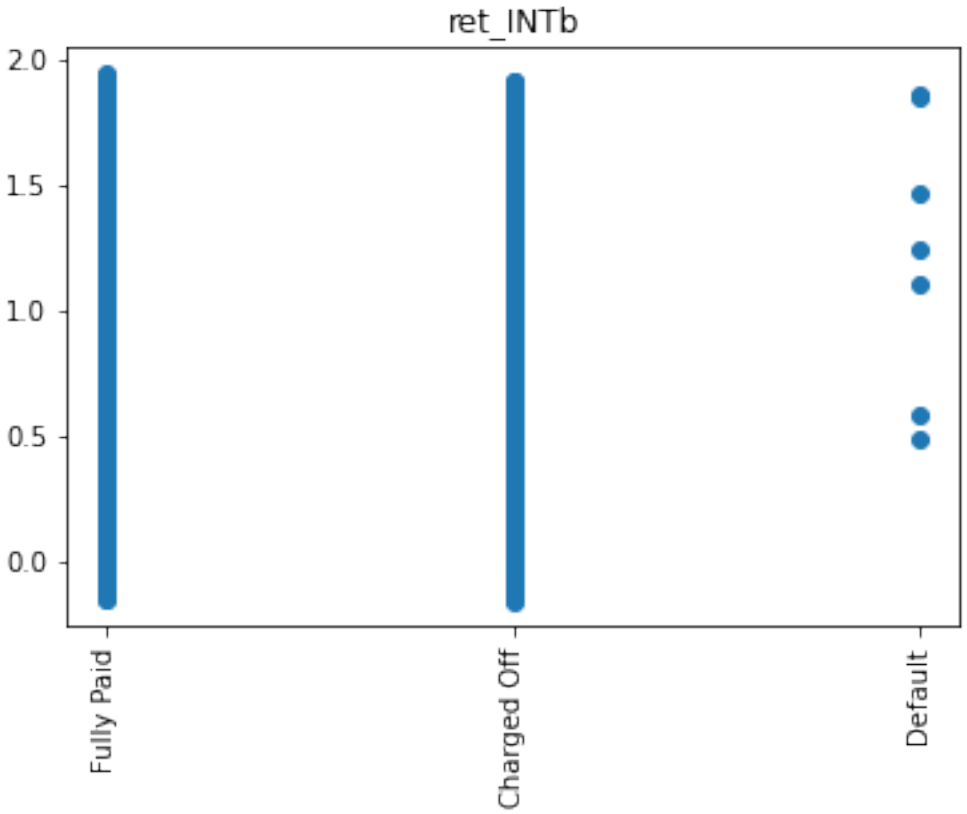

loan_length
70
60
50
40
30
20
10
0
Fully Paid
Charged Off
Default


term_num
60
55
50
45
40
35
Fully Paid
Charged Off
Default


ret_PESS
0.1
0.0
−0.1
−0.2
−0.3
Fully Paid
Charged Off
Default


ret_OPT
0.6
0.4
0.2
0.0
−0.2
Fully Paid
Charged Off
Default


ret_INTa
0.6
0.4
0.2
0.0
−0.2
Fully Paid
Charged Off
Default



1.4.10 Data Exploration

Solution to Q.7 from the handout

[39]:
```
# Find the percentage of loans by grade, the default by grade,
# and the return of each grade
perc_by_grade = (final_data.grade.value_counts()*100/len(final_data)).
 ↪sort_index()

default_by_grade = final_data.groupby("grade").apply(lambda x : (x.loan_status !
 ↪= "Fully Paid").sum()*100/len(x) )
ret_by_grade_OPT = final_data.groupby("grade")['ret_OPT'].mean() * 100 #␣
 ↪average return for M2-Optimistic for each loan grade
ret_by_grade_PESS = final_data.groupby("grade")['ret_PESS'].mean() * 100 #␣
 ↪average return for M1-Pessimistic for each loan grade
ret_by_grade_INTa = final_data.groupby("grade")['ret_INTa'].mean() * 100 #␣
 ↪average return for M3
ret_by_grade_INTb = final_data.groupby("grade")['ret_INTb'].mean() * 100 #␣
 ↪average return for M3
```

```
int_rate_by_grade = final_data.groupby("grade")['int_rate'].mean() # average␣
↪interest rate for each grade

combined = pd.DataFrame(perc_by_grade)
combined.columns = ['perc_of_loans']
combined['perc_default'] = default_by_grade
combined['avg_int_rate'] = int_rate_by_grade
combined['return_OPT'] = ret_by_grade_OPT
combined['return_PESS'] = ret_by_grade_PESS
combined['return_INTa'] = ret_by_grade_INTa
combined['return_INTb'] = ret_by_grade_INTb

combined
```

