

Post Graduate Program - Data Science

In Partnership With Purdue University

Course Project – Machine Learning
Income Qualification Prediction



Submitted by:
Lavkush Singh

Submitted to:
Purdue University – Simplilearn

Agenda


- Introduction
- Dataset Summary
- Exploratory Data Analysis – Column Variables
- Outliers and its Treatment
- Scaling and PCA
- Predictive Model Summary
- Appendix

Introduction

- Latin America's social programs have a hard time ensuring that the right people are given enough aid.
- Therefore, the objective is to identify the level of income qualification needed for the families in Latin America, via Machine Learning Model
- Dataset has 9557 observations for each 143 variable (columns).
- Columns captures info like house materials, number of mobiles each house has, education level of members in house and etc.
- This is **Supervised – Classification** Problem, which 4 levels of income to predict.
- Random Forest is used to report the results, as per the project requirement. Grid Search along with Cross Validation is also used for result optimization.

Dataset Summary

- 9557 observations (rows) of 143 variables (columns) in “train.csv” file.
- Data-type of variables are “int” and “float”. “object” type columns had inconsistent datapoints, which were treated and column dtype was changed to float.
- “target” column is of “int” type, with 4 levels of output.
- “v2a1”, “v18q1” “rez_esc” columns were dropped as they had more than 70% missing values.
- ‘meanneduc’, ‘SQBmeanned’ columns were imputed with respective ‘median’ values as they had less than 0.15% missing values.
- Few variables have skewed distributions, which are taken care in outlier analysis.
- Columns have varied scales of values. All the values are scaled using “MinMax scaler” (except for the columns already having 0 and 1 as values), because not all the variables follow normal distribution in the provided dataset.
- “test.csv” file has more or less the same exploratory data analysis results as that of “train.csv”, therefore, the same treatments of “train.csv” are employed for predictions of “test.csv” variables.



Exploratory Data Analysis

Column Variables

Understanding Data – Datatypes, Dimension, Null Values Summary

```
In [7]: train_data.shape # checking rows and cols of the train dataset
```

```
Out[7]: (9557, 143)
```

```
In [8]: train_data.dtypes.value_counts() # count of distinct datatype c
```

```
Out[8]: int64      130  
float64         8  
object          5  
dtype: int64
```

```
In [9]: test_data.shape # checking rows and cols of the test dataset
```

```
Out[9]: (23856, 142)
```

```
In [10]: test_data.dtypes.value_counts()
```

```
Out[10]: int64      129  
float64         8  
object          5  
dtype: int64
```


Understanding Data – Datatypes, Dimension, Null Values Summary

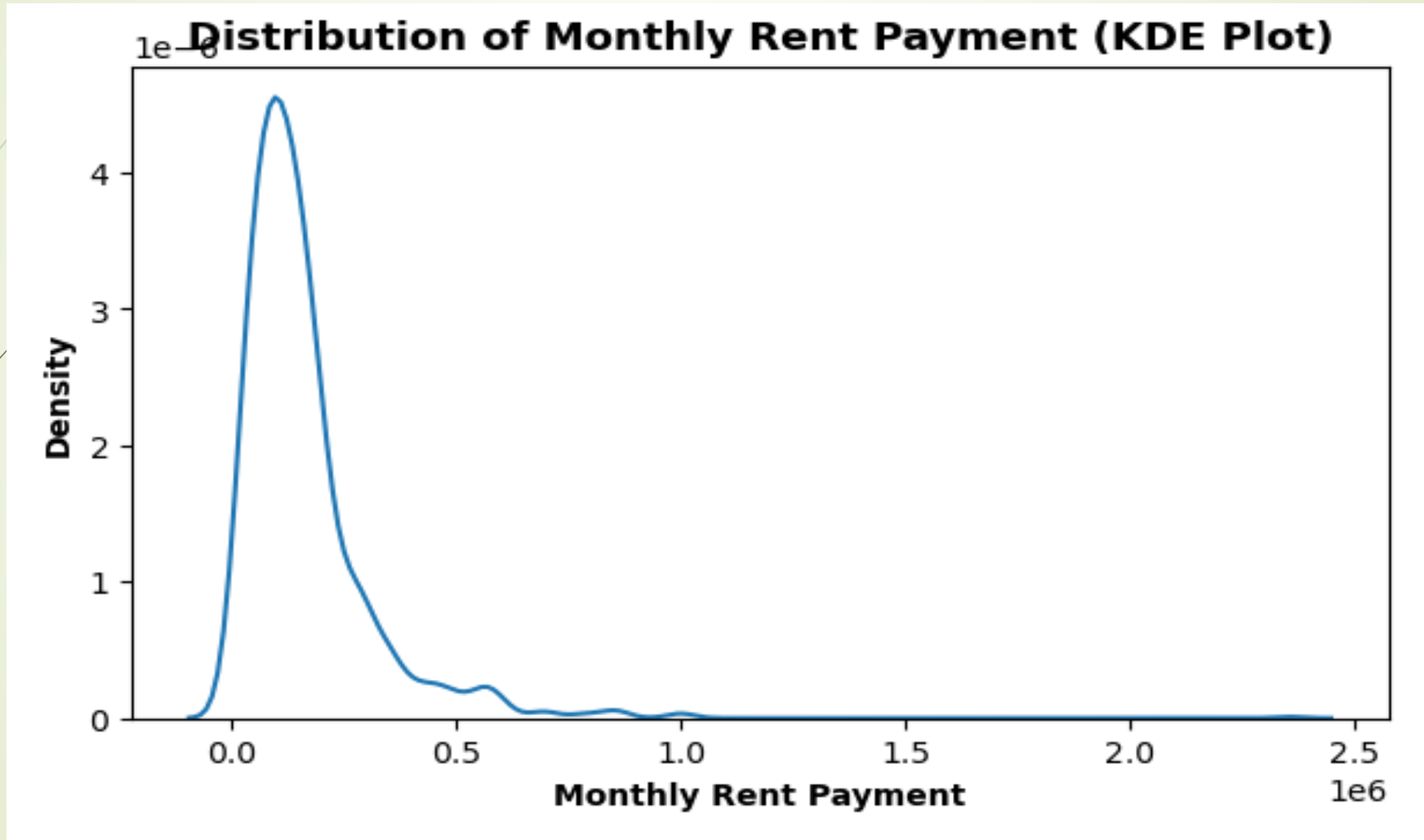
```
In [78]: train_data.isnull().sum()[train_data.isnull().sum() > 0]/train_data.shape[0]*100
```

```
Out[78]: v2a1          71.779847  
         v18q1        76.823271  
         rez_esc      82.954902  
         meaneduc      0.052318  
         SQBmeaned     0.052318  
         dtype: float64
```

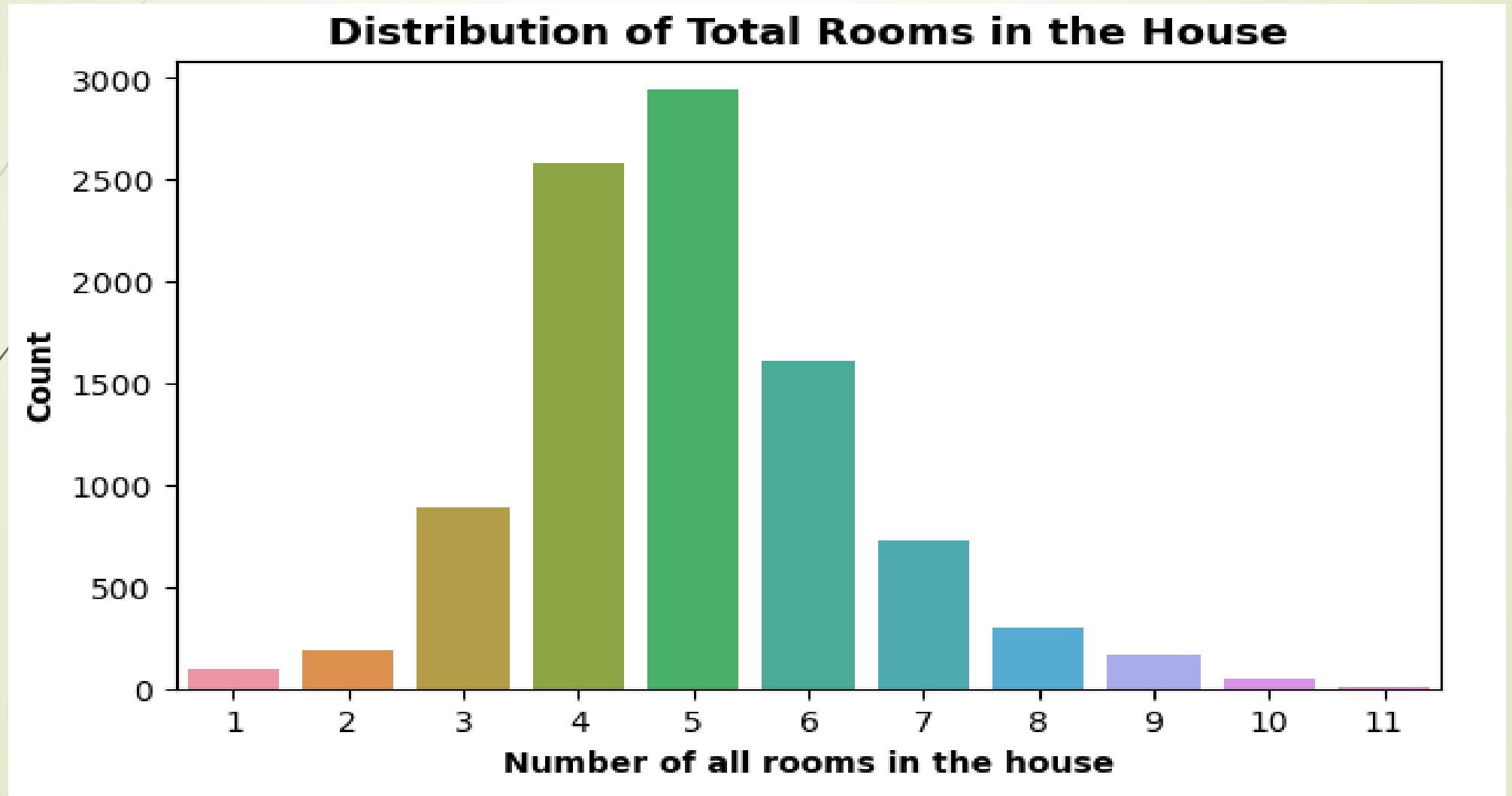
```
In [79]: test_data.isnull().sum()[test_data.isnull().sum() > 0]/test_data.shape[0]*100 # %
```

```
Out[79]: v2a1          72.950201  
         v18q1        75.980885  
         rez_esc      82.381791  
         meaneduc      0.129946  
         SQBmeaned     0.129946  
         dtype: float64
```

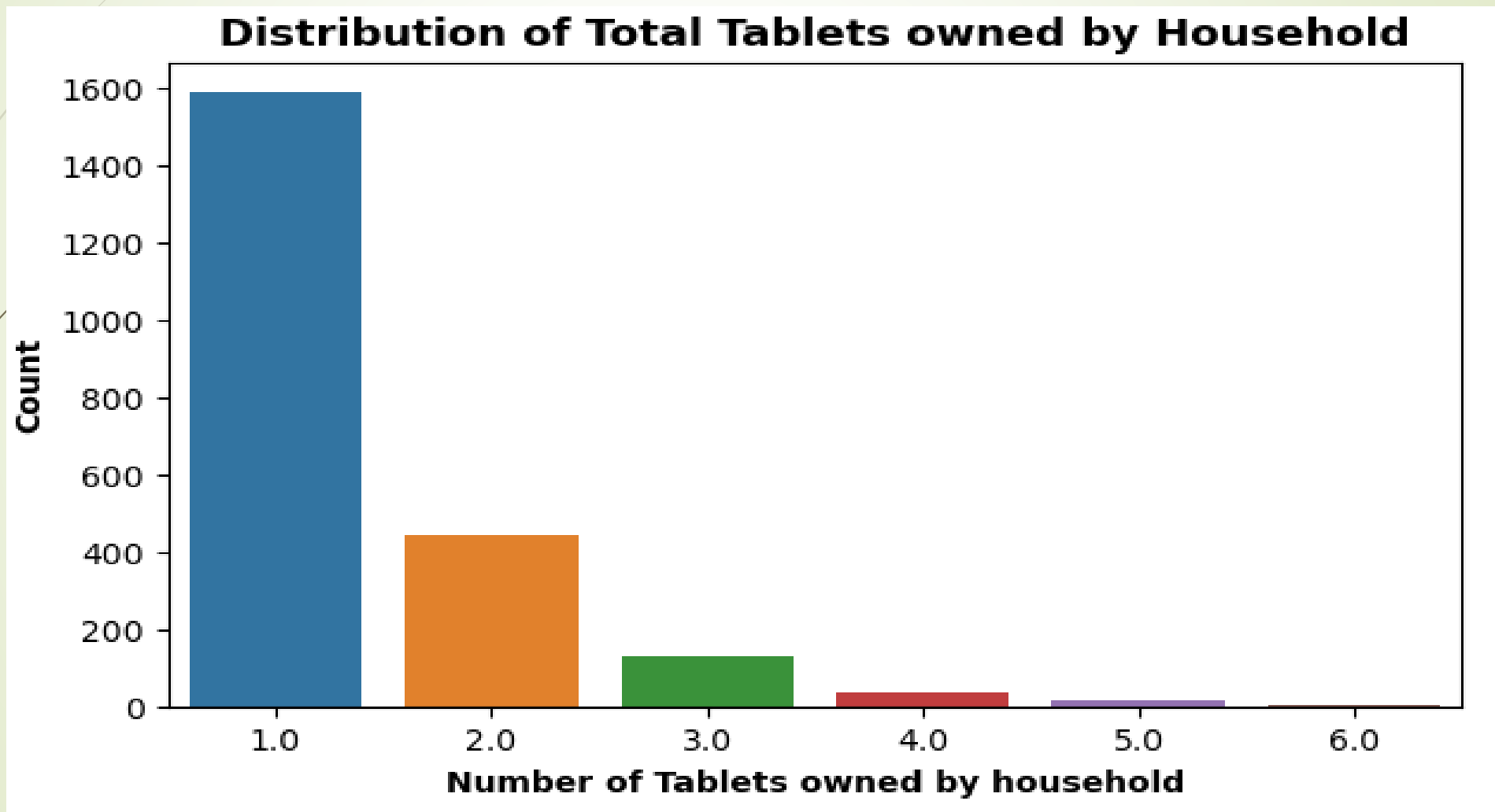
Distribution – ‘v2a1’



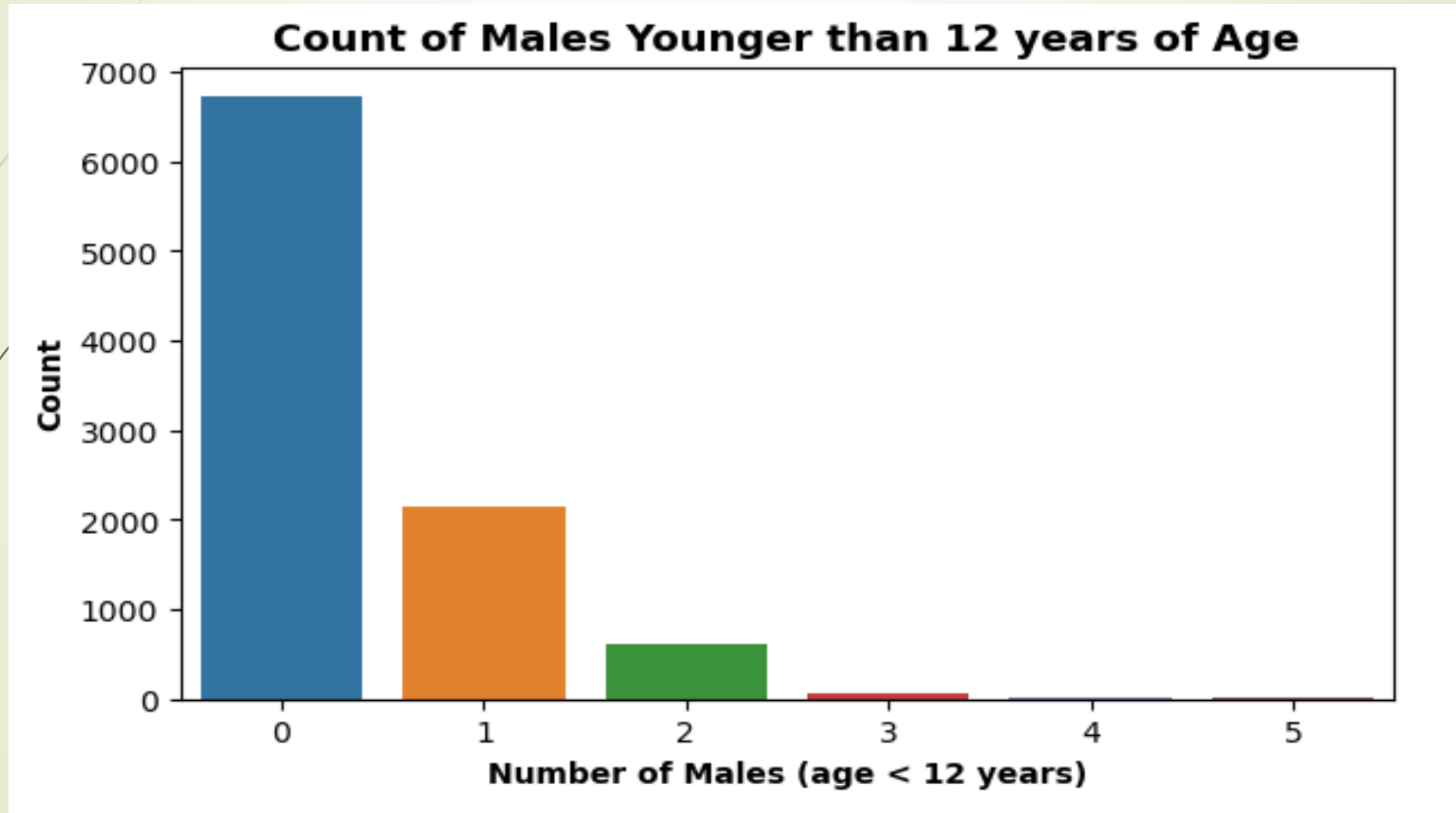
Distribution – ‘rooms’



