Lending Club – Case Study

Lending Club – an online agency which hosts a marketplace to mediate between investors and borrowers wants to analyse their data to minimise the risk of losing money while lending to customers.

We were offered a small subset of data between 2007 and 2011 (5 years) to identify driving factors for a profitable business.

James Jeyabalan / Akik Ranade

Observation - 1

- *There are couple of borrower's Annual Income values that are very high but it doesn't skew the mean or median. Hence those values were* ***not considered as outliers***

```
loan.annual_inc.quantile([0.70, 0.80, 0.90, 0.99, 1])
```

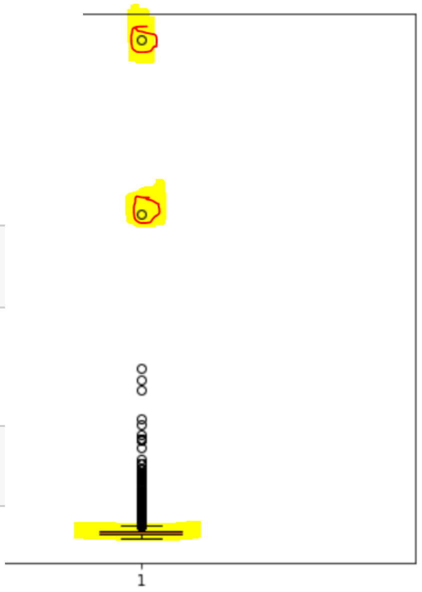
```
0.70      75000.0
0.80      90000.0
0.90     115000.0
0.99     234000.0
1.00    6000000.0
Name: annual_inc, dtype: float64
```

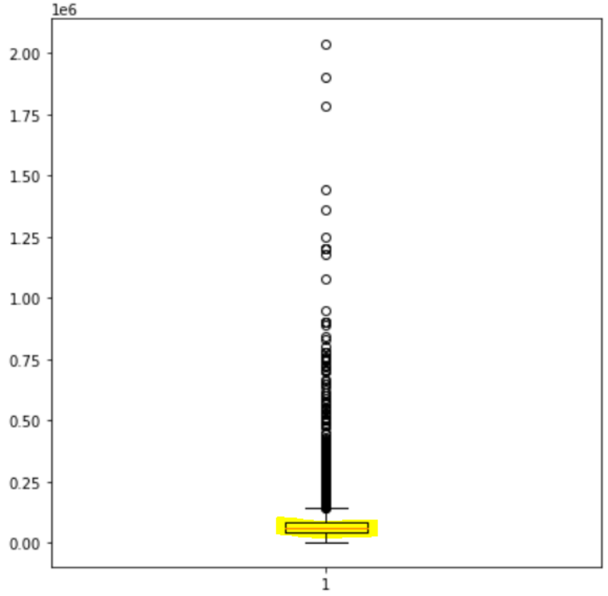
```
loan.annual_inc.mean()
```

```
68809.22861110396
```

```
loan[loan['annual_inc'] < 3850000].annual_inc.mean()
```

```
68555.82480726806
```





Observation – 2

- *There is no correlation between Borrower's Income & Grade/Sub Grade (Financial Rating)*

