Predicting Closed Loans

April 26, 2018

1 Final Project

1.1 Setting up our environment

```
In [1]: import pandas as pd
        from pandas.core.dtypes.common import is_numeric_dtype
        import numpy as np
        from scipy import stats
        from scipy.spatial.distance import cdist
        from datetime import datetime


        ## Import plotting libraries
        import seaborn as sns
        import matplotlib as mpl
        import matplotlib.pyplot as plt
        plt.style.use('seaborn-whitegrid')
        %matplotlib inline

        import warnings
        from sklearn import preprocessing

        ## Data loading packages
        import os
        from PyBI import sql as s

        ## Formatting
        from IPython.core.display import display, HTML

        notebook_style = os.path.join('.','style','style-notebook.css')

        css = open(notebook_style).read()
        HTML('<style>{}</style>'.format(css))

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

1.2 Exploratory Data Analysis

In [2]:
```
queryPath = os.path.join('.','Refi Closings 2017.sql')
loans = s.script_to_df(queryPath)

# Change Index to loanID
loans.set_index('loanID', inplace = True)
df = loans.copy()
```

In [3]:
```
loans = df.copy()

n_rows, n_cols = loans.shape
for n, col in enumerate(loans.items()):
    print('{0:3} {1:30} {2}'.format(str(n),col[1].name,str(col[1].dtype)))

# View Data
loans.head()
```

```
0   daysInProcess                  int64
1   daysToPullCredit               float64
2   daysToStartApplication         float64
3   daysToCompleteApplication      int64
4   daysForInitialCheck            int64
5   inProcessStartDate             datetime64[ns]
6   leadGrade                      object
7   leadSource                     object
8   loanReason                     object
9   refinanceReason                object
10  jumboFlag                      object
11  loanProduct                    object
12  loanProductDescription         object
13  productBucket                  object
14  heloc                          object
15  channel                        object
16  cema                           object
17  docType                        object
18  dti                            float64
19  ltv                            object
20  fico                           object
21  income                         object
22  selfEmployed                   object
23  assets                         object
24  currentlyServicedLoans         int64
25  previouslyServicedLoans        int64
26  loanAmount                     object
27  shortage                       object
28  estimatedProfit                object
29  appraisalWaiver                object
30  Deposit                        object
```

```
31  cashOutAmount                 object
32  interestRate                  float64
33  treasury10YrYield             float64
34  dow                           float64
35  monthlyPayment                object
36  pmi                           object
37  lockPeriod                    object
38  lockLength                    float64
39  loanTerm                      object
40  requiredPoints                object
41  clientPoints                  object
42  originationCharges            float64
43  thirdPartyCosts               float64
44  cureCosts                     object
45  prepaidEscrows                float64
46  escrowedAmount                object
47  occupancyType                 object
48  propertyType                  object
49  state                         object
50  county                        object
51  countyPopulation              float64
52  zipCode                       object
53  latitude                      float64
54  longitude                     float64
55  ruralInd                      object
56  bankerLocation                object
57  bankerSpeciality              object
58  bankerTier                    object
59  bankerTenure                  float64
60  bankerContinuousTenure        float64
```

Out[3]:
```
        daysInProcess  daysToPullCredit  daysToStartApplication  \
loanID
520               234               0.0                    28.0
7214               90               NaN                     NaN
1480               95               0.0                    13.0
17231              28               0.0                     4.0
917               157               1.0                     0.0

        daysToCompleteApplication  daysForInitialCheck  inProcessStartDate  \
loanID
520                             0                    3 2016-07-29 09:37:44
7214                           15                    0 2016-11-23 14:35:23
1480                            0                    3 2016-10-06 17:07:45
17231                           0                    2 2016-12-09 16:42:27
917                             0                    4 2016-08-31 21:24:21
```

loanID	leadGrade	leadSource	loanReason	refinanceReason	jumboFlag
520	None	LOLA	Refinance	Ltd Cash Out (rate/term)	None
7214	None	QLMS Broker	Refinance	Cash Out	None
1480	None	LOLA	Refinance	Ltd Cash Out (rate/term)	None
17231	None	App Call	Refinance	Ltd Cash Out (rate/term)	None
917	A	CARI	Refinance	Cash Out	None

loanID	loanProduct	loanProductDescription	productBucket	heloc
520	230	230 - 20 yr Conforming Fixed	Conventional	None
7214	F30	F30 - FHA 30 Yr Fixed	FHA	None
1480	130P	130P - 30 yr Conforming Fixed	Conventional	None
17231	587L	587L - 7/1 Arm Conforming	Conventional	None
917	330	330 - 15 yr Conforming Fixed	Conventional	None

loanID	channel	cema	docType	dti	ltv	fico	income	selfEmployed
520	Forward	None	Full	36.756431	80	803	2609.04	None
7214	QLMS	None	Full	37.980032	75	678	2504.00	None
1480	Forward	None	Full	44.901039	58	722	2829.89	None
17231	Forward	None	Full	33.936200	64	655	5447.31	None
917	Forward	None	Full	13.956481	55	716	9262.22	None

loanID	assets	currentlyServicedLoans	previouslyServicedLoans	loanAmount
520	1054.40	1	0	92000.00
7214	1510.00	0	0	132750.00
1480	0.00	0	0	196837.00
17231	2896.00	0	0	227300.00
917	0.00	1	0	155000.00

loanID	shortage	estimatedProfit	appraisalWaiver	Deposit	cashOutAmount
520	15.00	2.601	None	0.00	None
7214	0.00	2.250	None	0.00	None
1480	-3731.10	4.000	Appraisal Waived	500.00	None
17231	-1900.00	3.375	None	500.00	None
917	-1250.00	4.000	None	0.00	None

loanID	interestRate	treasury10YrYield	dow	monthlyPayment	pmi
520	3.990	1.4531	18432.24	933.99	None
7214	3.875	2.3498	19083.18	912.02	None
1480	3.625	1.7372	18268.50	1270.65	None
17231	3.500	2.4675	19756.85	1320.12	None
917	3.250	1.5800	18400.88	1292.68	None

loanID	lockPeriod	lockLength	loanTerm	requiredPoints	clientPoints
520	UnKnown	NaN	240	-115.00	0.00
7214	UnKnown	NaN	360	-2923.16	0.00
1480	40-Day Commitment	40.0	360	1230.23	0.00
17231	40-Day Commitment	40.0	360	6534.88	6534.88
917	40-Day Commitment	40.0	180	1550.00	1550.00

loanID	originationCharges	thirdPartyCosts	cureCosts	prepaidEscrows
520	1049.00	2496.18	0.00	10.20
7214	-1585.03	5059.11	35.00	843.64
1480	1049.00	1672.36	0.00	1118.89
17231	7583.88	2045.36	0.00	1566.33
917	2599.00	1872.70	0.00	652.86

loanID	escrowedAmount	occupancyType	propertyType	state
520	None	Primary Residence	Single Family	NEW YORK
7214	0.00	Primary Residence	Single Family	FLORIDA
1480	None	Primary Residence	Single Family	OREGON
17231	None	Primary Residence	PUD	CALIFORNIA
917	None	Primary Residence	Single Family	ARIZONA

loanID	county	countyPopulation	zipCode	latitude	longitude	ruralInd
520	ERIE	950265.0	14059	42.8356	78.6390	None
7214	HILLSBOROUGH	998948.0	33594	27.9129	82.2419	None
1480	CLACKAMAS	338391.0	97045	45.3407	122.5771	None
17231	CONTRA COSTA	948816.0	94561	37.9915	121.7145	None
917	MARICOPA	3072149.0	85226	33.3095	111.9288	None

loanID	bankerLocation	bankerSpeciality	bankerTier
520	Detroit - Chase	WB - Refi - CARI	TripleCrown
7214	Third Party	Third Party	None
1480	Detroit - Woodward	WB - Refi - CHAT	None
17231	Cleveland - Higbee	WB - Refi - Power Tenured	None
917	Arizona - 1 North Central	WB - Refi - CARI	None

loanID	bankerTenure	bankerContinuousTenure
520	3893.0	3883.0
7214	NaN	NaN
1480	950.0	946.0
17231	69.0	68.0
917	427.0	425.0

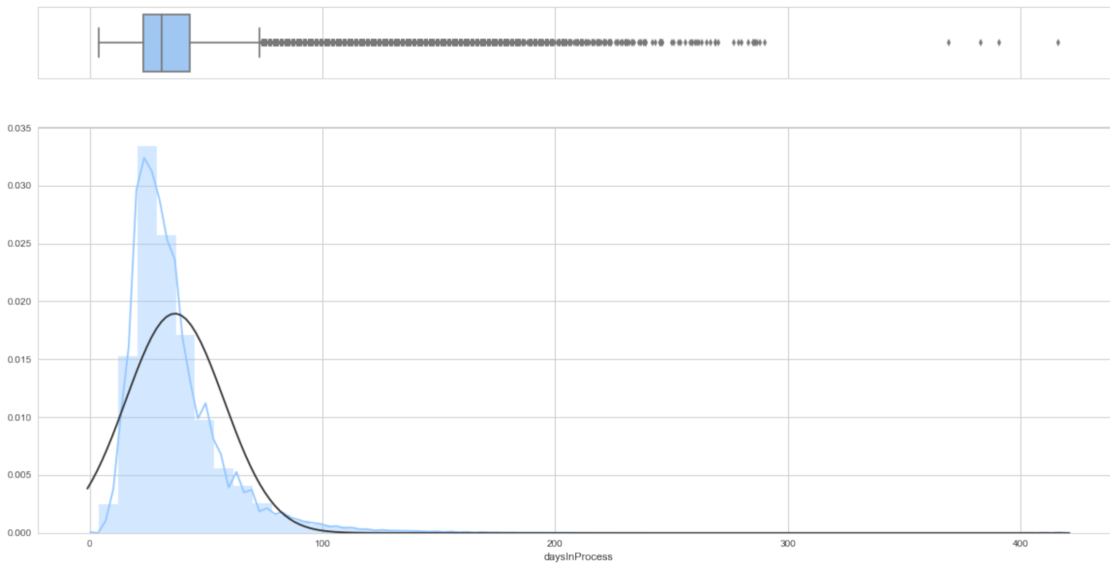
```
In [4]: # Set target, and set categorical vs numeric

        # Exclude Product, county, HELOC, zipCode, lat, long, Banker Location, Banker Speciali

        target = [0]
        cont_index = [1,2,3,4,18,19,20,21,23,24,25,26,27,28,30,31,32,33,34,35,36,38,39,40,41,
                      42,43,44,45,46,51,59,60,]
        cat_index = [6,7,8,9,10,13,15,16,17,22,29,37,47,48,49,55,58,]
        binary = []

In [5]: # Create functions to describe and visualize the data
        def target_desc(data, varname):
            dist_list = [['lognorm', 'Lognormal'], ['invgauss', 'Inverse Gaussian'], ['gamma',

            for distribution in dist_list:
                print('{:*^65}'.format('{} Distributional Fit'.format(distribution[1])))
                dist = getattr(stats, distribution[0])
                if distribution[1] == 'Normal':
                    params = dist.fit(data[varname])
                else:
                    params = dist.fit(data[varname],floc=0)
                data_target_ord = data[varname].sort_values()
                p1 = [(i+0.5)/len(data_target_ord) for i in range(len(data_target_ord))]
                p2 = dist.cdf(data_target_ord,*params[:-2], loc=params[-2], scale=params[-1])
                plt.title('PP-Plot')
                plt.xlabel('Empirical')
                plt.ylabel('Fitted')
                plt.plot(p1,p2)
                plt.xlim([0,1])
                plt.ylim([0,1])
                plt.show()

                print('{} Kolmogorov-Smirnov Test:'.format(distribution[1]))
                print(stats.kstest(data_target_ord,distribution[0],[*params[:-2], params[-2], p

                n, bins, patches = plt.hist(data_target_ord, 100, normed=True, color='g', alpha
                y = dist.pdf(bins, *params[:-2], loc=params[-2], scale=params[-1])
                plt.title('Empirical vs. Fitted PDF')
                plt.plot(bins, y, 'r-')
                plt.ylim([0,max(n)])
                plt.show()

        le = preprocessing.LabelEncoder()

        def cat_desc(data, varname):
            data_count = data.groupby(by=[varname])[varname].count().sort_values(ascending = Fa
            data_prop = data_count / n_rows
            data[[varname]] = data[[varname]].fillna('zNULL')
```

```
    if len(data_prop) == 1:
        print('{} contains a single level\n'.format(varname))
    else:
        data_prop.index = data_prop.index.map(str)
        for idx in data_prop.index:
            if data_prop[idx] < 0.005:
                print('{} has thin data in {}: {} out of {} ({:.2g})\n'.format(varname

    if data_count.index.size <= 50:
        for j, idx in enumerate(data_count.index):
            if j == 0:
                data_count = data_count.reset_index(drop=True)
                print('{0:50} {1}'.format('Level','Weight'))
            if j <= 50:
                print('{0:50} {1}'.format(str(idx),data_count[j]))

        fig,ax = plt.subplots(figsize=(6, 4))
        ax = sns.countplot(x=varname, data=data)
        ax.set(ylabel='Count')
        plt.title('Loan Counts by '+varname, fontsize=16)
        plt.xticks(rotation=90)
        fig.set_figheight(10)
        fig.set_figwidth(20)
        plt.show()

    if data_count.index.size == 1:
        binary.append(data.columns.get_loc(varname))

    data[[varname]] = le.fit_transform(data[varname].ravel())


def num_desc(data, varname):
    new_df = data[pd.notnull(data[varname])]

    fig, (ax_box, ax_hist) = plt.subplots(2, figsize=(6,4), sharex=True, gridspec_kw={
    # Add a graph in each part
    sns.boxplot(new_df[varname], ax=ax_box)
    sns.distplot(new_df[varname], ax=ax_hist, fit=stats.norm, kde=True)
    # Remove x axis name for the boxplot
    ax_box.set(xlabel='')
    fig.set_figheight(10)
    fig.set_figwidth(20)
    plt.show()


def print_desc(data, varname):
    print('\n')
```

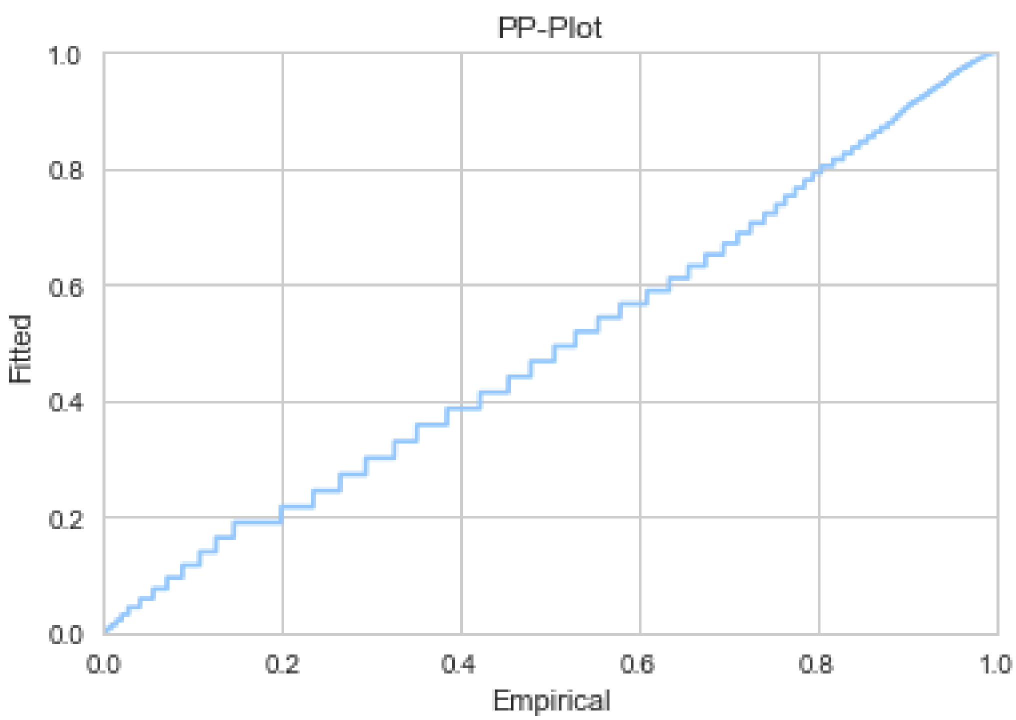
```
        print('EDA for: '+varname + '\n')
        print(data[varname].describe())
        n_miss = np.count_nonzero(data[varname].isnull().values)
        print('Missing Values: ' + str(n_miss) + '\n')


    def all_desc(data, varname):
        print_desc(data, varname)
        if data.columns.get_loc(varname) in target:
            if is_numeric_dtype(data[varname]):
                data[varname] = pd.to_numeric(data[varname])
                num_desc(data, varname)
                target_desc(data, varname)
            else:
                cat_desc(data, varname)
        elif data.columns.get_loc(varname) in cont_index:
            data[varname] = pd.to_numeric(data[varname])
            num_desc(data, varname)
        elif data.columns.get_loc(varname) in cat_index:
            cat_desc(data, varname)


    def desc_df(data):
        all_desc(data, loans.columns[target].values.any())
        for i, col in enumerate(data.columns):
            if i in np.concatenate((cont_index, cat_index), axis=0):
                if i not in target:
                    all_desc(data, col)
```