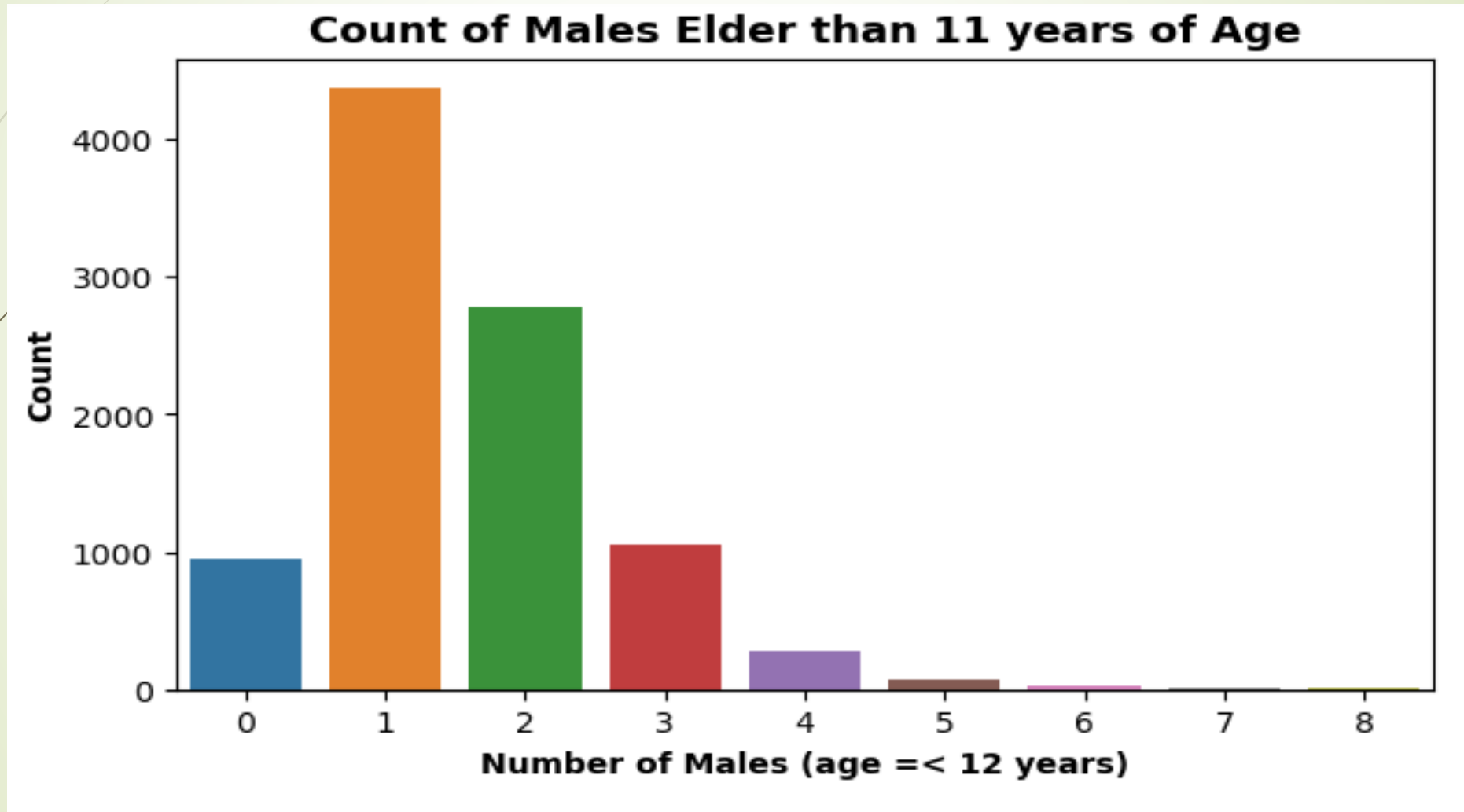
Distribution – ‘v18q1’



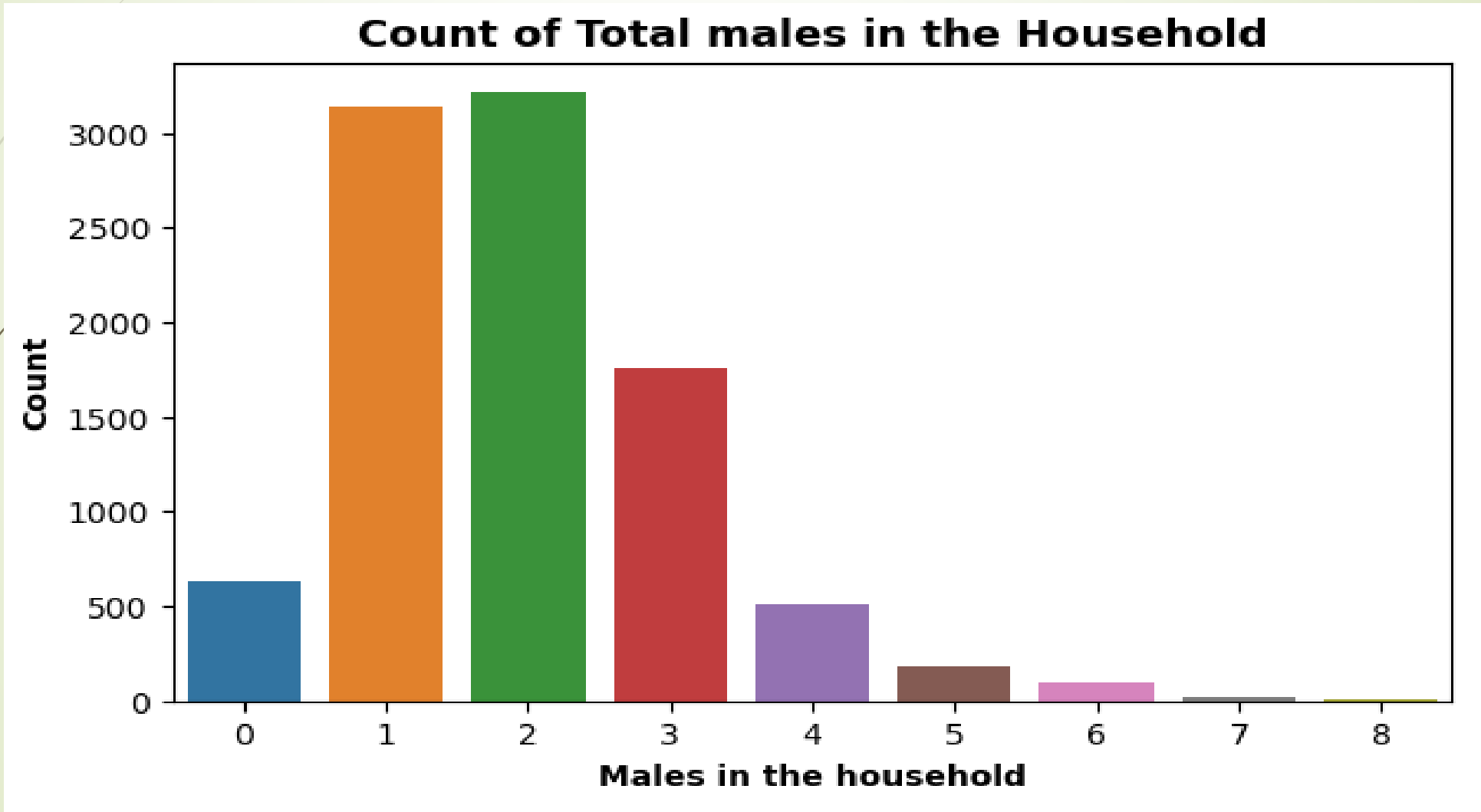
Distribution – 'r4h1'



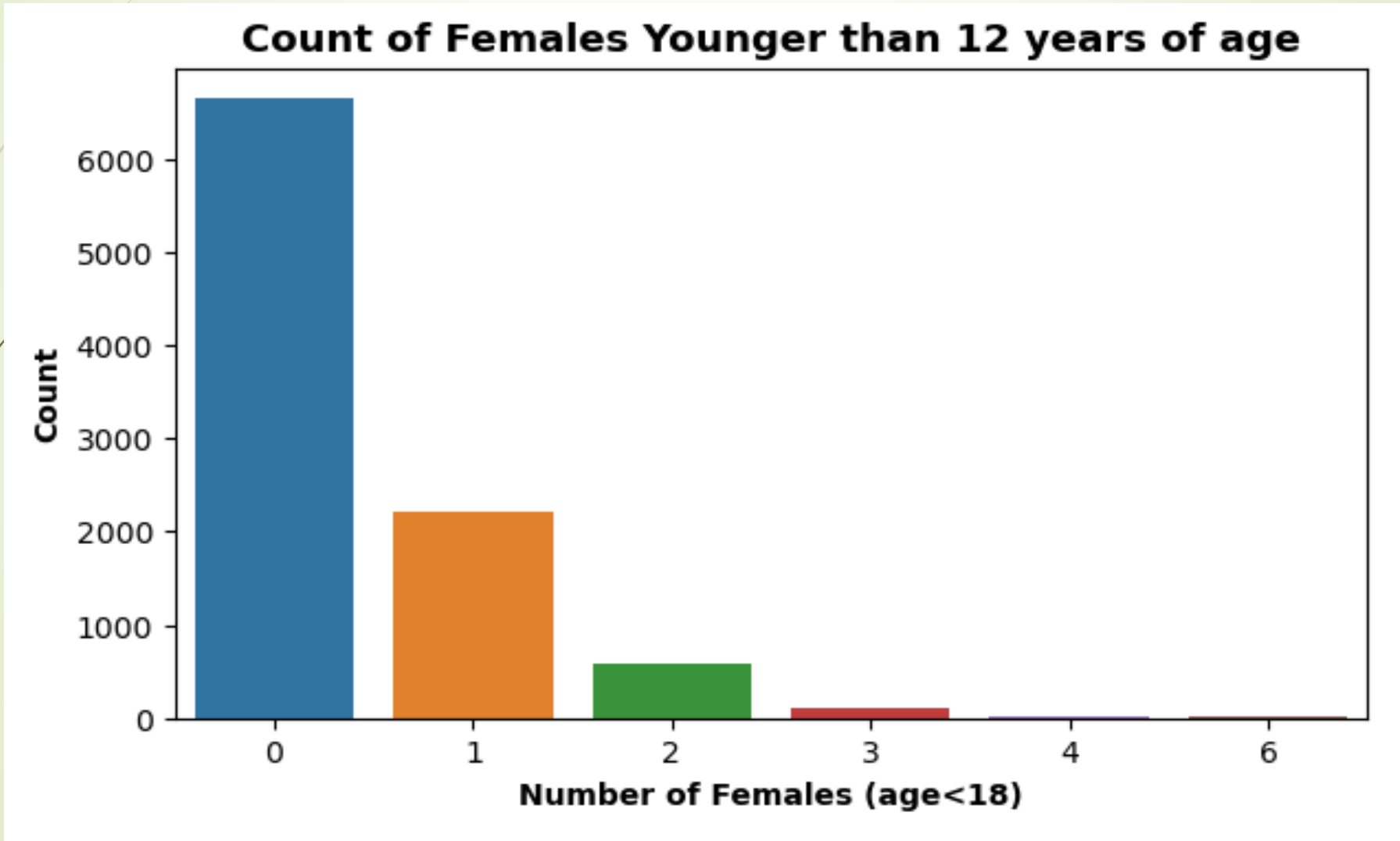
Distribution – 'r4h2'



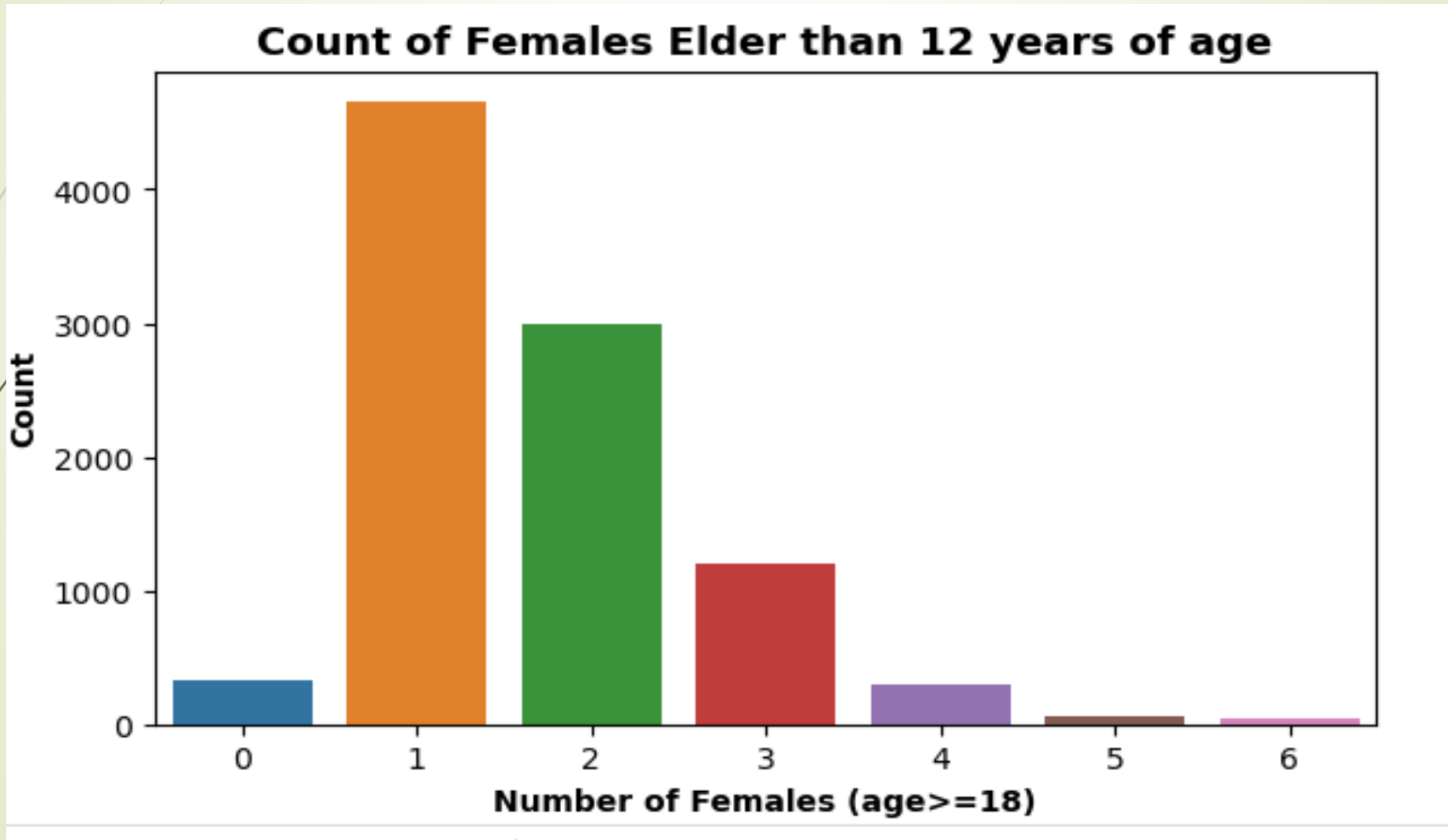
Distribution – 'r4h3'