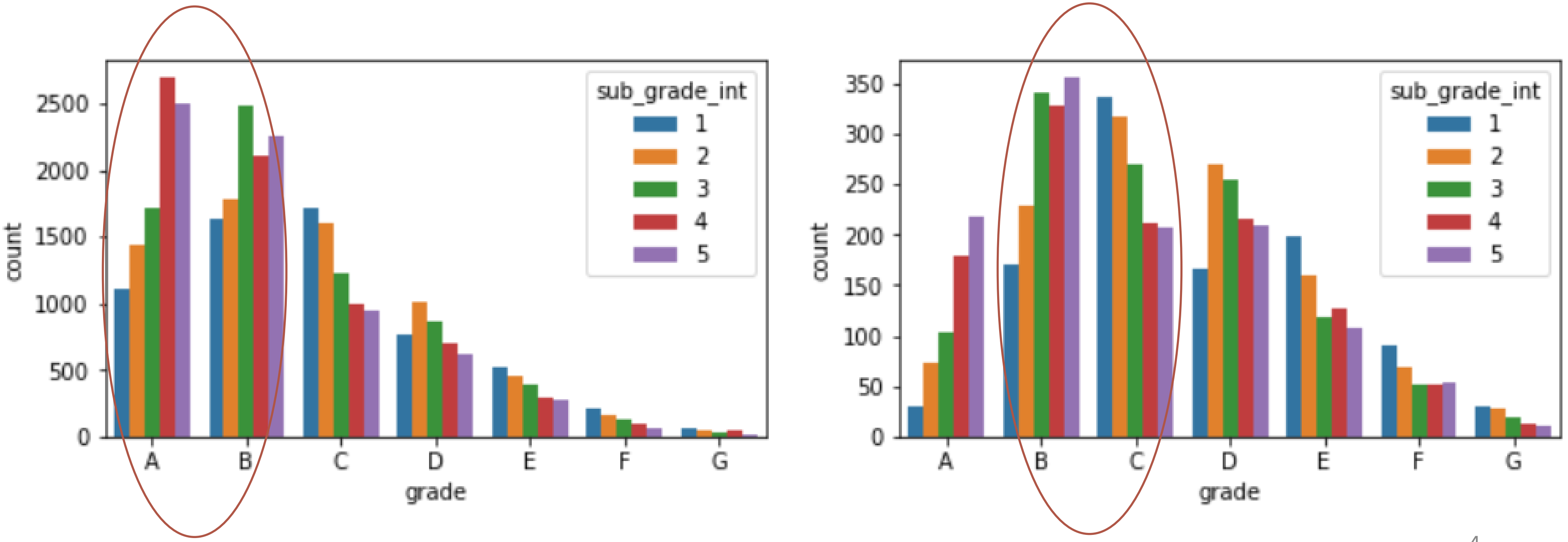
```
corr = loan.emp_length.corr(loan.grade_int)
print(corr)
corr = loan.emp_length.corr(loan.sub_grade_int)
print(corr)
```

```
-0.009659424947207684
-0.016887063002591185
```

Observation – 3

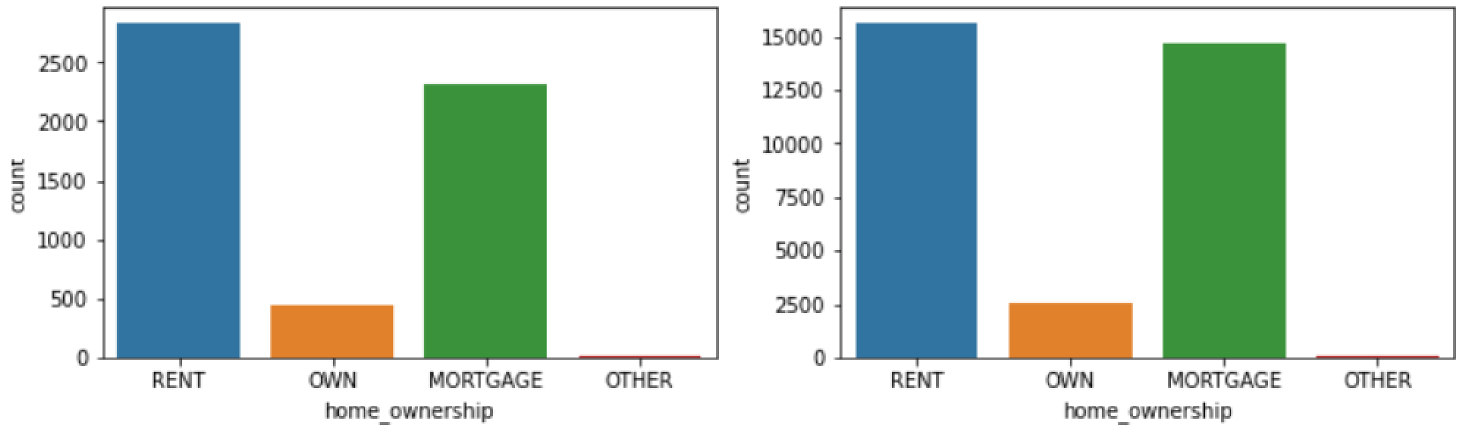
- *“Fully Paid” Category has borrowers in grade A & B predominantly*
- *“Charged Off” Category borrowers shift towards grade B & C*

```
ax = plt.subplots(figsize = (12,7))
subplot(221)
countplot(x='grade', order=(['A','B','C','D','E','F','G']),data=loan[loan['loan_status']=='Fully Paid'], hue='sub_grade_int')
subplot(222)
countplot(x='grade', order=(['A','B','C','D','E','F','G']), data=loan[loan.loan_status=='Charged Off'], hue='sub_grade_int')
```



