In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
import os
os.getcwd()
```

Out[2]: '/Users/abhinavkumar/Desktop/python/risk model'

In [6]:
```
loan_data_backup = pd.read_csv('loan_data_2007_2014.csv')
```

```
/Applications/anaconda3/lib/python3.8/site-packages/IPython/core/interactivesh
ell.py:3146: DtypeWarning: Columns (20) have mixed types.Specify dtype option
on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

In [7]:
```
loan_data_backup
```

Out[7]:

	Unnamed: 0	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term
0	0	1077501	1296599	5000	5000	4975.0	36 months
1	1	1077430	1314167	2500	2500	2500.0	60 months
2	2	1077175	1313524	2400	2400	2400.0	36 months
3	3	1076863	1277178	10000	10000	10000.0	36 months
4	4	1075358	1311748	3000	3000	3000.0	60 months
...	...	...	...	...	...	...	...
466280	466280	8598660	1440975	18400	18400	18400.0	60 months
466281	466281	9684700	11536848	22000	22000	22000.0	60 months
466282	466282	9584776	11436914	20700	20700	20700.0	60 months
466283	466283	9604874	11457002	2000	2000	2000.0	36 months
466284	466284	9199665	11061576	10000	10000	9975.0	36 months

466285 rows × 75 columns

In [8]:
```
loan_data = loan_data_backup.copy()
```

In [9]:
```
#information about data
loan_data
```

Out[9]:

	Unnamed: 0	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term
0	0	1077501	1296599	5000	5000	4975.0	36 months

	Unnamed: 0	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term
1	1	1077430	1314167	2500	2500	2500.0	60 months
2	2	1077175	1313524	2400	2400	2400.0	36 months
3	3	1076863	1277178	10000	10000	10000.0	36 months
4	4	1075358	1311748	3000	3000	3000.0	60 months
...	...	...	...	...	...	...	...
466280	466280	8598660	1440975	18400	18400	18400.0	60 months
466281	466281	9684700	11536848	22000	22000	22000.0	60 months
466282	466282	9584776	11436914	20700	20700	20700.0	60 months
466283	466283	9604874	11457002	2000	2000	2000.0	36 months
466284	466284	9199665	11061576	10000	10000	9975.0	36 months

466285 rows × 75 columns

In [10]:
```
pd.options.display.max_columns = None
```

In [11]:
```
loan_data.head()
```

Out[11]:

	Unnamed: 0	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_r
0	0	1077501	1296599	5000	5000	4975.0	36 months	10
1	1	1077430	1314167	2500	2500	2500.0	60 months	15
2	2	1077175	1313524	2400	2400	2400.0	36 months	15
3	3	1076863	1277178	10000	10000	10000.0	36 months	13

	Unnamed: 0	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_r
4	4	1075358	1311748	3000	3000	3000.0	60 months	12

In [12]:
```
loan_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 466285 entries, 0 to 466284
Data columns (total 75 columns):
 #   Column                       Non-Null Count   Dtype  
---  ------                       --------------   -----  
 0   Unnamed: 0                   466285 non-null  int64  
 1   id                           466285 non-null  int64  
 2   member_id                    466285 non-null  int64  
 3   loan_amnt                    466285 non-null  int64  
 4   funded_amnt                  466285 non-null  int64  
 5   funded_amnt_inv              466285 non-null  float64
 6   term                         466285 non-null  object 
 7   int_rate                     466285 non-null  float64
 8   installment                  466285 non-null  float64
 9   grade                        466285 non-null  object 
 10  sub_grade                    466285 non-null  object 
 11  emp_title                    438697 non-null  object 
 12  emp_length                   445277 non-null  object 
 13  home_ownership               466285 non-null  object 
 14  annual_inc                   466281 non-null  float64
 15  verification_status          466285 non-null  object 
 16  issue_d                      466285 non-null  object 
 17  loan_status                  466285 non-null  object 
 18  pymnt_plan                   466285 non-null  object 
 19  url                          466285 non-null  object 
 20  desc                         125983 non-null  object 
 21  purpose                      466285 non-null  object 
 22  title                        466265 non-null  object 
 23  zip_code                     466285 non-null  object 
 24  addr_state                   466285 non-null  object 
 25  dti                          466285 non-null  float64
 26  delinq_2yrs                  466256 non-null  float64
 27  earliest_cr_line             466256 non-null  object 
 28  inq_last_6mths               466256 non-null  float64
 29  mths_since_last_delinq       215934 non-null  float64
 30  mths_since_last_record       62638 non-null   float64
 31  open_acc                     466256 non-null  float64
 32  pub_rec                      466256 non-null  float64
 33  revol_bal                    466285 non-null  int64  
 34  revol_util                   465945 non-null  float64
 35  total_acc                    466256 non-null  float64
 36  initial_list_status          466285 non-null  object 
 37  out_prncp                    466285 non-null  float64
 38  out_prncp_inv                466285 non-null  float64
 39  total_pymnt                  466285 non-null  float64
 40  total_pymnt_inv              466285 non-null  float64
 41  total_rec_prncp              466285 non-null  float64
 42  total_rec_int                466285 non-null  float64
 43  total_rec_late_fee           466285 non-null  float64
 44  recoveries                   466285 non-null  float64
 45  collection_recovery_fee      466285 non-null  float64
 46  last_pymnt_d                 465909 non-null  object 
 47  last_pymnt_amnt              466285 non-null  float64
 48  next_pymnt_d                 239071 non-null  object 
 49  last_credit_pull_d           466243 non-null  object 
```

```
 50  collections_12_mths_ex_med   466140 non-null  float64
 51  mths_since_last_major_derog  98974 non-null   float64
 52  policy_code                  466285 non-null  int64
 53  application_type             466285 non-null  object
 54  annual_inc_joint             0 non-null       float64
 55  dti_joint                    0 non-null       float64
 56  verification_status_joint    0 non-null       float64
 57  acc_now_delinq               466256 non-null  float64
 58  tot_coll_amt                 396009 non-null  float64
 59  tot_cur_bal                  396009 non-null  float64
 60  open_acc_6m                  0 non-null       float64
 61  open_il_6m                   0 non-null       float64
 62  open_il_12m                  0 non-null       float64
 63  open_il_24m                  0 non-null       float64
 64  mths_since_rcnt_il           0 non-null       float64
 65  total_bal_il                 0 non-null       float64
 66  il_util                      0 non-null       float64
 67  open_rv_12m                  0 non-null       float64
 68  open_rv_24m                  0 non-null       float64
 69  max_bal_bc                   0 non-null       float64
 70  all_util                     0 non-null       float64
 71  total_rev_hi_lim             396009 non-null  float64
 72  inq_fi                       0 non-null       float64
 73  total_cu_tl                  0 non-null       float64
 74  inq_last_12m                 0 non-null       float64
dtypes: float64(46), int64(7), object(22)
memory usage: 266.8+ MB
```

In [13]:
```python
#preprocessing of data
#To see the values that emp_length takes
loan_data['emp_length'].unique()
```

Out[13]:
```
array(['10+ years', '< 1 year', '1 year', '3 years', '8 years', '9 years',
       '4 years', '5 years', '6 years', '2 years', '7 years', nan],
      dtype=object)
```

In [14]:
```python
loan_data['emp_length_int'] = loan_data['emp_length'].str.replace('\+ years',
loan_data['emp_length_int'] = loan_data['emp_length_int'].str.replace('< 1 ye
loan_data['emp_length_int'] = loan_data['emp_length_int'].str.replace('n/a',
loan_data['emp_length_int'] = loan_data['emp_length_int'].str.replace(' years
loan_data['emp_length_int'] = loan_data['emp_length_int'].str.replace(' year'
```

In [15]:
```python
#string type

type(loan_data['emp_length_int'][0])
```

Out[15]:
```
str
```

In [16]:
```python
#Converts a series into numeric type
loan_data['emp_length_int'] = pd.to_numeric(loan_data['emp_length_int'])
```

In [17]:
```python
type(loan_data['emp_length_int'][0])
```

Out[17]:
```
numpy.float64
```

In [18]:
```python
loan_data['term'].unique()
```

Out[18]:
```
array([' 36 months', ' 60 months'], dtype=object)
```

In [19]:
```python
loan_data['term_int'] = loan_data['term'].str.replace(' months', '')
loan_data['term_int'] = loan_data['term_int'].str.replace(' ', '')
```

In [20]:
```python
type(loan_data['term_int'][0])
```

Out[20]: str

In [21]:
```
loan_data['term_int'] = pd.to_numeric(loan_data['term_int'])
```

In [22]:
```
type(loan_data['term_int'][0])
```

Out[22]: numpy.int64

In [23]:
```
loan_data['earliest_cr_line']
```

Out[23]:
```
0         Jan-85
1         Apr-99
2         Nov-01
3         Feb-96
4         Jan-96
           ...
466280    Apr-03
466281    Jun-97
466282    Dec-01
466283    Feb-03
466284    Feb-00
Name: earliest_cr_line, Length: 466285, dtype: object
```

In [24]:
```
loan_data['earliest_cr_line_date'] = pd.to_datetime(loan_data['earliest_cr_li
#%b indicates first 3 letters of the month, %y indicates last 2 digits of the
```

In [25]:
```
type(loan_data['earliest_cr_line_date'][0])
```

Out[25]: pandas._libs.tslibs.timestamps.Timestamp

In [26]:
```
pd.to_datetime('2017-12-01') - loan_data['earliest_cr_line_date']
```

Out[26]:
```
0        12022 days
1         6819 days
2         5874 days
3         7974 days
4         8005 days
            ...
466280    5358 days
466281    7488 days
466282    5844 days
466283    5417 days
466284    6513 days
Name: earliest_cr_line_date, Length: 466285, dtype: timedelta64[ns]
```