[39]:

	perc_of_loans	perc_default	avg_int_rate	return_OPT	return_PESS	\
A	18.539689	6.476934	7.219852	3.986987	1.201130	
B	29.205654	13.461626	10.885972	5.121760	1.152820	
C	27.763913	21.839563	14.205114	5.657940	0.264614	
D	15.337104	29.096783	17.988371	6.367915	-0.354900	
E	6.531939	36.587276	21.227476	6.800662	-1.001098	
F	2.095674	41.742393	24.767634	7.624870	-1.253729	
G	0.526027	45.154899	27.401644	7.994172	-2.586403	

	return_INTa	return_INTb
A	45.577675	134.107600
B	45.014553	132.149286
C	43.763157	130.057631
D	42.404218	127.204594
E	40.568384	122.744257
F	39.460259	120.123653
G	37.707052	118.320612

Question 7i

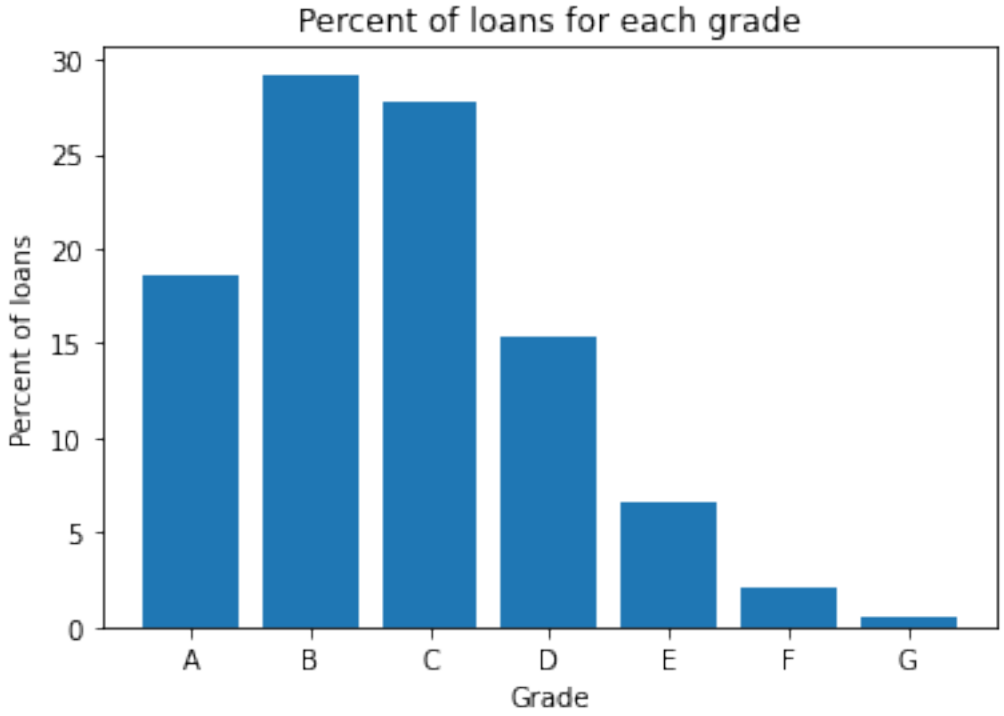
[40]:
```
pd.DataFrame(combined['perc_of_loans'])
```

[40]:

	perc_of_loans
A	18.539689
B	29.205654
C	27.763913
D	15.337104
E	6.531939
F	2.095674
G	0.526027