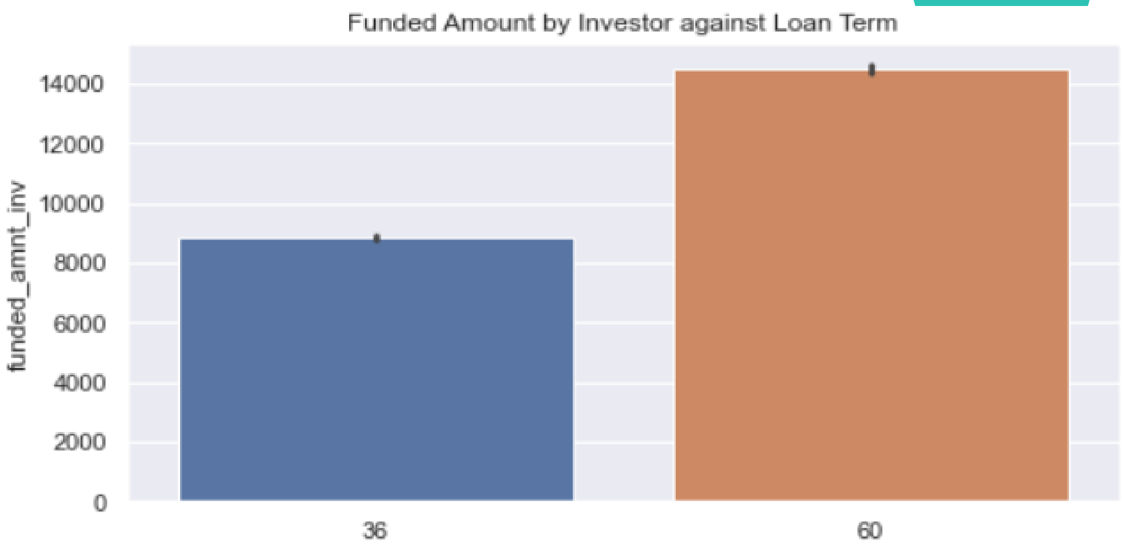
Observation – 4

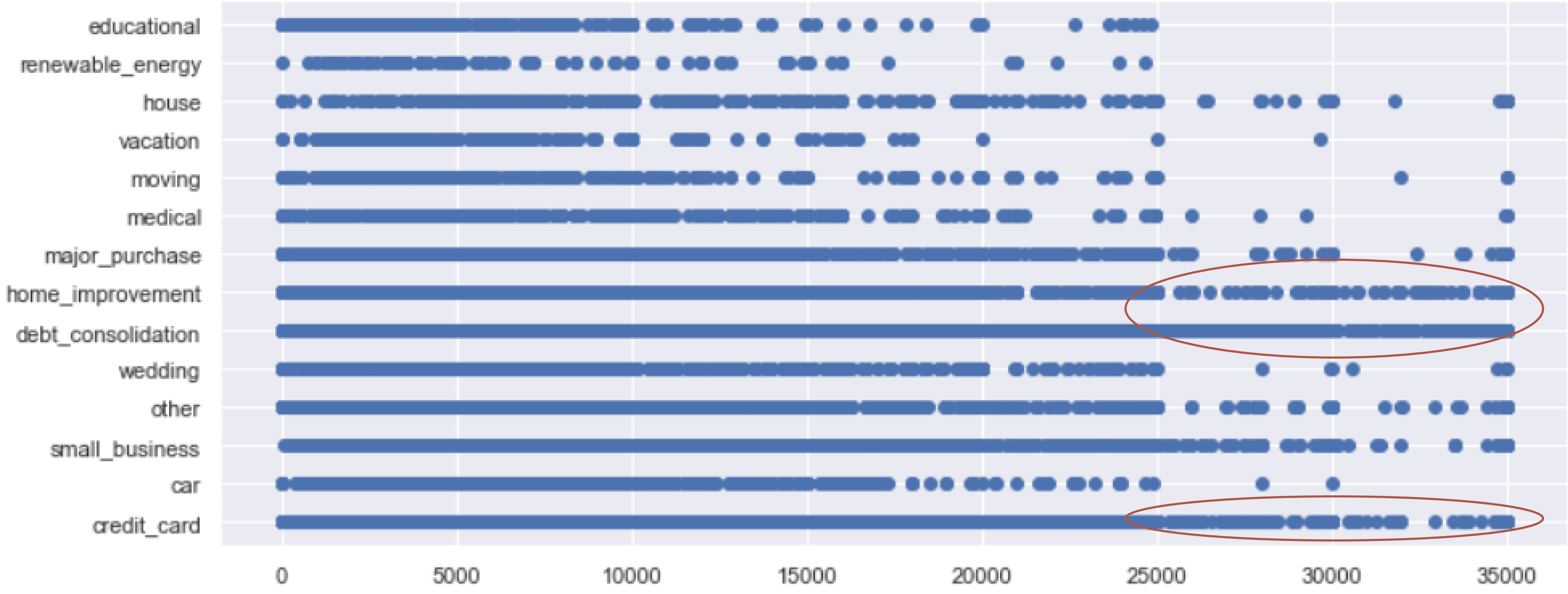
- *Home Ownership Pattern doesn't signify the risk of loan default as there is hardly any difference*