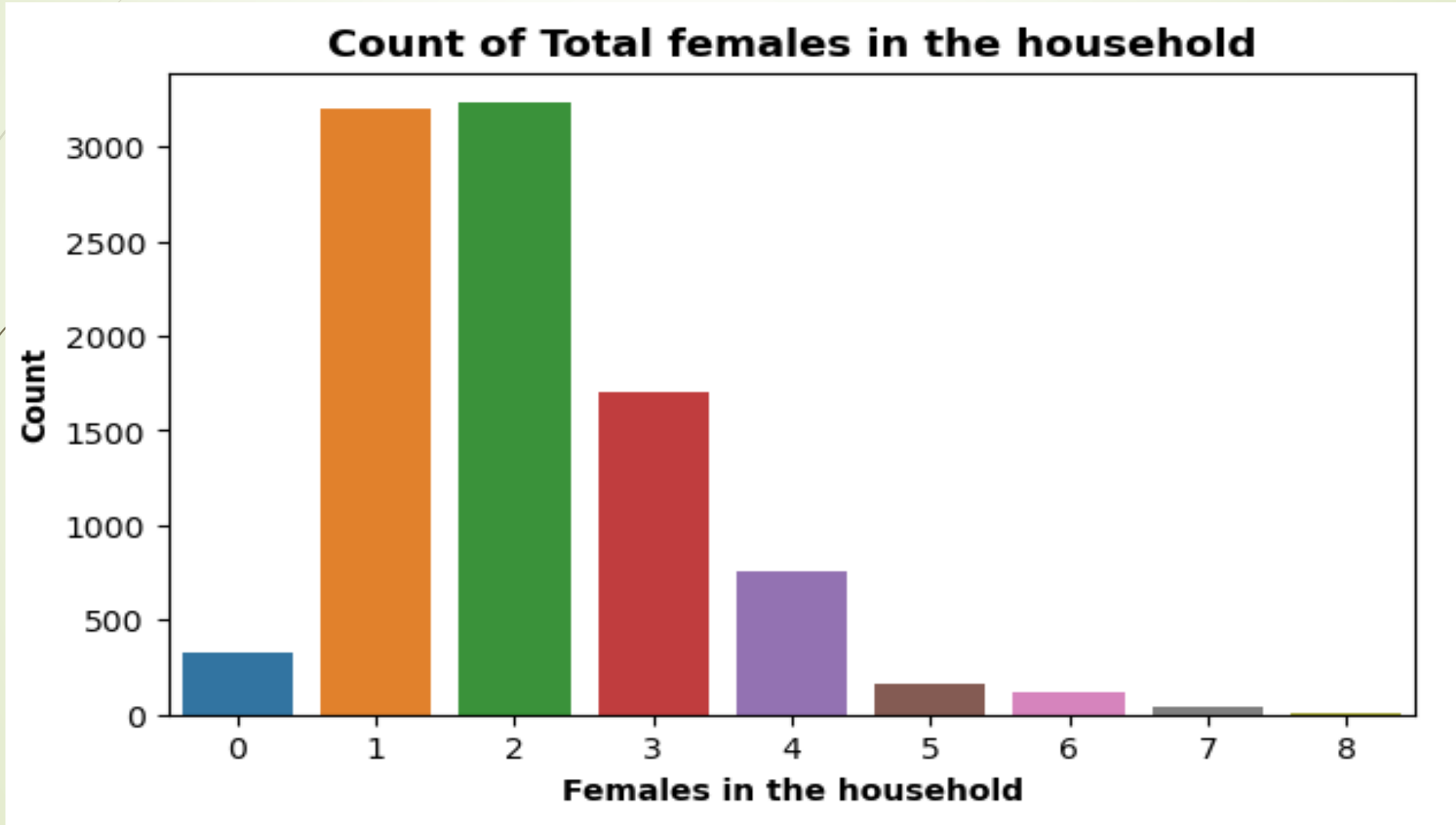
Distribution – 'r4m1'



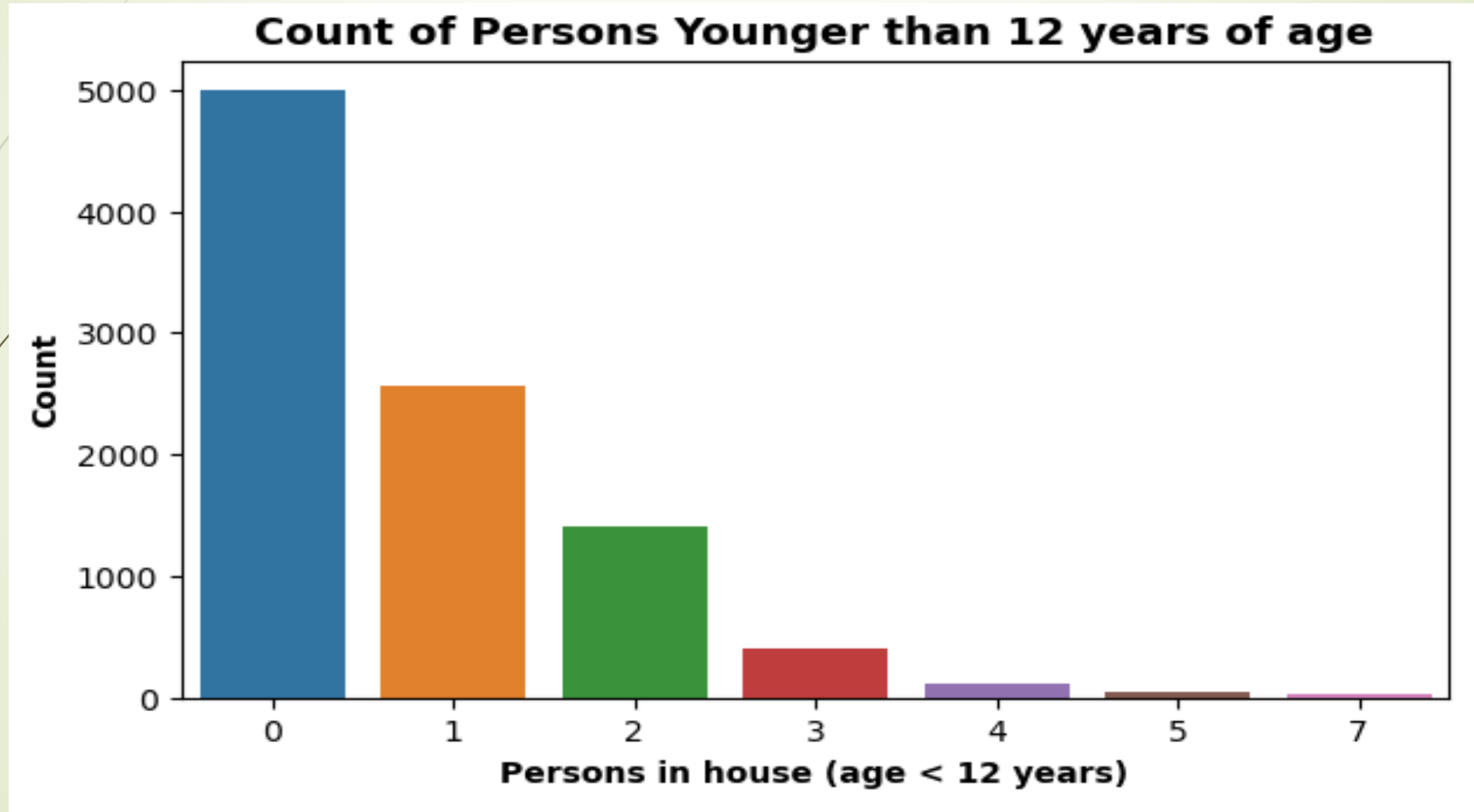
Distribution – 'r4m2'



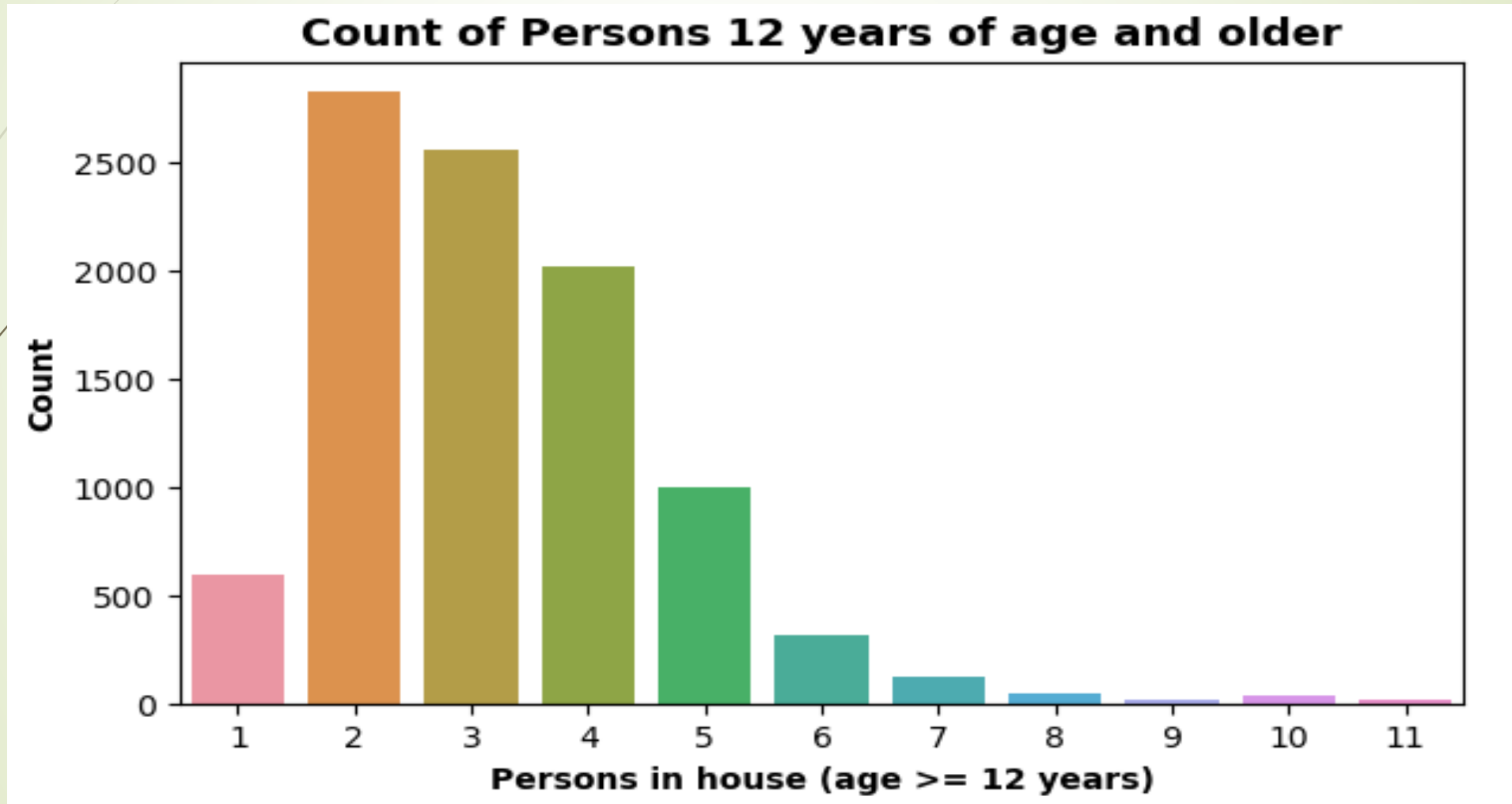
Distribution – 'r4m3'



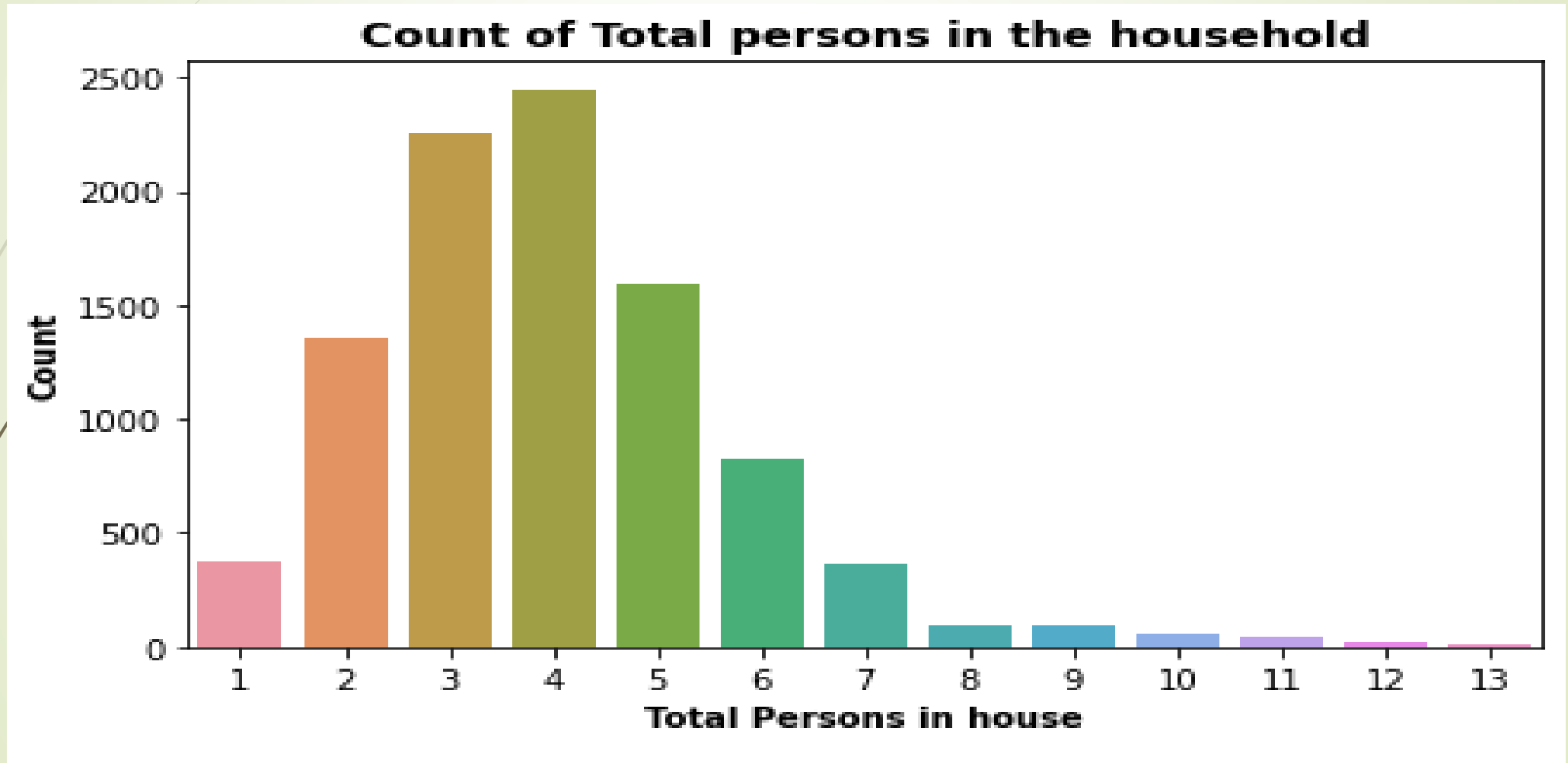
Distribution – 'r4t1'