[41]:
```
# Bar chart of percent of loans by grade
grades = sorted(final_data['grade'].unique())
```

```
plt.bar(range(len(grades)), combined['perc_of_loans'])
plt.xticks(range(len(grades)), grades)
plt.title('Percent of loans for each grade')
plt.xlabel('Grade')
plt.ylabel('Percent of loans')
plt.show()
```



Question 7ii

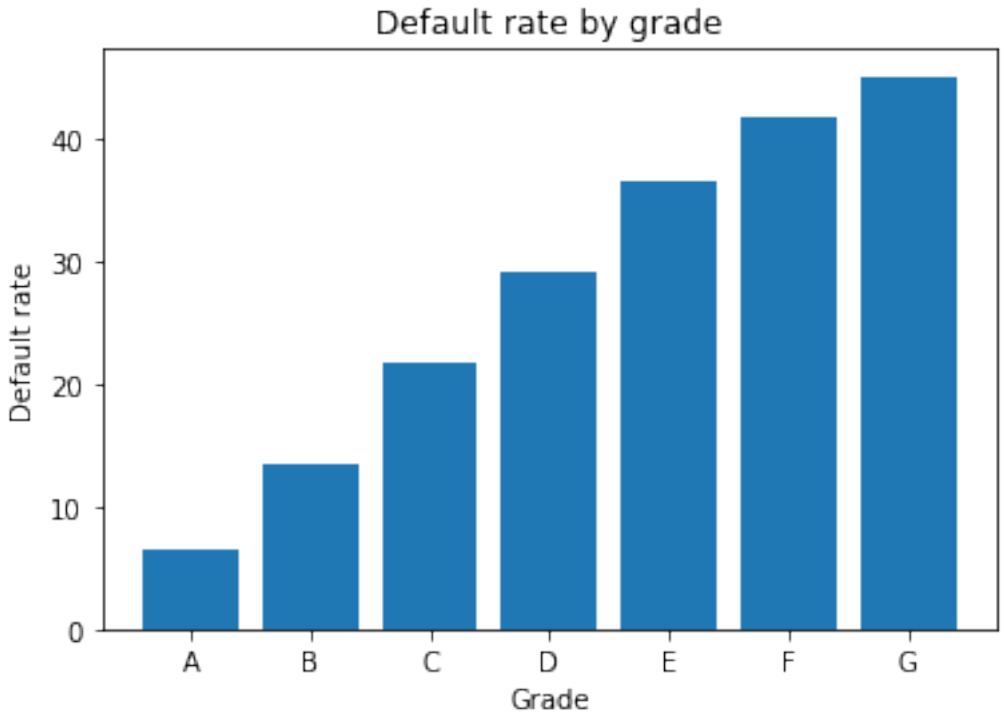
[42]:
```
pd.DataFrame(combined['perc_default'])
```

[42]:

	perc_default
A	6.476934
B	13.461626
C	21.839563
D	29.096783
E	36.587276
F	41.742393
G	45.154899

[43]:
```
# Bar chart of percent of default rate by grade
plt.bar(range(len(grades)), combined['perc_default'])
plt.xticks(range(len(grades)), grades)
plt.title('Default rate by grade')
```

```
plt.xlabel('Grade')
plt.ylabel('Default rate')
plt.show()
```