Observation – 5

- *Longer the term, higher the loan amount taken*
- *Higher loan amounts are predominantly taken for Debt Consolidation, Home Improvement & Credit Card*





Observation – 6

- *Delinquent in the last 2 years shows 10% default rate in the Not Funded category*

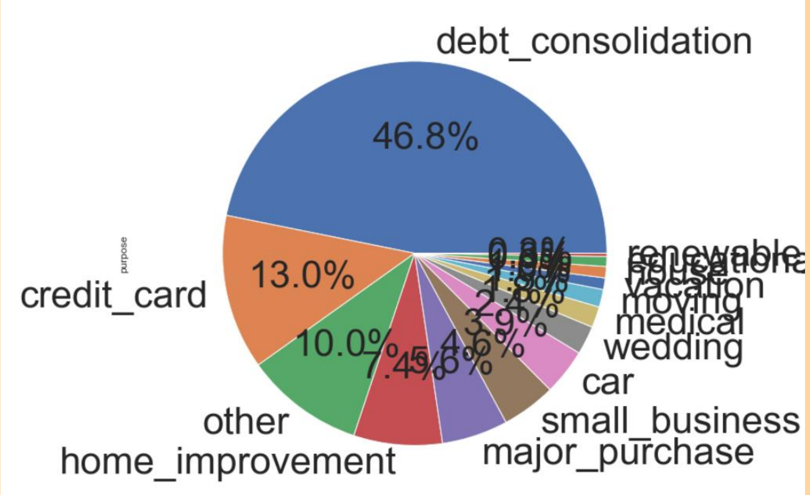
```
loan[loan['funded_amnt_inv_bin']=='Not Funded'].filter(['delinq_2yrs']).value_counts()
```

```
delinq_2yrs
0              130
1               12
2                4
3                2
dtype: int64
```