In [27]:
```
loan_data['mths_since_earliest_cr_line'] = round(pd.to_numeric((pd.to_datetim
```

In [28]:
```
#Descriptive statistics
loan_data['mths_since_earliest_cr_line'].describe()
```

Out[28]:
```
count    466256.000000
mean        239.482430
std          93.974829
min        -612.000000
25%         183.000000
50%         225.000000
75%         285.000000
max         587.000000
Name: mths_since_earliest_cr_line, dtype: float64
```

In [29]:
```
loan_data.loc[: , ['earliest_cr_line', 'earliest_cr_line_date', 'mths_since_e
```

Out[29]:

	earliest_cr_line	earliest_cr_line_date	mths_since_earliest_cr_line

	earliest_cr_line	earliest_cr_line_date	mths_since_earliest_cr_line
1580	Sep-62	2062-09-01	-537.0
1770	Sep-68	2068-09-01	-609.0
2799	Sep-64	2064-09-01	-561.0
3282	Sep-67	2067-09-01	-597.0
3359	Feb-65	2065-02-01	-566.0
...	...	...	...
464003	Jan-68	2068-01-01	-601.0
464260	Jul-66	2066-07-01	-583.0
465100	Oct-67	2067-10-01	-598.0
465500	Sep-67	2067-09-01	-597.0
465655	Jan-56	2056-01-01	-457.0

1169 rows × 3 columns

In [30]:
```
loan_data['mths_since_earliest_cr_line'][loan_data['mths_since_earliest_cr_li
```

```
<ipython-input-30-60169add49d1>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  loan_data['mths_since_earliest_cr_line'][loan_data['mths_since_earliest_cr_l
ine'] < 0] = loan_data['mths_since_earliest_cr_line'].max()
```

In [31]:
```
#cleaning data variable
loan_data['issue_d']
```

Out[31]:
```
0         Dec-11
1         Dec-11
2         Dec-11
3         Dec-11
4         Dec-11
           ...
466280    Jan-14
466281    Jan-14
466282    Jan-14
466283    Jan-14
466284    Jan-14
Name: issue_d, Length: 466285, dtype: object
```

In [32]:
```
loan_data['issue_d_date'] = pd.to_datetime(loan_data['issue_d'], format = '%b-
```

In [33]:
```
type(loan_data['issue_d_date'][0])
```

Out[33]:
```
pandas._libs.tslibs.timestamps.Timestamp
```

In [34]:
```
pd.to_datetime('2017-12-01') - loan_data['issue_d_date']
```

Out[34]:
```
0        2192 days
1        2192 days
2        2192 days
3        2192 days
4        2192 days
           ...
466280   1430 days
```

```
466281   1430 days
466282   1430 days
466283   1430 days
466284   1430 days
Name: issue_d_date, Length: 466285, dtype: timedelta64[ns]
```

In [35]:
```
loan_data['mths_since_issue_d_date'] = round(pd.to_numeric((pd.to_datetime('2
```

In [36]:
```
loan_data['mths_since_issue_d_date'].describe()
```

Out[36]:
```
count    466285.000000
mean         51.255187
std          14.340154
min          36.000000
25%          41.000000
50%          47.000000
75%          57.000000
max         126.000000
Name: mths_since_issue_d_date, dtype: float64
```

In [37]:
```
# working on discrete variable
pd.get_dummies(loan_data['grade'])
```

Out[37]:

	A	B	C	D	E	F	G
0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0
2	0	0	1	0	0	0	0
3	0	0	1	0	0	0	0
4	0	1	0	0	0	0	0
...	...	...	...	...	...	...	...
466280	0	0	1	0	0	0	0
466281	0	0	0	1	0	0	0
466282	0	0	0	1	0	0	0
466283	1	0	0	0	0	0	0
466284	0	0	0	1	0	0	0

466285 rows × 7 columns

In [38]:
```
pd.get_dummies(loan_data['grade'], prefix = 'Grade', prefix_sep = ' : ')
```

Out[38]:

	Grade : A	Grade : B	Grade : C	Grade : D	Grade : E	Grade : F	Grade : G
0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0
2	0	0	1	0	0	0	0
3	0	0	1	0	0	0	0
4	0	1	0	0	0	0	0
...	...	...	...	...	...	...	...
466280	0	0	1	0	0	0	0
466281	0	0	0	1	0	0	0
466282	0	0	0	1	0	0	0

	Grade : A	Grade : B	Grade : C	Grade : D	Grade : E	Grade : F	Grade : G
466283	1	0	0	0	0	0	0
466284	0	0	0	1	0	0	0

466285 rows × 7 columns

In [39]:
```
loan_data_dummies = [pd.get_dummies(loan_data['grade'], prefix = 'Grade', pre
                     pd.get_dummies(loan_data['sub_grade'], prefix = 'sub_gra
                     pd.get_dummies(loan_data['home_ownership'], prefix = 'ho
                     pd.get_dummies(loan_data['verification_status'], prefix 
                     pd.get_dummies(loan_data['loan_status'], prefix = 'loan_
                     pd.get_dummies(loan_data['purpose'], prefix = 'purpose',
                     pd.get_dummies(loan_data['addr_state'], prefix = 'addr_s
                     pd.get_dummies(loan_data['initial_list_status'], prefix 
                    ]
```

In [40]:
```
#Converting list into a dataframe
loan_data_dummies = pd.concat(loan_data_dummies, axis = 1)
```

In [41]:
```
type(loan_data_dummies)
```

Out[41]:
```
pandas.core.frame.DataFrame
```

In [42]:
```
#Appending this dataframe to the original one
loan_data = pd.concat([loan_data, loan_data_dummies], axis = 1)
```

In [43]:
```
loan_data.columns.values
```

Out[43]:
```
array(['Unnamed: 0', 'id', 'member_id', 'loan_amnt', 'funded_amnt',
       'funded_amnt_inv', 'term', 'int_rate', 'installment', 'grade',
       'sub_grade', 'emp_title', 'emp_length', 'home_ownership',
       'annual_inc', 'verification_status', 'issue_d', 'loan_status',
       'pymnt_plan', 'url', 'desc', 'purpose', 'title', 'zip_code',
       'addr_state', 'dti', 'delinq_2yrs', 'earliest_cr_line',
       'inq_last_6mths', 'mths_since_last_delinq',
       'mths_since_last_record', 'open_acc', 'pub_rec', 'revol_bal',
       'revol_util', 'total_acc', 'initial_list_status', 'out_prncp',
       'out_prncp_inv', 'total_pymnt', 'total_pymnt_inv',
       'total_rec_prncp', 'total_rec_int', 'total_rec_late_fee',
       'recoveries', 'collection_recovery_fee', 'last_pymnt_d',
       'last_pymnt_amnt', 'next_pymnt_d', 'last_credit_pull_d',
       'collections_12_mths_ex_med', 'mths_since_last_major_derog',
       'policy_code', 'application_type', 'annual_inc_joint', 'dti_joint',
       'verification_status_joint', 'acc_now_delinq', 'tot_coll_amt',
       'tot_cur_bal', 'open_acc_6m', 'open_il_6m', 'open_il_12m',
       'open_il_24m', 'mths_since_rcnt_il', 'total_bal_il', 'il_util',
       'open_rv_12m', 'open_rv_24m', 'max_bal_bc', 'all_util',
       'total_rev_hi_lim', 'inq_fi', 'total_cu_tl', 'inq_last_12m',
       'emp_length_int', 'term_int', 'earliest_cr_line_date',
       'mths_since_earliest_cr_line', 'issue_d_date',
       'mths_since_issue_d_date', 'Grade : A', 'Grade : B', 'Grade : C',
       'Grade : D', 'Grade : E', 'Grade : F', 'Grade : G',
       'sub_grade : A1', 'sub_grade : A2', 'sub_grade : A3',
       'sub_grade : A4', 'sub_grade : A5', 'sub_grade : B1',
       'sub_grade : B2', 'sub_grade : B3', 'sub_grade : B4',
       'sub_grade : B5', 'sub_grade : C1', 'sub_grade : C2',
       'sub_grade : C3', 'sub_grade : C4', 'sub_grade : C5',
       'sub_grade : D1', 'sub_grade : D2', 'sub_grade : D3',
       'sub_grade : D4', 'sub_grade : D5', 'sub_grade : E1',
       'sub_grade : E2', 'sub_grade : E3', 'sub_grade : E4',
       'sub_grade : E5', 'sub_grade : F1', 'sub_grade : F2',
       'sub_grade : F3', 'sub_grade : F4', 'sub_grade : F5',
```

```
       'sub_grade : G1', 'sub_grade : G2', 'sub_grade : G3',
       'sub_grade : G4', 'sub_grade : G5', 'home_ownership : ANY',
       'home_ownership : MORTGAGE', 'home_ownership : NONE',
       'home_ownership : OTHER', 'home_ownership : OWN',
       'home_ownership : RENT', 'verification_status : Not Verified',
       'verification_status : Source Verified',
       'verification_status : Verified', 'loan_status : Charged Off',
       'loan_status : Current', 'loan_status : Default',
       'loan_status : Does not meet the credit policy. Status:Charged Off',
       'loan_status : Does not meet the credit policy. Status:Fully Paid',
       'loan_status : Fully Paid', 'loan_status : In Grace Period',
       'loan_status : Late (16-30 days)',
       'loan_status : Late (31-120 days)', 'purpose : car',
       'purpose : credit_card', 'purpose : debt_consolidation',
       'purpose : educational', 'purpose : home_improvement',
       'purpose : house', 'purpose : major_purchase', 'purpose : medical',
       'purpose : moving', 'purpose : other',
       'purpose : renewable_energy', 'purpose : small_business',
       'purpose : vacation', 'purpose : wedding', 'addr_state : AK',
       'addr_state : AL', 'addr_state : AR', 'addr_state : AZ',
       'addr_state : CA', 'addr_state : CO', 'addr_state : CT',
       'addr_state : DC', 'addr_state : DE', 'addr_state : FL',
       'addr_state : GA', 'addr_state : HI', 'addr_state : IA',
       'addr_state : ID', 'addr_state : IL', 'addr_state : IN',
       'addr_state : KS', 'addr_state : KY', 'addr_state : LA',
       'addr_state : MA', 'addr_state : MD', 'addr_state : ME',
       'addr_state : MI', 'addr_state : MN', 'addr_state : MO',
       'addr_state : MS', 'addr_state : MT', 'addr_state : NC',
       'addr_state : NE', 'addr_state : NH', 'addr_state : NJ',
       'addr_state : NM', 'addr_state : NV', 'addr_state : NY',
       'addr_state : OH', 'addr_state : OK', 'addr_state : OR',
       'addr_state : PA', 'addr_state : RI', 'addr_state : SC',
       'addr_state : SD', 'addr_state : TN', 'addr_state : TX',
       'addr_state : UT', 'addr_state : VA', 'addr_state : VT',
       'addr_state : WA', 'addr_state : WI', 'addr_state : WV',
       'addr_state : WY', 'initial_list_status : f',
       'initial_list_status : w'], dtype=object)
```