Question 7iii

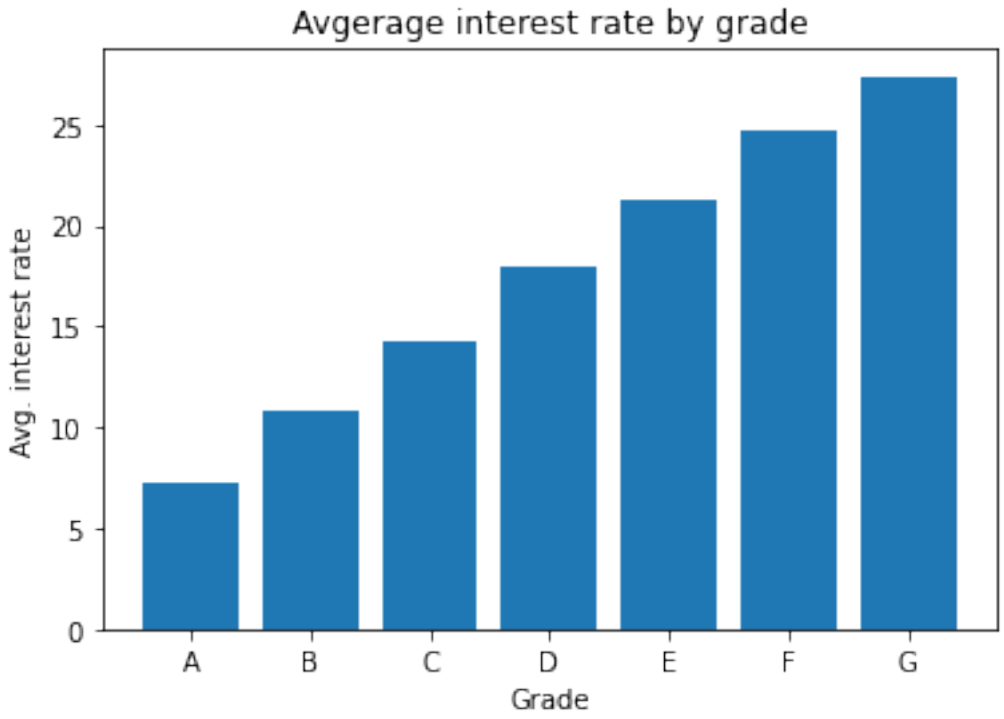
[44]:
```
pd.DataFrame(combined['avg_int_rate'])
```

[44]:

	avg_int_rate
A	7.219852
B	10.885972
C	14.205114
D	17.988371
E	21.227476
F	24.767634
G	27.401644

[45]:
```
# Bar chart of percent of loans by grade
grades = sorted(final_data['grade'].unique())

plt.bar(range(len(grades)), combined['avg_int_rate'])
plt.xticks(range(len(grades)), grades)
plt.title('Avgerage interest rate by grade')
plt.xlabel('Grade')
```

```
plt.ylabel('Avg. interest rate')
plt.show()
```



Question 7vi

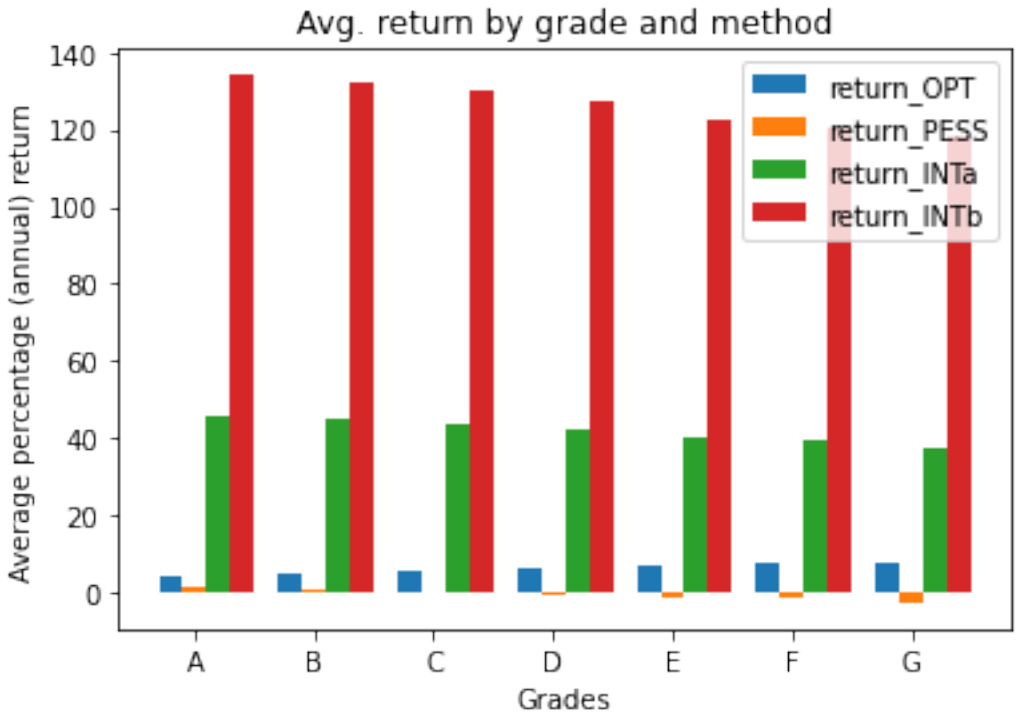
[46]:
```
pd.DataFrame(combined[['return_OPT', 'return_PESS', 'return_INTa',␣
↪'return_INTb']])
```