```
In [6]: desc_df(loans)
```

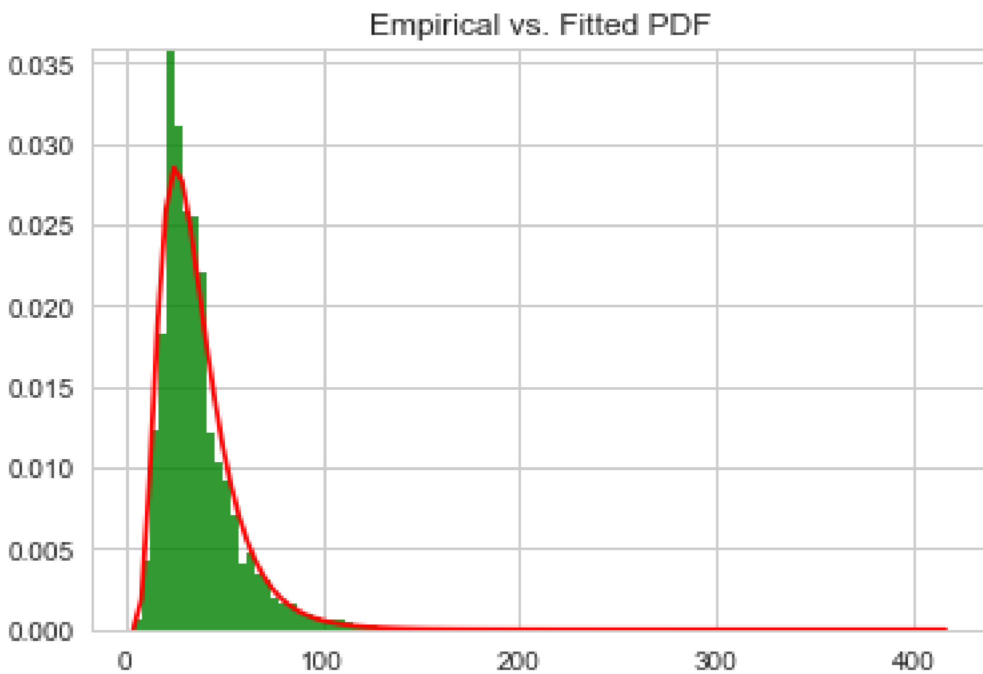
```
EDA for: daysInProcess

count    296268.000000
mean         36.575999
std          21.048530
min           4.000000
25%          23.000000
50%          31.000000
75%          43.000000
max         416.000000
Name: daysInProcess, dtype: float64
Missing Values: 0
```



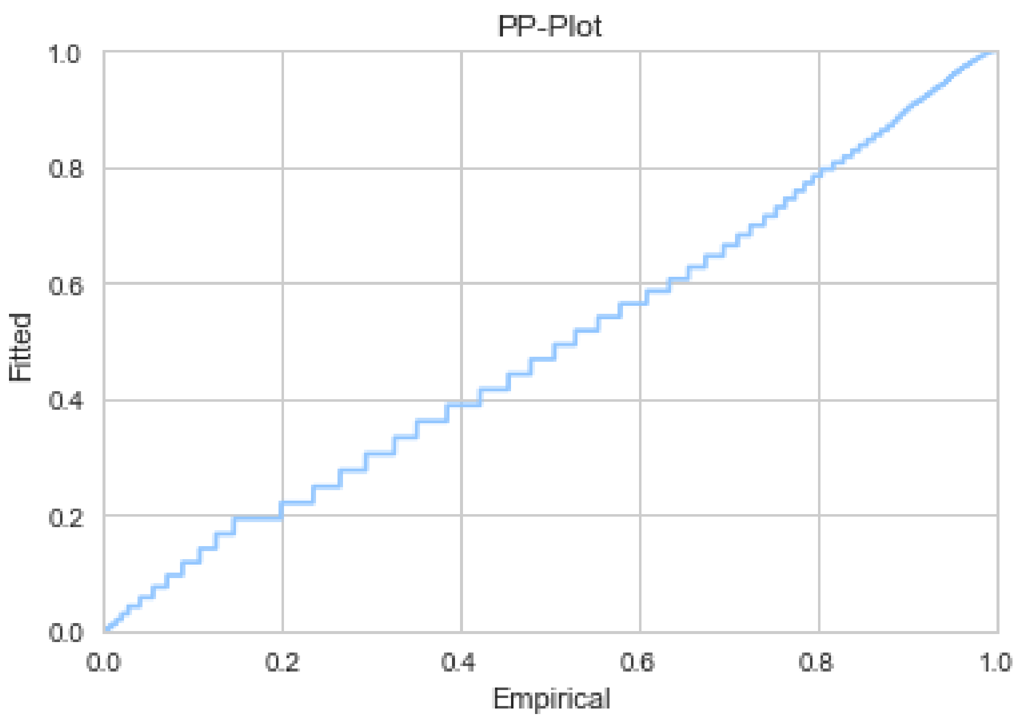
******************Lognormal Distributional Fit*******************



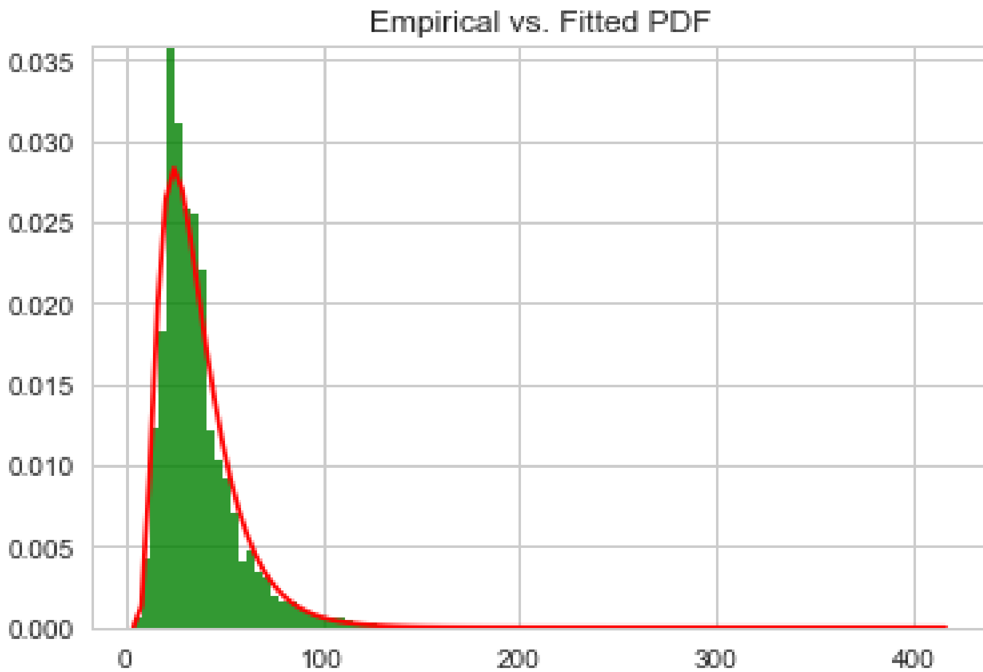
Lognormal Kolmogorov-Smirnov Test:
0.044182713945



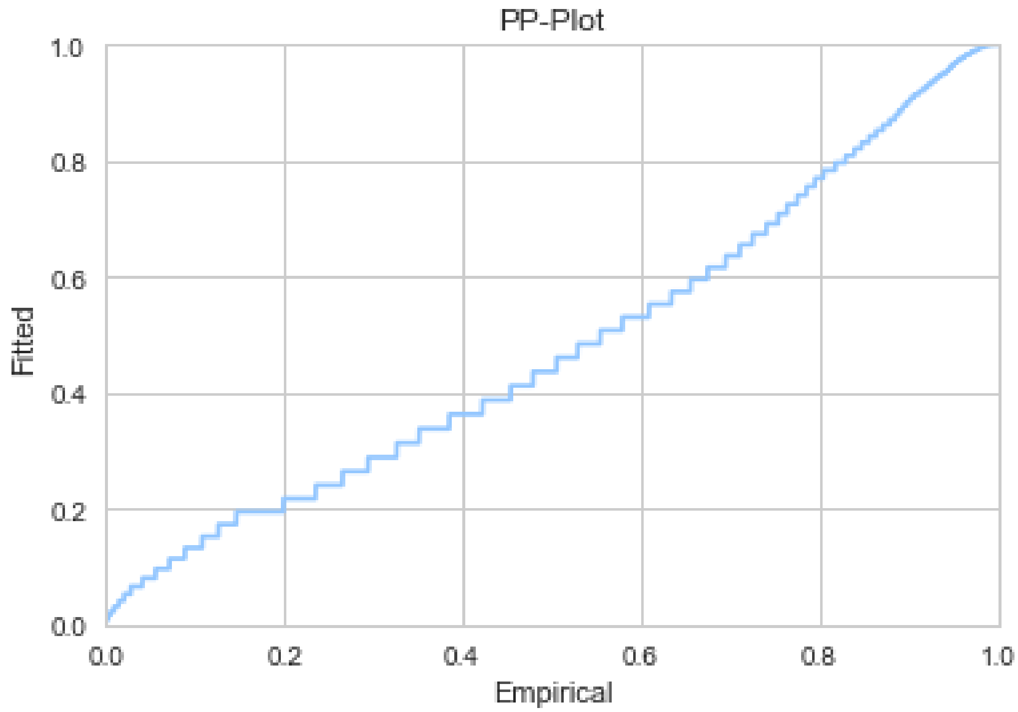
***************Inverse Gaussian Distributional Fit***************



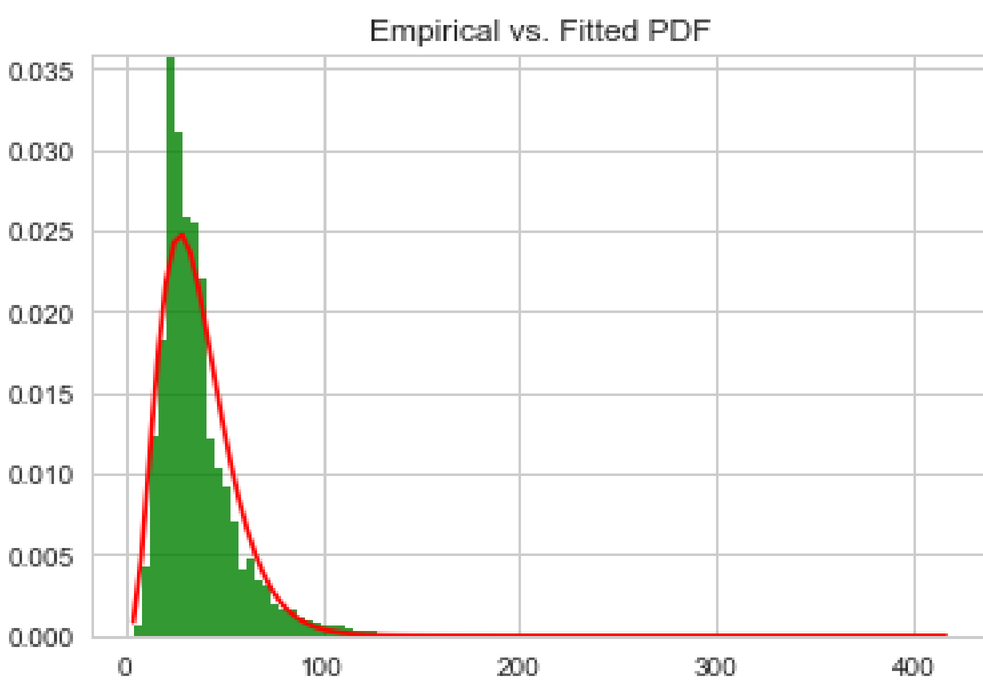
Inverse Gaussian Kolmogorov-Smirnov Test:
0.0481900642596



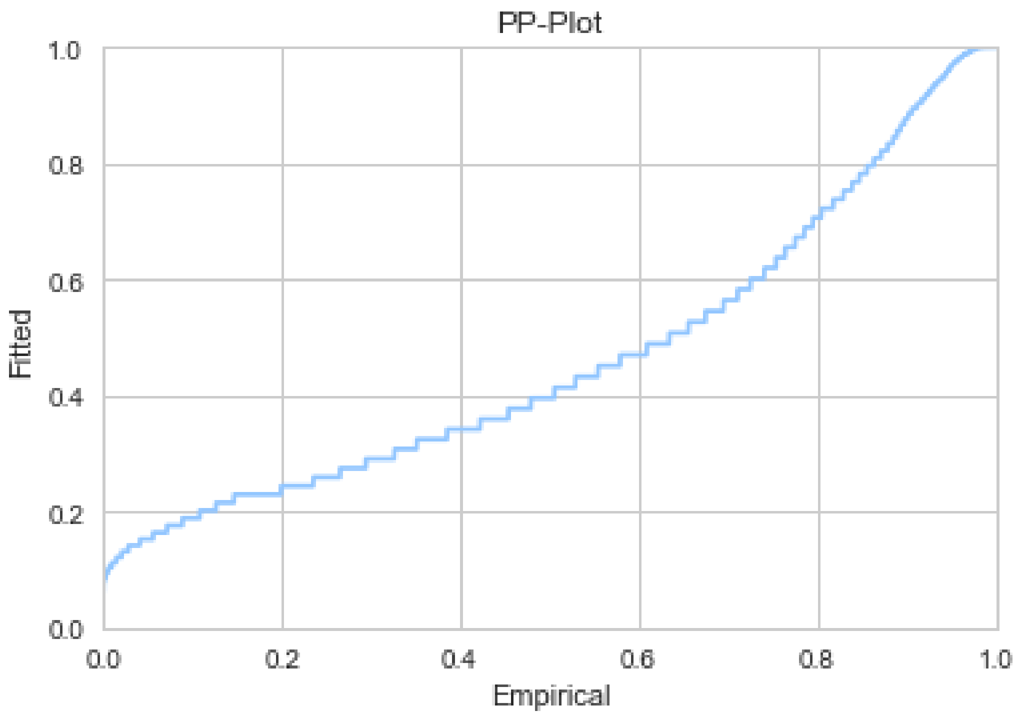
********************Gamma Distributional Fit*********************



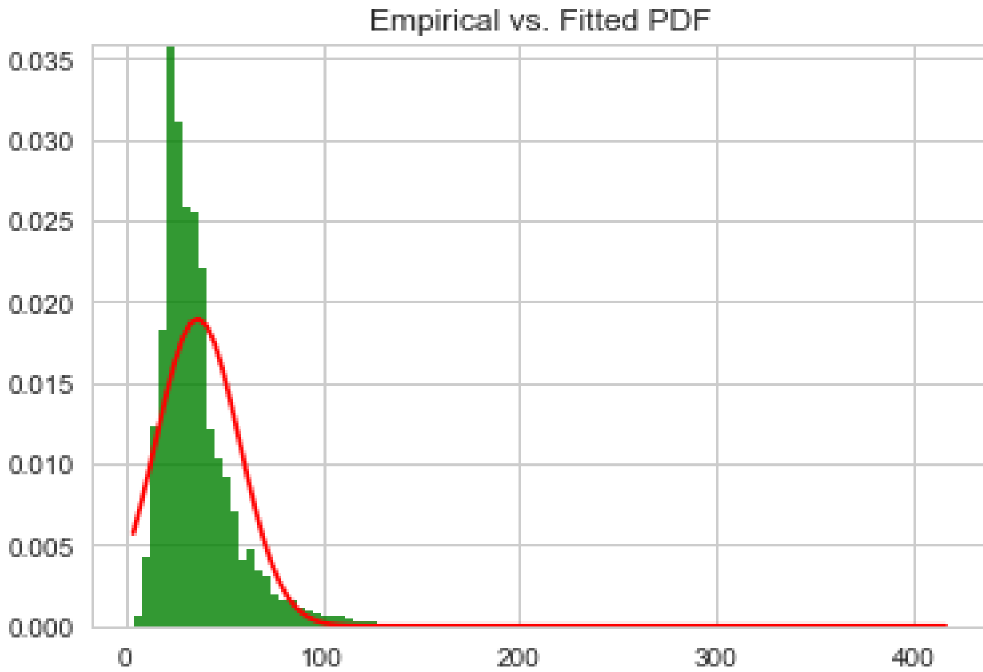

```
Gamma Kolmogorov-Smirnov Test:
0.0806152243123
```



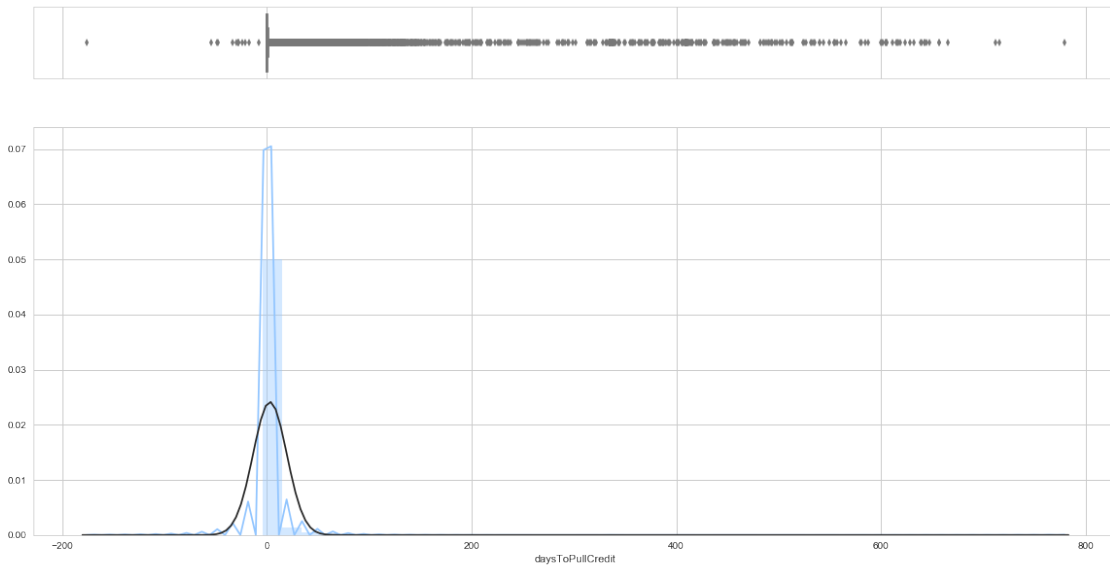
```
********************Normal Distributional Fit********************
```



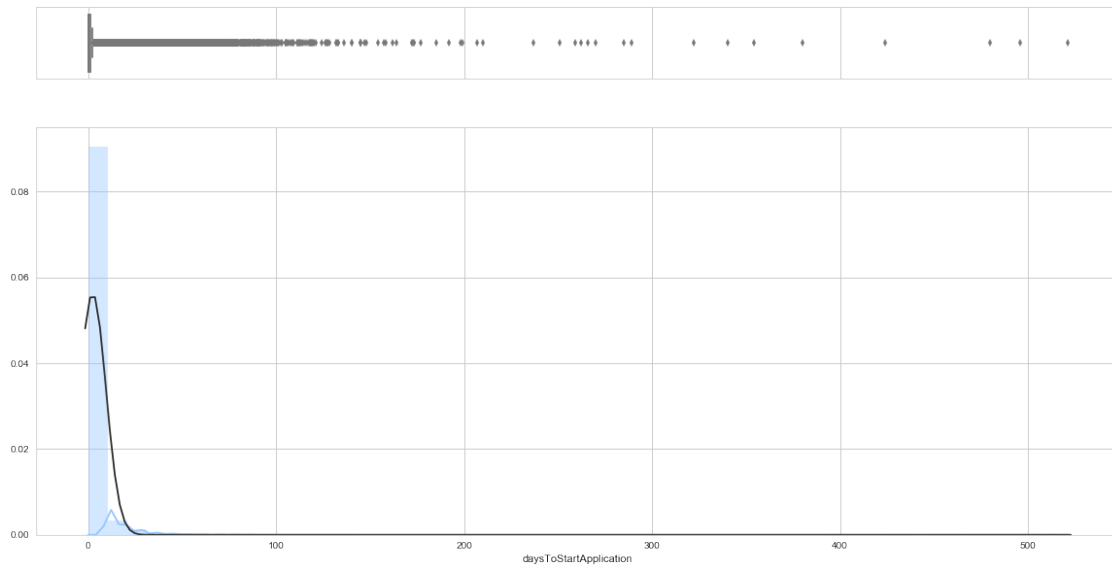

```
Normal Kolmogorov-Smirnov Test:
0.14823237028
```



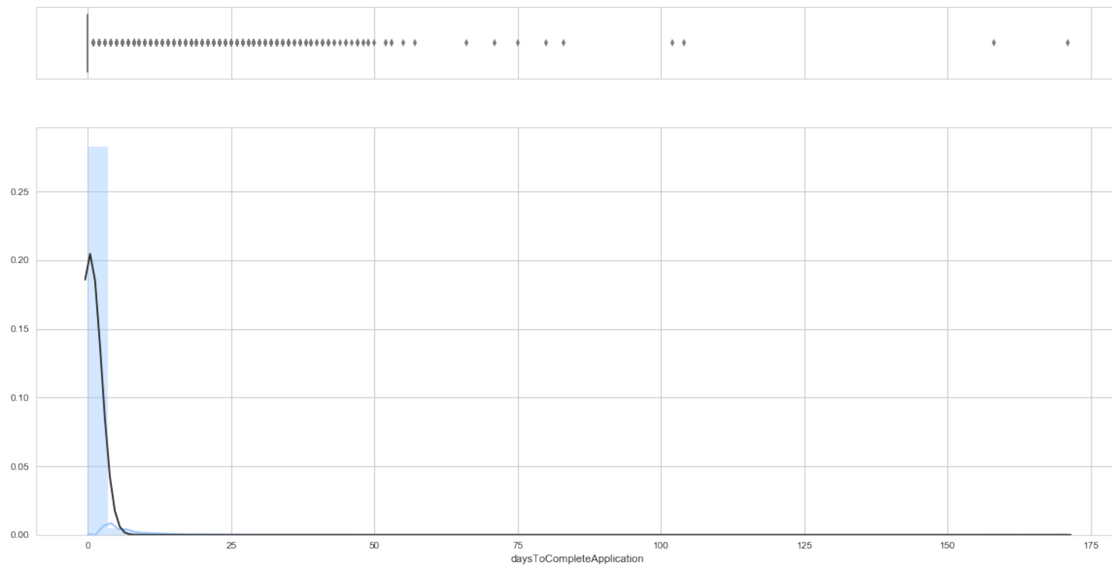