In [44]:

```python
#finding missing value and preposseing
loan_data.isnull()
```

Out[44]:

	Unnamed: 0	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_
0	False	False	False	False	False	False	False	I
1	False	False	False	False	False	False	False	I
2	False	False	False	False	False	False	False	I
3	False	False	False	False	False	False	False	I
4	False	False	False	False	False	False	False	I
...	...	...	...	...	...	...	...	
466280	False	False	False	False	False	False	False	I
466281	False	False	False	False	False	False	False	I
466282	False	False	False	False	False	False	False	I
466283	False	False	False	False	False	False	False	I
466284	False	False	False	False	False	False	False	I

466285 rows × 207 columns

In [45]:
```
#number of missing values in all columns
pd.options.display.max_rows = None
loan_data.isnull().sum()
```

Out[45]:
```
Unnamed: 0                          0
id                                  0
member_id                           0
loan_amnt                           0
funded_amnt                         0
funded_amnt_inv                     0
term                                0
int_rate                            0
installment                         0
grade                               0
sub_grade                           0
emp_title                       27588
emp_length                      21008
home_ownership                      0
annual_inc                          4
verification_status                 0
issue_d                             0
loan_status                         0
pymnt_plan                          0
url                                 0
desc                           340302
purpose                             0
title                              20
zip_code                            0
addr_state                          0
dti                                 0
delinq_2yrs                        29
earliest_cr_line                   29
inq_last_6mths                     29
mths_since_last_delinq         250351
mths_since_last_record         403647
open_acc                           29
pub_rec                            29
revol_bal                           0
revol_util                        340
total_acc                          29
initial_list_status                 0
out_prncp                           0
out_prncp_inv                       0
total_pymnt                         0
total_pymnt_inv                     0
total_rec_prncp                     0
total_rec_int                       0
total_rec_late_fee                  0
recoveries                          0
collection_recovery_fee             0
last_pymnt_d                      376
last_pymnt_amnt                     0
next_pymnt_d                   227214
last_credit_pull_d                 42
collections_12_mths_ex_med        145
mths_since_last_major_derog    367311
policy_code                         0
application_type                    0
annual_inc_joint               466285
dti_joint                      466285
verification_status_joint      466285
acc_now_delinq                     29
tot_coll_amt                    70276
tot_cur_bal                     70276
open_acc_6m                    466285
open_il_6m                     466285
```

```
open_il_12m                                 466285
open_il_24m                                 466285
mths_since_rcnt_il                          466285
total_bal_il                                466285
il_util                                     466285
open_rv_12m                                 466285
open_rv_24m                                 466285
max_bal_bc                                  466285
all_util                                    466285
total_rev_hi_lim                             70276
inq_fi                                      466285
total_cu_tl                                 466285
inq_last_12m                                466285
emp_length_int                               21008
term_int                                         0
earliest_cr_line_date                           29
mths_since_earliest_cr_line                     29
issue_d_date                                     0
mths_since_issue_d_date                          0
Grade : A                                        0
Grade : B                                        0
Grade : C                                        0
Grade : D                                        0
Grade : E                                        0
Grade : F                                        0
Grade : G                                        0
sub_grade : A1                                   0
sub_grade : A2                                   0
sub_grade : A3                                   0
sub_grade : A4                                   0
sub_grade : A5                                   0
sub_grade : B1                                   0
sub_grade : B2                                   0
sub_grade : B3                                   0
sub_grade : B4                                   0
sub_grade : B5                                   0
sub_grade : C1                                   0
sub_grade : C2                                   0
sub_grade : C3                                   0
sub_grade : C4                                   0
sub_grade : C5                                   0
sub_grade : D1                                   0
sub_grade : D2                                   0
sub_grade : D3                                   0
sub_grade : D4                                   0
sub_grade : D5                                   0
sub_grade : E1                                   0
sub_grade : E2                                   0
sub_grade : E3                                   0
sub_grade : E4                                   0
sub_grade : E5                                   0
sub_grade : F1                                   0
sub_grade : F2                                   0
sub_grade : F3                                   0
sub_grade : F4                                   0
sub_grade : F5                                   0
sub_grade : G1                                   0
sub_grade : G2                                   0
sub_grade : G3                                   0
sub_grade : G4                                   0
sub_grade : G5                                   0
home_ownership : ANY                             0
home_ownership : MORTGAGE                        0
home_ownership : NONE                            0
home_ownership : OTHER                           0
home_ownership : OWN                             0
home_ownership : RENT                            0
verification_status : Not Verified               0
verification_status : Source Verified            0
```

```
verification_status : Verified                                          0
loan_status : Charged Off                                               0
loan_status : Current                                                   0
loan_status : Default                                                   0
loan_status : Does not meet the credit policy. Status:Charged Off       0
loan_status : Does not meet the credit policy. Status:Fully Paid        0
loan_status : Fully Paid                                                0
loan_status : In Grace Period                                           0
loan_status : Late (16-30 days)                                         0
loan_status : Late (31-120 days)                                        0
purpose : car                                                           0
purpose : credit_card                                                   0
purpose : debt_consolidation                                            0
purpose : educational                                                   0
purpose : home_improvement                                              0
purpose : house                                                         0
purpose : major_purchase                                                0
purpose : medical                                                       0
purpose : moving                                                        0
purpose : other                                                         0
purpose : renewable_energy                                              0
purpose : small_business                                                0
purpose : vacation                                                      0
purpose : wedding                                                       0
addr_state : AK                                                         0
addr_state : AL                                                         0
addr_state : AR                                                         0
addr_state : AZ                                                         0
addr_state : CA                                                         0
addr_state : CO                                                         0
addr_state : CT                                                         0
addr_state : DC                                                         0
addr_state : DE                                                         0
addr_state : FL                                                         0
addr_state : GA                                                         0
addr_state : HI                                                         0
addr_state : IA                                                         0
addr_state : ID                                                         0
addr_state : IL                                                         0
addr_state : IN                                                         0
addr_state : KS                                                         0
addr_state : KY                                                         0
addr_state : LA                                                         0
addr_state : MA                                                         0
addr_state : MD                                                         0
addr_state : ME                                                         0
addr_state : MI                                                         0
addr_state : MN                                                         0
addr_state : MO                                                         0
addr_state : MS                                                         0
addr_state : MT                                                         0
addr_state : NC                                                         0
addr_state : NE                                                         0
addr_state : NH                                                         0
addr_state : NJ                                                         0
addr_state : NM                                                         0
addr_state : NV                                                         0
addr_state : NY                                                         0
addr_state : OH                                                         0
addr_state : OK                                                         0
addr_state : OR                                                         0
addr_state : PA                                                         0
addr_state : RI                                                         0
addr_state : SC                                                         0
addr_state : SD                                                         0
addr_state : TN                                                         0
addr_state : TX                                                         0
addr_state : UT                                                         0
addr_state : VA                                                         0
```

```
addr_state : VT                                           0
addr_state : WA                                           0
addr_state : WI                                           0
addr_state : WV                                           0
addr_state : WY                                           0
initial_list_status : f                                   0
initial_list_status : w                                   0
dtype: int64
```

In [46]:
```
#filling missing values with the funded amount in the same place (variable)
loan_data['total_rev_hi_lim'].fillna(loan_data['funded_amnt'], inplace = True
```

In [47]:
```
loan_data['total_rev_hi_lim'].isnull().sum()
```

Out[47]: 0

In [48]:
```
loan_data['annual_inc'].isnull().sum()
```

Out[48]: 4

In [49]:
```
mean_annual_income = loan_data['annual_inc'].mean()
loan_data['annual_inc'].fillna(mean_annual_income, inplace = True)
```

In [50]:
```
loan_data['annual_inc'].isnull().sum()
```

Out[50]: 0

In [51]:
```
loan_data['mths_since_earliest_cr_line'].fillna(0, inplace = True)
loan_data['acc_now_delinq'].fillna(0, inplace = True)
loan_data['total_acc'].fillna(0, inplace = True)
loan_data['pub_rec'].fillna(0, inplace = True)
loan_data['open_acc'].fillna(0, inplace = True)
loan_data['inq_last_6mths'].fillna(0, inplace = True)
loan_data['delinq_2yrs'].fillna(0, inplace = True)
loan_data['emp_length_int'].fillna(0, inplace = True)
```