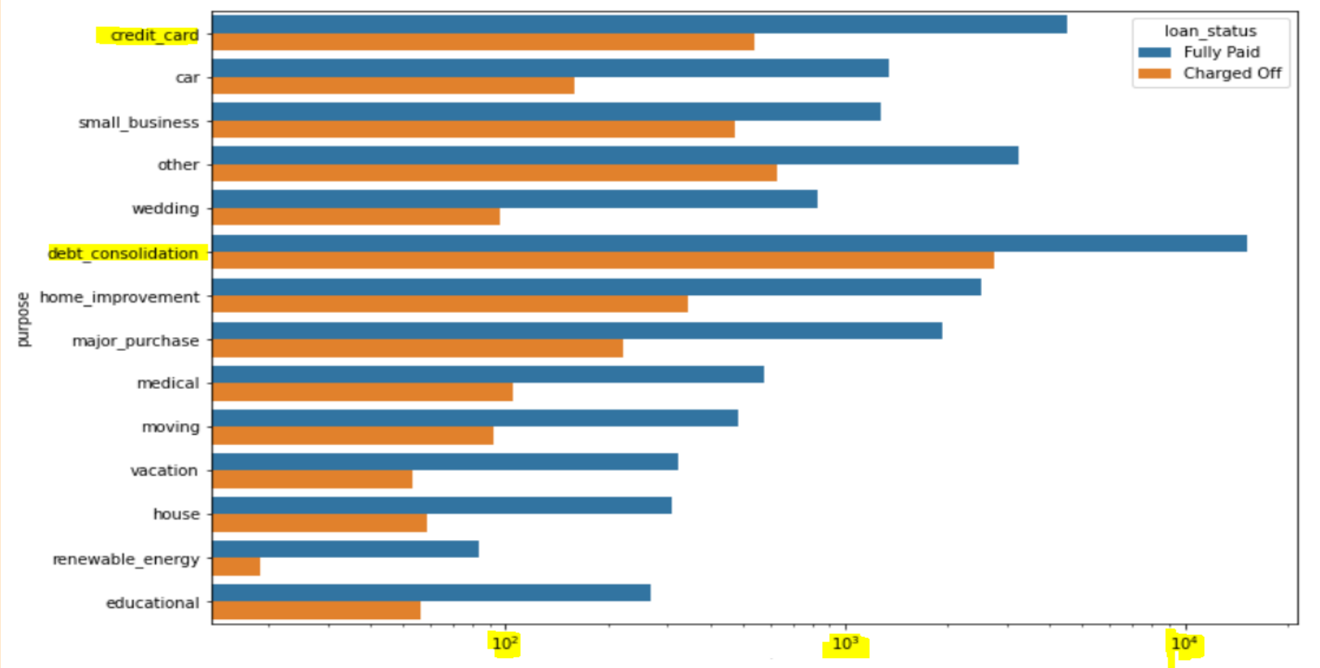
Appendix

Analyse: Distribution Patterns

- *Distribution of loan by "**Purpose**"*



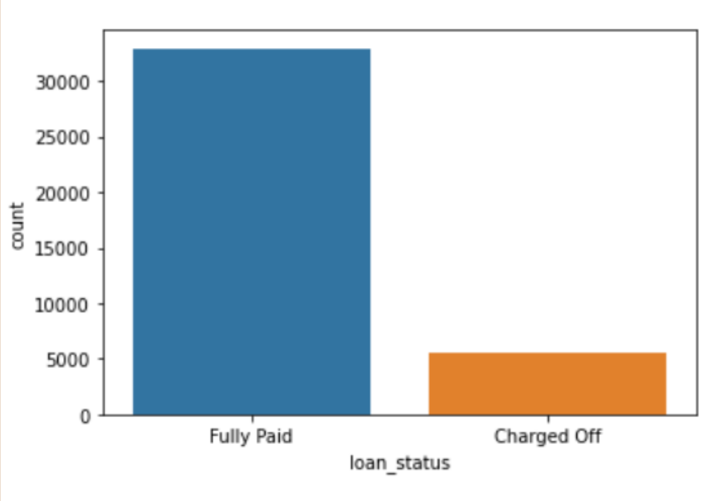
*"**Debt Consolidation**" is the single largest loan purpose across "Fully Paid" and "Charged-Off" borrowers*



- *Distribution by "**Loan Status**"*
  (after removing 'Current' accounts)

```
loan.loan_status.value_counts()

Fully Paid      32916
Charged Off      5611
```



Cleanse: Row-Column Elimination Reasoning

- *Raw data size (Rows: 39717, Columns: 111)*
    1. *After removing "NA" columns (Columns: 57)*
        - *Effective memory utilization technique as the data set size reduced 50% (34 Mb to 17 Mb)*
    2. *After removing descriptive columns (Columns: 50)*
    3. *After removing UniValue (only 1 value or NaN) columns (Columns: 41)*
    4. *After removing above80% missing columns (Columns: 38)*
- *Eliminating Current accounts as they don't authoritatively say whether the borrower is a defaulter or not.*
    - *# of Current accounts is 1140, of which only 121 rows show delinquency, and the* ***last*** *delinquency is an average 3 Years*
    - *Hence considering this subset of data has insignificant for any concrete decision and removing the "Current" loan accounts. Rows,*
        - *Rows after dropping this subset:(38577, 38)*
    - *Remove 50 rows with 'NA' value for revol_util column since we don't want to impute this 'Ratio' field*
        - *Rows after dropping this subset:(38527, 38)*

Cleanse Curing the data

- ***Remove*** *%, +* ***symbols****, String objects which could potentially be Numeric data after removing the unwanted text suffixing the number value*
- ***Add*** *2 derived attributes/categories like Year/Month*
- ***Add*** *derived metrics like 'Annual_****Installment*** *Amount' to 'Annual* ***Income****' ratio (i2i) for analysis*
- ***Add*** *fields to dataset by casting AlphaNumeric Codes in 'grade', 'subgrade' columns as "****Int****" for correlation analysis*
- ***Impute*** *Employment Length of borrower with "****mode****" value for ~1000 rows which have NaN*
- ***Manual Reassign*** *Home Ownership from one bin to another to reduce the # of categories*

```
loan.home_ownership.value_counts()

RENT        18448
MORTGAGE    17010
OWN          2970
OTHER          96
NONE            3
Name: home_ownership, dtype: int64
```

```
loan.home_ownership.value_counts()

RENT        18448
MORTGAGE    17010
OWN          2970
OTHER          99
Name: home_ownership, dtype: int64
```

Thank you