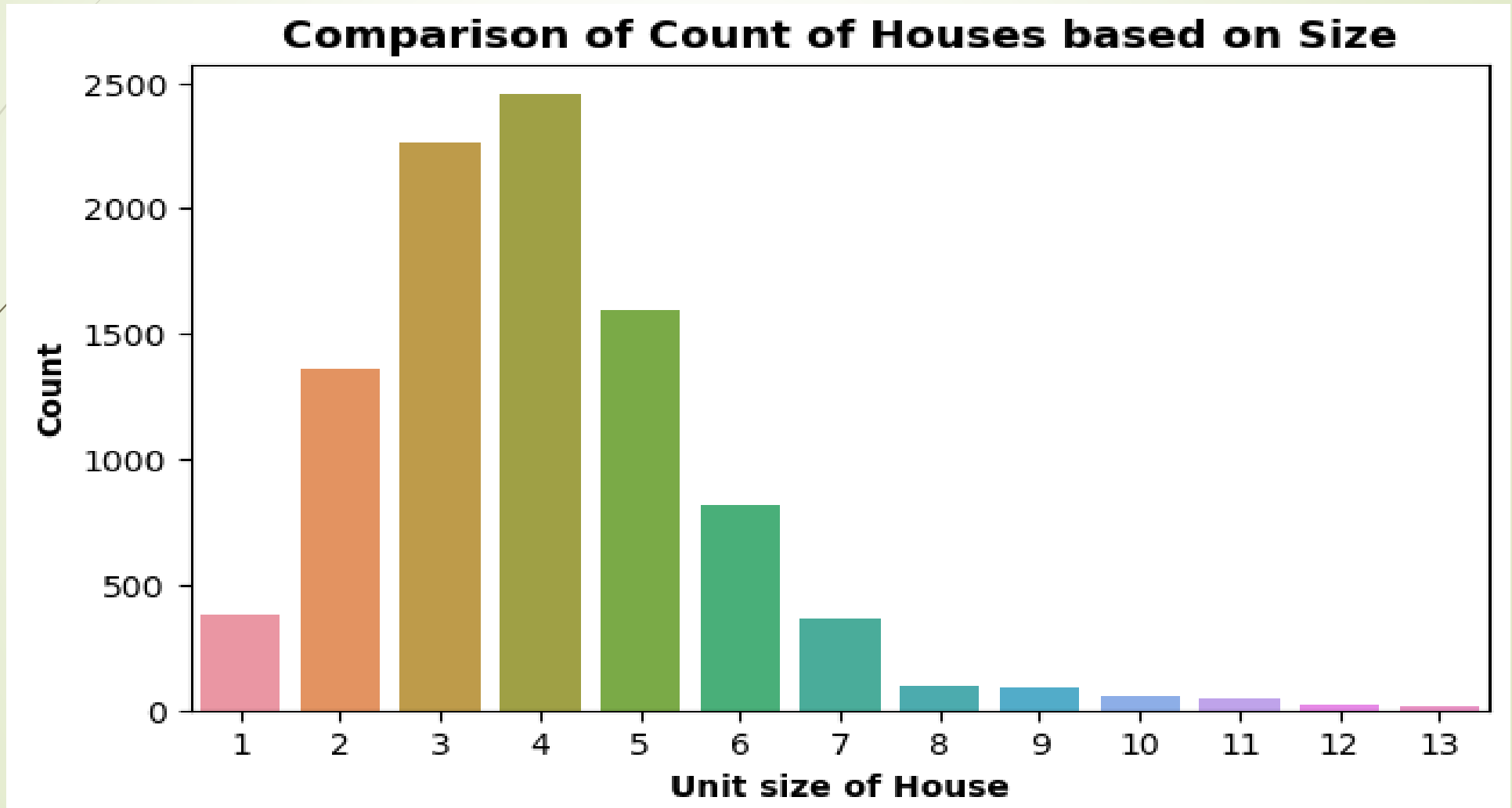
Distribution – 'r4t2'



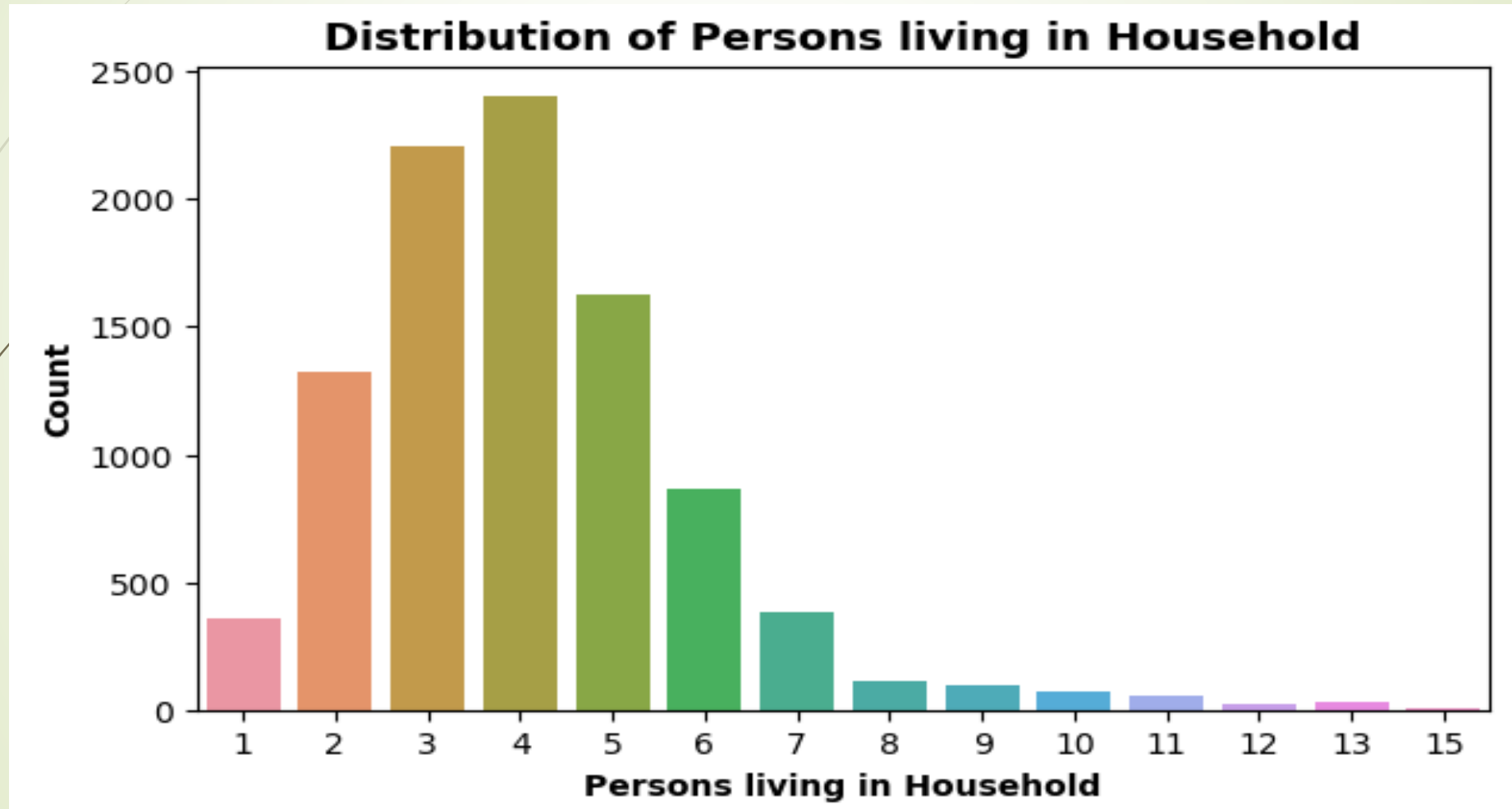
Distribution – 'r4t3'



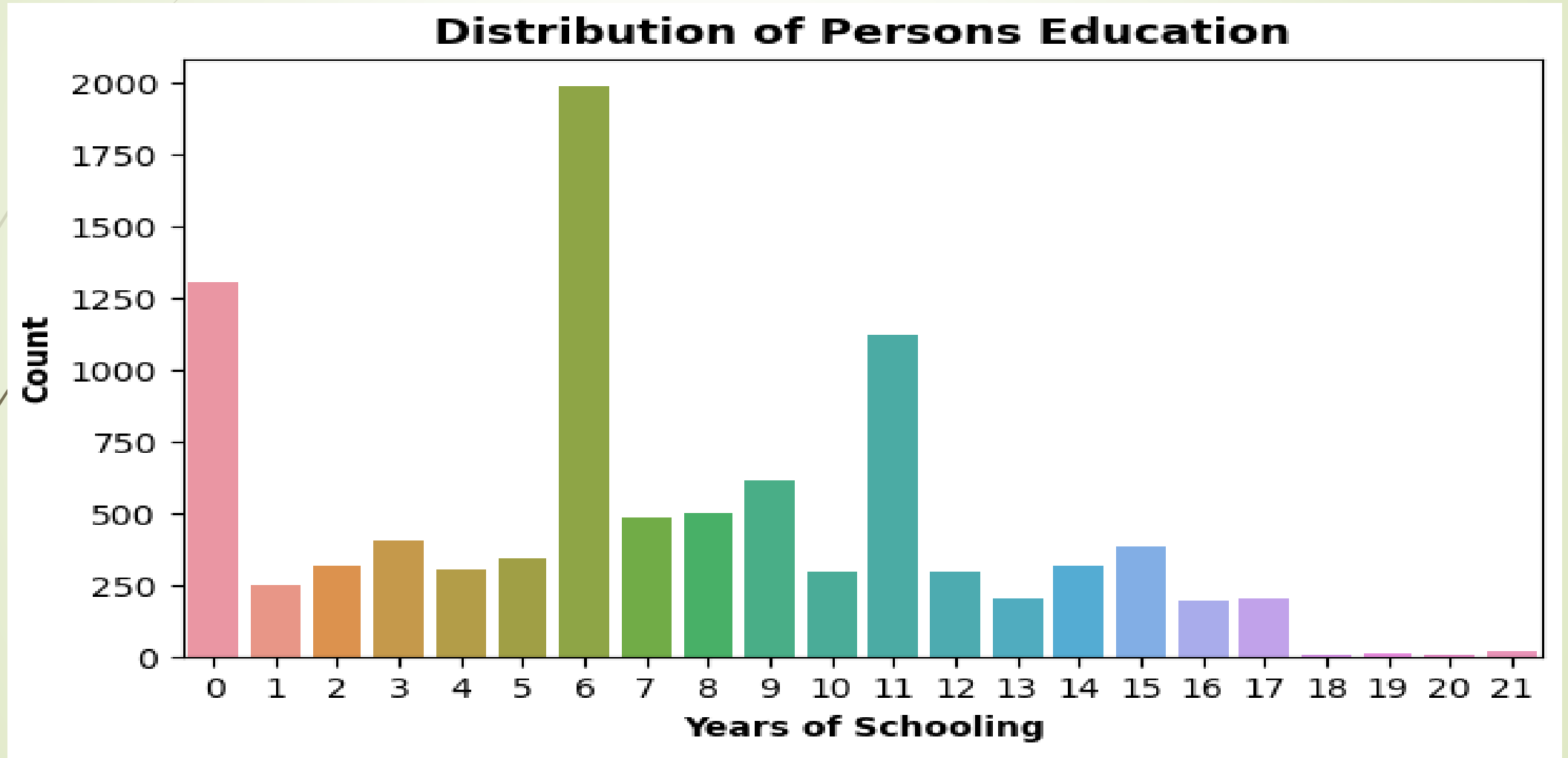
Distribution – ‘tamhog’



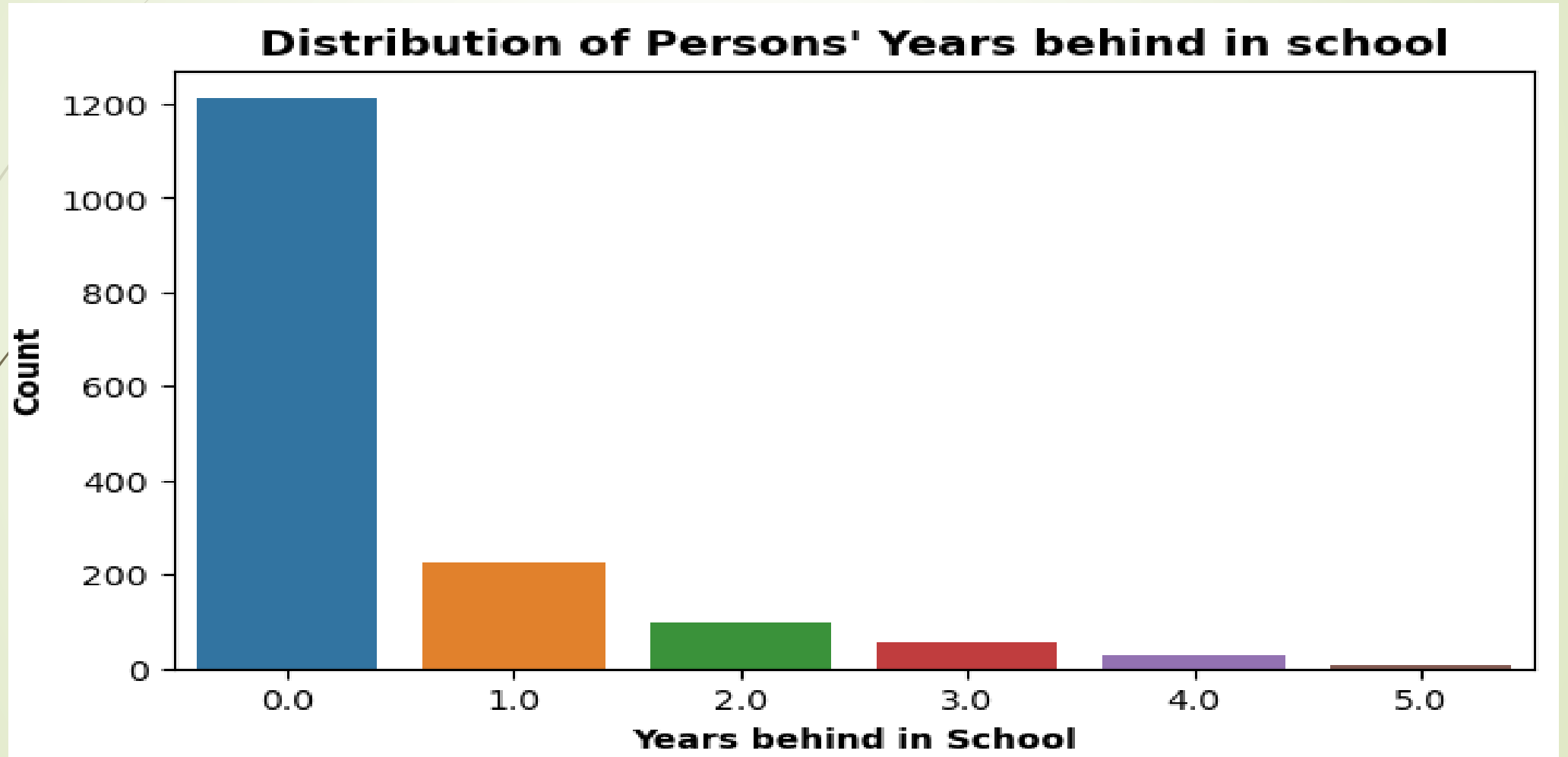
Distribution – ‘tamviv’



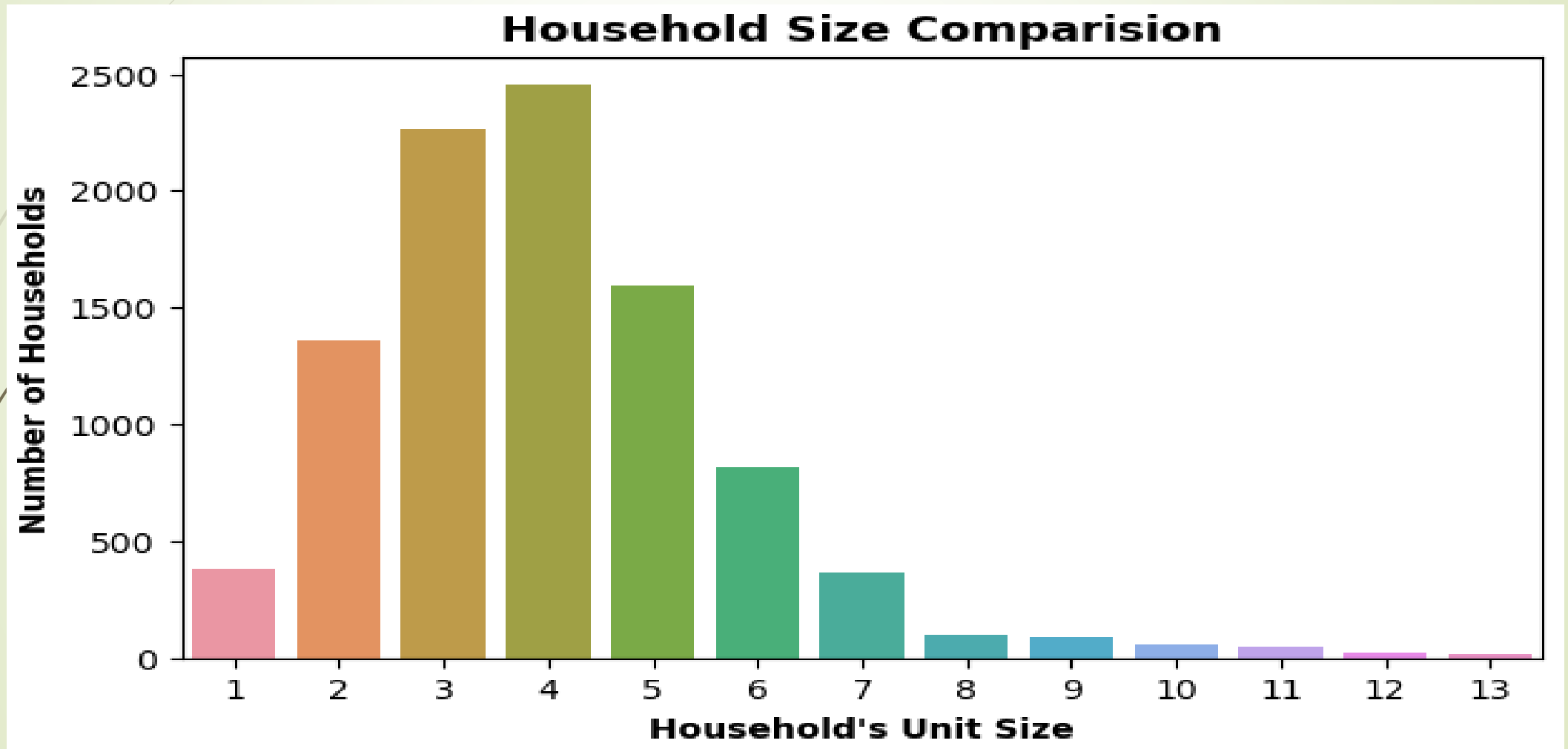
Distribution – ‘escolari’



