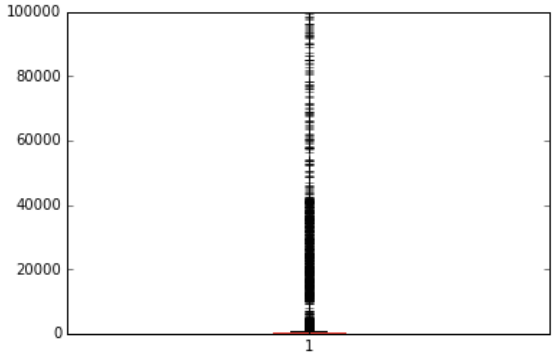
Metadata for data newly created:

Field	Type	Description
Loan_Amount_Level	String	Categorizes each loan amount into groups: 'Super High' - Grater than $10000k 'High' - [$347k, $10000k] 'Moderate' - ($153k, $347k) 'Low' - ($10k, $153k] 'Super Low' - (0, $10k]

Data Quality Concerns:

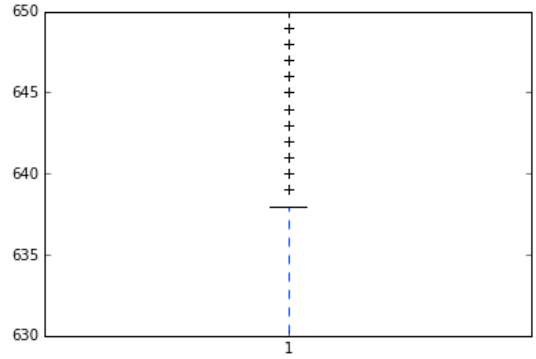
- **Rules for Data Quality Assessment:**
  - Detect missing values;
  - Detect abnormal inputs: such as non-digit input in numeric feature;
  - Calculate descriptive statistics for numeric feature to see its mean, range, distribution, etc;
  - Draw boxplot to see outliers for numeric feature;
  - Draw histogram to see distribution for numeric feature;
  - Draw frequency table to see patterns in categorical feature
  - ……
- **'Loan_Amount_000':**
  - There is no missing value;
  - There is no non-digit input;
  - They are very widely ranged (Descriptive statistics):
    - count 1.321158e+06
    - mean 2.902331e+02
    - std 9.657792e+02
    - min 1.000000e+00
    - 25% 1.530000e+02
    - 50% 2.350000e+02
    - 75% 3.470000e+02
    - max 9.962500e+04
  - Boxplot to see outliers:

```
In [81]: plt.boxplot(amount)
    ...:
Out[81]:
{'boxes': [<matplotlib.lines.Line2D at 0x11c7909e8>],
 'caps': [<matplotlib.lines.Line2D at 0x11c797cf8>,
  <matplotlib.lines.Line2D at 0x11c79cc50>],
 'fliers': [<matplotlib.lines.Line2D at 0x11c7a1cc0>],
 'means': [],
 'medians': [<matplotlib.lines.Line2D at 0x11c79cd68>],
 'whiskers': [<matplotlib.lines.Line2D at 0x11c790b70>,
  <matplotlib.lines.Line2D at 0x11c797be0>]}
```



- It is very long tailed. And there are seemingly a lot of outliers. Try to find the upper bound:

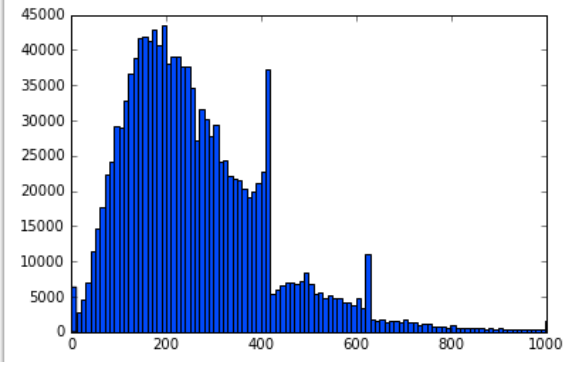
```
In [60]: plt.boxplot(amount)
    ...: plt.ylim(630, 650)
Out[60]: (630, 650)
```



- Find out that there are about 38983 points greater than 638 and they take about 2.96% of total points. They can be removed but I do not think we should. Because they represent rich people's loan.