[46]:

	return_OPT	return_PESS	return_INTa	return_INTb
A	3.986987	1.201130	45.577675	134.107600
B	5.121760	1.152820	45.014553	132.149286
C	5.657940	0.264614	43.763157	130.057631
D	6.367915	-0.354900	42.404218	127.204594
E	6.800662	-1.001098	40.568384	122.744257
F	7.624870	-1.253729	39.460259	120.123653
G	7.994172	-2.586403	37.707052	118.320612

[47]:
```
X_axis = np.arange(len(grades))

plt.bar(X_axis - 0.2, combined['return_OPT'], 0.2, label = 'return_OPT')
plt.bar(X_axis, combined['return_PESS'], 0.2, label = 'return_PESS')
plt.bar(X_axis + 0.2, combined['return_INTa'], 0.2, label = 'return_INTa')
plt.bar(X_axis + 0.4, combined['return_INTb'], 0.2, label = 'return_INTb')
```

```
plt.xticks(range(len(grades)), grades)
plt.xlabel("Grades")
plt.ylabel("Average percentage (annual) return")
plt.title("Avg. return by grade and method")
plt.legend()
plt.show()
```



1.4.11 Further exploratory data analysis

[48]:
```
final_data.groupby(by=['purpose', 'grade'])[ret_cols].mean()
```

[48]:

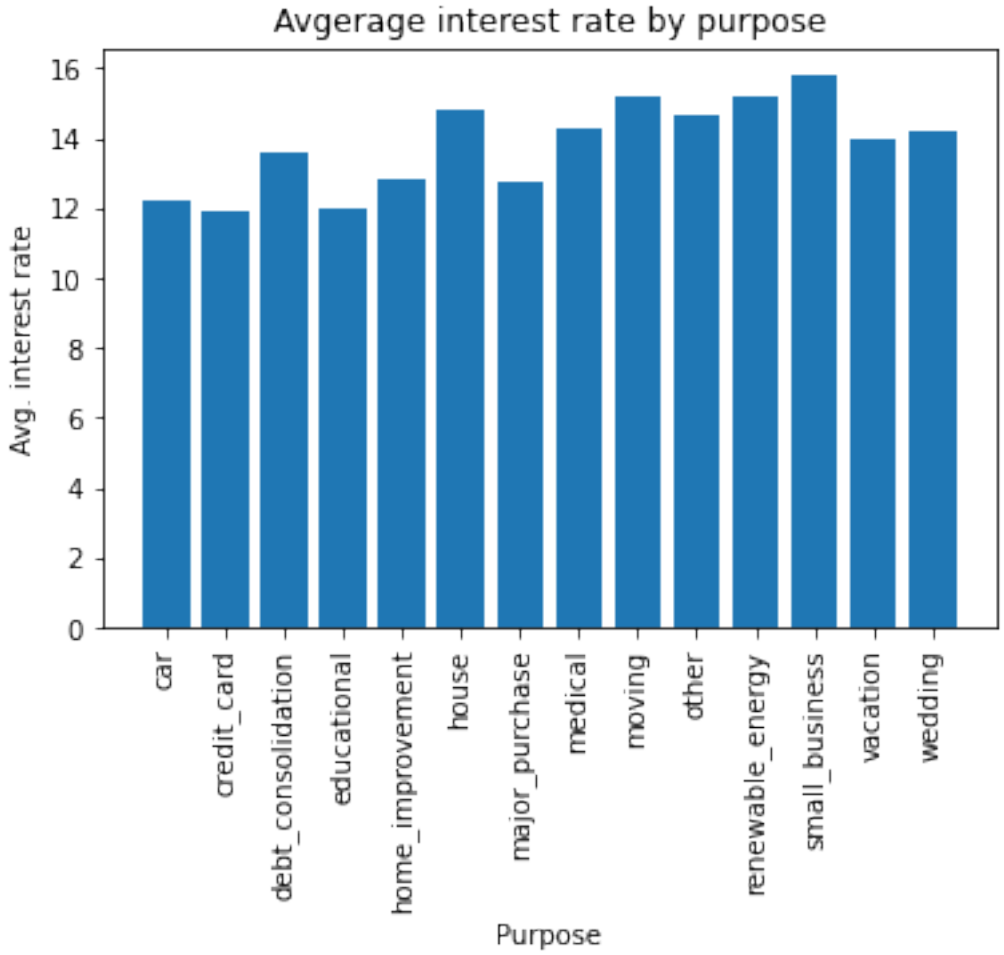
purpose	grade	ret_PESS	ret_OPT	ret_INTa	ret_INTb
car	A	0.014939	0.041262	0.459399	1.349549
	B	0.013777	0.053434	0.456891	1.340972
	C	0.008084	0.061987	0.451628	1.337866
	D	0.003008	0.069504	0.441516	1.316390
	E	-0.001766	0.076869	0.427468	1.281313
	F	-0.019647	0.060348	0.374060	1.152521
	G	-0.043962	0.028788	0.317932	1.032874
credit_card	A	0.012617	0.037953	0.448664	1.311998
	B	0.012840	0.049737	0.446446	1.303427
	C	0.002608	0.054883	0.432919	1.283236