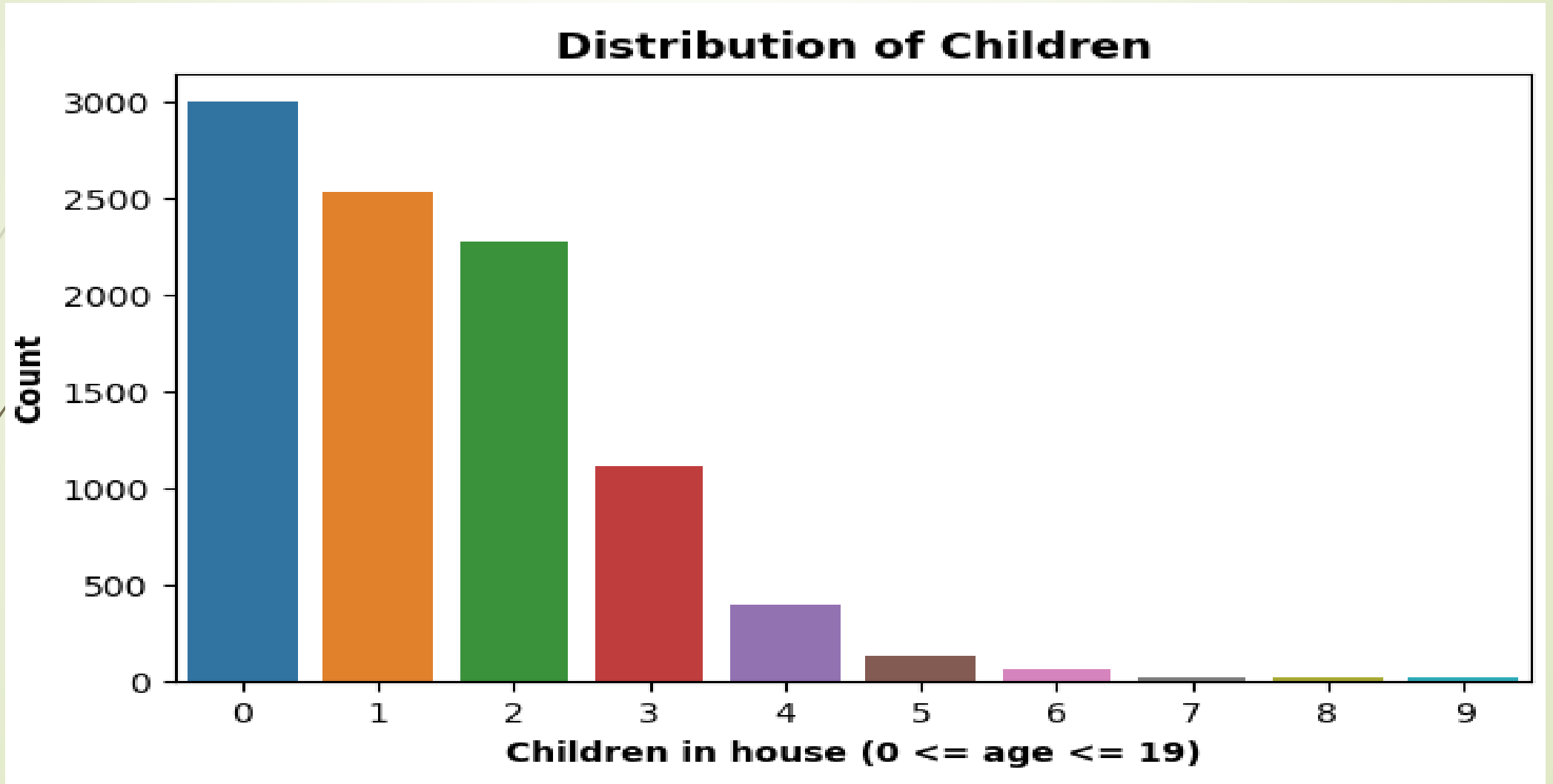
Distribution – 'rez_esc'



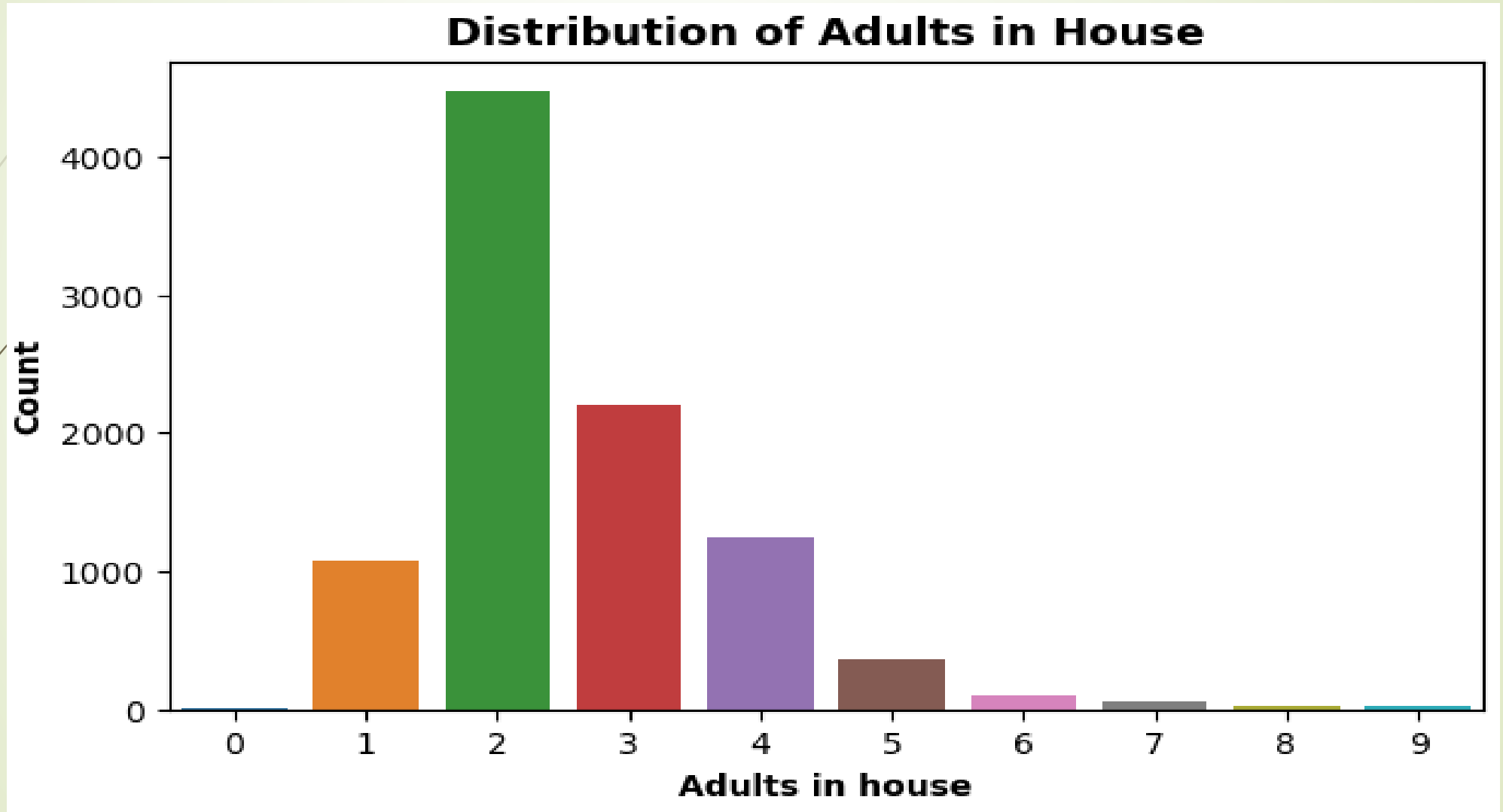
Distribution – 'hhsized'



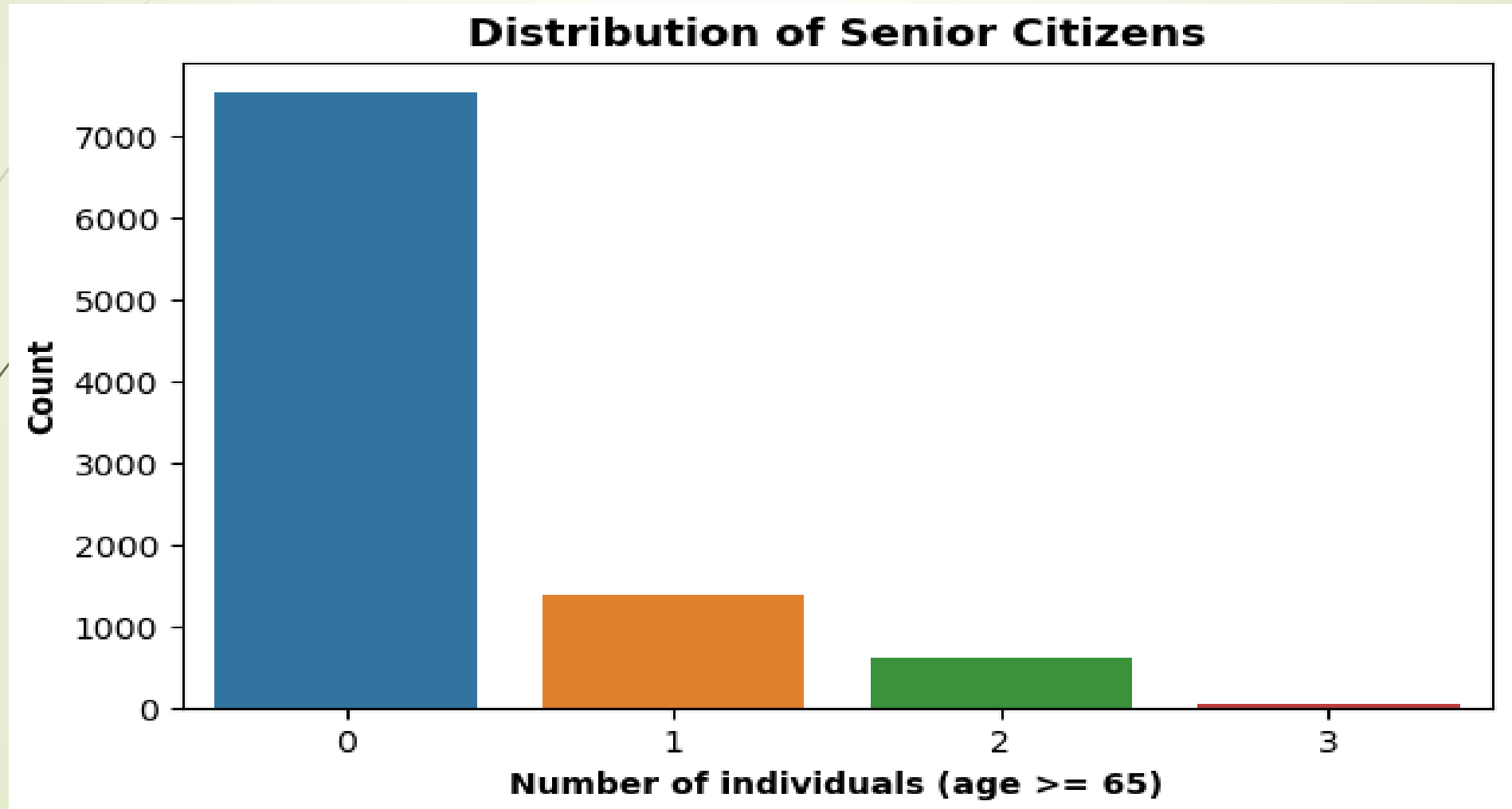
Distribution – 'hogar_nin'



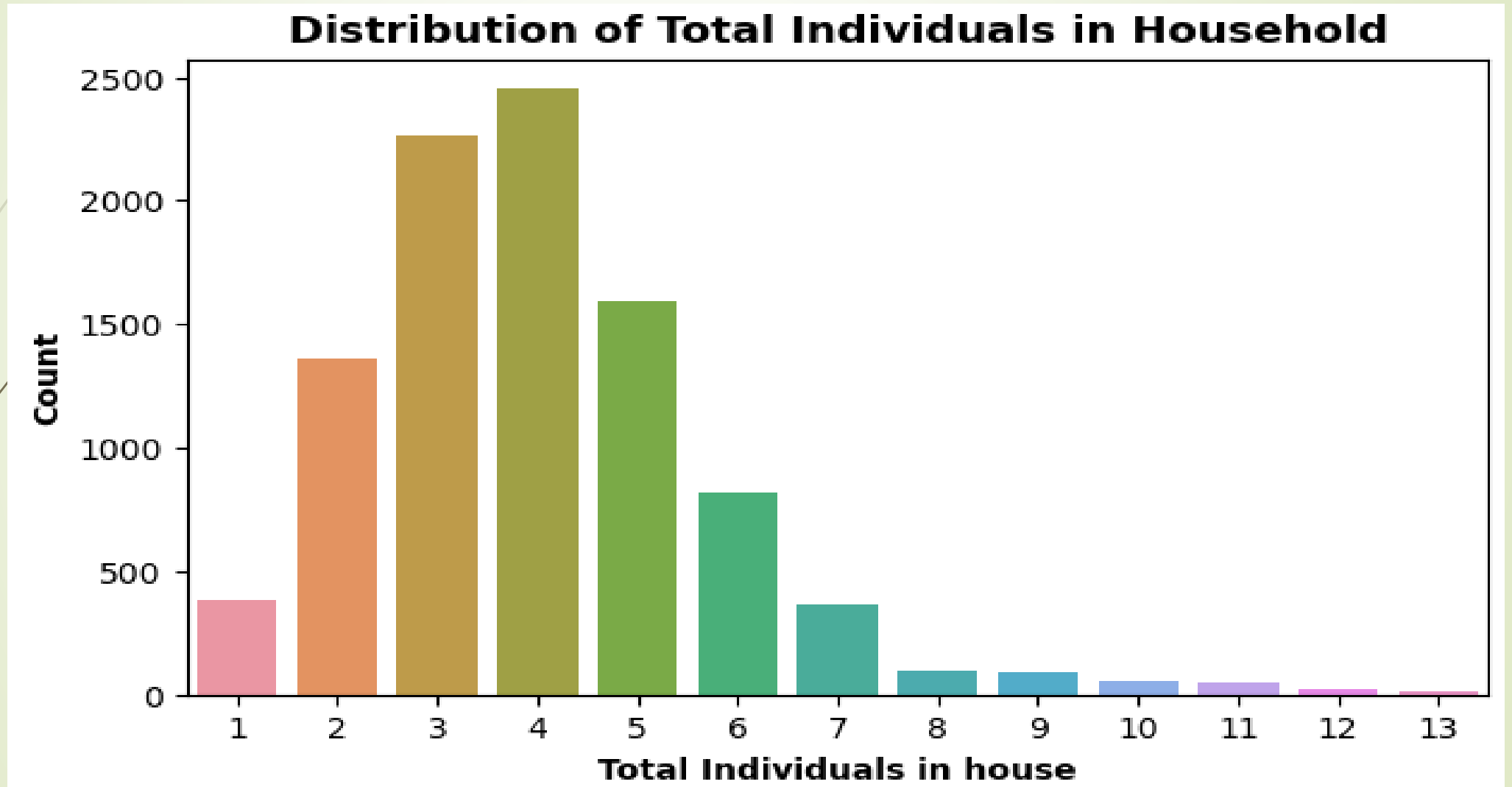
Distribution – 'hogar_adul'