```
EDA for: daysToPullCredit

count    284449.000000
mean          3.092997
std          16.511423
min        -176.000000
25%           0.000000
50%           0.000000
75%           1.000000
max         779.000000
Name: daysToPullCredit, dtype: float64
Missing Values: 11819
```



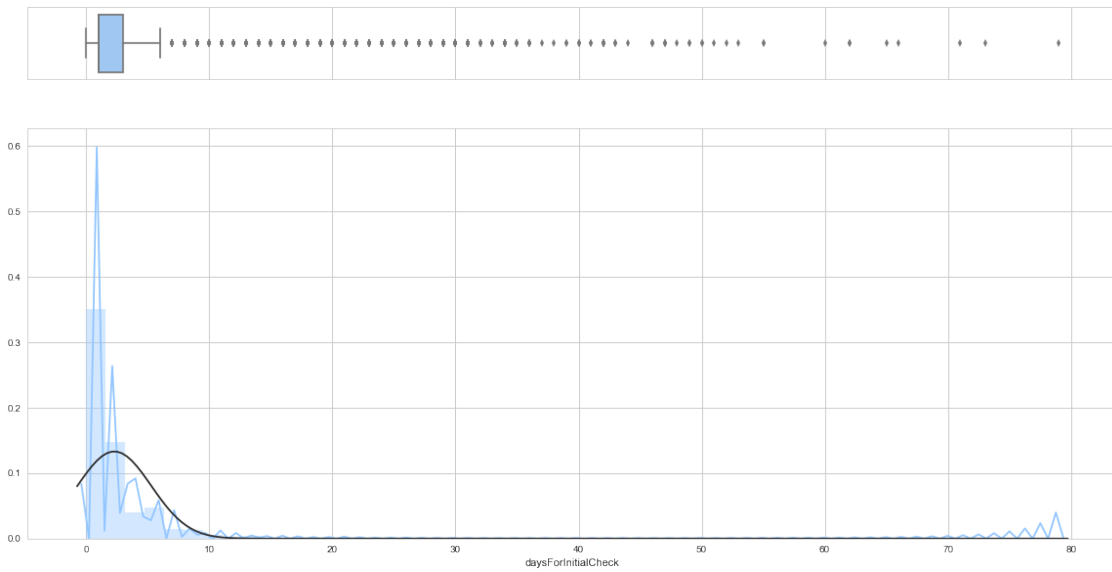

```
EDA for: daysToStartApplication

count    284449.000000
mean          2.254647
std           7.081166
min           0.000000
25%           0.000000
50%           0.000000
75%           1.000000
max         521.000000
Name: daysToStartApplication, dtype: float64
Missing Values: 11819
```




```
EDA for: daysToCompleteApplication

count    296268.000000
mean          0.387264
std           1.949410
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max         171.000000
Name: daysToCompleteApplication, dtype: float64
Missing Values: 0
```

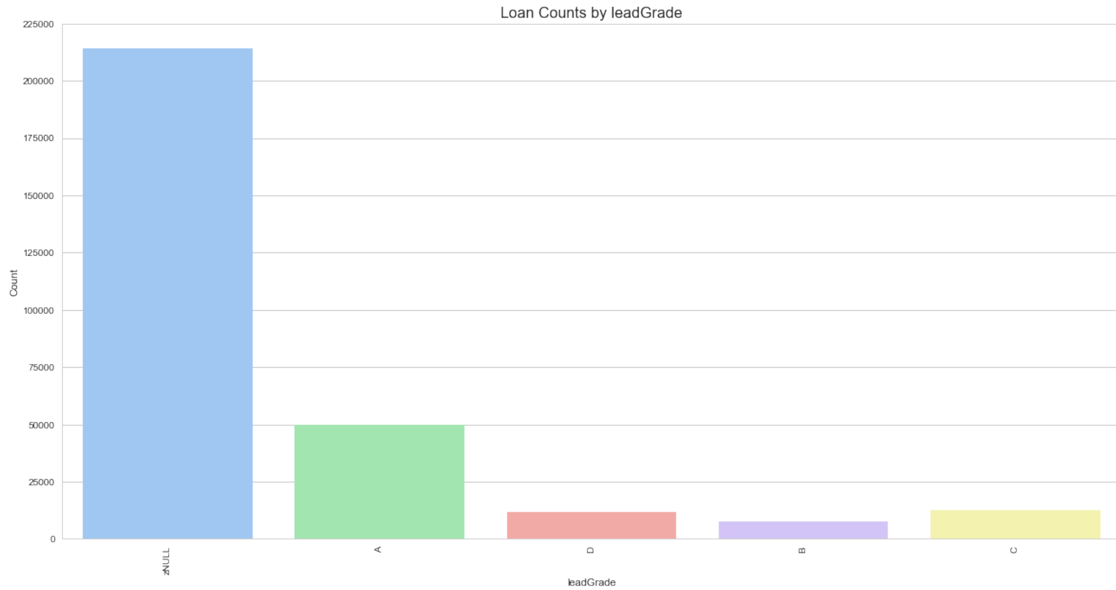



```
EDA for: daysForInitialCheck

count    296268.000000
mean          2.302966
std           2.993631
min           0.000000
25%           1.000000
50%           1.000000
75%           3.000000
max          79.000000
Name: daysForInitialCheck, dtype: float64
Missing Values: 0
```




```
EDA for: leadGrade

count     82040
unique        4
top           A
freq      49978
Name: leadGrade, dtype: object
Missing Values: 214228
```

Level	Weight
A	49978
C	12609
D	11650
B	7803

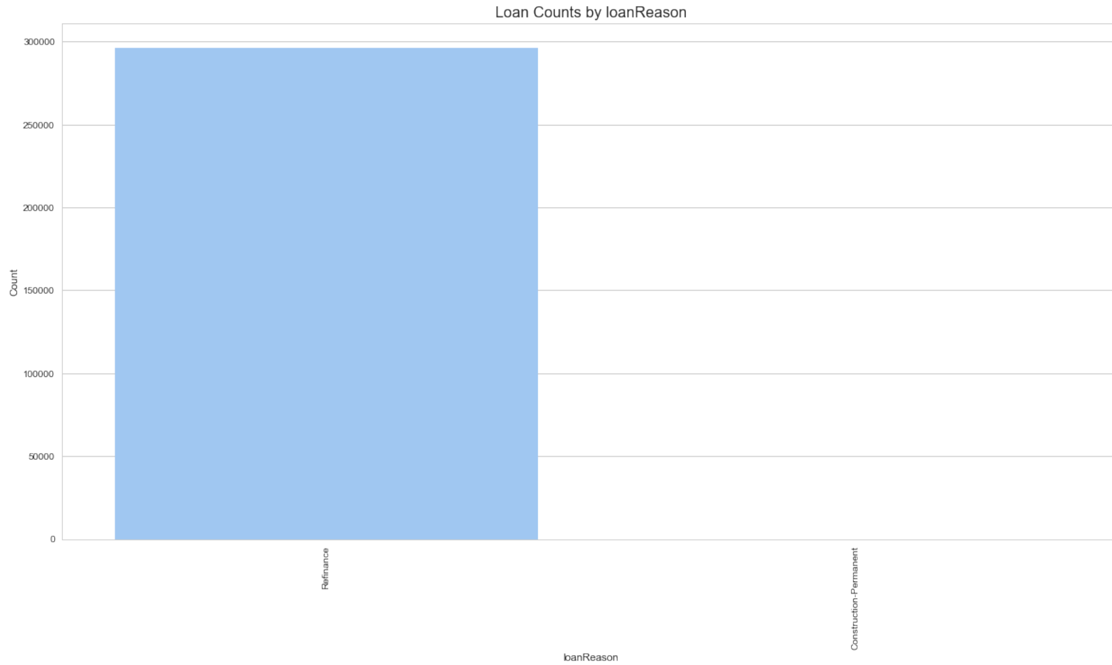



```
EDA for: leadSource

count                296268
unique                   63
top       Quickenloans.com
freq                  47525
Name: leadSource, dtype: object
Missing Values: 0

leadSource has thin data in FreeRateUpdate.com: 1324 out of 296268 (0.0045)

leadSource has thin data in Lakewood: 1071 out of 296268 (0.0036)

leadSource has thin data in Seterus: 923 out of 296268 (0.0031)

leadSource has thin data in MortgageInsiders.com: 769 out of 296268 (0.0026)

leadSource has thin data in Brokermatch.com: 769 out of 296268 (0.0026)

leadSource has thin data in HGTV.com: 533 out of 296268 (0.0018)

leadSource has thin data in Flagstar: 494 out of 296268 (0.0017)

leadSource has thin data in SmartQuote.com: 465 out of 296268 (0.0016)

leadSource has thin data in Re-Marketing: 454 out of 296268 (0.0015)
```

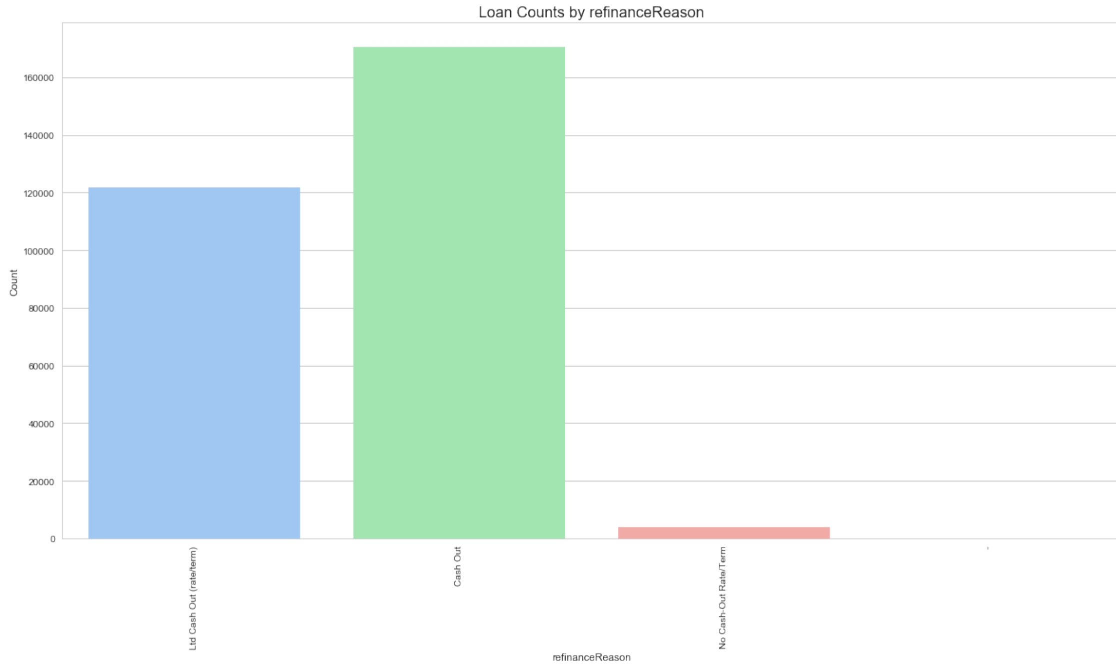
```
leadSource has thin data in Banker Transfer: 352 out of 296268 (0.0012)

leadSource has thin data in Ally Bank: 333 out of 296268 (0.0011)

leadSource has thin data in LeadCloud: 314 out of 296268 (0.0011)

leadSource has thin data in Home Loan Benefit Program: 280 out of 296268 (0.00095)

leadSource has thin data in Sweepstakes: 266 out of 296268 (0.0009)

leadSource has thin data in Revi Media: 220 out of 296268 (0.00074)

leadSource has thin data in ReallyGreatRate.com: 205 out of 296268 (0.00069)

leadSource has thin data in Direct Mail: 130 out of 296268 (0.00044)

leadSource has thin data in Quicken Loans - Repurposed: 126 out of 296268 (0.00043)

leadSource has thin data in CHASE: 123 out of 296268 (0.00042)

leadSource has thin data in Sales Force: 115 out of 296268 (0.00039)

leadSource has thin data in Quizzle.com: 114 out of 296268 (0.00038)

leadSource has thin data in Email: 109 out of 296268 (0.00037)

leadSource has thin data in QLCredit: 102 out of 296268 (0.00034)

leadSource has thin data in BestRateReferral.com: 88 out of 296268 (0.0003)

leadSource has thin data in BoostUp: 80 out of 296268 (0.00027)

leadSource has thin data in MediaForce: 52 out of 296268 (0.00018)

leadSource has thin data in Deck Hand: 48 out of 296268 (0.00016)

leadSource has thin data in The Lenders Network: 45 out of 296268 (0.00015)

leadSource has thin data in TPCARI: 40 out of 296268 (0.00014)

leadSource has thin data in MLS Trigger: 18 out of 296268 (6.1e-05)

leadSource has thin data in Saxum Partners: 16 out of 296268 (5.4e-05)

leadSource has thin data in Pingora: 15 out of 296268 (5.1e-05)

leadSource has thin data in SMHL: 9 out of 296268 (3e-05)
```

```
leadSource has thin data in First Tennessee: 7 out of 296268 (2.4e-05)

leadSource has thin data in Facebook: 6 out of 296268 (2e-05)

leadSource has thin data in Homes.com: 6 out of 296268 (2e-05)

leadSource has thin data in Zing Blog Artical - Q1 2017: 6 out of 296268 (2e-05)

leadSource has thin data in Other: 5 out of 296268 (1.7e-05)

leadSource has thin data in Bankrate.com: 2 out of 296268 (6.8e-06)

leadSource has thin data in MyPerfectHome: 2 out of 296268 (6.8e-06)

leadSource has thin data in ReferXpress: 1 out of 296268 (3.4e-06)

leadSource has thin data in OneSource Relocation: 1 out of 296268 (3.4e-06)

leadSource has thin data in Lead Buy Provider: 1 out of 296268 (3.4e-06)

leadSource has thin data in NRI Relocation: 1 out of 296268 (3.4e-06)


EDA for: loanReason

count        296268
unique            2
top        Refinance
freq         296267
Name: loanReason, dtype: object
Missing Values: 0

loanReason has thin data in Construction-Permanent: 1 out of 296268 (3.4e-06)
```

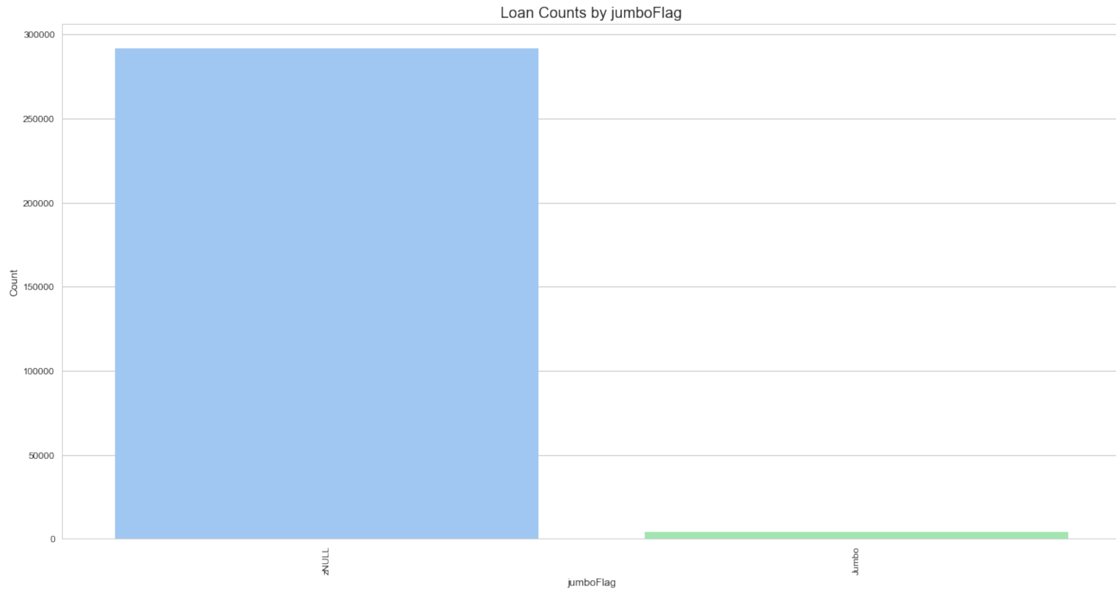
Level	Weight
Refinance	296267
Construction-Permanent	1




```
EDA for: refinanceReason

count        296268
unique            4
top        Cash Out
freq         170446
Name: refinanceReason, dtype: object
Missing Values: 0

refinanceReason has thin data in -: 1 out of 296268 (3.4e-06)

Level                                                 Weight
Cash Out                                              170446
Ltd Cash Out (rate/term)                              121906
No Cash-Out Rate/Term                                 3915
-                                                     1
```

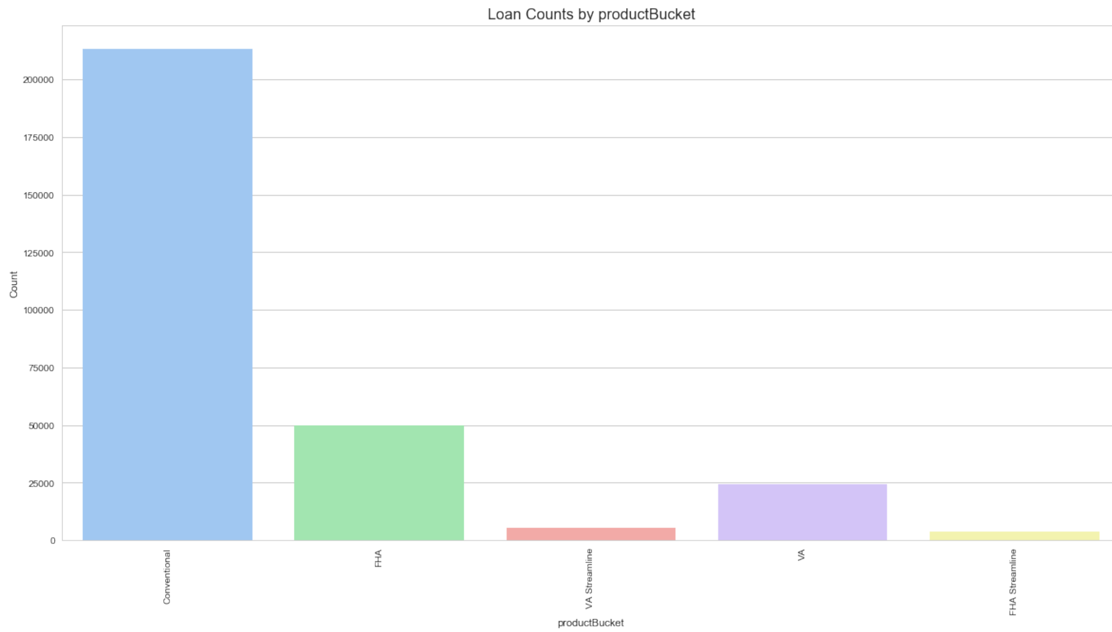



```
EDA for: jumboFlag

count       4302
unique         1
top        Jumbo
freq        4302
Name: jumboFlag, dtype: object
Missing Values: 291966

jumboFlag contains a single level

Level                                                Weight
Jumbo                                                4302
```