In [52]:
```
loan_data['mths_since_earliest_cr_line'].isnull().sum()
```

Out[52]: 0

In [53]:
```
loan_data['acc_now_delinq'].isnull().sum()
```

Out[53]: 0

In [54]:
```
loan_data['pub_rec'].isnull().sum()
```

Out[54]: 0

In [55]:
```
loan_data['open_acc'].isnull().sum()
```

Out[55]: 0

In [56]:
```
loan_data['inq_last_6mths'].isnull().sum()
```

Out[56]: 0

In [57]:
```
loan_data['delinq_2yrs'].isnull().sum()
```

Out[57]: 0

In [58]:
```
loan_data['emp_length_int'].isnull().sum()
```

Out[58]: 0

In [59]:
```
#-----------------------------
```

In [60]:
```
#PD MODELS
```

In [61]:
```
loan_data['loan_status'].unique()
```

Out[61]:
```
array(['Fully Paid', 'Charged Off', 'Current', 'Default',
       'Late (31-120 days)', 'In Grace Period', 'Late (16-30 days)',
       'Does not meet the credit policy. Status:Fully Paid',
       'Does not meet the credit policy. Status:Charged Off'],
      dtype=object)
```

In [62]:
```
#Number of people with each loan_status
loan_data['loan_status'].value_counts()
```

Out[62]:
```
Current                                                224226
Fully Paid                                             184739
Charged Off                                             42475
Late (31-120 days)                                       6900
In Grace Period                                          3146
Does not meet the credit policy. Status:Fully Paid       1988
Late (16-30 days)                                        1218
Default                                                   832
Does not meet the credit policy. Status:Charged Off       761
Name: loan_status, dtype: int64
```

In [63]:
```
#Ratio of each counts
loan_data['loan_status'].value_counts() / loan_data['loan_status'].count()
```

Out[63]:
```
Current                                                0.480878
Fully Paid                                             0.396193
Charged Off                                            0.091092
Late (31-120 days)                                     0.014798
In Grace Period                                        0.006747
Does not meet the credit policy. Status:Fully Paid     0.004263
Late (16-30 days)                                      0.002612
Default                                                0.001784
Does not meet the credit policy. Status:Charged Off    0.001632
Name: loan_status, dtype: float64
```

In [64]:
```
#Applying loan default and non-default definition
#Where works like if, else
#isin checks if values are in a list
#2nd arg (0): if condition is True, returns 0, else 1 (3rd arg)
loan_data['good_bad'] = np.where(loan_data['loan_status'].isin(['Charged Off'
                                                                 'Late (31-120
```

In [65]:
```
pd.options.display.max_rows = 10
loan_data['good_bad']
```

Out[65]:
```
0         1
1         0
2         1
3         1
4         1
         ..
466280    1
466281    0
466282    1
466283    1
```

```
466284    1
Name: good_bad, Length: 466285, dtype: int64
```

In [66]:
```
from sklearn.model_selection import train_test_split
```

In [67]:
```
loan_data_inputs_train, loan_data_inputs_test, loan_data_targets_train, loan_
```

In [68]:
```
loan_data_inputs_train.shape
```

Out[68]:
```
(349713, 207)
```

In [69]:
```
loan_data_targets_train.shape
```

Out[69]:
```
(349713,)
```

In [70]:
```
loan_data_inputs_test.shape
```

Out[70]:
```
(116572, 207)
```

In [71]:
```
loan_data_targets_test.shape
```

Out[71]:
```
(116572,)
```

In [72]:
```
loan_data_inputs_train, loan_data_inputs_test, loan_data_targets_train, loan_
    loan_data.drop('good_bad', axis = 1), loan_data['good_bad'], test_size =
```

In [73]:
```
loan_data_inputs_train, loan_data_inputs_test, loan_data_targets_train, loan_
    loan_data.drop('good_bad', axis = 1), loan_data['good_bad'], test_size =
```

In [74]:
```
loan_data_inputs_train.shape
```

Out[74]:
```
(373028, 207)
```

In [75]:
```
loan_data_targets_train.shape
```

Out[75]:
```
(373028,)
```

In [76]:
```
loan_data_targets_test.shape
```

Out[76]:
```
(93257,)
```

In [77]:
```
loan_data_inputs_test.shape
```

Out[77]:
```
(93257, 207)
```

In [78]:
```
df_inputs_prepr = loan_data_inputs_train
df_targets_prepr = loan_data_targets_train
```

In [79]:
```
df_inputs_prepr.columns.values
```

Out[79]:
```
array(['Unnamed: 0', 'id', 'member_id', 'loan_amnt', 'funded_amnt',
       'funded_amnt_inv', 'term', 'int_rate', 'installment', 'grade',
       'sub_grade', 'emp_title', 'emp_length', 'home_ownership',
       'annual_inc', 'verification_status', 'issue_d', 'loan_status',
       'pymnt_plan', 'url', 'desc', 'purpose', 'title', 'zip_code',
       'addr_state', 'dti', 'delinq_2yrs', 'earliest_cr_line',
       'inq_last_6mths', 'mths_since_last_delinq',
       'mths_since_last_record', 'open_acc', 'pub_rec', 'revol_bal',
       'revol_util', 'total_acc', 'initial_list_status', 'out_prncp',
```

```
'out_prncp_inv', 'total_pymnt', 'total_pymnt_inv',
'total_rec_prncp', 'total_rec_int', 'total_rec_late_fee',
'recoveries', 'collection_recovery_fee', 'last_pymnt_d',
'last_pymnt_amnt', 'next_pymnt_d', 'last_credit_pull_d',
'collections_12_mths_ex_med', 'mths_since_last_major_derog',
'policy_code', 'application_type', 'annual_inc_joint', 'dti_joint',
'verification_status_joint', 'acc_now_delinq', 'tot_coll_amt',
'tot_cur_bal', 'open_acc_6m', 'open_il_6m', 'open_il_12m',
'open_il_24m', 'mths_since_rcnt_il', 'total_bal_il', 'il_util',
'open_rv_12m', 'open_rv_24m', 'max_bal_bc', 'all_util',
'total_rev_hi_lim', 'inq_fi', 'total_cu_tl', 'inq_last_12m',
'emp_length_int', 'term_int', 'earliest_cr_line_date',
'mths_since_earliest_cr_line', 'issue_d_date',
'mths_since_issue_d_date', 'Grade : A', 'Grade : B', 'Grade : C',
'Grade : D', 'Grade : E', 'Grade : F', 'Grade : G',
'sub_grade : A1', 'sub_grade : A2', 'sub_grade : A3',
'sub_grade : A4', 'sub_grade : A5', 'sub_grade : B1',
'sub_grade : B2', 'sub_grade : B3', 'sub_grade : B4',
'sub_grade : B5', 'sub_grade : C1', 'sub_grade : C2',
'sub_grade : C3', 'sub_grade : C4', 'sub_grade : C5',
'sub_grade : D1', 'sub_grade : D2', 'sub_grade : D3',
'sub_grade : D4', 'sub_grade : D5', 'sub_grade : E1',
'sub_grade : E2', 'sub_grade : E3', 'sub_grade : E4',
'sub_grade : E5', 'sub_grade : F1', 'sub_grade : F2',
'sub_grade : F3', 'sub_grade : F4', 'sub_grade : F5',
'sub_grade : G1', 'sub_grade : G2', 'sub_grade : G3',
'sub_grade : G4', 'sub_grade : G5', 'home_ownership : ANY',
'home_ownership : MORTGAGE', 'home_ownership : NONE',
'home_ownership : OTHER', 'home_ownership : OWN',
'home_ownership : RENT', 'verification_status : Not Verified',
'verification_status : Source Verified',
'verification_status : Verified', 'loan_status : Charged Off',
'loan_status : Current', 'loan_status : Default',
'loan_status : Does not meet the credit policy. Status:Charged Off',
'loan_status : Does not meet the credit policy. Status:Fully Paid',
'loan_status : Fully Paid', 'loan_status : In Grace Period',
'loan_status : Late (16-30 days)',
'loan_status : Late (31-120 days)', 'purpose : car',
'purpose : credit_card', 'purpose : debt_consolidation',
'purpose : educational', 'purpose : home_improvement',
'purpose : house', 'purpose : major_purchase', 'purpose : medical',
'purpose : moving', 'purpose : other',
'purpose : renewable_energy', 'purpose : small_business',
'purpose : vacation', 'purpose : wedding', 'addr_state : AK',
'addr_state : AL', 'addr_state : AR', 'addr_state : AZ',
'addr_state : CA', 'addr_state : CO', 'addr_state : CT',
'addr_state : DC', 'addr_state : DE', 'addr_state : FL',
'addr_state : GA', 'addr_state : HI', 'addr_state : IA',
'addr_state : ID', 'addr_state : IL', 'addr_state : IN',
'addr_state : KS', 'addr_state : KY', 'addr_state : LA',
'addr_state : MA', 'addr_state : MD', 'addr_state : ME',
'addr_state : MI', 'addr_state : MN', 'addr_state : MO',
'addr_state : MS', 'addr_state : MT', 'addr_state : NC',
'addr_state : NE', 'addr_state : NH', 'addr_state : NJ',
'addr_state : NM', 'addr_state : NV', 'addr_state : NY',
'addr_state : OH', 'addr_state : OK', 'addr_state : OR',
'addr_state : PA', 'addr_state : RI', 'addr_state : SC',
'addr_state : SD', 'addr_state : TN', 'addr_state : TX',
'addr_state : UT', 'addr_state : VA', 'addr_state : VT',
'addr_state : WA', 'addr_state : WI', 'addr_state : WV',
'addr_state : WY', 'initial_list_status : f',
'initial_list_status : w'], dtype=object)
```