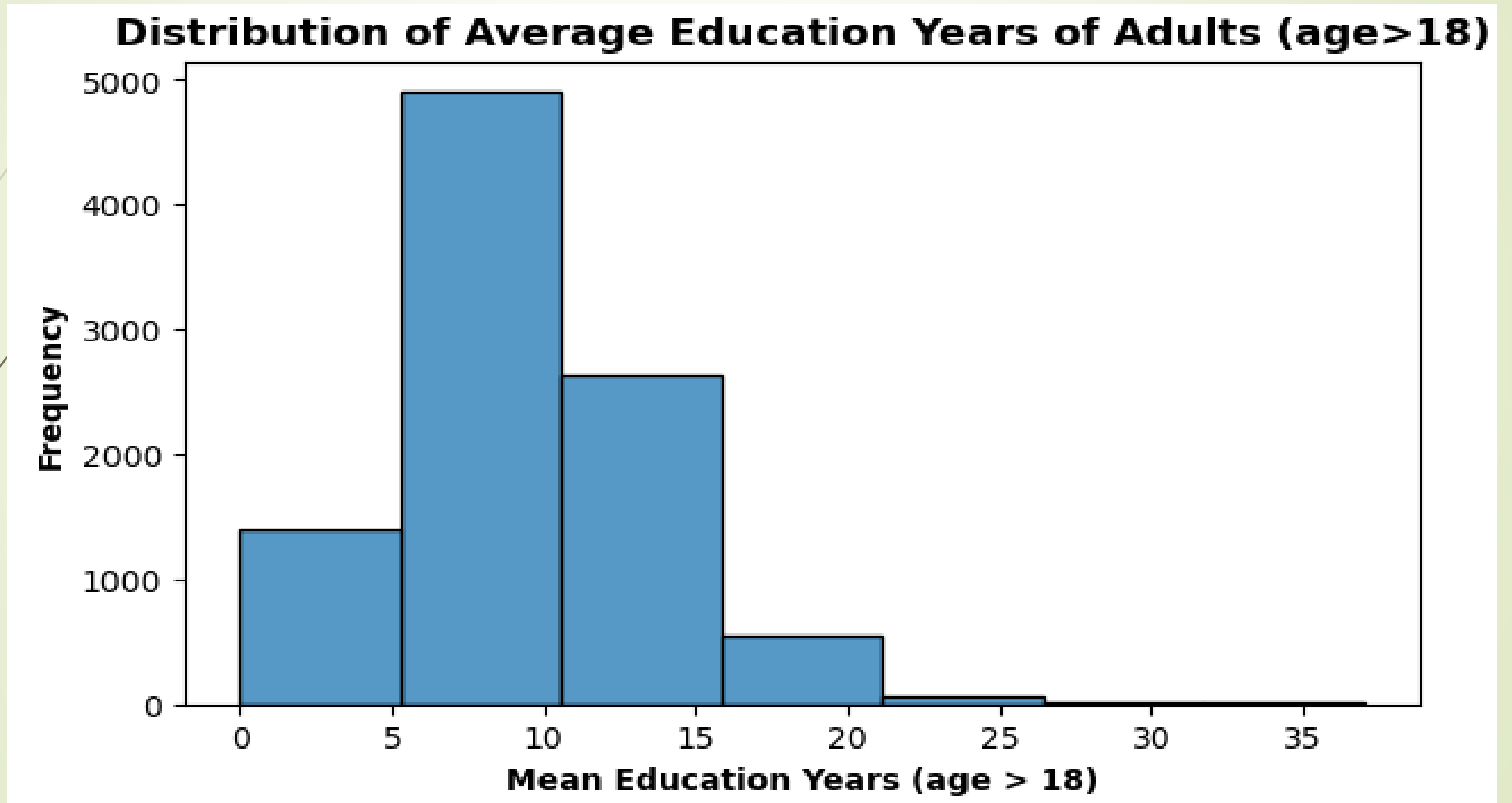
Distribution – 'hogar_mayor'



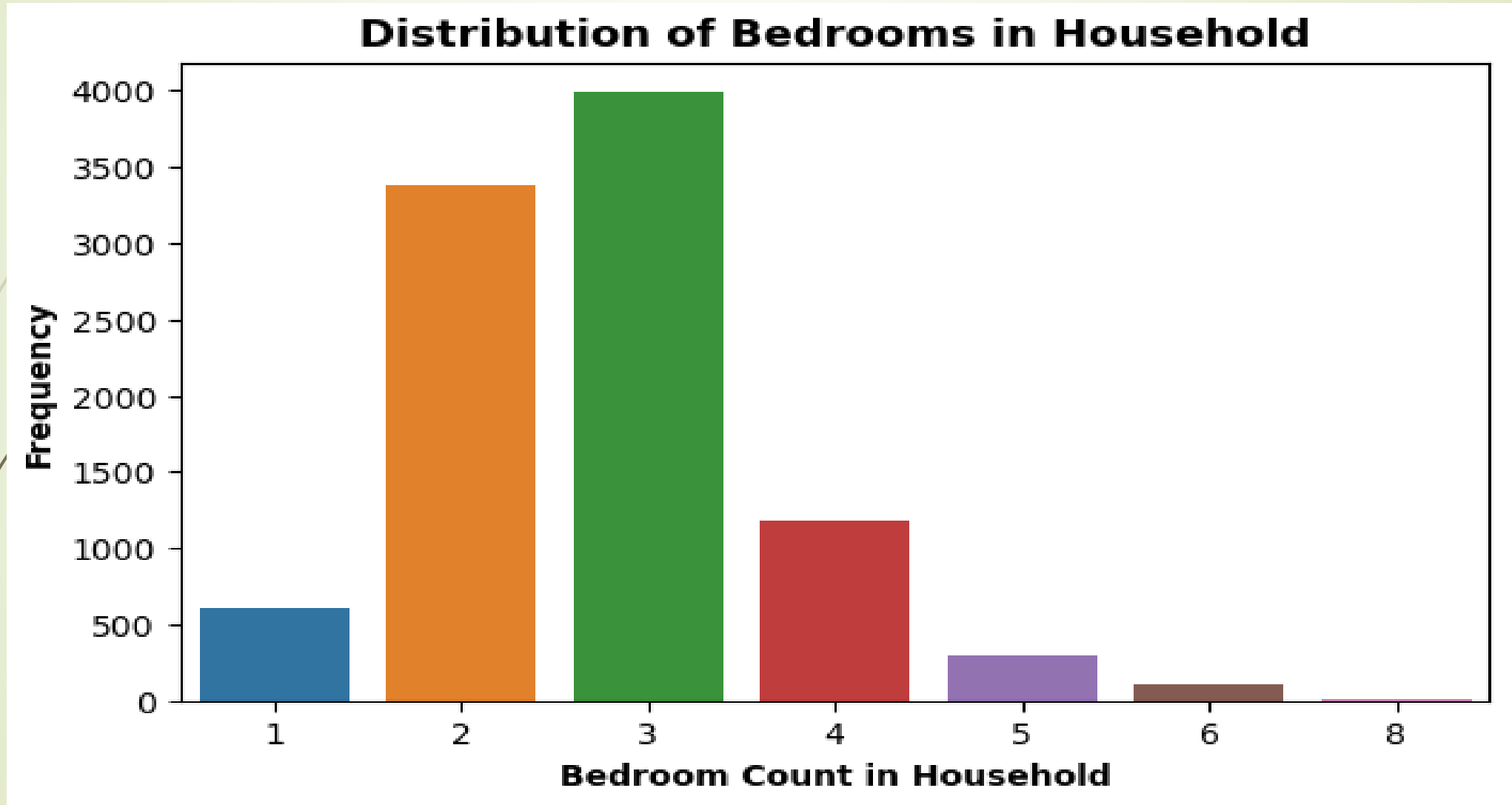
Distribution – 'hogar_total'



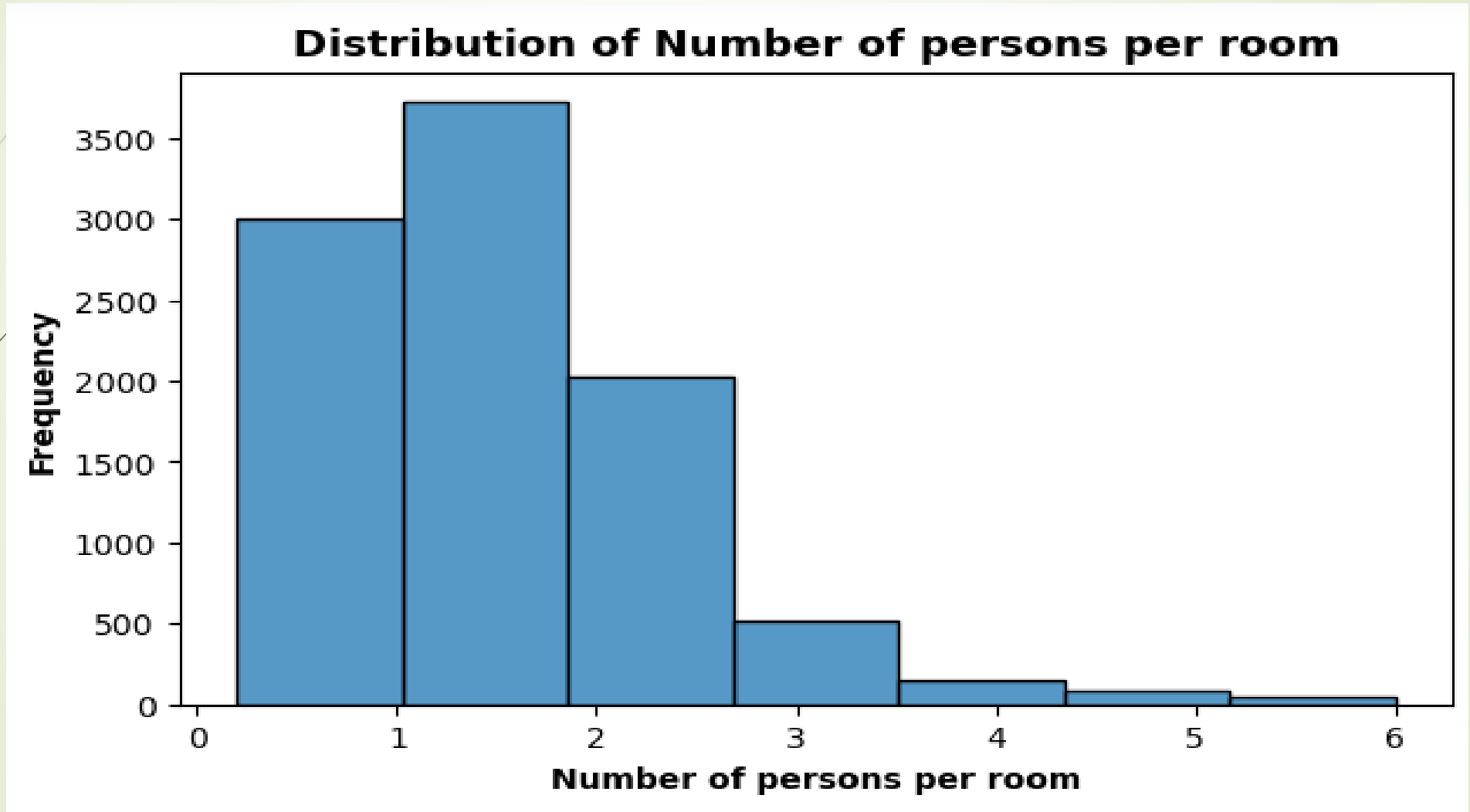
Distribution – ‘meanneduc’



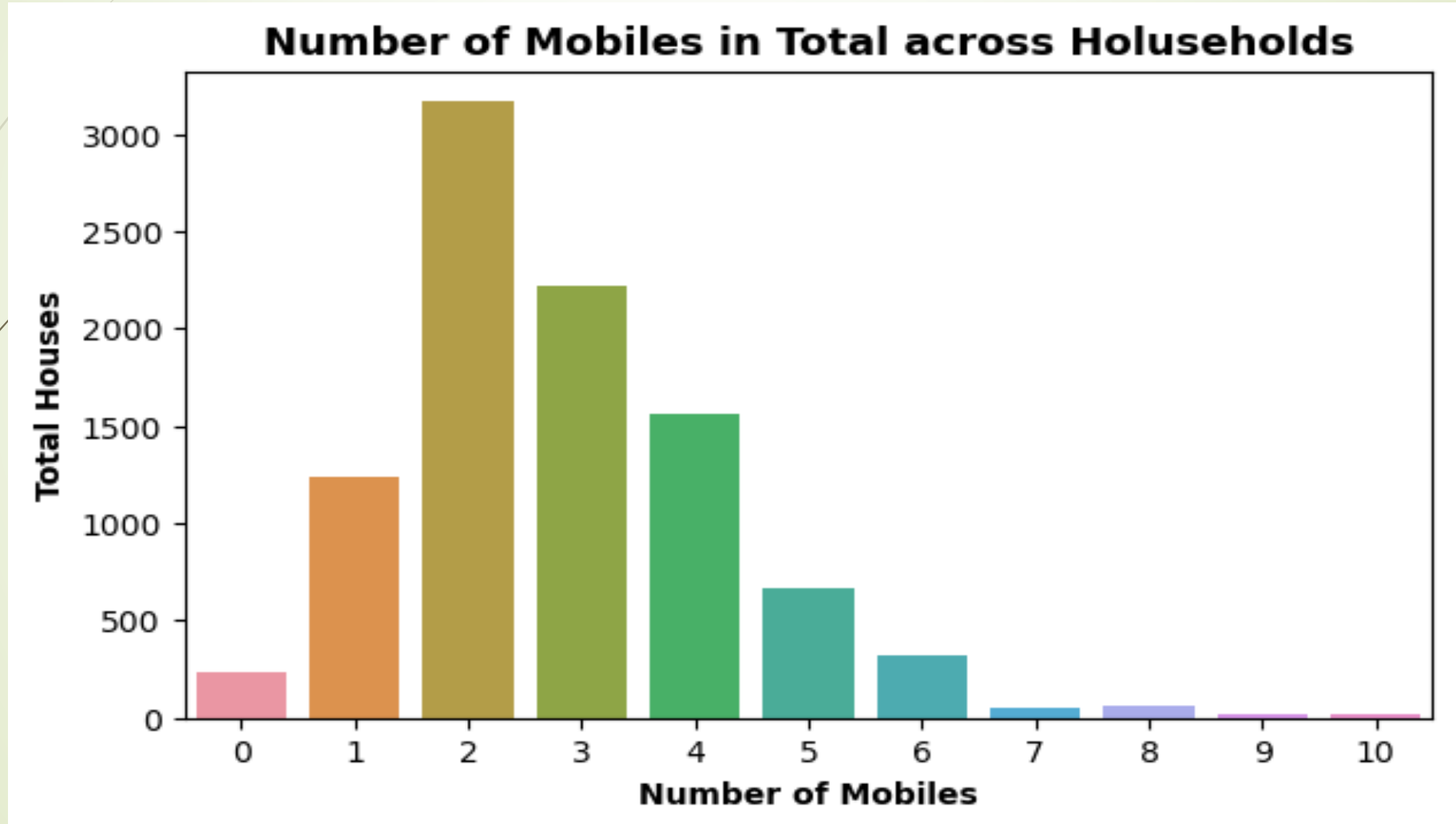
Distribution – ‘bedrooms’



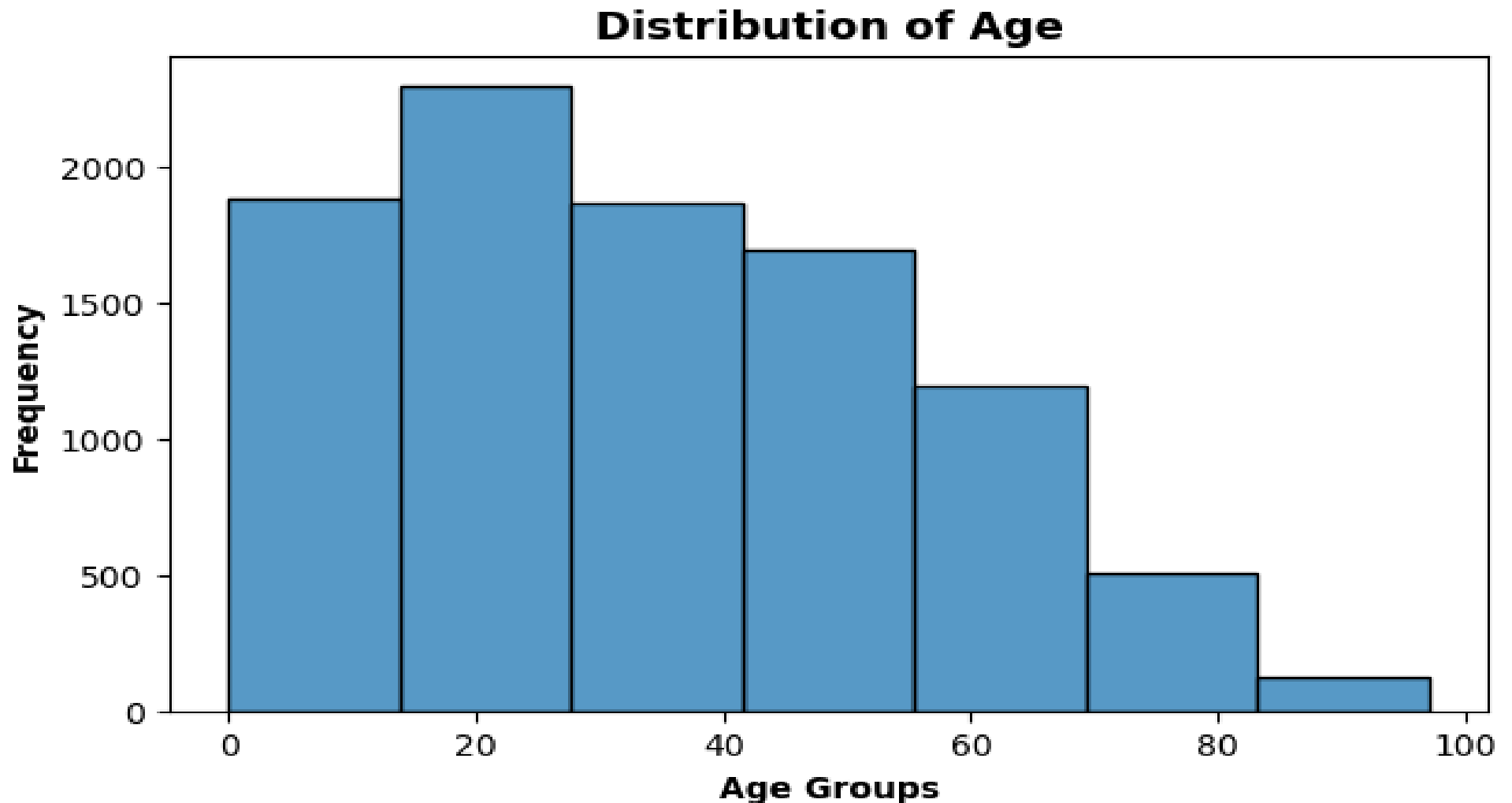
Distribution – ‘overcrowding’