```
EDA for: productBucket

count           296268
unique               5
top       Conventional
freq            213031
Name: productBucket, dtype: object
Missing Values: 0
```

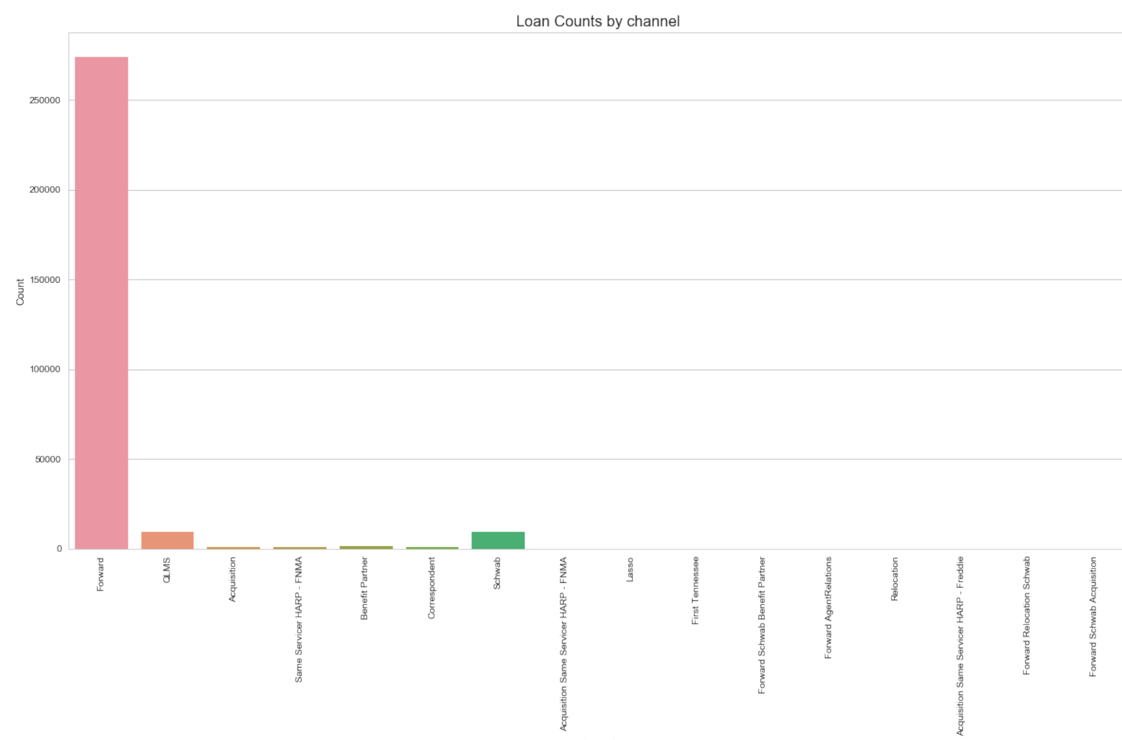
Level	Weight
Conventional	213031
FHA	49798
VA	24306
VA Streamline	5434
FHA Streamline	3699



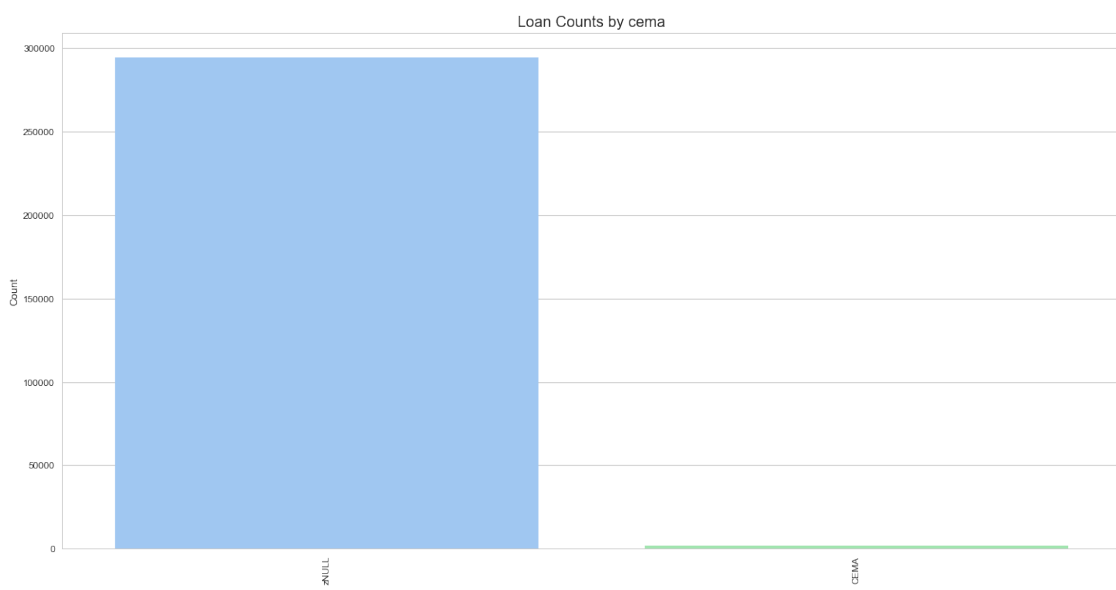

```
EDA for: channel

count      296268
unique         16
top       Forward
freq       273890
Name: channel, dtype: object
Missing Values: 0

channel has thin data in Same Servicer HARP - FNMA: 929 out of 296268 (0.0031)

channel has thin data in Acquisition: 811 out of 296268 (0.0027)

channel has thin data in Correspondent: 569 out of 296268 (0.0019)

channel has thin data in Forward AgentRelations: 22 out of 296268 (7.4e-05)

channel has thin data in Lasso: 17 out of 296268 (5.7e-05)

channel has thin data in Acquisition Same Servicer HARP - FNMA: 9 out of 296268 (3e-05)

channel has thin data in First Tennessee: 8 out of 296268 (2.7e-05)

channel has thin data in Acquisition Same Servicer HARP - Freddie: 6 out of 296268 (2e-05)
```

```
channel has thin data in Forward Schwab Benefit Partner: 2 out of 296268 (6.8e-06)

channel has thin data in Relocation: 1 out of 296268 (3.4e-06)

channel has thin data in Forward Schwab Acquisition: 1 out of 296268 (3.4e-06)

channel has thin data in Forward Relocation Schwab: 1 out of 296268 (3.4e-06)
```

Level	Weight
Forward	273890
QLMS	9451
Schwab	9046
Benefit Partner	1505
Same Servicer HARP - FNMA	929
Acquisition	811
Correspondent	569
Forward AgentRelations	22
Lasso	17
Acquisition Same Servicer HARP - FNMA	9
First Tennessee	8
Acquisition Same Servicer HARP - Freddie	6
Forward Schwab Benefit Partner	2
Relocation	1
Forward Schwab Acquisition	1
Forward Relocation Schwab	1



```
EDA for: cema

count     1882
unique       1
top       CEMA
freq      1882
Name: cema, dtype: object
Missing Values: 294386

cema contains a single level

Level                                                 Weight
CEMA                                                  1882
```




```
EDA for: docType

count     296268
unique         7
top         Full
freq      285259
```

Name: docType, dtype: object
Missing Values: 0

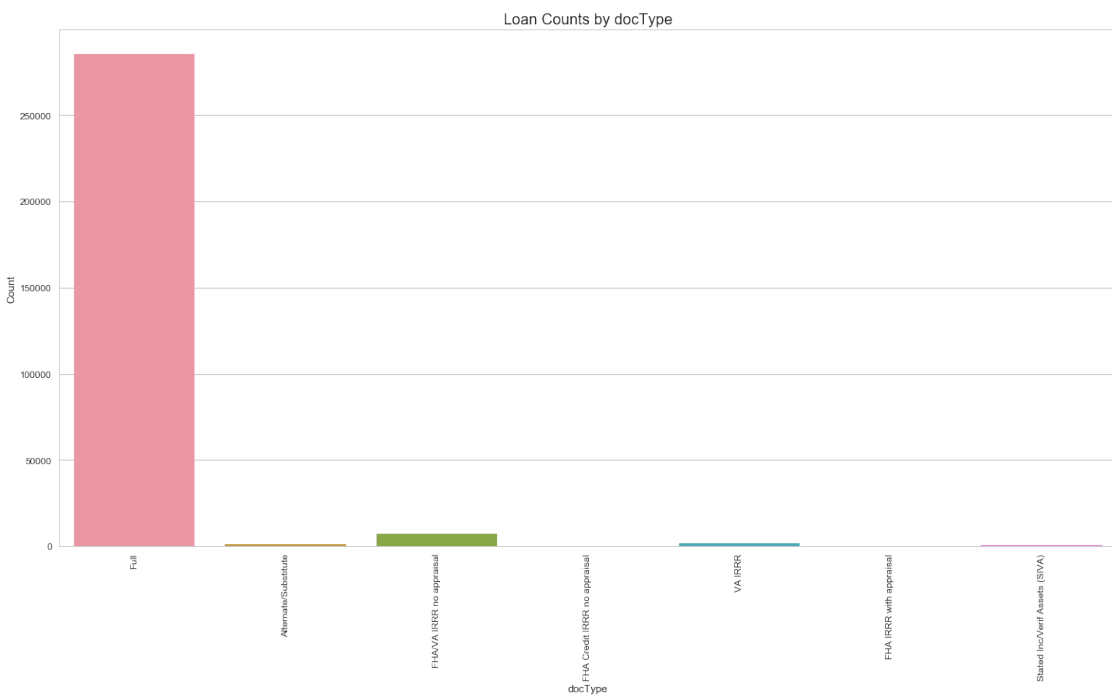
docType has thin data in Alternate/Substitute: 1022 out of 296268 (0.0034)

docType has thin data in Stated Inc/Verif Assets (SIVA): 854 out of 296268 (0.0029)

docType has thin data in FHA Credit IRRR no appraisal: 209 out of 296268 (0.00071)

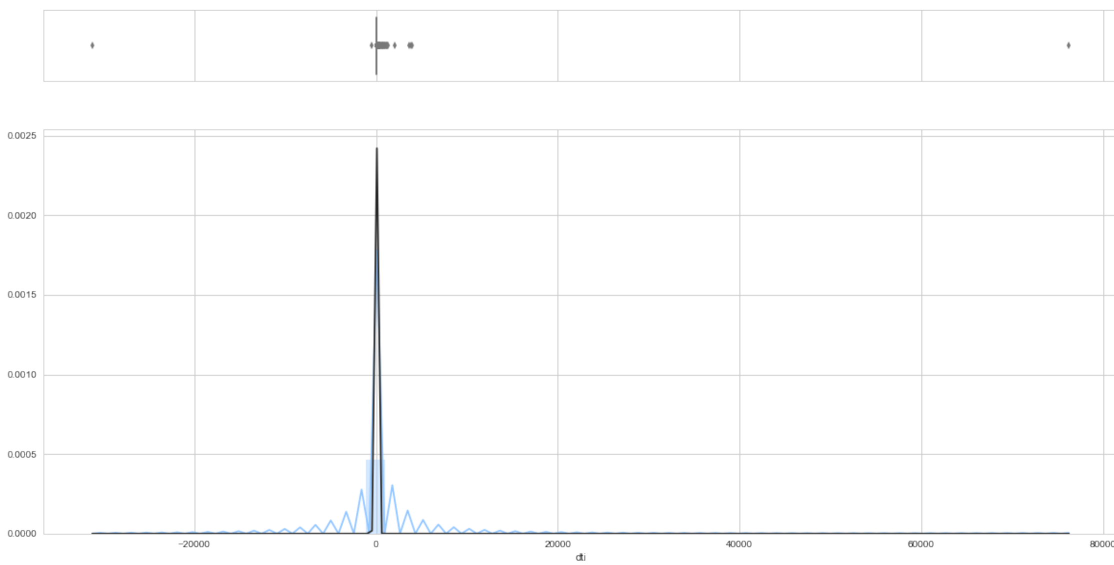
docType has thin data in FHA IRRR with appraisal: 18 out of 296268 (6.1e-05)

Level	Weight
Full	285259
FHA/VA IRRR no appraisal	7302
VA IRRR	1604
Alternate/Substitute	1022
Stated Inc/Verif Assets (SIVA)	854
FHA Credit IRRR no appraisal	209
FHA IRRR with appraisal	18

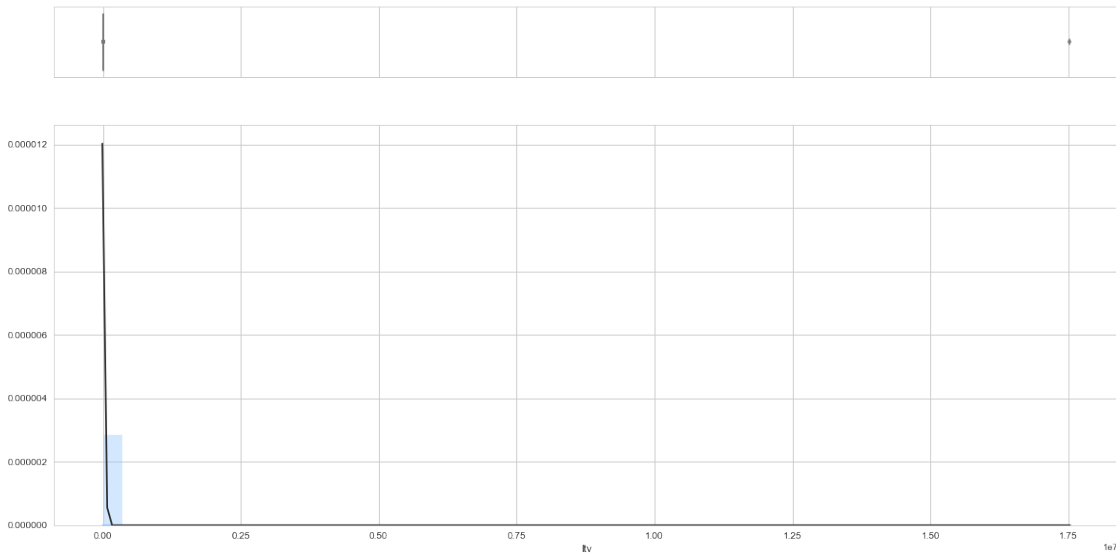


EDA for: dti

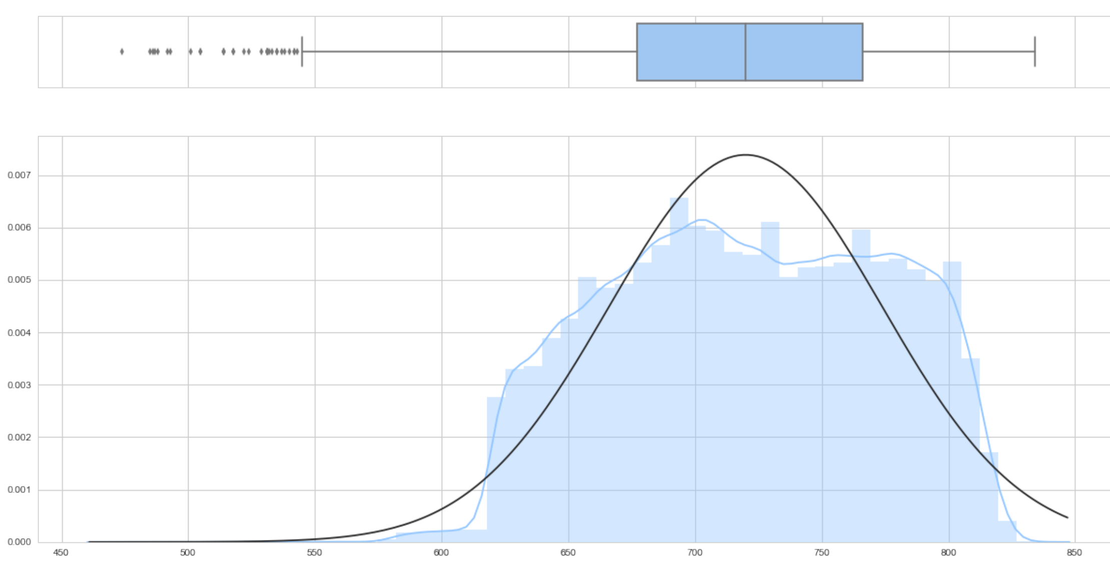
```
count    296268.000000
mean         35.725531
std         152.399824
min      -31187.736645
25%          28.607335
50%          36.730605
75%          43.433152
max       76200.792254
Name: dti, dtype: float64
Missing Values: 0
```



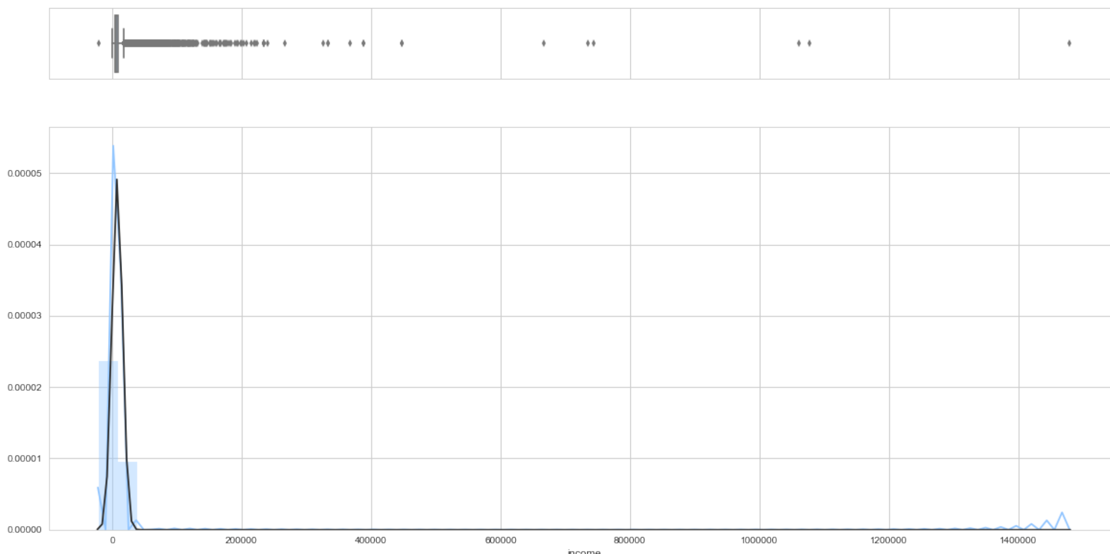

```
EDA for: ltv

count     296268
unique       192
top           80
freq       27348
Name: ltv, dtype: object
Missing Values: 0
```




```
EDA for: fico

count     296268
unique       307
top          700
freq        1988
Name: fico, dtype: object
Missing Values: 0
```

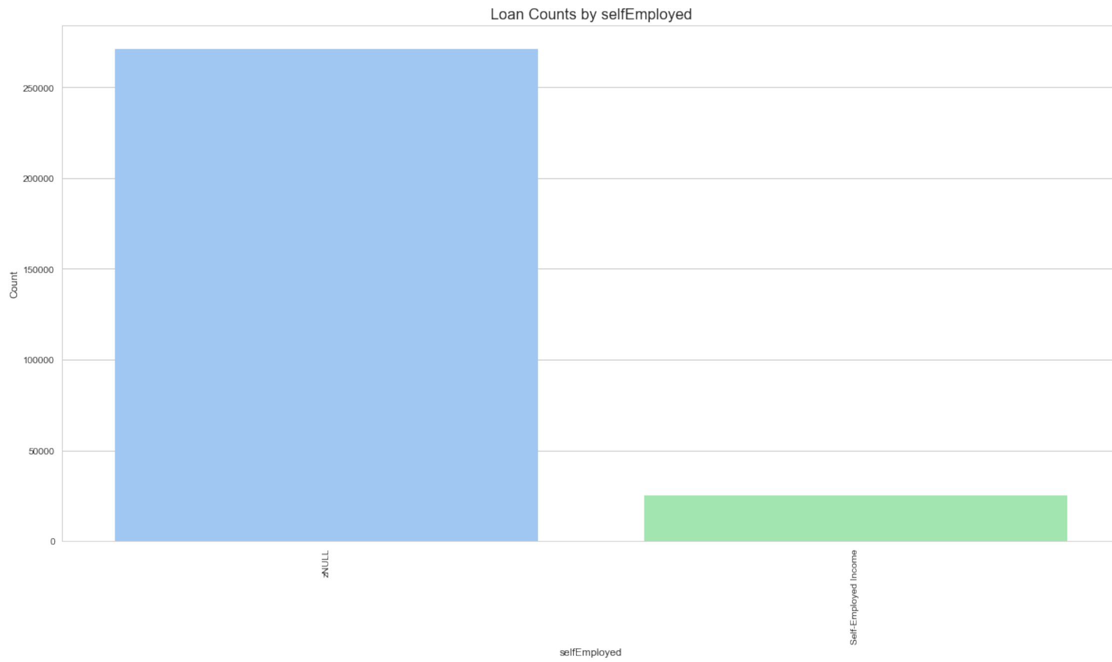