- Histogram to see distribution:

```
In [49]: plt.hist(amount,bins=10000)
    ...: plt.xlim(0,1000)
Out[49]: (0, 1000)
```



        - Obviously not normally distributed. Very right skewed.
        - Kind of chi-squared distributed.
    - Check normality:
        - Perform normality tests to determine:
        - All p-values are nearly 0. So it is not normally distributed.

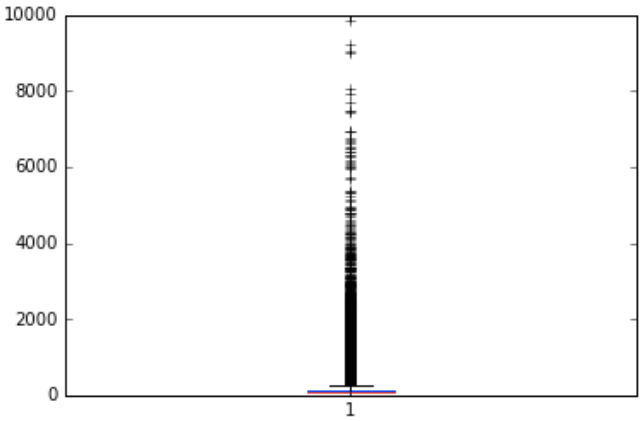
- **'Respondent_Name'**
    - Check and conclude that 'As_of_Year', 'Agency_Code' and 'Respondent_ID' uniquely determine 'Respondent_Name'
    - Look into those 'Respondent_Name' with same 'Respondent_ID' and find out some issues like:
        - ('22-3603829', ['ABSOLUTE HOME MORTGAGE CORPORA', 'ABSOLUTE HOME MORTGAGE CORP', 'AHMC'])
        - ID '22-3603829' corresponeds to three institutions in different locations in 3 successive years.
        - They have similar names and two have same locations NJ, but I think they are actually 3 institutions.
    - Create frequency table and find that 'FIRST NATIONAL BANK' has the most applications. So more research into this institution may be of great help.
- **'Applicant_Income_000'**
    - Income is a key factor to consider in loan approving. It is also an indicator for financial institutions making revenue. People with higher income can afford higher loan, from which institutions can make more profit.
    - There are 117853 non-digit inputs. They are all 'NA' – not null but string 'NA'. We should remove them.
    - Also very widely ranged (Descriptive statistics):

        count 1.203305e+06
        mean 1.180193e+02
        std 1.226957e+02
        min 0.000000e+00
        25% 6.000000e+01

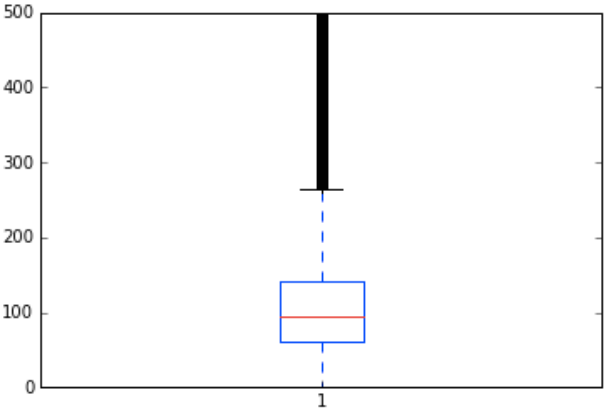
50% 9.400000e+01
75% 1.420000e+02
max 9.999000e+03

- Boxplot:

```
In [32]: plt.boxplot(income)
    ...:
Out[32]:
{'boxes': [<matplotlib.lines.Line2D at 0x11b810240>],
 'caps': [<matplotlib.lines.Line2D at 0x11b816c50>,
  <matplotlib.lines.Line2D at 0x11b816d68>],
 'fliers': [<matplotlib.lines.Line2D at 0x11b81bdd8>],
 'means': [],
 'medians': [<matplotlib.lines.Line2D at 0x11b81b5c0>],
 'whiskers': [<matplotlib.lines.Line2D at 0x11b810be0>,
  <matplotlib.lines.Line2D at 0x11b810cf8>]}
```




```
In [33]: plt.boxplot(income)
    ...: plt.ylim(1, 500)
Out[33]: (1, 500)
```



- Similar pattern as Loan amount above.

- **'Sequence_Number'**
  - Check if 'Sequence_Number' is uniquely assigned given 'As_of_Year' and 'Respondent_ID':
    - In 2013, there is a 'Sequence_Number' wrongly assigned to two applications.