	D	-0.004641	0.062733	0.418633	1.257422
	E	-0.013811	0.066827	0.396107	1.207576
	F	-0.012126	0.084747	0.398925	1.217282
	G	-0.016908	0.096796	0.402258	1.241581
debt_consolidation	A	0.012725	0.040933	0.456820	1.342580
	B	0.012532	0.052200	0.451217	1.322291
	C	0.003490	0.057486	0.438470	1.300640
	D	-0.004391	0.063624	0.422671	1.268876
	E	-0.011910	0.067369	0.402313	1.220275
	F	-0.017286	0.072891	0.384805	1.181867
	G	-0.029885	0.078214	0.368337	1.168243
educational	A	0.032174	0.051163	0.451567	1.270231
	B	0.038990	0.052700	0.428229	1.174746
	C	0.014456	0.033526	0.390975	1.111907
	D	0.020316	0.047485	0.394472	1.115192
	E	0.000810	0.003447	0.317922	0.835572
	F	0.035132	0.175779	0.596221	1.734782
	G	0.126461	0.126475	0.489219	1.095023
home_improvement	A	0.010586	0.042109	0.467454	1.390020
	B	0.009072	0.054005	0.456762	1.354250
	C	0.002874	0.061692	0.445733	1.330612
	D	-0.000023	0.073250	0.439055	1.315461
	E	-0.003222	0.082667	0.428176	1.287853
	F	-0.003005	0.100029	0.429953	1.293739
	G	-0.009434	0.104570	0.419182	1.280376
house	A	0.004021	0.040943	0.477568	1.448959
	B	0.003489	0.056575	0.466094	1.412787
	C	-0.005145	0.058750	0.445855	1.360727
	D	-0.012617	0.060656	0.423063	1.295770
	E	-0.006192	0.078230	0.433717	1.316682
	F	-0.002207	0.091202	0.436120	1.316266
	G	-0.019039	0.086526	0.408594	1.274008
major_purchase	A	0.011996	0.040227	0.461160	1.363560
	B	0.006825	0.048105	0.448186	1.330020
	C	0.001236	0.054837	0.437948	1.310248
	D	-0.005179	0.060730	0.422023	1.272955
	E	-0.018658	0.058387	0.390261	1.198218
	F	-0.017986	0.067234	0.387980	1.188664
	G	-0.018865	0.086868	0.391844	1.213197
medical	A	0.004178	0.037112	0.464926	1.399445
	B	-0.000972	0.043082	0.443475	1.334181
	C	-0.006349	0.048709	0.430339	1.300490
	D	-0.007953	0.055443	0.418057	1.261873
	E	-0.001739	0.071014	0.423356	1.266491
	F	0.003120	0.080341	0.418385	1.237854
	G	-0.017110	0.092246	0.415486	1.275436
moving	A	0.003748	0.038094	0.467041	1.412024

```
                    B     -0.005153  0.039419  0.439150  1.329296
                    C     -0.006909  0.048378  0.430994  1.305601
                    D     -0.005479  0.051791  0.418822  1.254949
                    E     -0.004144  0.060231  0.415601  1.243477
                    F     -0.001241  0.075824  0.415545  1.244402
                    G     -0.022238  0.064584  0.367475  1.133025
other               A      0.006341  0.040236  0.469946  1.413020
                    B      0.003995  0.049810  0.454248  1.360327
                    C     -0.001495  0.053117  0.438985  1.319501
                    D      0.001311  0.065319  0.434395  1.297310
                    E     -0.000953  0.070165  0.421132  1.258094
                    F     -0.000595  0.078603  0.411943  1.224429
                    G     -0.025345  0.073426  0.375548  1.168865
renewable_energy    A      0.006180  0.034720  0.449033  1.336707
                    B     -0.010082  0.029998  0.417193  1.259104