```
EDA for: income

count     296268
unique    207496
top         0.00
freq        6002
Name: income, dtype: object
Missing Values: 0
```

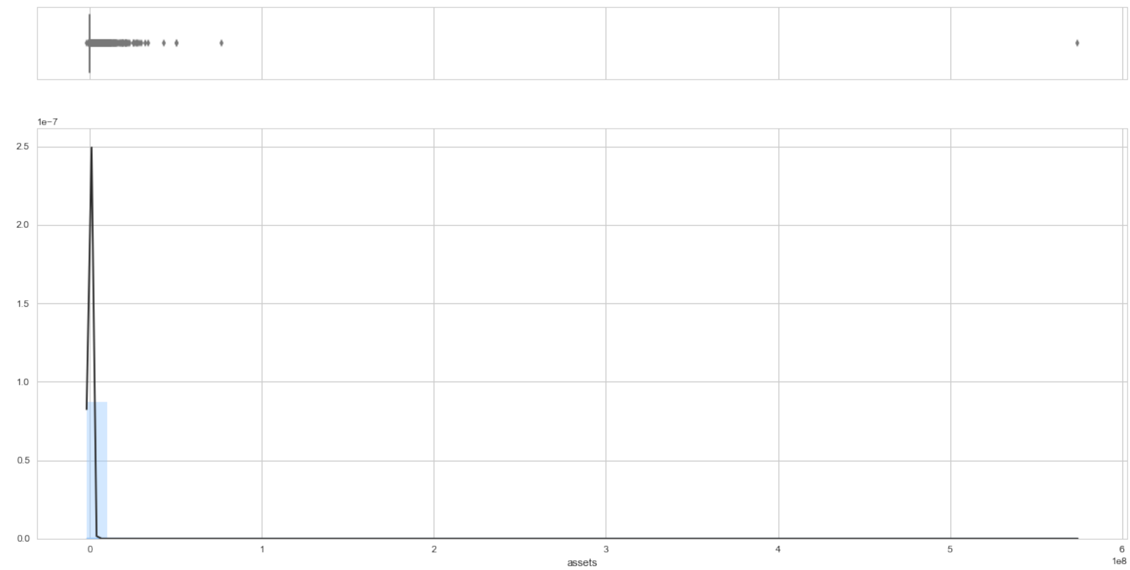



```
EDA for: selfEmployed

count                   25413
unique                      1
top       Self-Employed Income
freq                    25413
Name: selfEmployed, dtype: object
Missing Values: 270855

selfEmployed contains a single level
```

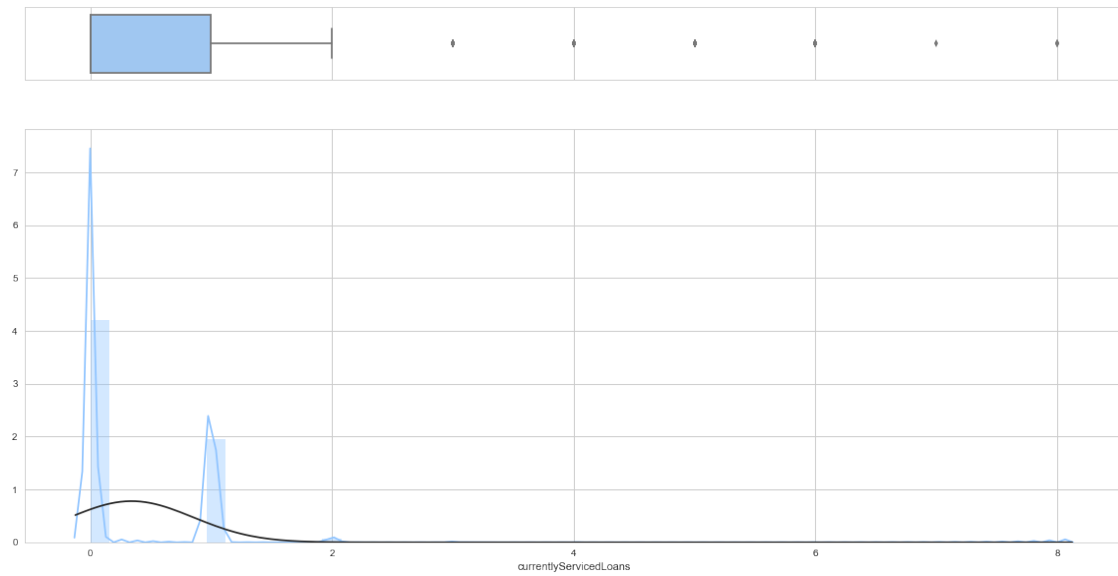
```
Level                                                Weight
Self-Employed Income                                 25413
```



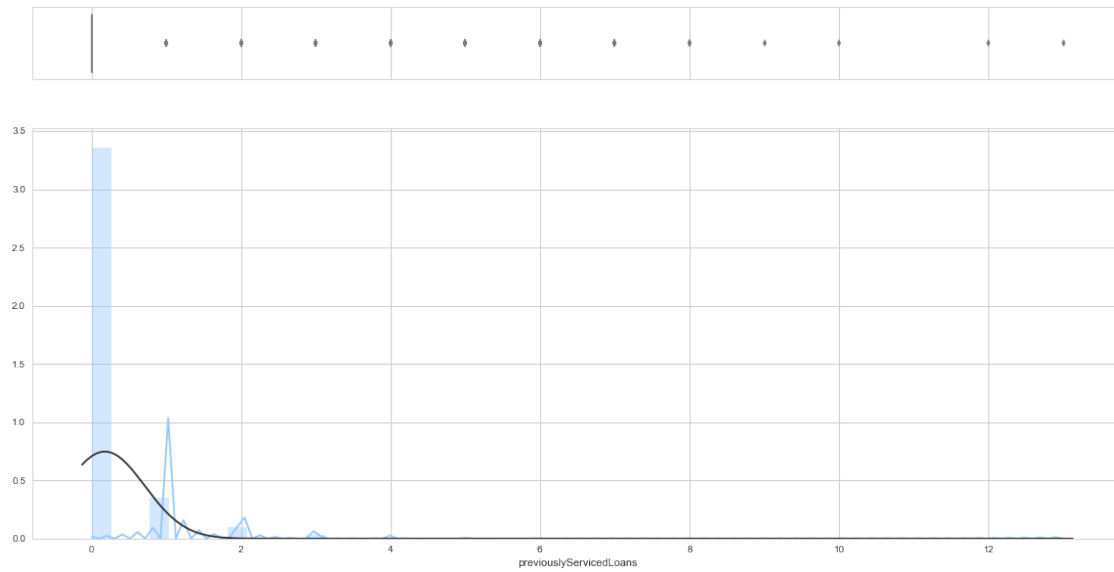

```
EDA for: assets

count     296265
unique     81938
top         0.00
freq      188735
Name: assets, dtype: object
Missing Values: 3
```



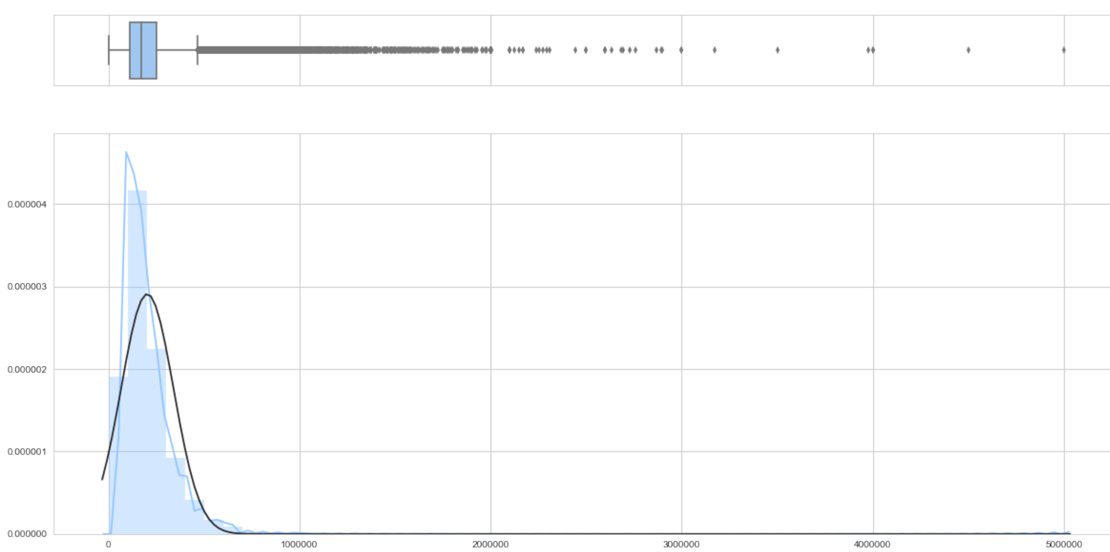

```
EDA for: currentlyServicedLoans

count    296268.000000
mean          0.341576
std           0.511916
min           0.000000
25%           0.000000
50%           0.000000
75%           1.000000
max           8.000000
Name: currentlyServicedLoans, dtype: float64
Missing Values: 0
```



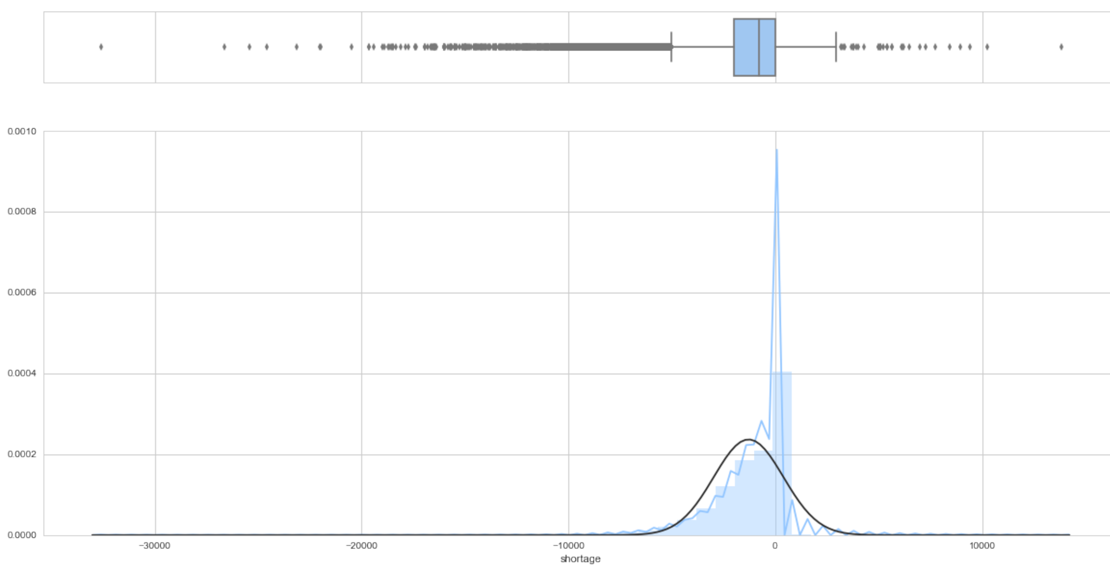

```
EDA for: previouslyServicedLoans

count    296268.000000
mean          0.176182
std           0.533255
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max          13.000000
Name: previouslyServicedLoans, dtype: float64
Missing Values: 0
```



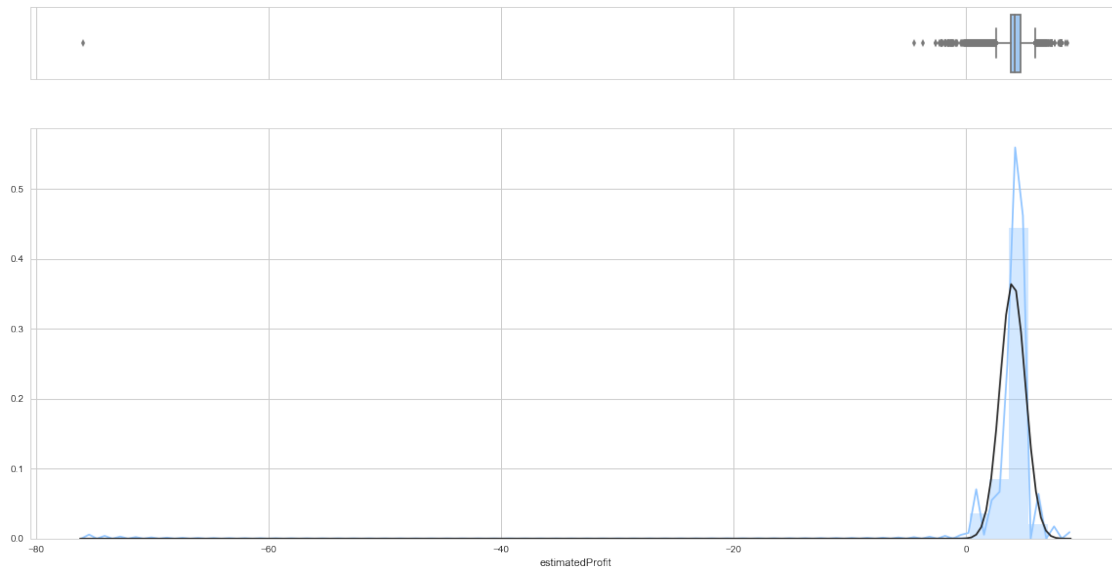

```
EDA for: loanAmount

count        296268
unique        95680
top        75000.00
freq           3115
Name: loanAmount, dtype: object
Missing Values: 0
```



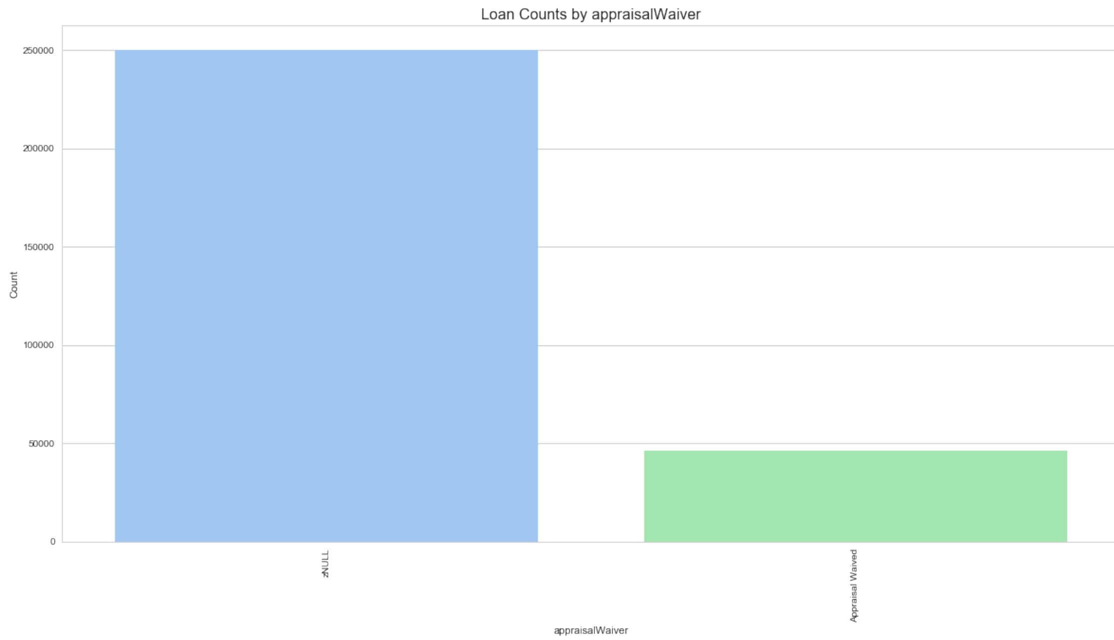
```
EDA for: shortage

count     296268
unique     88491
top         0.00
freq       89534
Name: shortage, dtype: object
Missing Values: 0
```



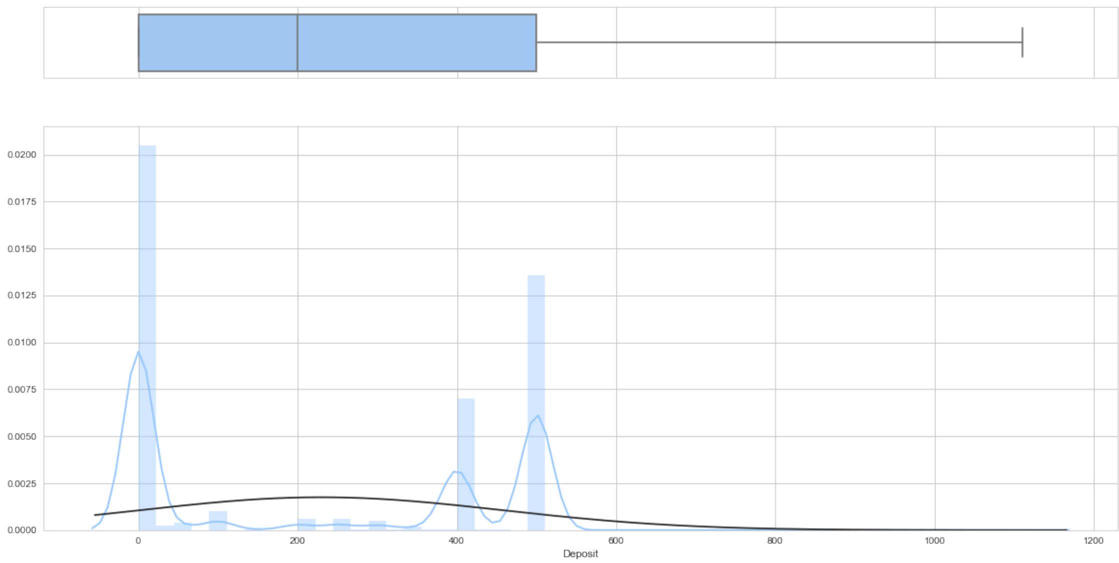

```
EDA for: estimatedProfit

count     296268
unique      4932
top        3.875
freq       28786
Name: estimatedProfit, dtype: object
Missing Values: 0
```



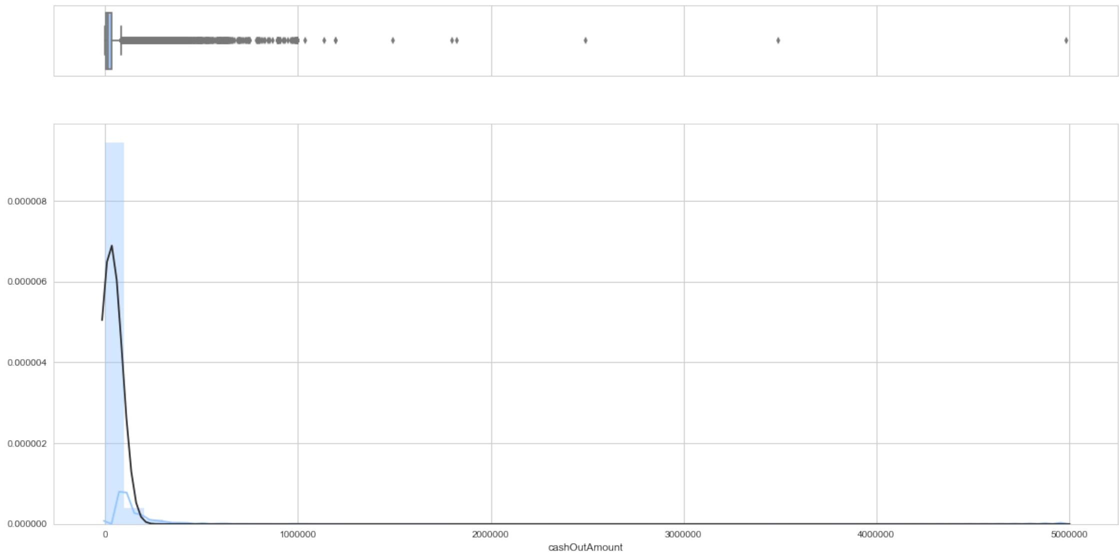