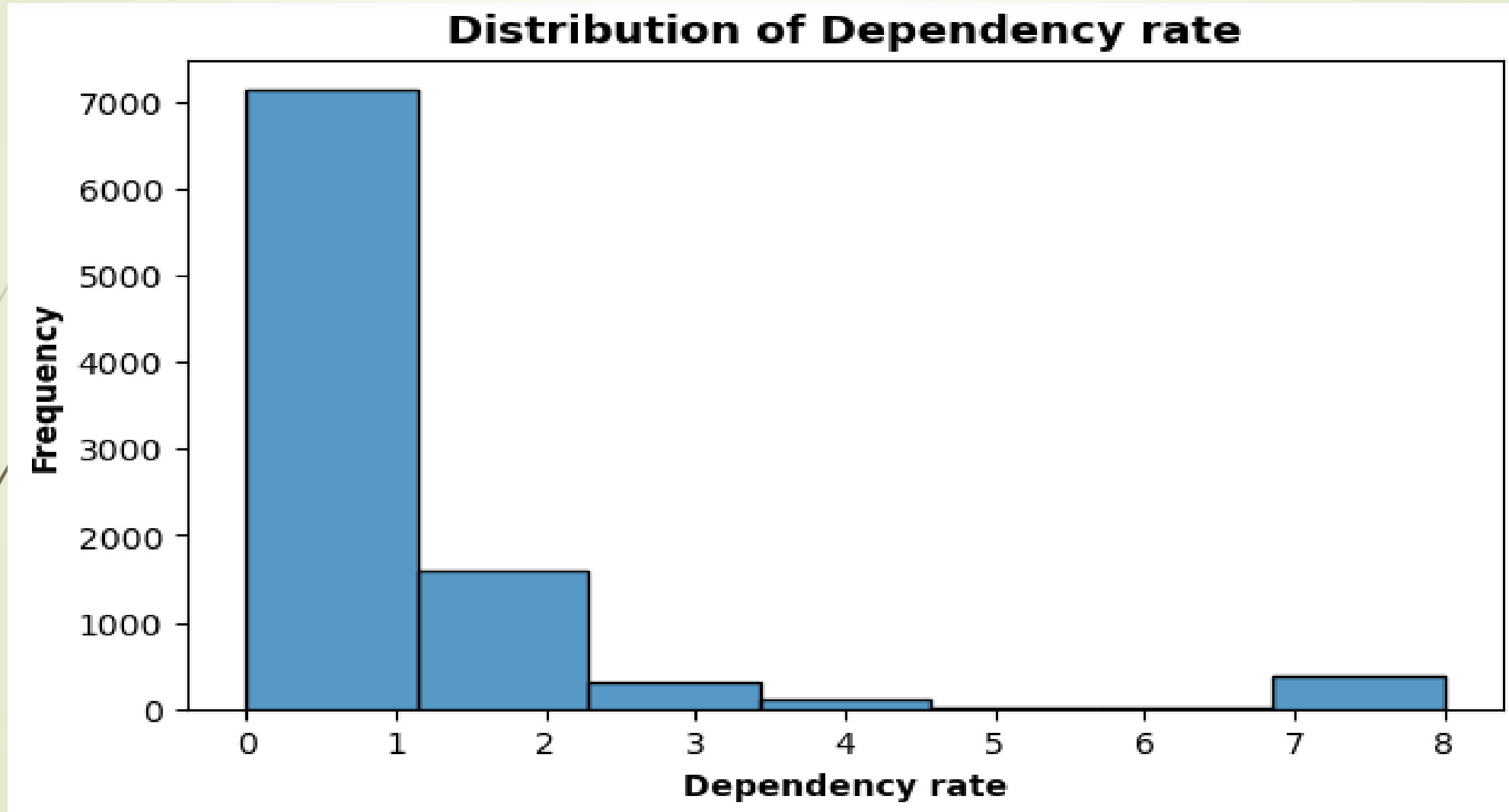
Distribution – ‘qmobilephone’



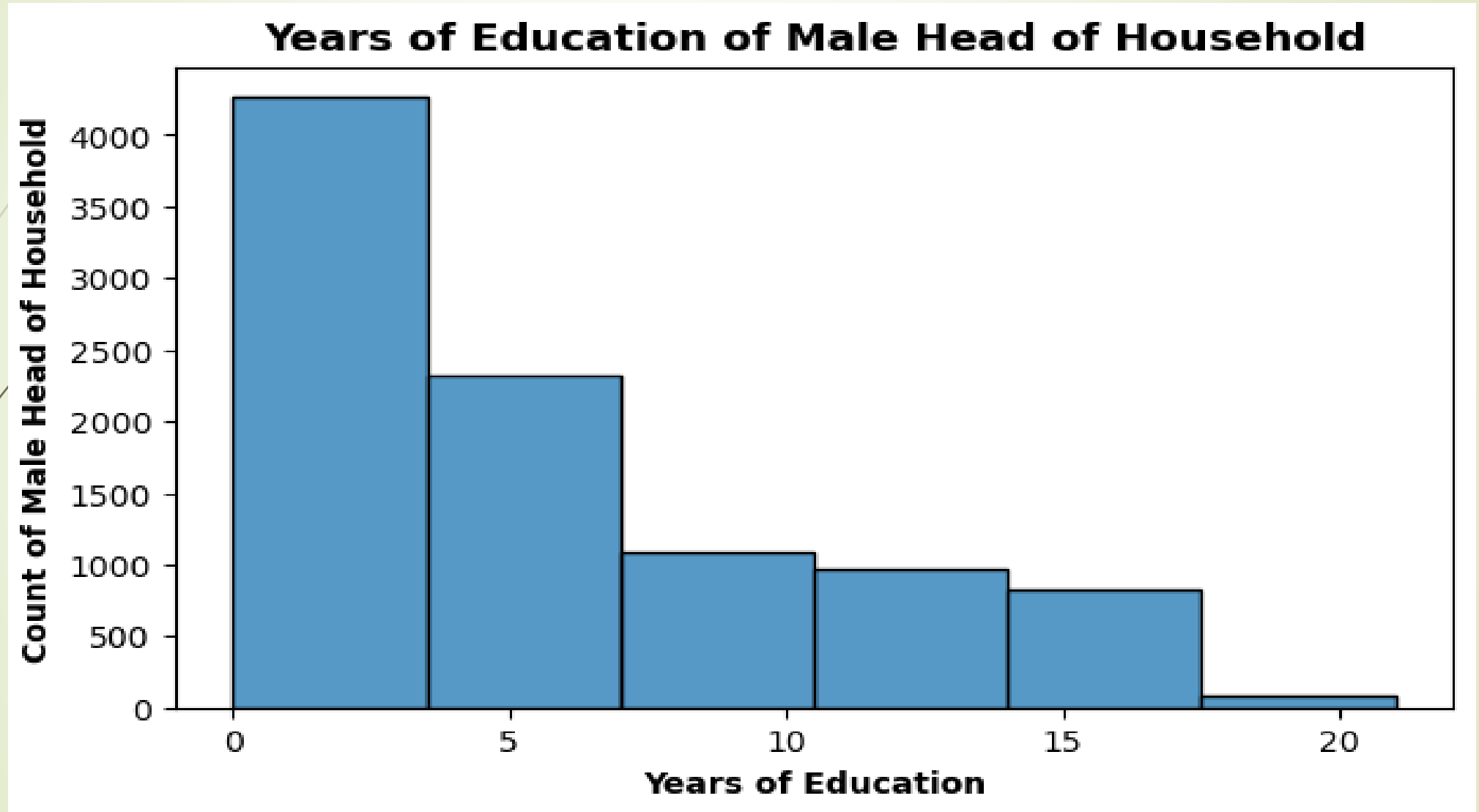
Distribution – ‘age’



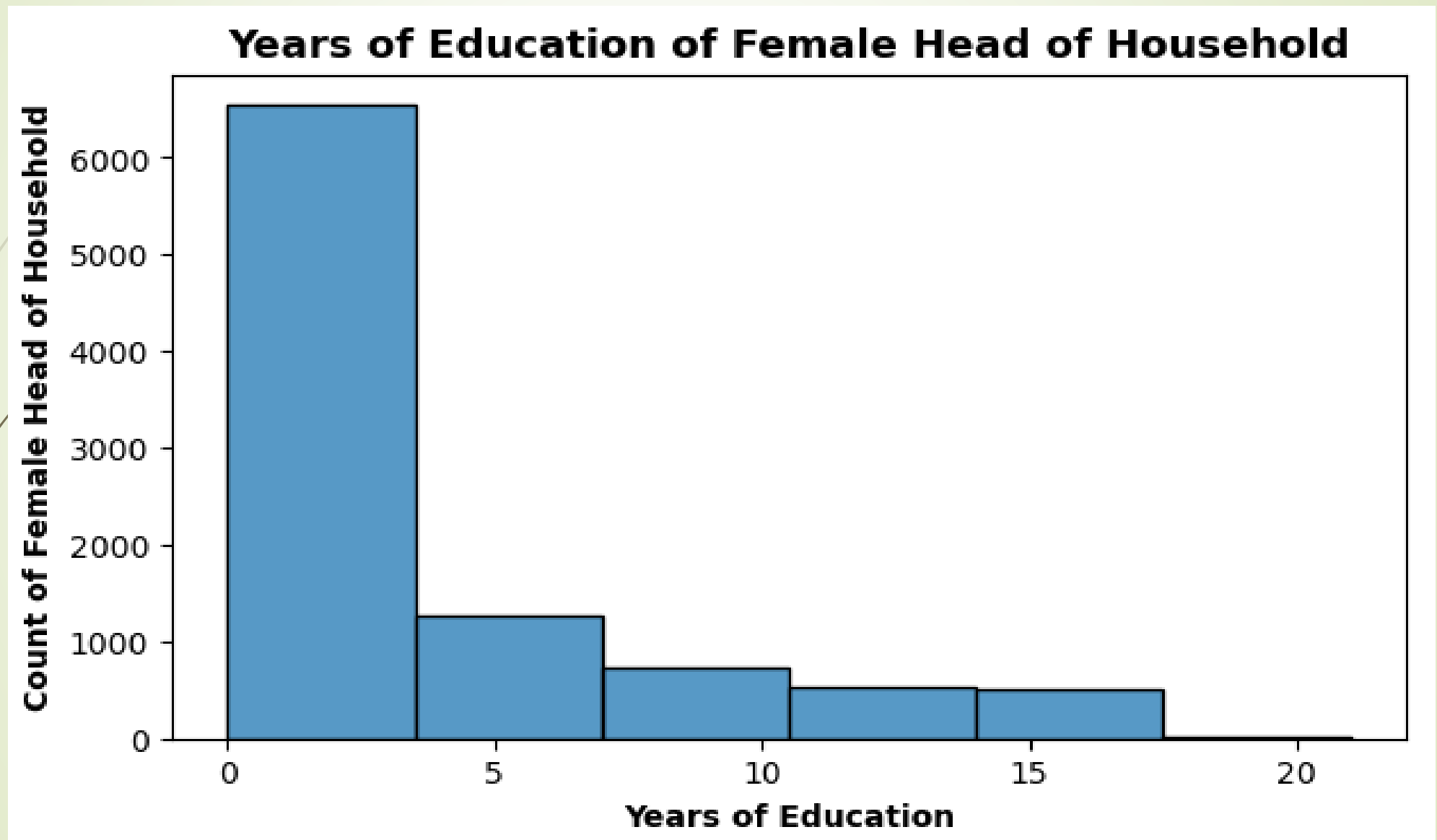
Distribution – ‘dependency’



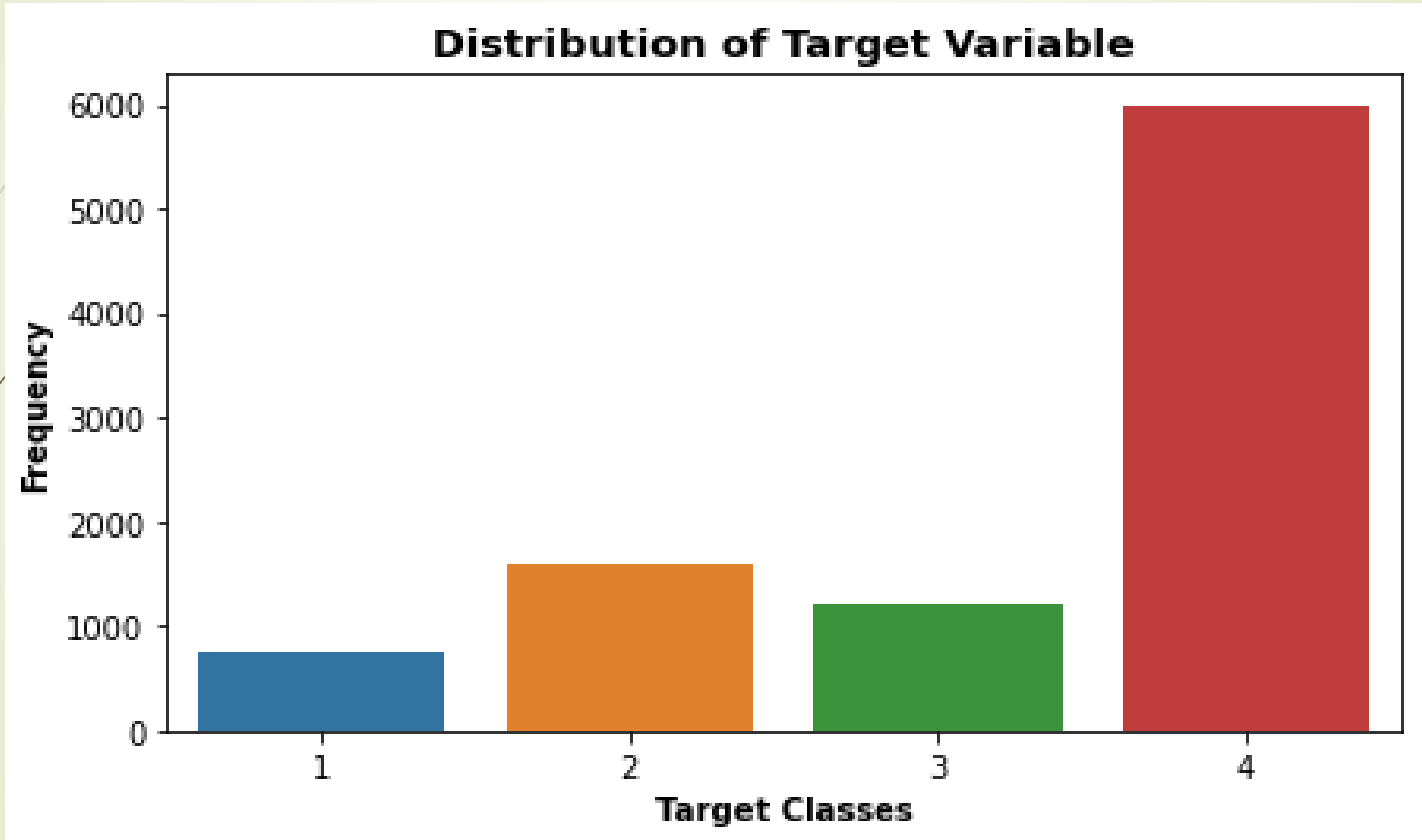
Distribution – ‘edjefe’



Distribution – ‘edjefa’



Distribution – ‘Target’



Exploratory Analysis Summary

- We have 143 columns and 9557 entries in dataset.
- 'dependency', 'edjefe' (years of education of male head of household), 'edjefa' (years of education of female head of household) columns had inconsistent datapoints due to which its datatype was reported as 'object'. However, appropriate treatment was applied and the datatype was restored to 'float'.
- Most of the houses have 4 or 5 rooms. At least 1 tablet is present in house. Fewer houses have more than 1 tablet.
- More than 6500 houses have no males younger than 12 years of age. Most of the houses have at least one male elder than 11 years of age.
- On an average, the houses have 1 or 2 males as well as females in family. Total family members in houses ranges between 1 to 13, where 3 and 4 members are likely to be found.
- As per the data, 6 years of schooling is most common, followed by 11 years. However, the count of persons who have never been to school is more compared to persons who have 11 years of schooling experience.
- 7000+ houses have no senior citizen in the family.