                    C      0.005554  0.063148  0.456006  1.362594
                    D     -0.002351  0.056091  0.423020  1.260345
                    E      0.011026  0.090157  0.448365  1.326945
                    F     -0.001971  0.043692  0.376424  1.122296
                    G     -0.023956  0.106786  0.393344  1.244930
small_business      A     -0.001722  0.021388  0.420207  1.248182
                    B     -0.002019  0.031607  0.412795  1.224163
                    C     -0.008008  0.034592  0.403891  1.208471
                    D     -0.005899  0.042281  0.400495  1.191945
                    E     -0.011200  0.043149  0.385438  1.154687
                    F     -0.005816  0.061322  0.396278  1.179844
                    G     -0.019988  0.068276  0.379969  1.162607
vacation            A      0.004040  0.040217  0.472042  1.427988
                    B      0.003740  0.051887  0.459632  1.381701
                    C     -0.000616  0.056265  0.447934  1.349784
                    D      0.004816  0.071640  0.447471  1.334773
                    E      0.006269  0.071849  0.429841  1.266862
                    F      0.021626  0.095401  0.458593  1.329753
                    G     -0.041043  0.014787  0.295252  0.937583
wedding             A      0.023552  0.035432  0.414249  1.157378
                    B      0.031347  0.053303  0.433274  1.205286
                    C      0.042193  0.070792  0.454196  1.253123
                    D      0.058297  0.090792  0.477618  1.296262
                    E      0.052266  0.100227  0.483442  1.321052
                    F      0.068016  0.136323  0.516296  1.363426
                    G      0.042053  0.134949  0.506826  1.395544
```

[49]:
```
purposes = sorted(final_data['purpose'].unique())
avg_int_rates = final_data.groupby(by=['purpose'])['int_rate'].mean()

plt.bar(range(len(purposes)), avg_int_rates)
plt.xticks(range(len(purposes)), purposes, rotation=90)
```

```
plt.title('Avgerage interest rate by purpose')
plt.xlabel('Purpose')
plt.ylabel('Avg. interest rate')
plt.show()
```



1.4.12 Save a Pickle

[50]:
```
# Remove the "total_pymnt" and "recoveries" from the list of continuous features
continuous_features = [x for x in continuous_features if x not in␣
↪['total_pymnt', 'recoveries']]
```

Why did we remove `total_pymnt` and `recoveries` from the data for the task of predicting whether to give loan or not, although these are highly predictive features?

We removed these features for prediction because these features are indicative of the values we are trying to predict. These features give information that we would not have when we are considering to invest in a new loan. Including these features in our dataset would be data leakage and could lead to a falsely high training error.

[51]:
```
# save the prepared data for modeling in next Phase.
pickle.dump( [final_data, discrete_features, continuous_features, ret_cols],␣
 ↪open(pickle_file, "wb") )
```