```
EDA for: appraisalWaiver

count                  46128
unique                     1
top         Appraisal Waived
freq                   46128
Name: appraisalWaiver, dtype: object
Missing Values: 250140

appraisalWaiver contains a single level

Level                                               Weight
Appraisal Waived                                    46128
```



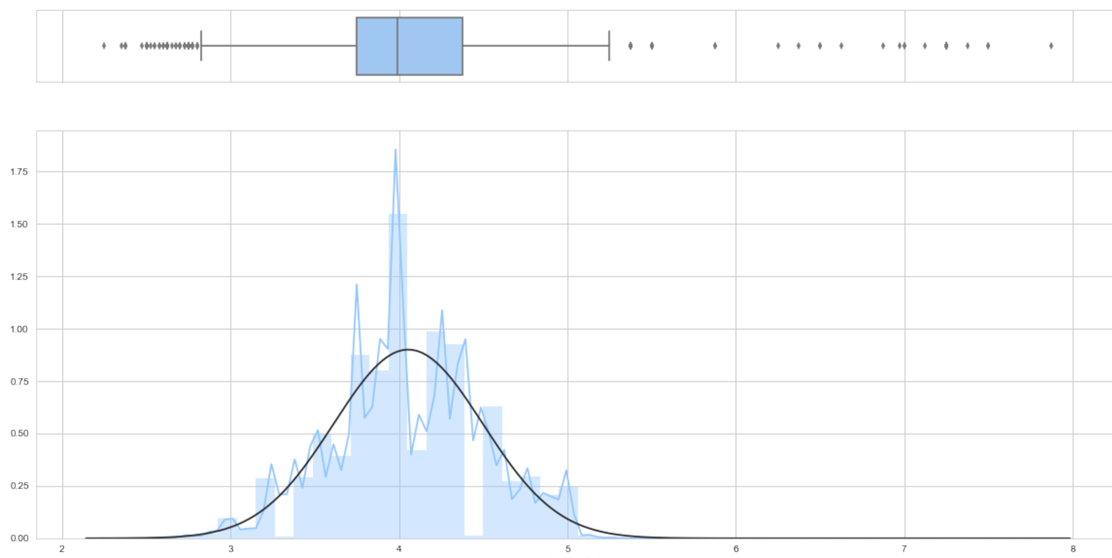

```
EDA for: Deposit

count      296268
unique        340
top          0.00
freq       134621
Name: Deposit, dtype: object
Missing Values: 0
```




```
EDA for: cashOutAmount

count      150089
unique     142777
top          0.63
freq           11
Name: cashOutAmount, dtype: object
Missing Values: 146179
```

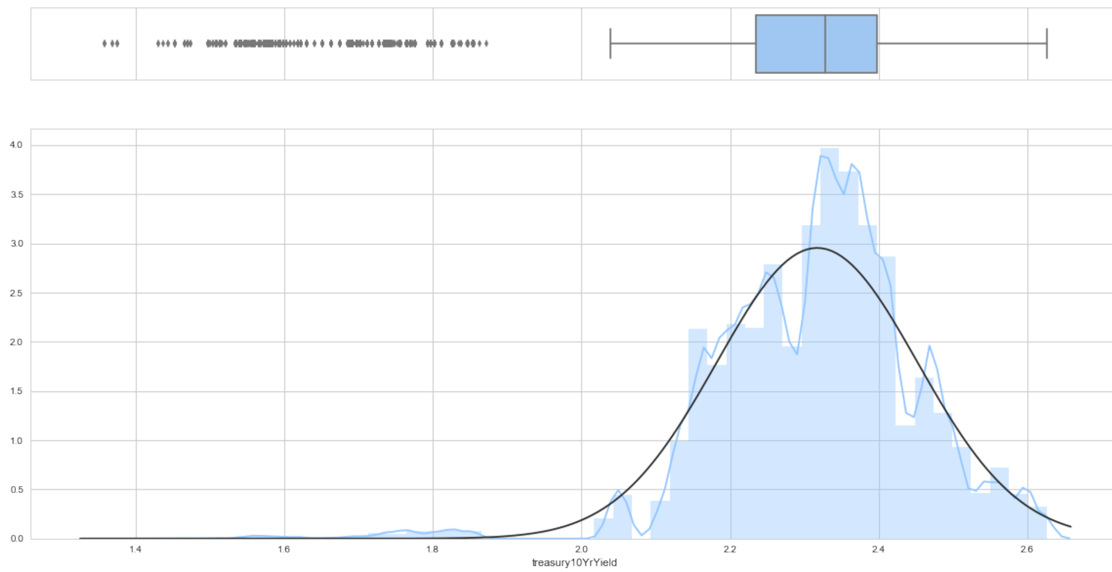


```
EDA for: interestRate

count    296268.000000
mean          4.054945
std           0.443155
min           2.250000
25%           3.750000
50%           3.990000
75%           4.375000
max           7.875000
Name: interestRate, dtype: float64
Missing Values: 0
```

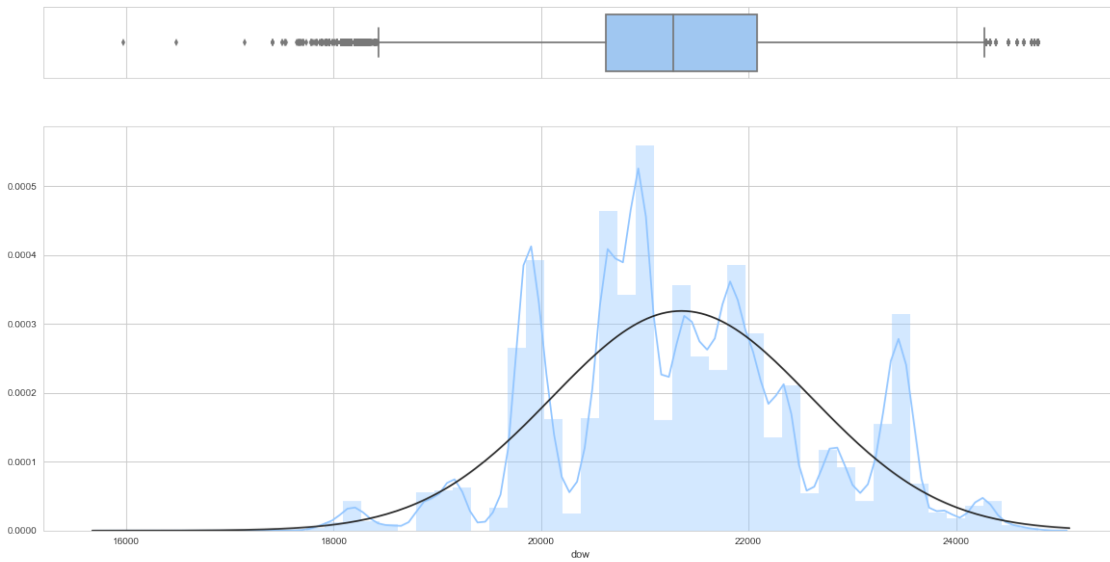



```
EDA for: treasury10YrYield

count    296268.000000
mean          2.317066
std           0.135039
min           1.357900
25%           2.234600
```

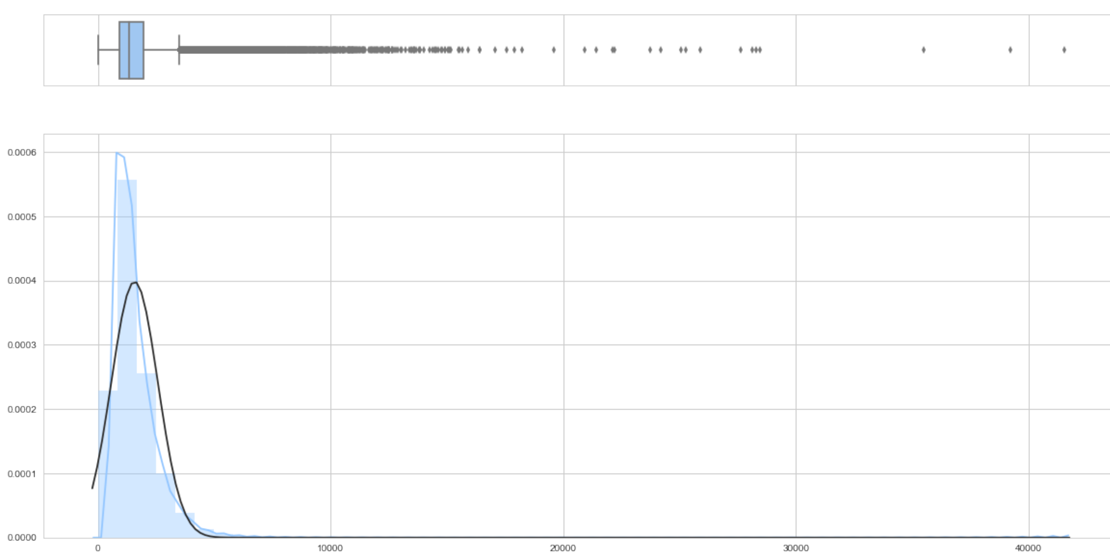
```
50%            2.327700
75%            2.397100
max            2.625800
Name: treasury10YrYield, dtype: float64
Missing Values: 0
```



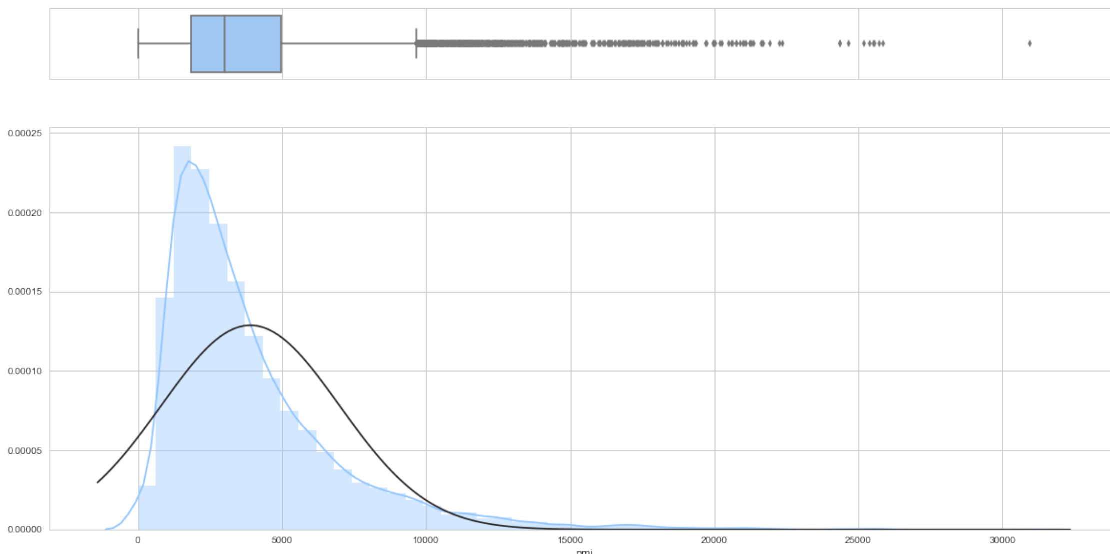

```
EDA for: dow

count    296268.000000
mean      21353.853978
std        1250.910164
min       15973.840000
25%       20624.050000
50%       21271.970000
75%       22085.340000
max       24792.200000
Name: dow, dtype: float64
Missing Values: 0
```




```
EDA for: monthlyPayment

count     296265
unique    174876
top         0.00
freq        1666
Name: monthlyPayment, dtype: object
Missing Values: 3
```

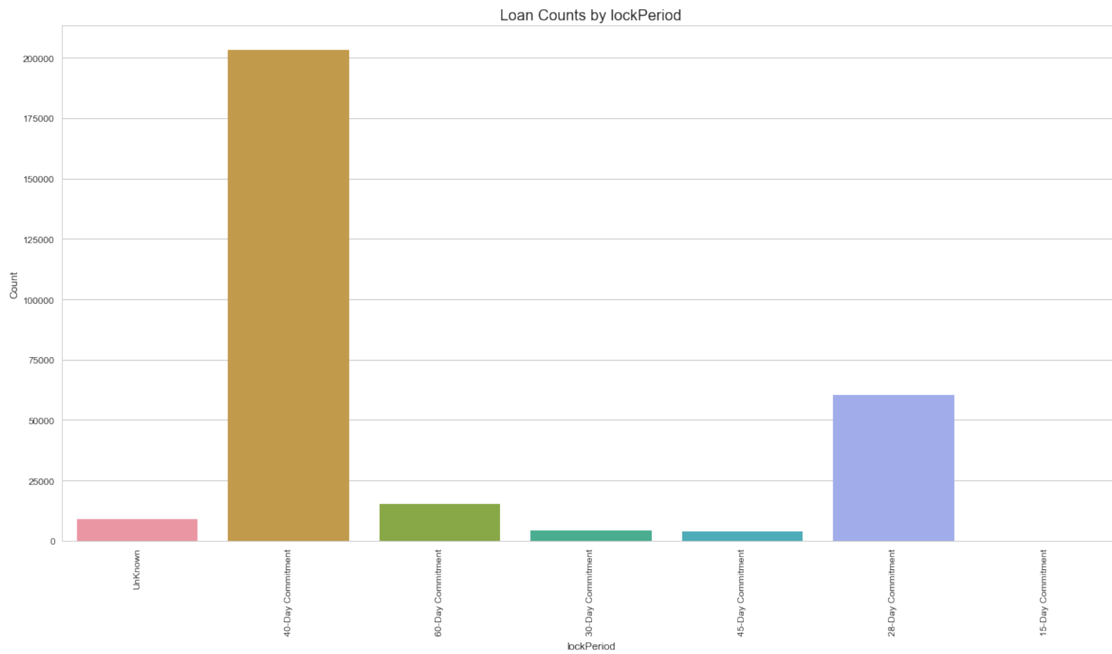


```
EDA for: pmi

count     12635
unique    11486
top           0
freq        157
Name: pmi, dtype: object
Missing Values: 283633
```

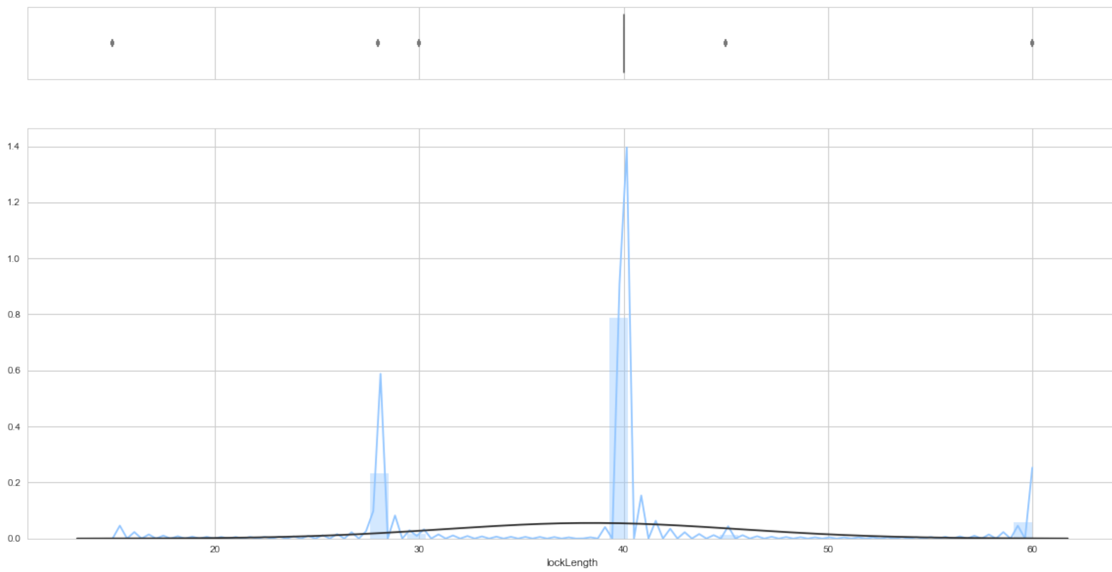



```
EDA for: lockPeriod

count                296268
unique                    7
top       40-Day Commitment
freq                 203134
Name: lockPeriod, dtype: object
Missing Values: 0

lockPeriod has thin data in  15-Day Commitment: 11 out of 296268 (3.7e-05)
```

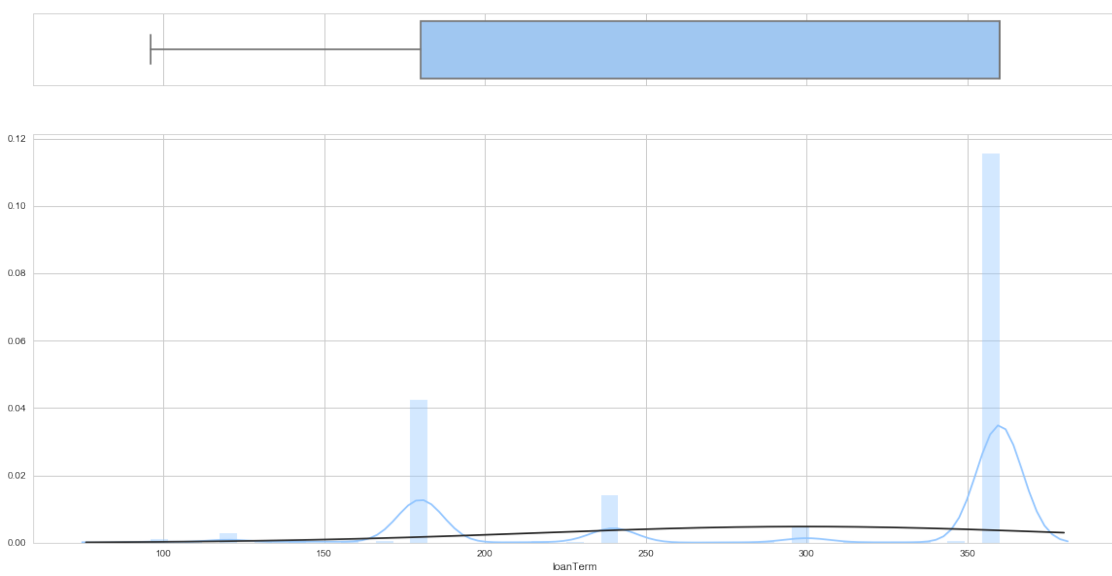
Level	Weight
40-Day Commitment	203134
28-Day Commitment	60523
60-Day Commitment	15349
UnKnown	9154
30-Day Commitment	4299
45-Day Commitment	3798
15-Day Commitment	11



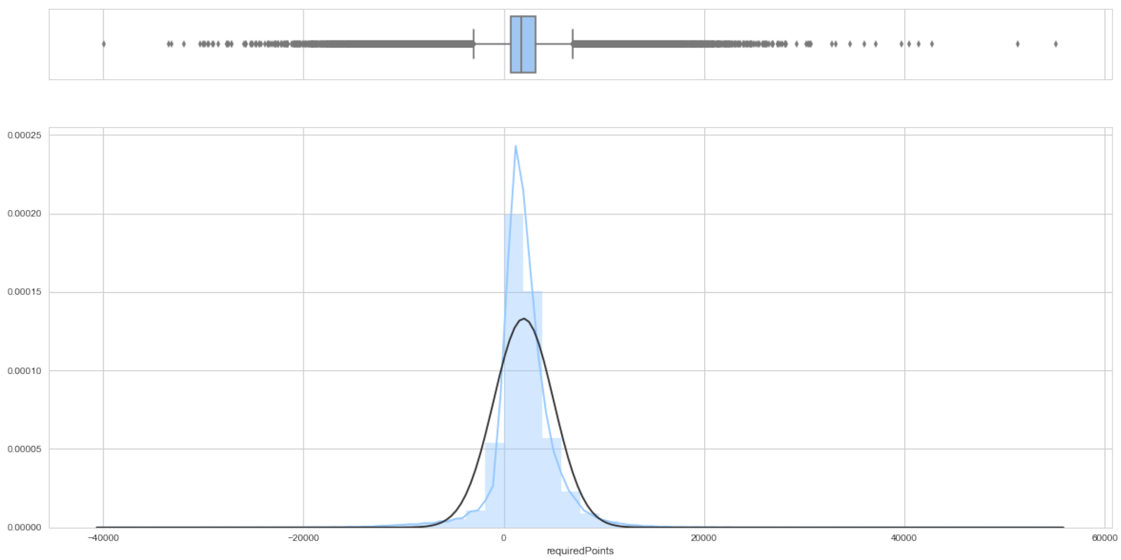