In [80]:
```
df_inputs_prepr['grade'].unique()
```

Out[80]:
```
array(['A', 'C', 'D', 'B', 'E', 'F', 'G'], dtype=object)
```

In [81]:
```
pd.options.display.max_columns = 5
```

```
df1 = pd.concat([df_inputs_prepr['grade'], df_targets_prepr], axis = 1)
df1.head()
```

Out[81]:

	grade	good_bad
427211	A	1
206088	C	1
136020	A	1
412305	D	0
36159	C	0

In [82]:

```
df1.groupby(df1.columns.values[0], as_index=False)[df1.columns.values[1]].cou
```

Out[82]:

	grade	good_bad
0	A	59759
1	B	109730
2	C	100245
3	D	61498
4	E	28612
5	F	10530
6	G	2654

In [83]:

```
df1.groupby(df1.columns.values[0], as_index=False)[df1.columns.values[1]].mea
```

Out[83]:

	grade	good_bad
0	A	0.961044
1	B	0.921015
2	C	0.885770
3	D	0.846304
4	E	0.805257
5	F	0.754416
6	G	0.727958

In [84]:

```
#Mergigng both the above outputs into 1 dataframe
df1 = pd.concat([df1.groupby(df1.columns.values[0], as_index=False)[df1.colum
                df1.groupby(df1.columns.values[0], as_index=False)[df1.column
```

In [85]:

```
df1
```

Out[85]:

	grade	good_bad	grade	good_bad
0	A	59759	A	0.961044
1	B	109730	B	0.921015
2	C	100245	C	0.885770
3	D	61498	D	0.846304
4	E	28612	E	0.805257

	grade	good_bad	grade	good_bad
5	F	10530	F	0.754416
6	G	2654	G	0.727958

In [86]:
```
#Renaming column names
df1 = df1.iloc[:, [0,1,3]]
df1.columns = [df1.columns.values[0], 'n_obs', 'prop_good']
df1
```

Out[86]:

	grade	n_obs	prop_good
0	A	59759	0.961044
1	B	109730	0.921015
2	C	100245	0.885770
3	D	61498	0.846304
4	E	28612	0.805257
5	F	10530	0.754416
6	G	2654	0.727958

In [87]:
```
#Calculating proportion of observations
df1['prop_n_obs'] = df1['n_obs'] / df1['n_obs'].sum()
```

In [88]:
```
df1
```

Out[88]:

	grade	n_obs	prop_good	prop_n_obs
0	A	59759	0.961044	0.160200
1	B	109730	0.921015	0.294160
2	C	100245	0.885770	0.268733
3	D	61498	0.846304	0.164862
4	E	28612	0.805257	0.076702
5	F	10530	0.754416	0.028228
6	G	2654	0.727958	0.007115

In [89]:
```
#Calculating number of good and bad borrowers for each grade
pd.options.display.max_columns = None
df1['n_good'] = df1['prop_good'] * df1['n_obs']
df1['n_bad'] = (1 - df1['prop_good']) * df1['n_obs']
df1
```

Out[89]:

	grade	n_obs	prop_good	prop_n_obs	n_good	n_bad
0	A	59759	0.961044	0.160200	57431.0	2328.0
1	B	109730	0.921015	0.294160	101063.0	8667.0
2	C	100245	0.885770	0.268733	88794.0	11451.0
3	D	61498	0.846304	0.164862	52046.0	9452.0
4	E	28612	0.805257	0.076702	23040.0	5572.0
5	F	10530	0.754416	0.028228	7944.0	2586.0

	grade	n_obs	prop_good	prop_n_obs	n_good	n_bad
6	G	2654	0.727958	0.007115	1932.0	722.0

In [90]:

```
#Calculating proportion of good and bad
df1['prop_n_good'] = df1['n_good'] / df1['n_good'].sum()
df1['prop_n_bad'] = df1['n_bad'] / df1['n_bad'].sum()
df1
```

Out[90]:

	grade	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop_n_bad
0	A	59759	0.961044	0.160200	57431.0	2328.0	0.172855	0.057090
1	B	109730	0.921015	0.294160	101063.0	8667.0	0.304178	0.212541
2	C	100245	0.885770	0.268733	88794.0	11451.0	0.267251	0.280813
3	D	61498	0.846304	0.164862	52046.0	9452.0	0.156647	0.231792
4	E	28612	0.805257	0.076702	23040.0	5572.0	0.069345	0.136642
5	F	10530	0.754416	0.028228	7944.0	2586.0	0.023910	0.063417
6	G	2654	0.727958	0.007115	1932.0	722.0	0.005815	0.017706

In [91]:

```
#Calculating Weight of evidence
df1['WOE'] = np.log(df1['prop_n_good'] / df1['prop_n_bad'])
df1
```

Out[91]:

	grade	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop_n_bad	
0	A	59759	0.961044	0.160200	57431.0	2328.0	0.172855	0.057090	1.1
1	B	109730	0.921015	0.294160	101063.0	8667.0	0.304178	0.212541	0.3
2	C	100245	0.885770	0.268733	88794.0	11451.0	0.267251	0.280813	-0.0
3	D	61498	0.846304	0.164862	52046.0	9452.0	0.156647	0.231792	-0.3
4	E	28612	0.805257	0.076702	23040.0	5572.0	0.069345	0.136642	-0.6
5	F	10530	0.754416	0.028228	7944.0	2586.0	0.023910	0.063417	-0.9
6	G	2654	0.727958	0.007115	1932.0	722.0	0.005815	0.017706	-1.1

In [92]:

```
df1 = df1.sort_values(['WOE'])
df1 = df1.reset_index(drop=True)
df1
```

Out[92]:

	grade	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop_n_bad	
0	G	2654	0.727958	0.007115	1932.0	722.0	0.005815	0.017706	-1.1
1	F	10530	0.754416	0.028228	7944.0	2586.0	0.023910	0.063417	-0.9
2	E	28612	0.805257	0.076702	23040.0	5572.0	0.069345	0.136642	-0.6
3	D	61498	0.846304	0.164862	52046.0	9452.0	0.156647	0.231792	-0.3
4	C	100245	0.885770	0.268733	88794.0	11451.0	0.267251	0.280813	-0.0
5	B	109730	0.921015	0.294160	101063.0	8667.0	0.304178	0.212541	0.3
6	A	59759	0.961044	0.160200	57431.0	2328.0	0.172855	0.057090	1.1

In [93]:

```
#Calculating the difference between rows (above - below)
df1['diff_prop_good'] = df1['prop_good'].diff().abs()
```

```
df1['diff_WOE'] = df1['WOE'].diff().abs()
df1
```

Out[93]:

	grade	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop_n_bad	
0	G	2654	0.727958	0.007115	1932.0	722.0	0.005815	0.017706	-1.1
1	F	10530	0.754416	0.028228	7944.0	2586.0	0.023910	0.063417	-0.9
2	E	28612	0.805257	0.076702	23040.0	5572.0	0.069345	0.136642	-0.6
3	D	61498	0.846304	0.164862	52046.0	9452.0	0.156647	0.231792	-0.3
4	C	100245	0.885770	0.268733	88794.0	11451.0	0.267251	0.280813	-0.0
5	B	109730	0.921015	0.294160	101063.0	8667.0	0.304178	0.212541	0.3
6	A	59759	0.961044	0.160200	57431.0	2328.0	0.172855	0.057090	1.1

In [94]:

```
#Calculating Information Value (IV)
df1['IV'] = (df1['prop_n_good'] - df1['prop_n_bad']) * df1['WOE']
df1['IV'] = df1['IV'].sum()
df1
```

Out[94]:

	grade	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop_n_bad	
0	G	2654	0.727958	0.007115	1932.0	722.0	0.005815	0.017706	-1.1
1	F	10530	0.754416	0.028228	7944.0	2586.0	0.023910	0.063417	-0.9
2	E	28612	0.805257	0.076702	23040.0	5572.0	0.069345	0.136642	-0.6
3	D	61498	0.846304	0.164862	52046.0	9452.0	0.156647	0.231792	-0.3
4	C	100245	0.885770	0.268733	88794.0	11451.0	0.267251	0.280813	-0.0
5	B	109730	0.921015	0.294160	101063.0	8667.0	0.304178	0.212541	0.3
6	A	59759	0.961044	0.160200	57431.0	2328.0	0.172855	0.057090	1.1

In [95]:

```
def woe_discrete(df, discrete_variable_name, good_bad_variable_df):
    df = pd.concat([df[discrete_variable_name], good_bad_variable_df], axis=1
    df = pd.concat([df.groupby(df.columns.values[0], as_index=False)[df.colum
                    df.groupby(df.columns.values[0], as_index=False)[df.column
    df = df.iloc[:, [0,1,3]]
    df.columns = [df.columns.values[0], 'n_obs', 'prop_good']
    df['prop_n_obs'] = df['n_obs'] / df['n_obs'].sum()
    df['n_good'] = df['prop_good'] * df['n_obs']
    df['n_bad'] = (1 - df['prop_good']) * df['n_obs']
    df['prop_n_good'] = df['n_good'] / df['n_good'].sum()
    df['prop_n_bad'] = df['n_bad'] / df['n_bad'].sum()
    df['WOE'] = np.log(df['prop_n_good'] / df['prop_n_bad'])
    df = df.sort_values(['WOE'])
    df = df.reset_index(drop=True)
    df['diff_prop_good'] = df['prop_good'].diff().abs()
    df['diff_WOE'] = df['WOE'].diff().abs()
    df['IV'] = (df['prop_n_good'] - df['prop_n_bad']) * df['WOE']
    df['IV'] = df['IV'].sum()
    return df
```

In [96]:

```
df_temp = woe_discrete(df_inputs_prepr, 'grade', df_targets_prepr)
df_temp
```

Out[96]:

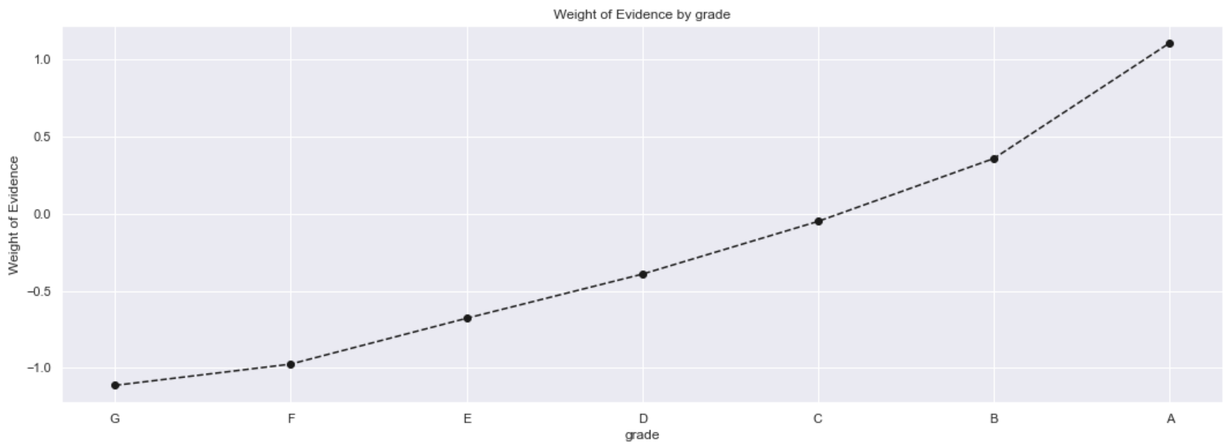
	grade	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop_n_bad	
0	G	2654	0.727958	0.007115	1932.0	722.0	0.005815	0.017706	-1.1

	grade	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop_n_bad	
1	F	10530	0.754416	0.028228	7944.0	2586.0	0.023910	0.063417	-0.9
2	E	28612	0.805257	0.076702	23040.0	5572.0	0.069345	0.136642	-0.6
3	D	61498	0.846304	0.164862	52046.0	9452.0	0.156647	0.231792	-0.3
4	C	100245	0.885770	0.268733	88794.0	11451.0	0.267251	0.280813	-0.0
5	B	109730	0.921015	0.294160	101063.0	8667.0	0.304178	0.212541	0.3
6	A	59759	0.961044	0.160200	57431.0	2328.0	0.172855	0.057090	1.1

In [97]:
```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

In [98]:
```
def plot_by_woe(df_WOE, rotation_of_x_axis_labels = 0):
    #converting the independent variable categories into strings and making a
    x = np.array(df_WOE.iloc[:,0].apply(str))
    y = df_WOE['WOE']
    #width = 18 inches, height = 6 inches
    plt.figure(figsize = (18,6))
    plt.plot(x, y, marker = 'o', linestyle = '--', color = 'k') #marker='o':
    plt.xlabel(df_WOE.columns[0])
    plt.ylabel('Weight of Evidence')
    plt.title(str('Weight of Evidence by ' + df_WOE.columns[0]))
    plt.xticks(rotation = rotation_of_x_axis_labels)
```

In [99]:
```
plot_by_woe(df_temp)
```

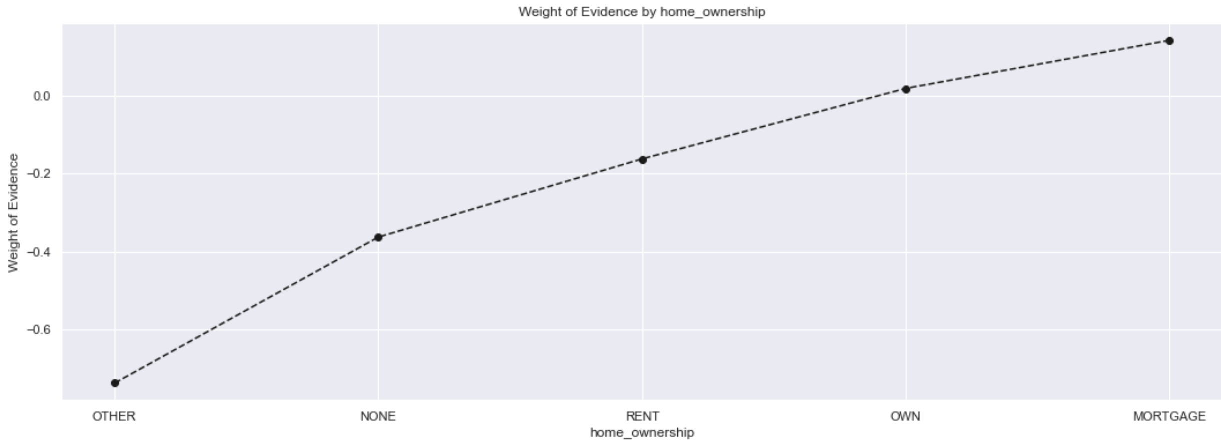


In [100...
```
df_temp = woe_discrete(df_inputs_prepr, 'home_ownership', df_targets_prepr)
df_temp
```

Out[100...

	home_ownership	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop_
0	OTHER	137	0.795620	0.000367	109.0	28.0	0.000328	0.0
1	NONE	40	0.850000	0.000107	34.0	6.0	0.000102	0.0
2	RENT	150599	0.873870	0.403720	131604.0	18995.0	0.396099	0.4
3	OWN	33295	0.892536	0.089256	29717.0	3578.0	0.089442	0.0
4	MORTGAGE	188956	0.903835	0.506546	170785.0	18171.0	0.514026	0.4
5	ANY	1	1.000000	0.000003	1.0	0.0	0.000003	0.0

In [101...
```
plot_by_woe(df_temp)
```



In [102…
```
df_inputs_prepr['home_ownership : RENT_OTHER_NONE_ANY'] = sum([df_inputs_prep
df_inputs_prepr['home_ownership : OTHER'], df_inputs_prepr['home_ownership : 
```

In [103…
```
df_inputs_prepr['addr_state'].unique()
```

Out[103…
```
array(['SC', 'NJ', 'GA', 'MA', 'CA', 'IL', 'NC', 'NY', 'TX', 'CT', 'FL',
       'VA', 'UT', 'AZ', 'MD', 'WI', 'MI', 'CO', 'TN', 'IN', 'AL', 'NV',
       'MT', 'RI', 'OR', 'MN', 'KS', 'AK', 'PA', 'OH', 'WA', 'KY', 'OK',
       'MO', 'NM', 'HI', 'WV', 'LA', 'VT', 'AR', 'DC', 'SD', 'NH', 'WY',
       'MS', 'DE', 'IA', 'NE', 'ID', 'ME'], dtype=object)
```

In [104…
```
pd.options.display.max_rows = None
df_temp = woe_discrete(df_inputs_prepr, 'addr_state', df_targets_prepr)
df_temp
```