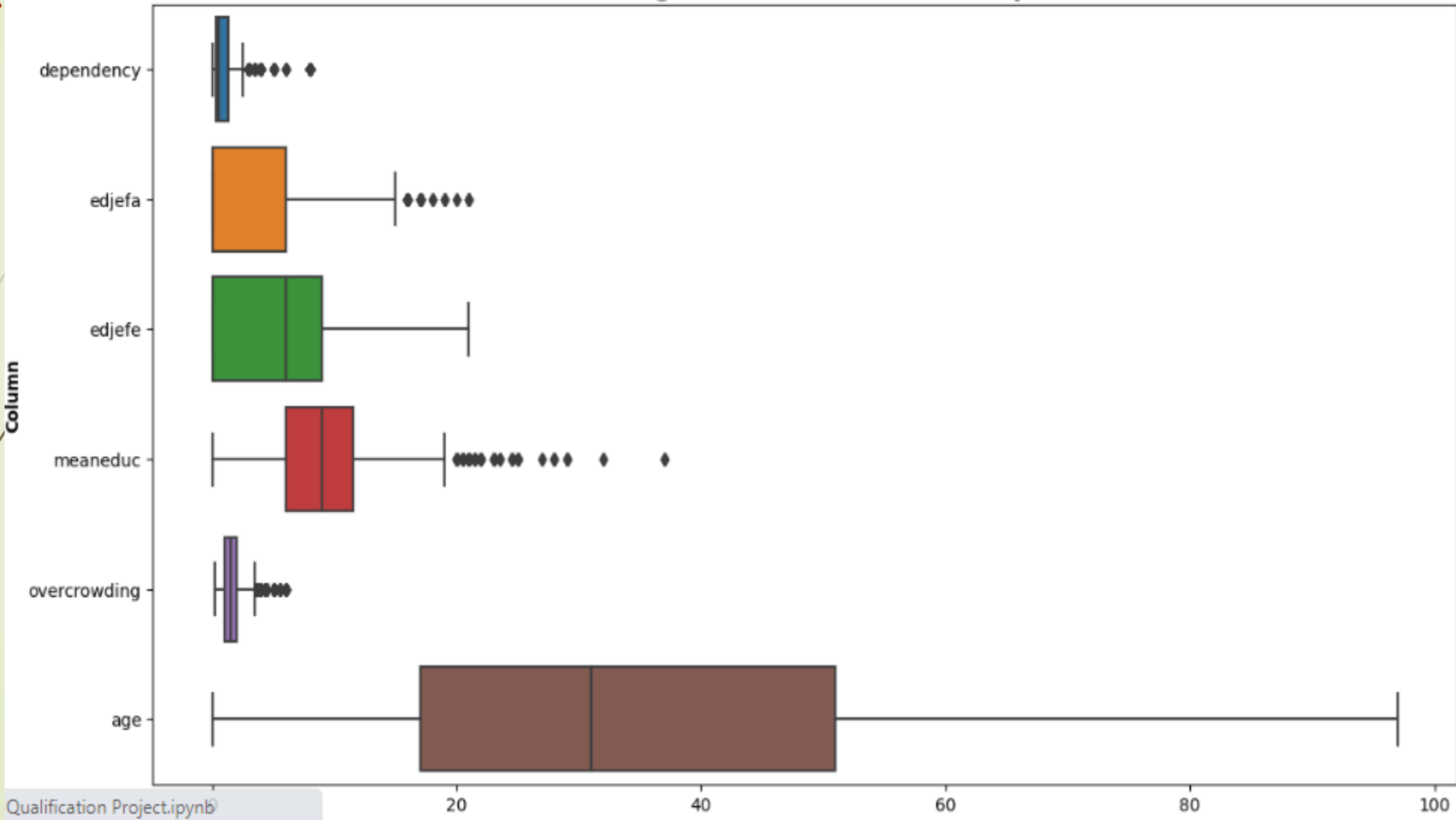
Exploratory Analysis Summary

- Most of the male and female head of family has lower education qualification.
- Dataset has significant null values for these columns - 'v2a1' (Monthly rent payment), 'v18q1' (no. of tablets household owns), 'rez_esc' (Years behind in school), and therefore, dropped.
- There are 6569 houses which has more than one entry in the dataset
- All the members of the house do not have same poverty level. There are 171 such houses which has different poverty level for same house ID.
- There are 2600 houses without a head.
- 'dependency', 'edjefa', 'meaneduc', 'overcrowding' columns have outliers.



Outlier Analysis

Outliers Visualization



Outliers Analysis Summary

- 'dependency', 'edjefa', 'meaneduc', 'overcrowding' columns have outliers.
- 'outliar_removal' function was created to remove the outliers.
- Values lower than $[75^{\text{th}} \text{ Percentile} + (1.5 * \text{IQR})]$ and greater than $[25^{\text{th}} \text{ Percentile} + (1.5 * \text{IQR})]$ are kept.
- Outlier analysis is done only for continuous variables (independent features).



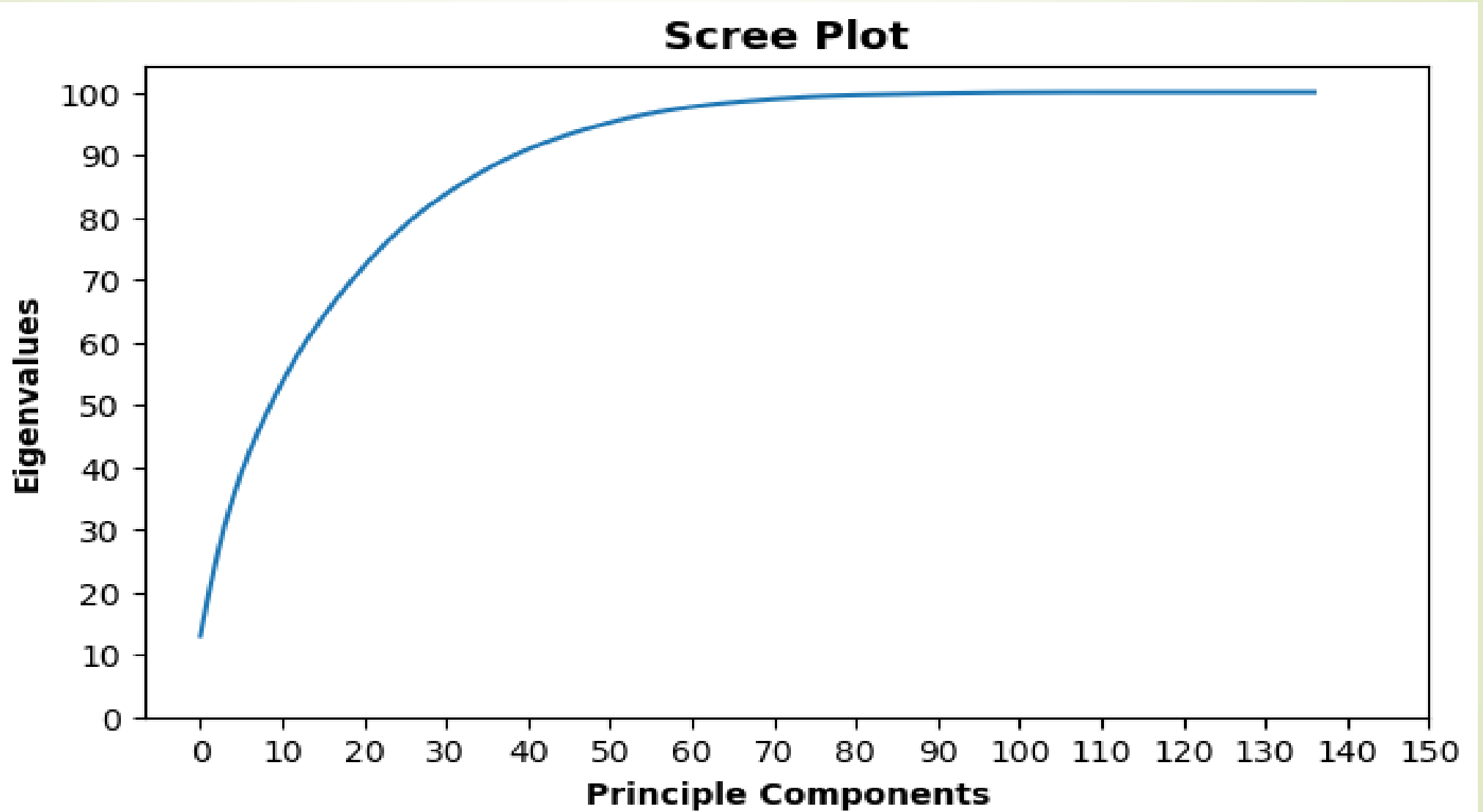
Data Scaling and Principle Component Analysis



Scaling and PCA Summary

- MinMax Scaler is used to scale the feature variables because not all the columns follow Normal Distribution
- Only those features are scaled which had values other than 0 and 1 in the column
- Because the dataset is of high dimensionality (142 independent features or columns), PCA is deployed to reduce the dimensions.
- Observed that 63 Principle Components captured more than 98% of dataset variance. Therefore, 62 components were used for training and testing dataset for training and prediction of Machine Learning Model

PCA Summary – Scree Plot





Predictive Model Summary: Random Forest Classifier

Predictive Model Summary

- As per the project requirement, Random Forest was used to build the model.
- Initial model “RandomForestClassifier(n_estimators=100, max_depth=7, oob_score=True)”, gave about Training Accuracy: 69.7% and Test Accuracy: 66.94%. This is a decent model with no overfitting.
- After hyper-parameter tuning, the best score obtained was 74.8% and best model obtained was ‘RandomForestClassifier(max_depth=30, min_samples_leaf=2, n_estimators=300, n_jobs=-1)’, however this model **overfits**.
- Further, Average CV Score obtained after 10 fold cross validation using the best model is 65.6%.

Random Forest Classifier - Analysis

```
In [140]: random_forest_clf = RandomForestClassifier(n_estimators=100, max_depth=7, oob_score=True)
```

```
In [144]: print(f'Training Accuracy: {round((accuracy_score(y_train, y_pred_train)*100),2)}%')  
          print(f'Test Accuracy: {round((accuracy_score(y_test, y_pred_test)*100),2)}%')
```

Training Accuracy: 69.7%
Test Accuracy: 66.94%

```
In [151]: grid_search.best_score_
```

```
Out[151]: 0.7478665632273079
```

```
In [152]: best_parameters = grid_search.best_params_  
          print(best_parameters)
```

```
{'max_depth': 30, 'min_samples_leaf': 2, 'n_estimators': 300}
```


Random Forest Classifier - Analysis

Sub-Task: Using best model obtained from Gridsearch for predictions

```
In [154]: rf_best.fit(X_train_pc, y_train)

y_pred_train = rf_best.predict(X_train_pc)
y_pred_test = rf_best.predict(X_test_pc)
```

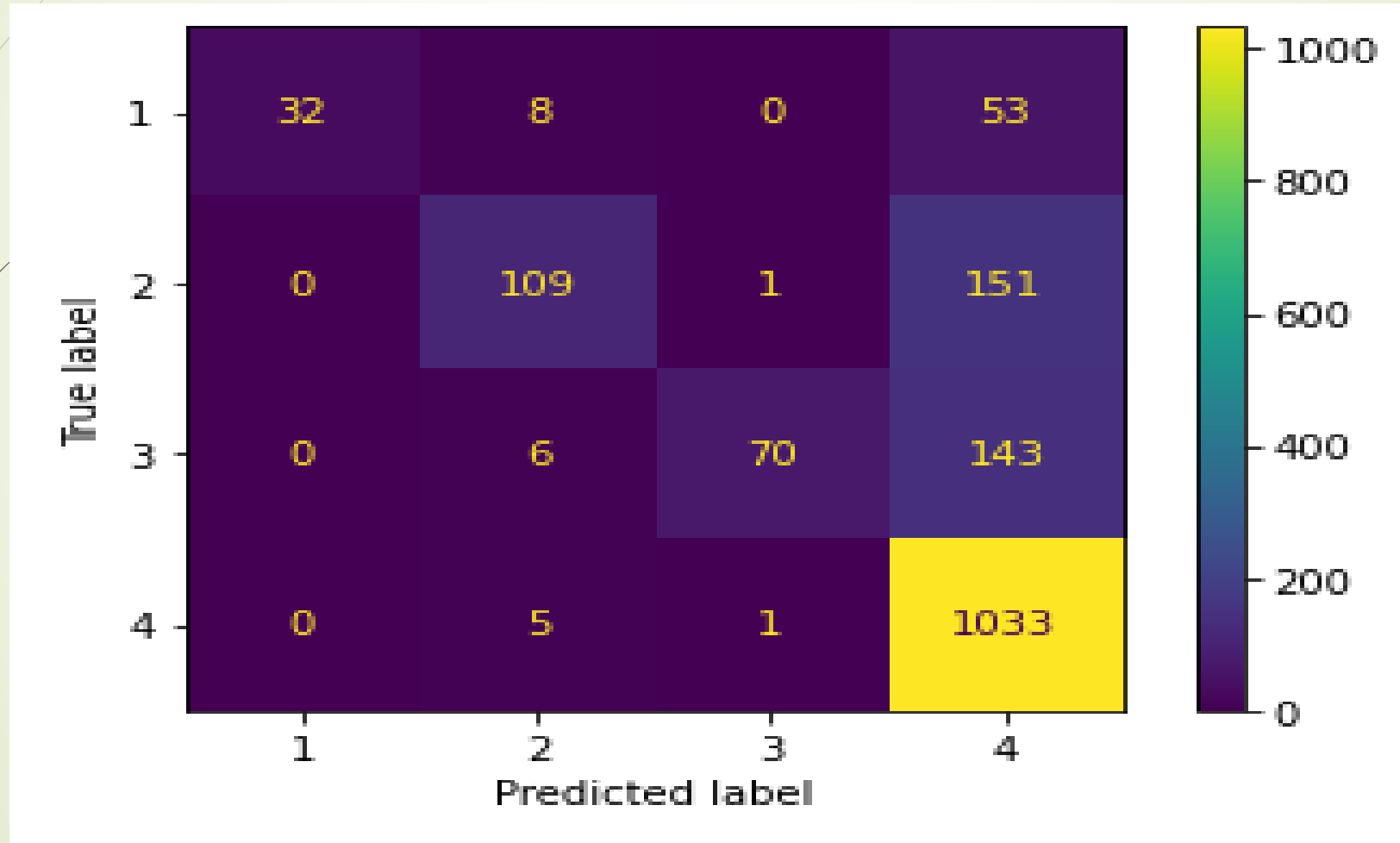
Sub-Task: Analysing Test Results

```
In [155]: print(f'Training Accuracy: {round((accuracy_score(y_train, y_pred_train)*100),2)}%')
print(f'Test Accuracy: {round((accuracy_score(y_test, y_pred_test)*100),2)}%')
```

Training Accuracy: 99.52%

Test Accuracy: 77.17%

Random Forest Classifier – Confusion Matrix



Random Forest Classifier Classification Report

```
In [159]: print(classification_report(y_test, y_pred_test))
```

	precision	recall	f1-score	support
1	1.00	0.34	0.51	93
2	0.85	0.42	0.56	261
3	0.97	0.32	0.48	219
4	0.75	0.99	0.85	1039
accuracy			0.77	1612
macro avg	0.89	0.52	0.60	1612
weighted avg	0.81	0.77	0.74	1612

Random Forest Classifier - Analysis

```
In [161]: print("Cross Validation Scores: ", scores)
          print("\nAverage CV Score: ", scores.mean())
          print("\nNumber of CV Scores used in Average: ", len(scores))
```

```
Cross Validation Scores: [0.79528536 0.73200993 0.74441687 0.68610422 0.66997519 0.64019851
0.57444169 0.54782609 0.54409938 0.62236025]
```

```
Average CV Score: 0.6556717476072313
```

```
Number of CV Scores used in Average: 10
```

Random Forest Classifier – test.csv

Predictions

```
In [162]: test_csv_pred = rf_best.predict(test_data_pc)
```

```
In [163]: test_csv_pred = pd.DataFrame(test_csv_pred, columns=['predicted_values'])  
test_csv_pred.head(7)
```

Out[163]:

	predicted_values
0	4
1	4
2	4
3	4
4	2
5	4
6	4



test.csv Predicted
values

Appendix

- Please refer 'Income Qualification Prediction Project_Lavkush.pdf' file, submitted along with this PPT
- Because the code was developed in jupyter notebook, it has source code along with the detailed analysis and report
- All the graphs included in this presentation can also be found in that project report
- This PPT is just a glimpse of the analysis done, for quick reference. Detailed work is present in the project report – "Income Qualification Prediction Project_Lavkush.pdf".
- Predicted values for 'test.csv' is attached as an csv file in slide number 51 of this PPT.



Thank you!