```
EDA for: lockLength

count    287114.000000
mean         38.455070
std           7.155701
min          15.000000
25%          40.000000
50%          40.000000
75%          40.000000
max          60.000000
Name: lockLength, dtype: float64
Missing Values: 9154
```



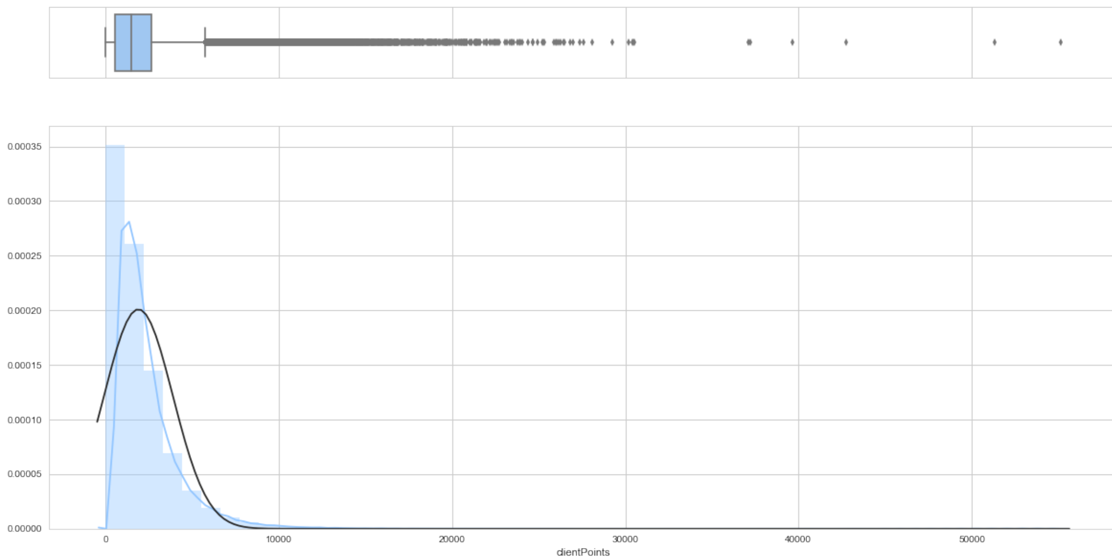

```
EDA for: loanTerm

count     296268
unique        28
top          360
freq      184528
Name: loanTerm, dtype: object
Missing Values: 0
```



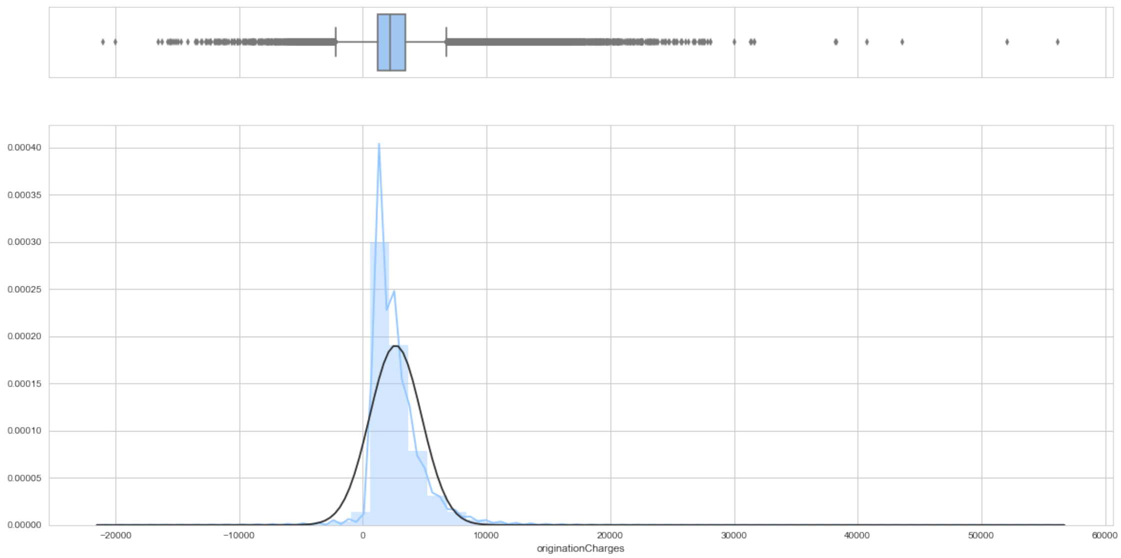
```
EDA for: requiredPoints

count     296268
unique    150701
top         0.00
freq        8130
Name: requiredPoints, dtype: object
Missing Values: 0
```



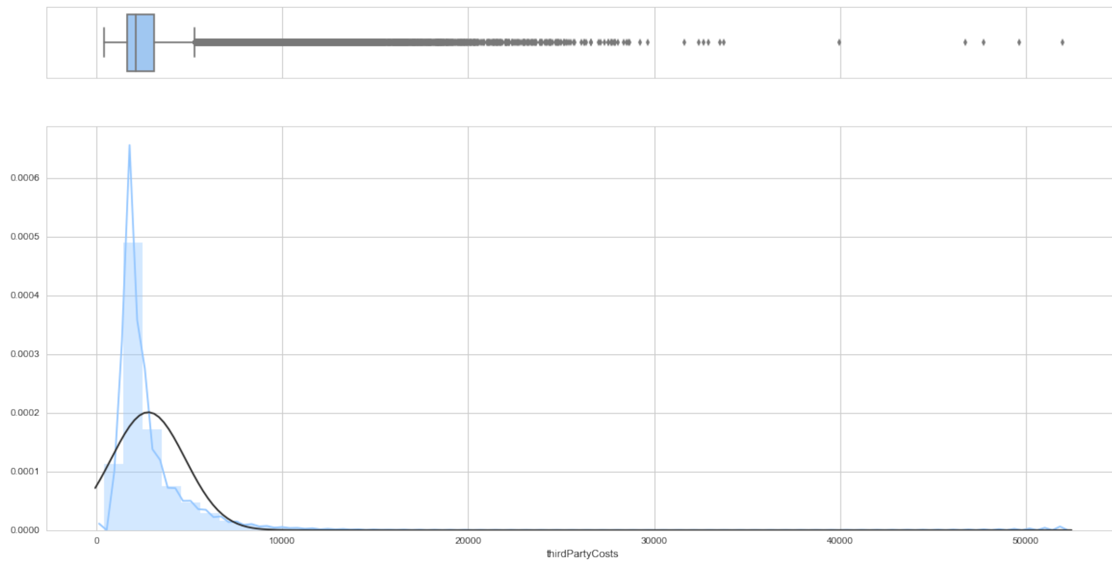

```
EDA for: clientPoints

count     296268
unique    110394
top         0.00
freq       58043
Name: clientPoints, dtype: object
Missing Values: 0
```



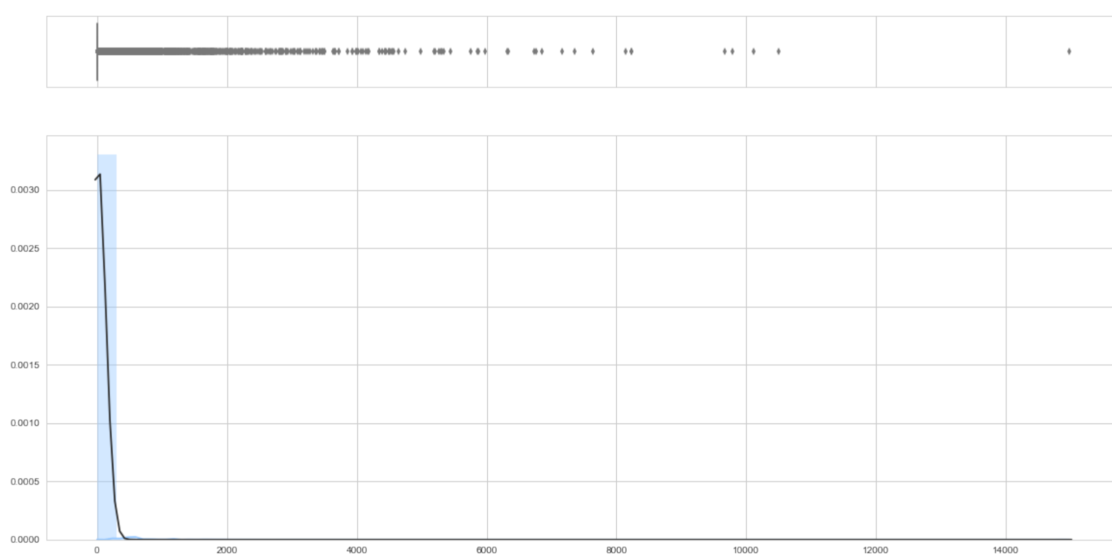

```
EDA for: originationCharges

count    293739.000000
mean       2648.121998
std        2094.951183
min      -21005.290000
25%        1170.000000
50%        2177.890000
75%        3408.420000
max       56197.250000
Name: originationCharges, dtype: float64
Missing Values: 2529
```



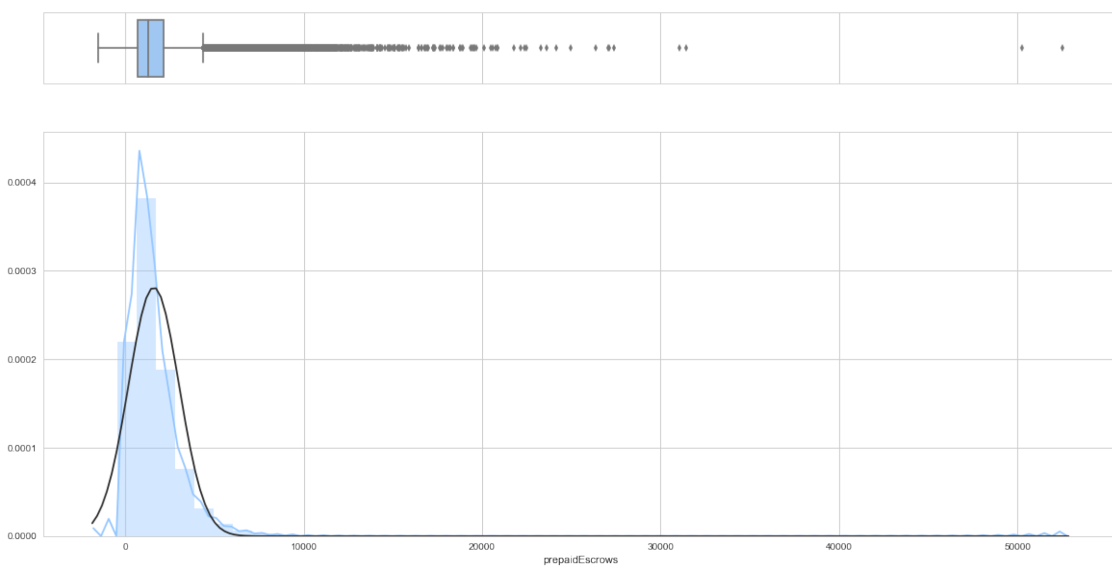

```
EDA for: thirdPartyCosts

count    293739.000000
mean       2784.951902
std        1987.573403
min         413.500000
25%        1674.205000
50%        2126.630000
75%        3106.235000
max       51970.500000
Name: thirdPartyCosts, dtype: float64
Missing Values: 2529
```




```
EDA for: cureCosts

count      293739
unique       8511
top          0.00
freq       257105
Name: cureCosts, dtype: object
Missing Values: 2529
```

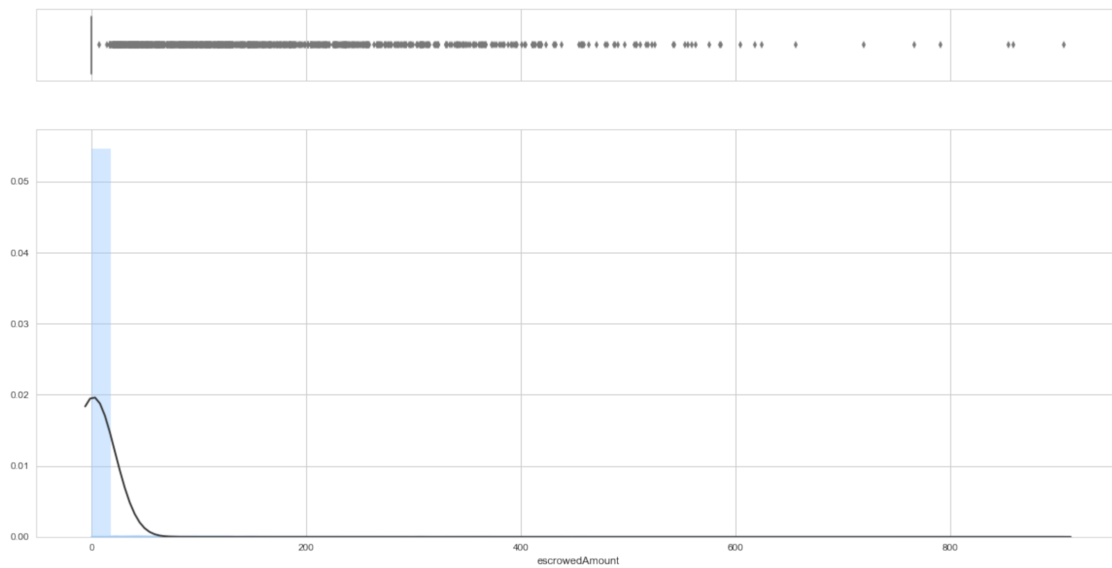


```
EDA for: prepaidEscrows

count    293737.000000
mean       1602.651093
std        1417.675602
min       -1504.980000
25%         686.330000
50%        1283.110000
75%        2148.130000
max       52520.970000
Name: prepaidEscrows, dtype: float64
Missing Values: 2531
```




```
EDA for: escrowedAmount

count      105770
unique        999
top          0.00
freq       104731
Name: escrowedAmount, dtype: object
```

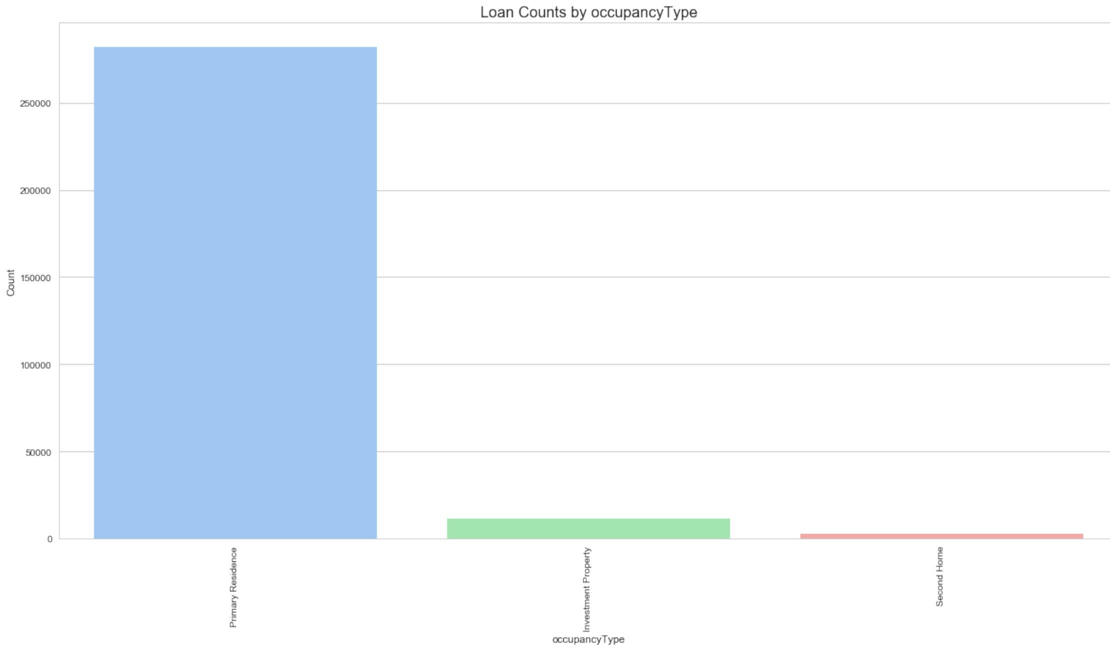
Missing Values: 190498



EDA for: occupancyType

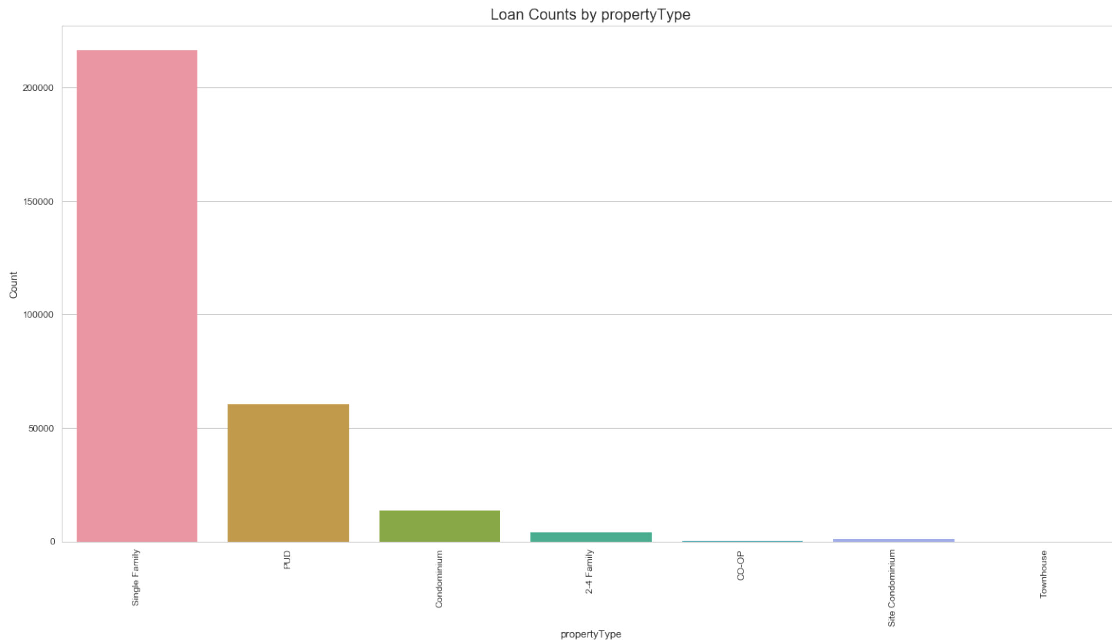
```
count                296268
unique                    3
top       Primary Residence
freq                 281755
Name: occupancyType, dtype: object
Missing Values: 0
```

Level	Weight
Primary Residence	281755
Investment Property	11693
Second Home	2820