```
/Applications/anaconda3/lib/python3.8/site-packages/pandas/core/algorithms.py:
1977: RuntimeWarning: invalid value encountered in subtract
  out_arr[res_indexer] = arr[res_indexer] - arr[lag_indexer]
```

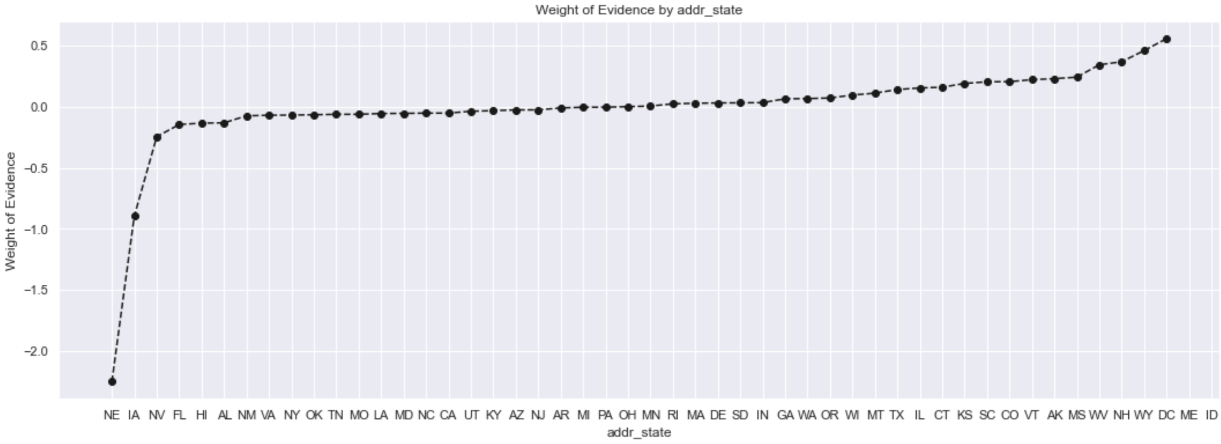
Out[104…

	addr_state	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop_n_bad
0	NE	13	0.461538	0.000035	6.0	7.0	0.000018	0.000172
1	IA	13	0.769231	0.000035	10.0	3.0	0.000030	0.000074
2	NV	5221	0.864585	0.013996	4514.0	707.0	0.013586	0.017338
3	FL	25211	0.875808	0.067585	22080.0	3131.0	0.066456	0.076782
4	HI	2001	0.877061	0.005364	1755.0	246.0	0.005282	0.006033
5	AL	4671	0.877328	0.012522	4098.0	573.0	0.012334	0.014052
6	NM	2075	0.883373	0.005563	1833.0	242.0	0.005517	0.005935
7	VA	11366	0.883864	0.030470	10046.0	1320.0	0.030236	0.032370
8	NY	32211	0.883984	0.086350	28474.0	3737.0	0.085701	0.091643
9	OK	3284	0.884287	0.008804	2904.0	380.0	0.008740	0.009319
10	TN	4845	0.884623	0.012988	4286.0	559.0	0.012900	0.013708
11	MO	6017	0.884660	0.016130	5323.0	694.0	0.016021	0.017019
12	LA	4359	0.885295	0.011685	3859.0	500.0	0.011615	0.012262
13	MD	8771	0.885418	0.023513	7766.0	1005.0	0.023374	0.024646
14	NC	10204	0.885633	0.027355	9037.0	1167.0	0.027199	0.028618
15	CA	57199	0.885645	0.153337	50658.0	6541.0	0.152470	0.160405
16	UT	2756	0.887155	0.007388	2445.0	311.0	0.007359	0.007627
17	KY	3587	0.887650	0.009616	3184.0	403.0	0.009583	0.009883

	addr_state	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop_n_bad
18	AZ	8645	0.888259	0.023175	7679.0	966.0	0.023112	0.023689
19	NJ	14385	0.888286	0.038563	12778.0	1607.0	0.038459	0.039409
20	AR	2760	0.889855	0.007399	2456.0	304.0	0.007392	0.007455
21	MI	9191	0.890436	0.024639	8184.0	1007.0	0.024632	0.024695
22	PA	13090	0.890451	0.035091	11656.0	1434.0	0.035082	0.035166
23	OH	12135	0.890894	0.032531	10811.0	1324.0	0.032539	0.032468
24	MN	6526	0.891358	0.017495	5817.0	709.0	0.017508	0.017387
25	RI	1647	0.893139	0.004415	1471.0	176.0	0.004427	0.004316
26	MA	8858	0.893543	0.023746	7915.0	943.0	0.023822	0.023125
27	DE	1064	0.893797	0.002852	951.0	113.0	0.002862	0.002771
28	SD	801	0.893883	0.002147	716.0	85.0	0.002155	0.002084
29	IN	5210	0.894050	0.013967	4658.0	552.0	0.014020	0.013537
30	GA	11960	0.896990	0.032062	10728.0	1232.0	0.032289	0.030212
31	WA	8372	0.897157	0.022443	7511.0	861.0	0.022606	0.021114
32	OR	4814	0.897590	0.012905	4321.0	493.0	0.013005	0.012090
33	WI	4740	0.899789	0.012707	4265.0	475.0	0.012837	0.011648
34	MT	1103	0.901179	0.002957	994.0	109.0	0.002992	0.002673
35	TX	29158	0.903800	0.078166	26353.0	2805.0	0.079317	0.068787
36	IL	14833	0.904874	0.039764	13422.0	1411.0	0.040397	0.034602
37	CT	5775	0.905455	0.015481	5229.0	546.0	0.015738	0.013390
38	KS	3360	0.908036	0.009007	3051.0	309.0	0.009183	0.007578
39	SC	4448	0.909173	0.011924	4044.0	404.0	0.012172	0.009907
40	CO	7823	0.909242	0.020972	7113.0	710.0	0.021409	0.017411
41	VT	727	0.910591	0.001949	662.0	65.0	0.001992	0.001594
42	AK	1003	0.911266	0.002689	914.0	89.0	0.002751	0.002183
43	MS	980	0.912245	0.002627	894.0	86.0	0.002691	0.002109
44	WV	1926	0.920042	0.005163	1772.0	154.0	0.005333	0.003777
45	NH	1830	0.921858	0.004906	1687.0	143.0	0.005078	0.003507
46	WY	919	0.928183	0.002464	853.0	66.0	0.002567	0.001619
47	DC	1129	0.934455	0.003027	1055.0	74.0	0.003175	0.001815
48	ME	2	1.000000	0.000005	2.0	0.0	0.000006	0.000000
49	ID	10	1.000000	0.000027	10.0	0.0	0.000030	0.000000

In [105…

```
plot_by_woe(df_temp)
```



In [106…

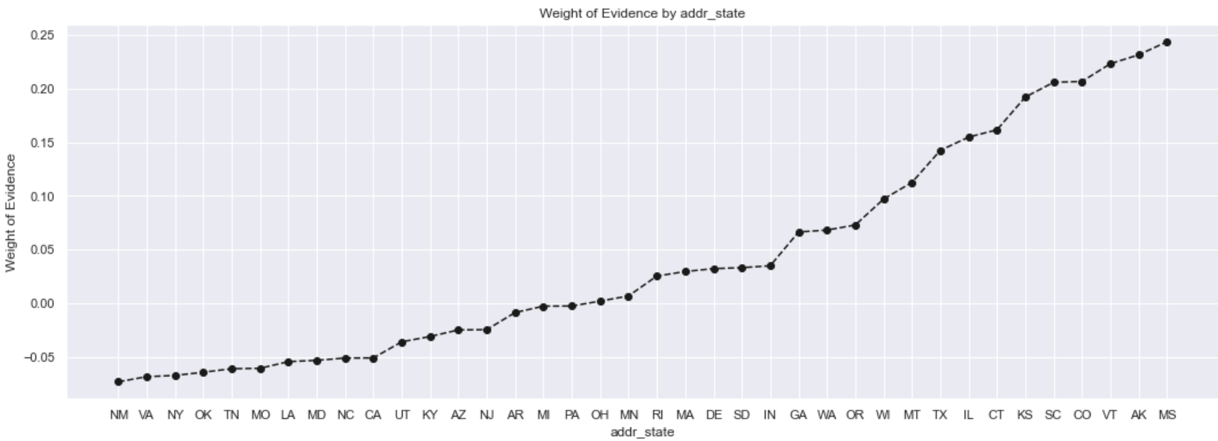
```
#set all values of the state ND to 0 if there is no such column, else pass
if ['addr_state : ND'] in df_inputs_prepr.columns.values:
    pass
else:
    df_inputs_prepr['addr_state : ND'] = 0
```

In [107…

```
#Plotting without first 2 and last 2 states
plot_by_woe(df_temp.iloc[2:-2, :])
```

In [108…

```
plot_by_woe(df_temp.iloc[6:-6, :])
```



In [109…

```
#Creating the actual dummy variables
df_inputs_prepr['addr_state : ND_NE_IA_NV_FL_HI_AL'] = sum([df_inputs_prepr['
df_inputs_prepr['addr_state : IA'], df_inputs_prepr['addr_state : NV'], df_in
df_inputs_prepr['addr_state : HI'], df_inputs_prepr['addr_state : AL']])

df_inputs_prepr['addr_state : NM_VA'] = sum([df_inputs_prepr['addr_state : NM

df_inputs_prepr['addr_state : OK_TN_MO_LA_MD_NC'] = sum([df_inputs_prepr['add
```

```
df_inputs_prepr['addr_state : TN'], df_inputs_prepr['addr_state : MO'], df_in
df_inputs_prepr['addr_state : MD'], df_inputs_prepr['addr_state : NC']])

df_inputs_prepr['addr_state : UT_KY_AZ_NJ'] = sum([df_inputs_prepr['addr_stat
df_inputs_prepr['addr_state : AZ'], df_inputs_prepr['addr_state : NJ']])

df_inputs_prepr['addr_state : AR_MI_PA_OH_MN'] = sum([df_inputs_prepr['addr_s
df_inputs_prepr['addr_state : PA'], df_inputs_prepr['addr_state : OH'], df_in

df_inputs_prepr['addr_state : RI_MA_DE_SD_IN'] = sum([df_inputs_prepr['addr_s
df_inputs_prepr['addr_state : DE'], df_inputs_prepr['addr_state : SD'], df_in

df_inputs_prepr['addr_state : GA_WA_OR'] = sum([df_inputs_prepr['addr_state :
df_inputs_prepr['addr_state : OR']])

df_inputs_prepr['addr_state : WI_MT'] = sum([df_inputs_prepr['addr_state : WI

df_inputs_prepr['addr_state : IL_CT'] = sum([df_inputs_prepr['addr_state : IL

df_inputs_prepr['addr_state : KS_SC_CO_VT_AK_MS'] = sum([df_inputs_prepr['add
df_inputs_prepr['addr_state : SC'], df_inputs_prepr['addr_state : CO'], df_in
df_inputs_prepr['addr_state : AK'], df_inputs_prepr['addr_state : MS']])

df_inputs_prepr['addr_state : WV_NH_WY_DC_ME_ID'] = sum([df_inputs_prepr['add
df_inputs_prepr['addr_state : NH'], df_inputs_prepr['addr_state : WY'], df_in
df_inputs_prepr['addr_state : ME'], df_inputs_prepr['addr_state : ID']])
```

In [110…

```
df_temp = woe_discrete(df_inputs_prepr, 'verification_status', df_targets_pre
df_temp
```

Out[110…

	verification_status	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop
0	Verified	134414	0.872781	0.360332	117314.0	17100.0	0.353090	0
1	Source Verified	120030	0.895918	0.321772	107537.0	12493.0	0.323663	0.
2	Not Verified	118584	0.905679	0.317896	107399.0	11185.0	0.323248	0

In [111…