```
EDA for: propertyType

count            296268
unique                7
top       Single Family
freq             216463
Name: propertyType, dtype: object
Missing Values: 0

propertyType has thin data in Site Condominium: 990 out of 296268 (0.0033)

propertyType has thin data in CO-OP: 365 out of 296268 (0.0012)

propertyType has thin data in Townhouse: 34 out of 296268 (0.00011)
```

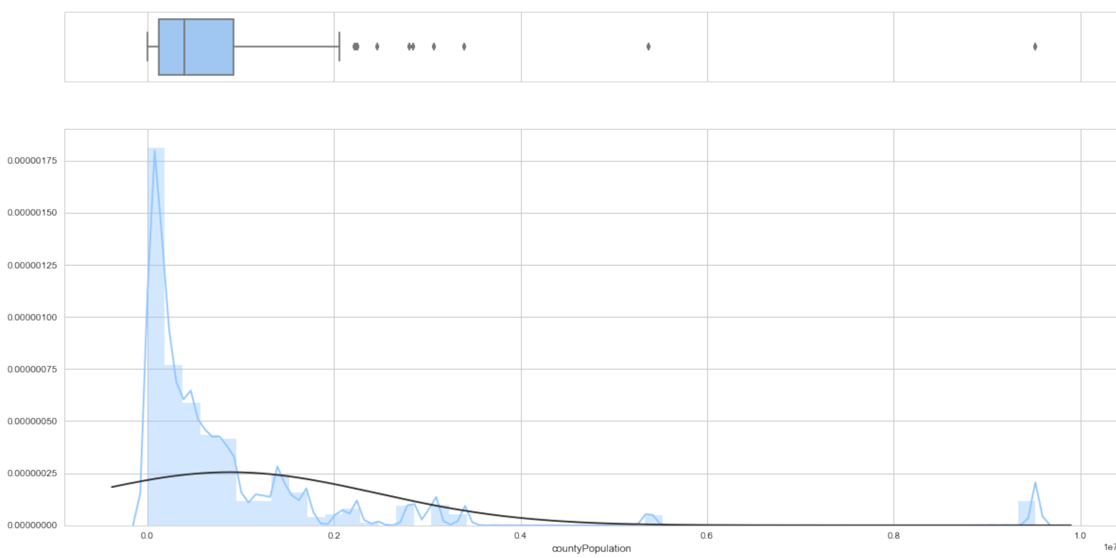
Level	Weight
Single Family	216463
PUD	60596
Condominium	13587
2-4 Family	4233
Site Condominium	990
CO-OP	365
Townhouse	34




```
EDA for: state

count          296248
unique             51
top        CALIFORNIA
freq            36845
Name: state, dtype: object
Missing Values: 20

state has thin data in MAINE : 1207 out of 296268 (0.0041)

state has thin data in DELAWARE : 1162 out of 296268 (0.0039)

state has thin data in HAWAII : 1149 out of 296268 (0.0039)

state has thin data in MONTANA : 1003 out of 296268 (0.0034)

state has thin data in ALASKA : 903 out of 296268 (0.003)

state has thin data in RHODE ISLAND : 873 out of 296268 (0.0029)

state has thin data in WYOMING : 769 out of 296268 (0.0026)

state has thin data in VERMONT : 702 out of 296268 (0.0024)
```

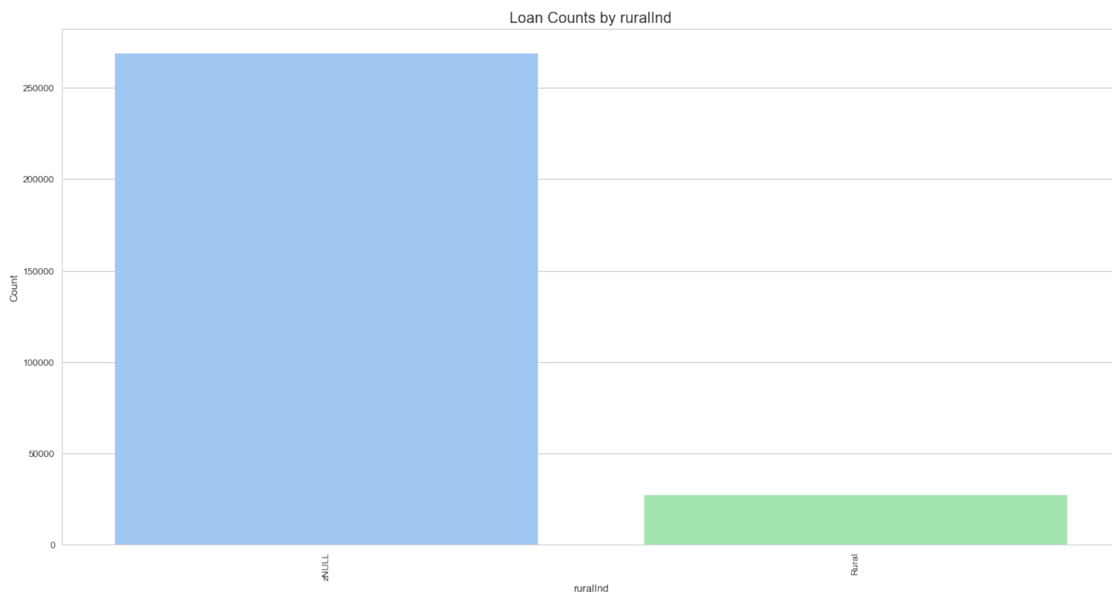
```
state has thin data in DISTRICT OF COLUMBIA : 654 out of 296268 (0.0022)

state has thin data in SOUTH DAKOTA : 529 out of 296268 (0.0018)

state has thin data in NORTH DAKOTA : 474 out of 296268 (0.0016)
```

```
EDA for: countyPopulation

count    2.924470e+05
mean     8.898770e+05
std      1.564763e+06
min      0.000000e+00
25%      1.226600e+05
50%      3.998430e+05
75%      9.214820e+05
max      9.519338e+06
Name: countyPopulation, dtype: float64
Missing Values: 3821
```

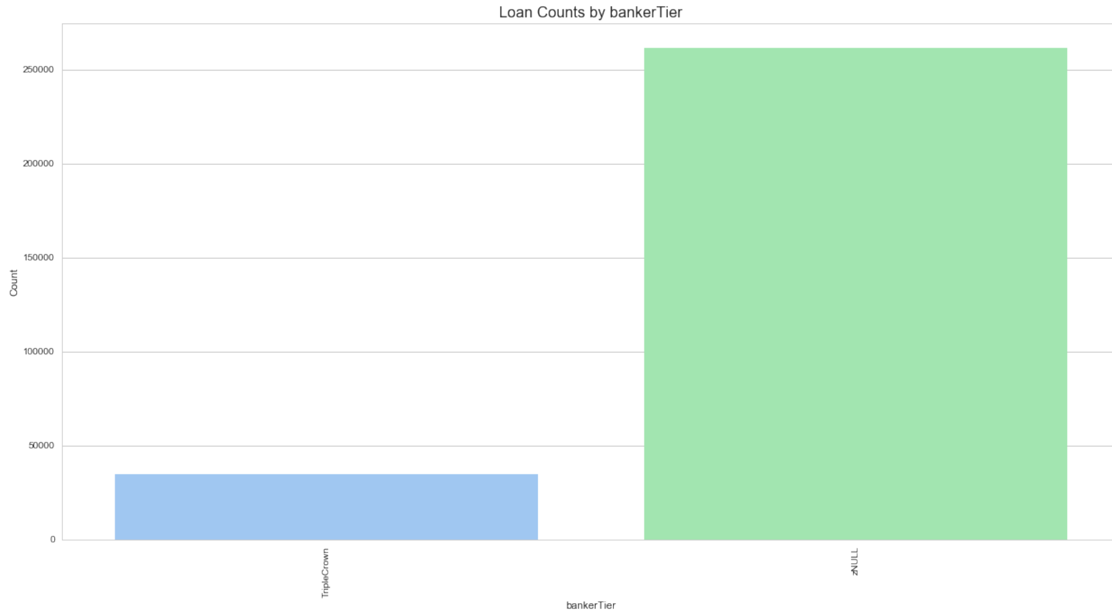



```
EDA for: ruralInd

count     27417
```

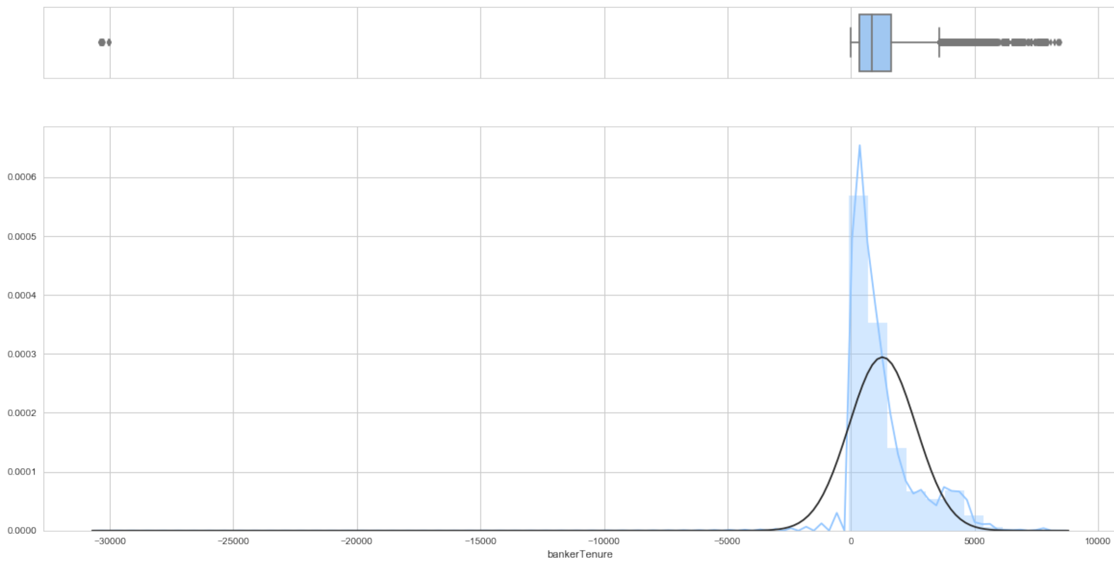
```
unique        1
top       Rural
freq      27417
Name: ruralInd, dtype: object
Missing Values: 268851

ruralInd contains a single level

Level                                              Weight
Rural                                              27417
```



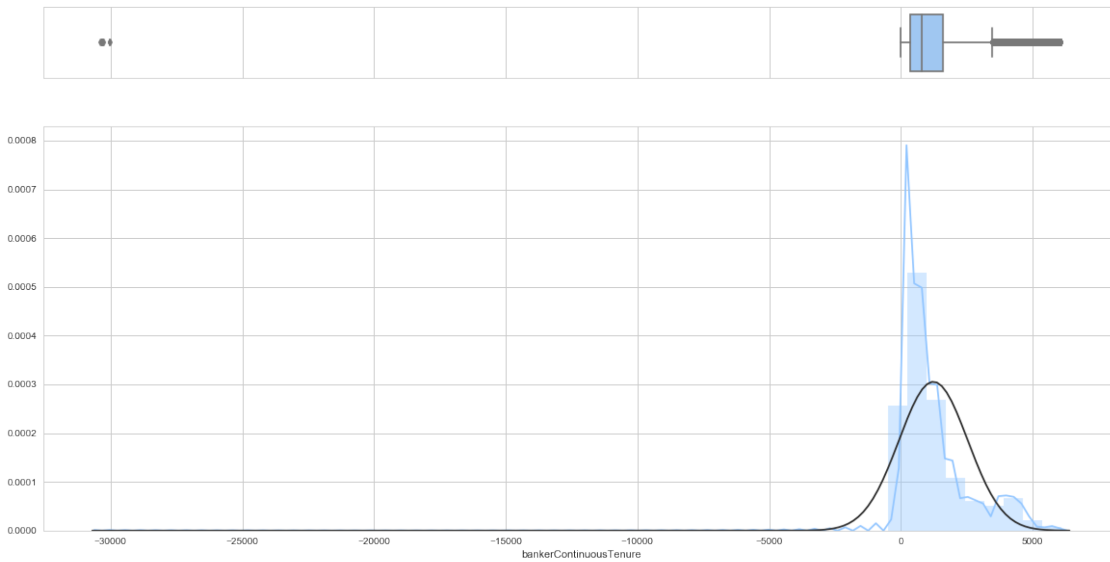

```
EDA for: bankerTier

count           34892
unique              1
top       TripleCrown
freq            34892
Name: bankerTier, dtype: object
Missing Values: 261376

bankerTier contains a single level

Level                                              Weight
TripleCrown                                        34892
```




```
EDA for: bankerTenure

count    286248.000000
mean       1265.925226
std        1354.389535
min      -30381.000000
25%         346.000000
50%         821.000000
75%        1635.000000
max        8467.000000
Name: bankerTenure, dtype: float64
Missing Values: 10020
```




```
EDA for: bankerContinuousTenure

count    286234.000000
mean       1227.887623
std        1305.759269
min      -30382.000000
25%         338.000000
50%         801.000000
75%        1592.000000
max        6078.000000
Name: bankerContinuousTenure, dtype: float64
Missing Values: 10034
```



1.3 Preprocessing

```
In [7]: y = loans.iloc[:, target].as_matrix()
        x_num = loans.iloc[:, cont_index].as_matrix()
        x_cat = loans.iloc[:, cat_index].as_matrix()
```

```
In [8]: from sklearn import preprocessing
        from sklearn.preprocessing import RobustScaler
        from sklearn.preprocessing import Imputer
        from sklearn.decomposition import PCA

        #Numeric Imputing
        imp = Imputer(strategy='median')
        x_num = imp.fit_transform(x_num)

        # Robust Standardization
        sca = RobustScaler()
        x_num = sca.fit_transform(x_num)

        # One Hot Encoding
        from sklearn.preprocessing import OneHotEncoder
        enc = OneHotEncoder()
        x_cat = enc.fit_transform(x_cat).toarray()

        x = np.concatenate((x_num, x_cat), axis = 1)
```

```
In [12]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state
```

```
In [ ]: from sklearn.linear_model import LinearRegression
        from sklearn.linear_model import RidgeCV
        from sklearn.linear_model import LassoCV
        from sklearn.linear_model import ElasticNetCV
        from sklearn.decomposition import PCA
        from sklearn.pipeline import Pipeline


        pipe_lr = Pipeline([('lr', LinearRegression())])

        pipe_lr_pca = Pipeline([('pca', PCA(n_components=10)),
                                ('lr', LinearRegression())])

        pipe_ridge = Pipeline([('lr', RidgeCV())])

        pipe_ridge_pca = Pipeline([('pca', PCA(n_components=10)),
                                ('lr', RidgeCV())])

        pipe_lasso = Pipeline([('lr', LassoCV())])

        pipe_lasso_pca = Pipeline([('pca', PCA(n_components=10)),
                                ('lr', LassoCV())])

        pipe_en = Pipeline([('lr', ElasticNetCV())])

        pipe_en_pca = Pipeline([('pca', PCA(n_components=10)),
                                ('lr', ElasticNetCV())])

        param_range_alpha = [0.1, 0.2, 0.5, 0.75, 1]

        # List of pipelines for ease of iteration
        grids = [pipe_lr, pipe_lr_pca, pipe_ridge, pipe_ridge_pca, pipe_lasso, pipe_lasso_pca,

        # Dictionary of pipelines and classifier types for ease of reference
        grid_dict = {0: 'Linear Regression', 1: 'Linear Regression w/PCA',
                     2: 'Ridge Regression', 3: 'Ridge Regression w/PCA',
                     4: 'Lasso Regression', 5: 'Lasso Regression w/PCA',
                     6: 'Elastic Net Regression', 7: 'Elastic Net Regression w/PCA'}



        from sklearn.metrics import mean_squared_error, r2_score
        print('Performing model optimizations...')
        best_acc = 0.0
        best_clf = 0
        best_gs = ''
        for idx, gs in enumerate(grids):
            print('\nEstimator: %s' % grid_dict[idx])
```

```
    # Fit grid search
    gs.fit(X_train, y_train)
    # Best training data accuracy
    y_pred = gs.predict(X_test)
    print('MSE: {:.2g}'.format(mean_squared_error(y_test, y_pred)))
    print('R^2: {:.2g}'.format(r2_score(y_test, y_pred)))


from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score

pipe_rf = Pipeline([('clf', RandomForestRegressor(random_state=42))])

pipe_rf_pca = Pipeline([('pca', PCA(n_components=10)),
                ('clf', RandomForestRegressor(random_state=42))])

param_range = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

grid_params_rf = [{'clf__criterion': ['mse', 'mae'],
                    'clf__max_depth': param_range}]

jobs = -1

gs_rf = GridSearchCV(estimator=pipe_rf,
                    param_grid=grid_params_rf,
                    scoring='accuracy',
                    cv=10,
                    n_jobs=jobs)

gs_rf_pca = GridSearchCV(estimator=pipe_rf_pca,
                    param_grid=grid_params_rf,
                    scoring='accuracy',
                    cv=10,
                    n_jobs=jobs)

# List of pipelines for ease of iteration
grids = [gs_rf, gs_rf_pca]

# Dictionary of pipelines and classifier types for ease of reference
grid_dict = {0: 'Random Forest', 1: 'Random Forest w/PCA'}

# Fit the grid search objects
best_acc = 0.0
best_clf = 0
best_gs = ''
for idx, gs in enumerate(grids):
    print('\nEstimator: %s' % grid_dict[idx])
```

```
        # Fit grid search
        gs.fit(X_train, y_train)
        # Best params
        print('Best params: %s' % gs.best_params_)
        # Best training data accuracy
        print('Best training accuracy: %.3f' % gs.best_score_)
        # Predict on test data with best params
        y_pred = gs.predict(X_test)
        # Test data accuracy of model with best params
        print('Test set accuracy score for best params: %.3f ' % accuracy_score(y_test, y_p
        # Track best (highest test accuracy) model
        if accuracy_score(y_test, y_pred) > best_acc:
            best_acc = accuracy_score(y_test, y_pred)
            best_gs = gs
            best_clf = idx
    print('\nClassifier with best test set accuracy: %s' % grid_dict[best_clf])
```

```
Performing model optimizations...

Estimator: Linear Regression
MSE: 3.2e+02
R^2: 0.26

Estimator: Linear Regression w/PCA
MSE: 4.1e+02
R^2: 0.053

Estimator: Ridge Regression
MSE: 3.2e+02
R^2: 0.26

Estimator: Ridge Regression w/PCA
MSE: 4.1e+02
R^2: 0.053

Estimator: Lasso Regression


c:\programdata\anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:1082: Da
  y = column_or_1d(y, warn=True)


MSE: 3.4e+02
R^2: 0.21

Estimator: Lasso Regression w/PCA


c:\programdata\anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:1082: Da
```

```
  y = column_or_1d(y, warn=True)


MSE: 4.1e+02
R^2: 0.056

Estimator: Elastic Net Regression


c:\programdata\anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:1082: Da
  y = column_or_1d(y, warn=True)


MSE: 3.6e+02
R^2: 0.19

Estimator: Elastic Net Regression w/PCA


c:\programdata\anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:1082: Da
  y = column_or_1d(y, warn=True)


MSE: 4.1e+02
R^2: 0.056

Estimator: Random Forest
```