```
plot_by_woe(df_temp)
```

In [112…

```
df_temp = woe_discrete(df_inputs_prepr, 'purpose', df_targets_prepr)
df_temp
```

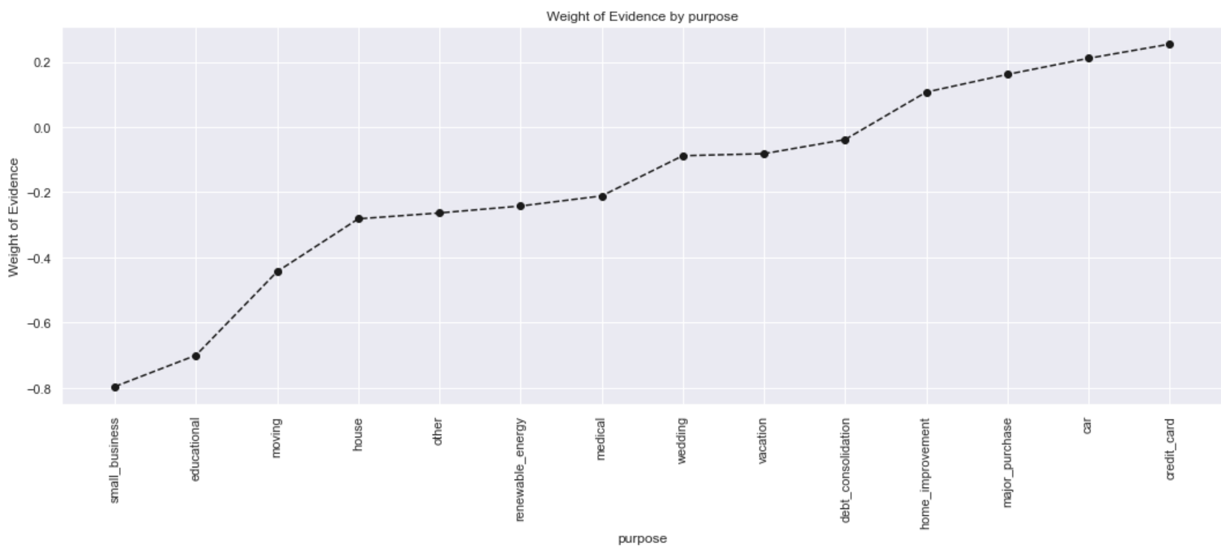
Out[112…

	purpose	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop
0	small_business	5582	0.786098	0.014964	4388.0	1194.0	0.013207	C
1	educational	333	0.801802	0.000893	267.0	66.0	0.000804	(
2	moving	2392	0.839465	0.006412	2008.0	384.0	0.006044	(

	purpose	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop
3	house	1830	0.860109	0.004906	1574.0	256.0	0.004737	0
4	other	18884	0.862264	0.050624	16283.0	2601.0	0.049008	0
5	renewable_energy	281	0.864769	0.000753	243.0	38.0	0.000731	0
6	medical	3684	0.868350	0.009876	3199.0	485.0	0.009628	(
7	wedding	1887	0.881823	0.005059	1664.0	223.0	0.005008	0
8	vacation	1991	0.882471	0.005337	1757.0	234.0	0.005288	0
9	debt_consolidation	219183	0.886884	0.587578	194390.0	24793.0	0.585071	0
10	home_improvement	21238	0.900697	0.056934	19129.0	2109.0	0.057574	
11	major_purchase	7837	0.905449	0.021009	7096.0	741.0	0.021357	
12	car	4325	0.909595	0.011594	3934.0	391.0	0.011840	0
13	credit_card	83581	0.913102	0.224061	76318.0	7263.0	0.229701	

In [113…

```
plot_by_woe(df_temp, rotation_of_x_axis_labels = 90)
```



In [114…

```
df_inputs_prepr['purpose : SB_ED'] = sum([df_inputs_prepr['purpose : small_bu

df_inputs_prepr['purpose : HO_OT_RE_ME'] = sum([df_inputs_prepr['purpose : ho
df_inputs_prepr['purpose : renewable_energy'], df_inputs_prepr['purpose : med

df_inputs_prepr['purpose : WE_VA_DC'] = sum([df_inputs_prepr['purpose : weddi
df_inputs_prepr['purpose : debt_consolidation']])

df_inputs_prepr['purpose : HI_MP_CA_CC'] = sum([df_inputs_prepr['purpose : ho
df_inputs_prepr['purpose : car'], df_inputs_prepr['purpose : credit_card']])
```

In [115…

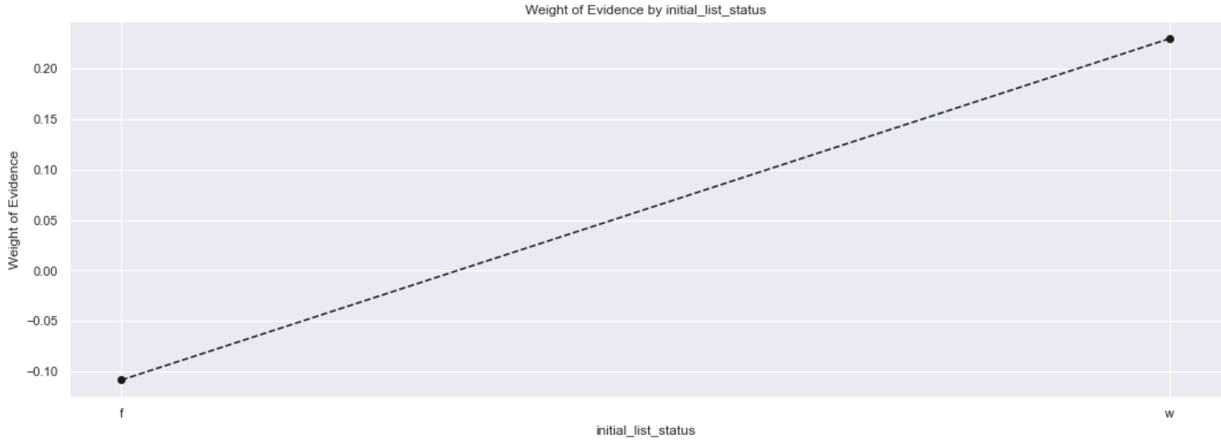
```
df_temp = woe_discrete(df_inputs_prepr, 'initial_list_status', df_targets_pre
df_temp
```

Out[115…

	initial_list_status	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop_
0	f	242514	0.879694	0.650123	213338.0	29176.0	0.642101	0.7
1	w	130514	0.911105	0.349877	118912.0	11602.0	0.357899	0.2

In [116…

```
plot_by_woe(df_temp)
```



In [117…

```
def woe_ordered_continuous(df, discrete_variable_name, good_bad_variable_df):
    df = pd.concat([df[discrete_variable_name], good_bad_variable_df], axis=1
    df = pd.concat([df.groupby(df.columns.values[0], as_index=False)[df.colum
                    df.groupby(df.columns.values[0], as_index=False)[df.column
    df = df.iloc[:, [0,1,3]]
    df.columns = [df.columns.values[0], 'n_obs', 'prop_good']
    df['prop_n_obs'] = df['n_obs'] / df['n_obs'].sum()
    df['n_good'] = df['prop_good'] * df['n_obs']
    df['n_bad'] = (1 - df['prop_good']) * df['n_obs']
    df['prop_n_good'] = df['n_good'] / df['n_good'].sum()
    df['prop_n_bad'] = df['n_bad'] / df['n_bad'].sum()
    df['WOE'] = np.log(df['prop_n_good'] / df['prop_n_bad'])
    df['diff_prop_good'] = df['prop_good'].diff().abs()
    df['diff_WOE'] = df['WOE'].diff().abs()
    df['IV'] = (df['prop_n_good'] - df['prop_n_bad']) * df['WOE']
    df['IV'] = df['IV'].sum()
    return df
```

In [118…

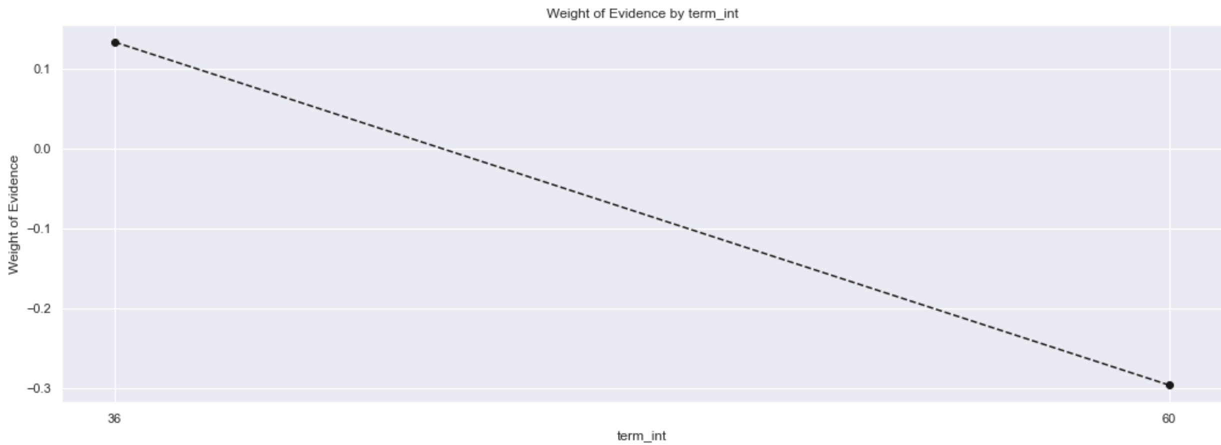
```
df_temp = woe_ordered_continuous(df_inputs_prepr, 'term_int', df_targets_prep
df_temp
```

Out[118…

	term_int	n_obs	prop_good	prop_n_obs	n_good	n_bad	prop_n_good	prop_n_bad
0	36	270419	0.902995	0.724929	244187.0	26232.0	0.73495	0.643288
1	60	102609	0.858239	0.275071	88063.0	14546.0	0.26505	0.356712

In [119…

```
plot_by_woe(df_temp)
```



In [120…

```
df_inputs_prepr['term : 36'] = np.where((df_inputs_prepr['term_int'] == 36),
df_inputs_prepr['term : 60'] = np.where((df_inputs_prepr['term_int'] == 60),
```

In [ ]: