# MAT 328 Project: Malique Russell

**Structure and Shape of the Data**

The dataset is structured in a rectangular (tabular) format, consisting of rows and columns. It includes a mix of quantitative and qualitative data.

**Quantitative Data:**

- Extremely Low Income Units: Units with rents at 0–30% of the area median income
- Very Low Income Units: Rents at 31–50% of the area median income
- Low Income Units: Rents at 51–80% of the area median income
- Moderate Income Units: Rents at 81–120% of the area median income
- Middle Income Units: Rents at 121–165% of the area median income
- Other Income Units: Units reserved for building superintendents
- Counted Rental Units: Units counted under the Housing New York plan where assistance was provided to landlords
- Counted Homeownership Units: Units counted under the Housing New York plan where assistance was provided directly to homeowners
- All Counted Units: Total affordable units counted under the Housing New York plan
- Total Units: Sum of all units in the dataset
- Senior Units: Units specifically designated for senior households

**Qualitative Data:**

- Project ID: Unique identifier for each project
- Project Name: Name assigned by the Housing Preservation and Development (HPD).
- Program Group: Type of housing initiative
- Project Start Date: Date of project loan or agreement closure
- Project Completion Date: Date of the last building completion in a project
- Extended Affordability Only: Indicates whether the project qualifies for extended affordability
- Prevailing Wage Status: Specifies if the project adheres to prevailing wage requirements (e.g., Davis-Bacon Act)

- Planned Tax Benefit: Expected tax incentives associated with the project

**Granularity of the Data:**

The dataset has a low level of granularity, as each row represents aggregated unit data rather than individual housing units. A more granular dataset would provide detailed information at the unit level rather than summaries by category

**Scope and Completeness of the Data**

The dataset is well-suited for analyzing affordable housing trends in New York City. However, its scope is too broad for hyper-localized questions (e.g., borough-specific trends) and too narrow for state-wide analysis

**Temporality of the Data**

The dataset spans eight years, covering January 1, 2014, to December 31, 2021. It is managed by the Department of Housing Preservation and Development (HPD) and was last updated on March 3, 2025

**Faithfulness of the Data**

The dataset appears highly reliable, as it is compiled by a reputable city agency with direct oversight and access to housing records, ensuring accuracy and completeness

In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import statsmodels.formula.api as smf
from collections import Counter
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import statsmodels.formula.api as smf
from scipy.special import expit
from scipy.stats import logistic
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_squared_error
```

In [4]:
```
affordable_housing = pd.read_csv("Affordable_Housing_Production_by_Project.csv")
affordable_housing["Project Completion Date"] = pd.to_datetime(affordable_housing["Project Completion Date"])
affordable_housing = affordable_housing.sort_values(by='Project Completion Date', ascending=True)
affordable_housing.head()
```

Out[4]:

| | Project ID | Project Name | Program Group | Project Start Date | Project Completion Date | Extended Affordability Only | Prevailing Wage Status | Planned Tax Benefit | Extremely Low Income Units | Very Low Income Units | Inc |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **549** | 55759 | CONFIDENTIAL | CONFIDENTIAL | 01/03/2014 | 2014-01-03 | No | Non Prevailing Wage | NaN | 0 | 0 | |
| **523** | 55647 | CONFIDENTIAL | CONFIDENTIAL | 01/07/2014 | 2014-01-07 | No | Non Prevailing Wage | NaN | 0 | 0 | |
| **555** | 55773 | CONFIDENTIAL | CONFIDENTIAL | 01/10/2014 | 2014-01-10 | No | Non Prevailing Wage | NaN | 0 | 0 | |
| **641** | 57341 | CONFIDENTIAL | CONFIDENTIAL | 01/10/2014 | 2014-01-10 | No | Non Prevailing Wage | NaN | 0 | 0 | |
| **533** | 55697 | CONFIDENTIAL | CONFIDENTIAL | 01/14/2014 | 2014-01-14 | No | Non Prevailing Wage | NaN | 0 | 0 | |

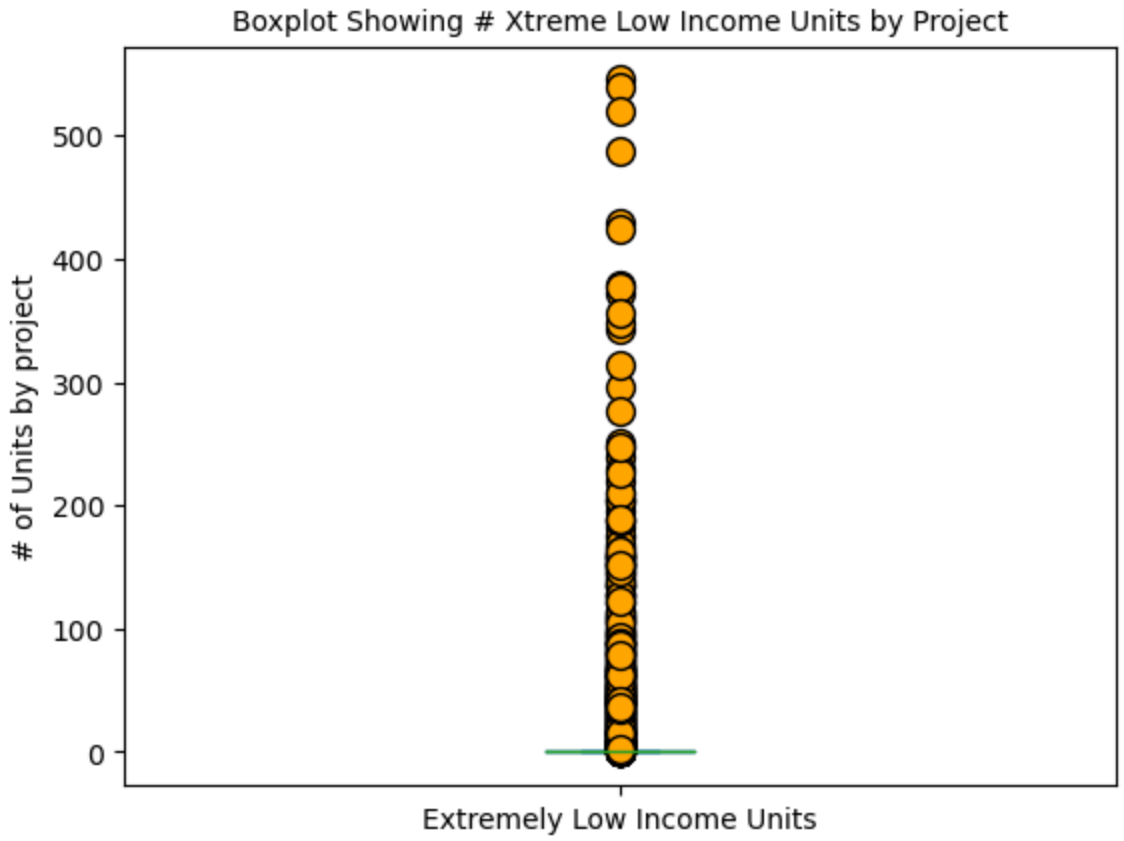
In [5]:
```
# Dropping Incomplete projects
complete_projects = affordable_housing.dropna(subset=['Project Completion Date'])
complete_projects.reset_index(drop=True, inplace=True)
complete_projects.head()
```

Out[5]:

| | Project ID | Project Name | Program Group | Project Start Date | Project Completion Date | Extended Affordability Only | Prevailing Wage Status | Planned Tax Benefit | Extremely Low Income Units | Very Low Income Units | Lo Incor Un |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 55759 | CONFIDENTIAL | CONFIDENTIAL | 01/03/2014 | 2014-01-03 | No | Non Prevailing Wage | NaN | 0 | 0 | |
| **1** | 55647 | CONFIDENTIAL | CONFIDENTIAL | 01/07/2014 | 2014-01-07 | No | Non Prevailing Wage | NaN | 0 | 0 | |
| **2** | 55773 | CONFIDENTIAL | CONFIDENTIAL | 01/10/2014 | 2014-01-10 | No | Non Prevailing Wage | NaN | 0 | 0 | |
| **3** | 57341 | CONFIDENTIAL | CONFIDENTIAL | 01/10/2014 | 2014-01-10 | No | Non Prevailing Wage | NaN | 0 | 0 | |
| **4** | 55697 | CONFIDENTIAL | CONFIDENTIAL | 01/14/2014 | 2014-01-14 | No | Non Prevailing Wage | NaN | 0 | 0 | |

In [6]:
```
xtreme = complete_projects["Extremely Low Income Units"]
very = complete_projects["Very Low Income Units"]
low = complete_projects["Low Income Units"]
moderate = complete_projects["Moderate Income Units"]
middle = complete_projects["Middle Income Units"]
other = complete_projects["Other Income Units"]
owned = complete_projects["Counted Homeownership Units"]
total = complete_projects["All Counted Units"]
```

In [7]:
```
xtreme.plot(kind = "box", flierprops=dict(marker='o', markersize=10, markerfacecolor ='orange'))
plt.ylabel("# of Units by project")
_=plt.title('Boxplot Showing # Xtreme Low Income Units by Project', fontsize = 10)
```




**This graph shows the distribution of completed extremely low-income units by project built in New York City from january 1, 2014 to December 30, 2025**

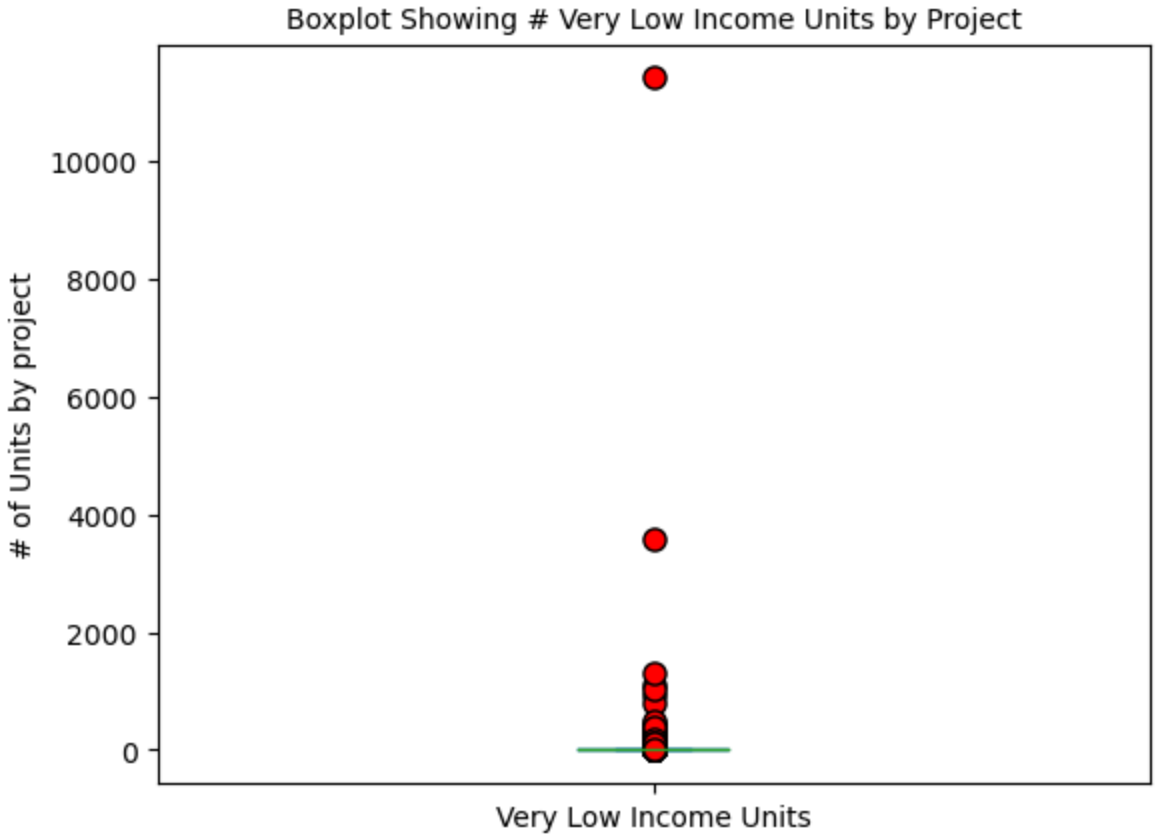
Some notable deductions:

- The majority of projects had under 300 extremely low-income unit
- Most projects had less than 250 of these units

In [14]:
```
# Percent of Completed Extremely Low Income Units
Xtreme =xtreme.sum()/total.sum()
xtreme_per = round(Xtreme*100,2)
xtreme_per
```

Out[14]: 16.84

In [16]:
```
very.plot(kind = "box", flierprops=dict(marker='o', markersize = 8, markerfacecolor ='red'))
plt.ylabel("# of Units by project")
_=plt.title('Boxplot Showing # Very Low Income Units by Project', fontsize = 10)
```




**This graph shows the distribution of completed very low-income units by project built in New York City from january 1, 2014 to December 30, 2025**

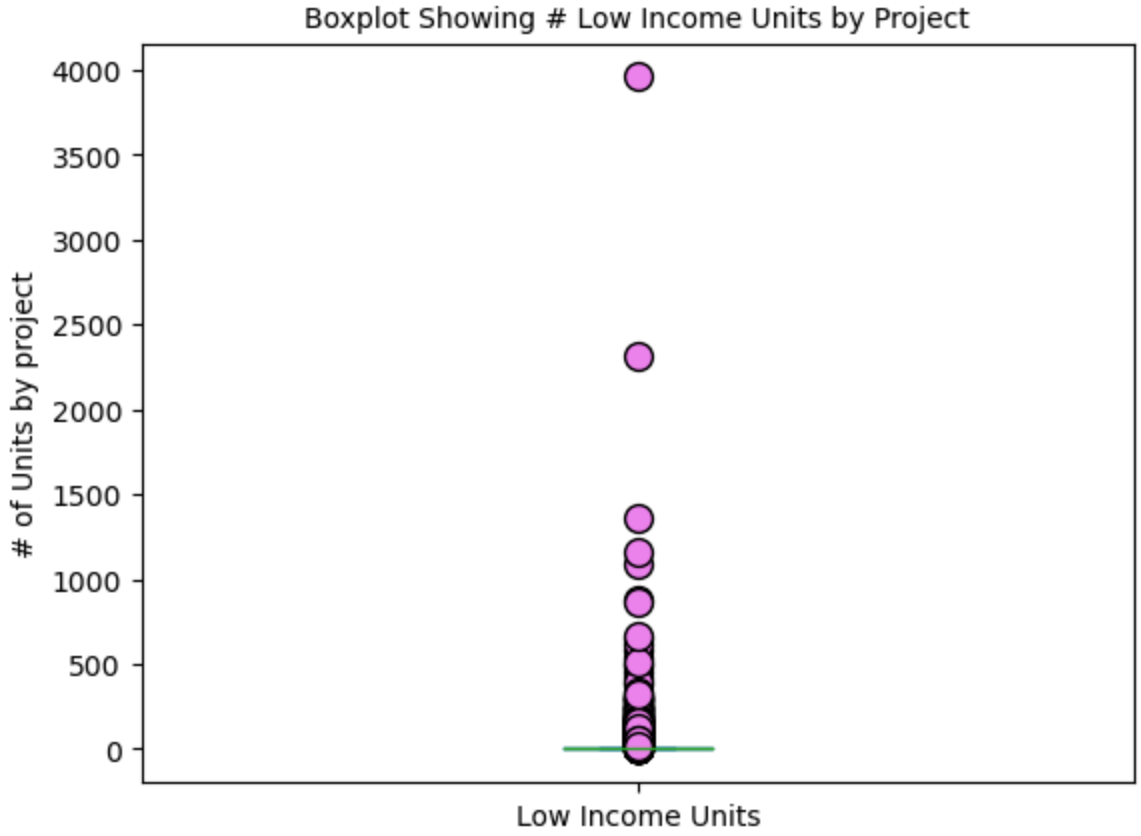
Some notable deductions:

- The majority of projects had under 2000 very low-income units
- One project had 10,000+ of these units

In [19]:
```
# Percent of Completed Very Low Income Units
Very = very.sum()/total.sum()
round(Very*100,2)
very_per = round(Very*100,2)
very_per
```

Out[19]: 24.88

In [21]:
```
low.plot(kind = "box", flierprops=dict(marker='o', markersize=10, markerfacecolor ='violet'))
plt.ylabel("# of Units by project")
_=plt.title('Boxplot Showing # Low Income Units by Project', fontsize = 10)
```




**This graph shows the distribution of completed low-income units by project built in New York City from January 1, 2014 to December 30, 2025**

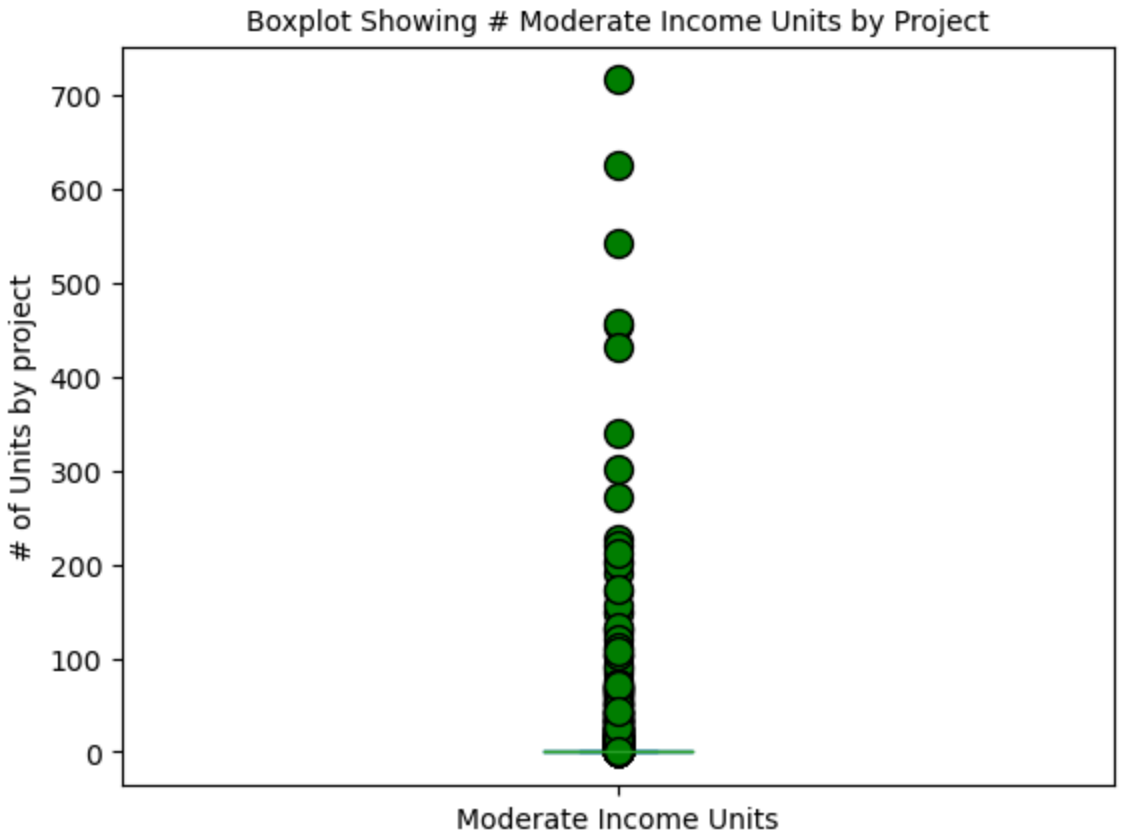
Some notable deductions:

- The majority of projects had under 1000 low-income unit
- Most projects had less than 600 of these units

In [24]:
```
# Percent of Completed Low Income Units
Low = low.sum()/total.sum()
round(Low*100,2)
low_per = round(Low*100,2)
low_per
```

Out[24]: 36.67

In [26]:
```
moderate.plot(kind = "box", flierprops=dict(marker='o', markersize=10, markerfacecolor ='green'))
plt.ylabel("# of Units by project")
_=plt.title('Boxplot Showing # Moderate Income Units by Project', fontsize = 10)
```




**This graph shows the distribution of completed moderate income units by project built in New York City from January 1, 2014 to December 30, 2025**

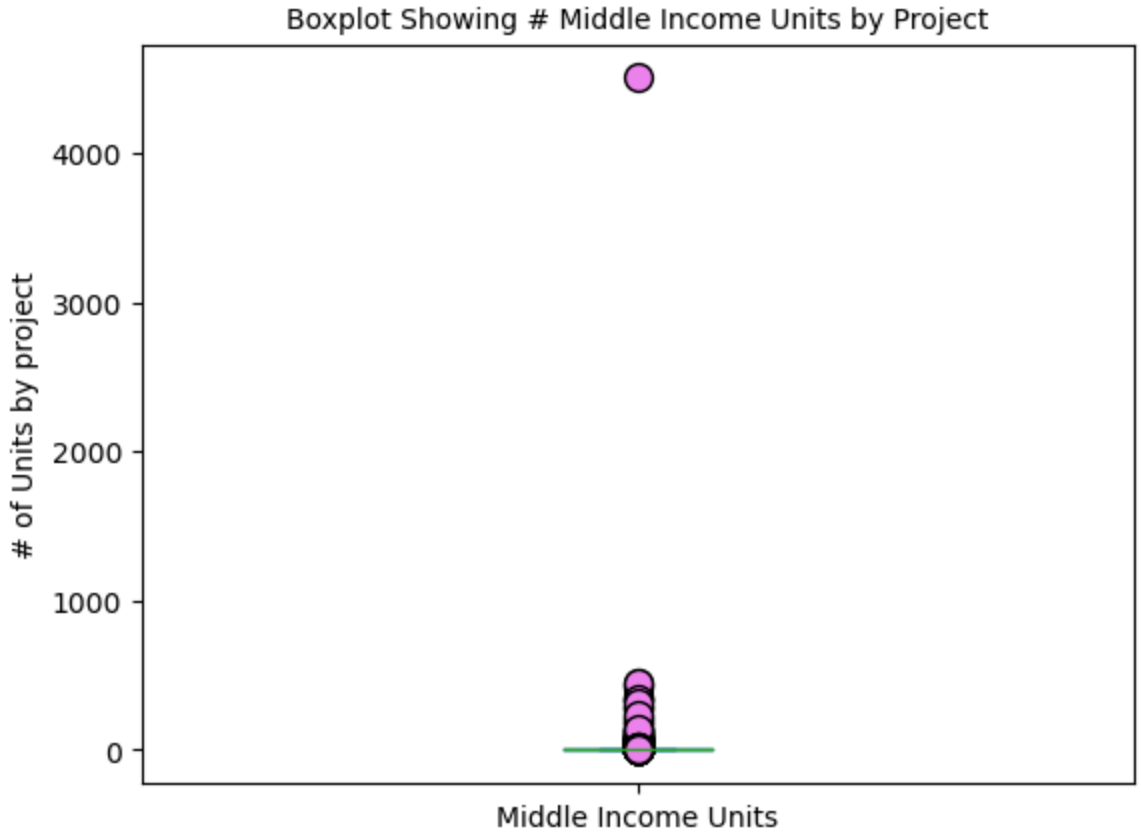
Some notable deductions:

- The majority of projects had under 250 moderate income unit
- Most projects had less than 100 of these units

In [29]:
```
# Percent of Completed Moderate Income Units
Moderate = moderate.sum()/total.sum()
round(Moderate*100,2)
moderate_per = round(Moderate*100,2)
moderate_per
```

Out[29]: 6.85

In [31]:
```
middle.plot(kind = "box", flierprops=dict(marker='o', markersize=10, markerfacecolor ='violet'))
plt.ylabel("# of Units by project")
_=plt.title('Boxplot Showing # Middle Income Units by Project', fontsize = 10)
```




**This graph shows the distribution of completed middle income units by project built in New York City from January 1, 2014 to December 30, 2025**

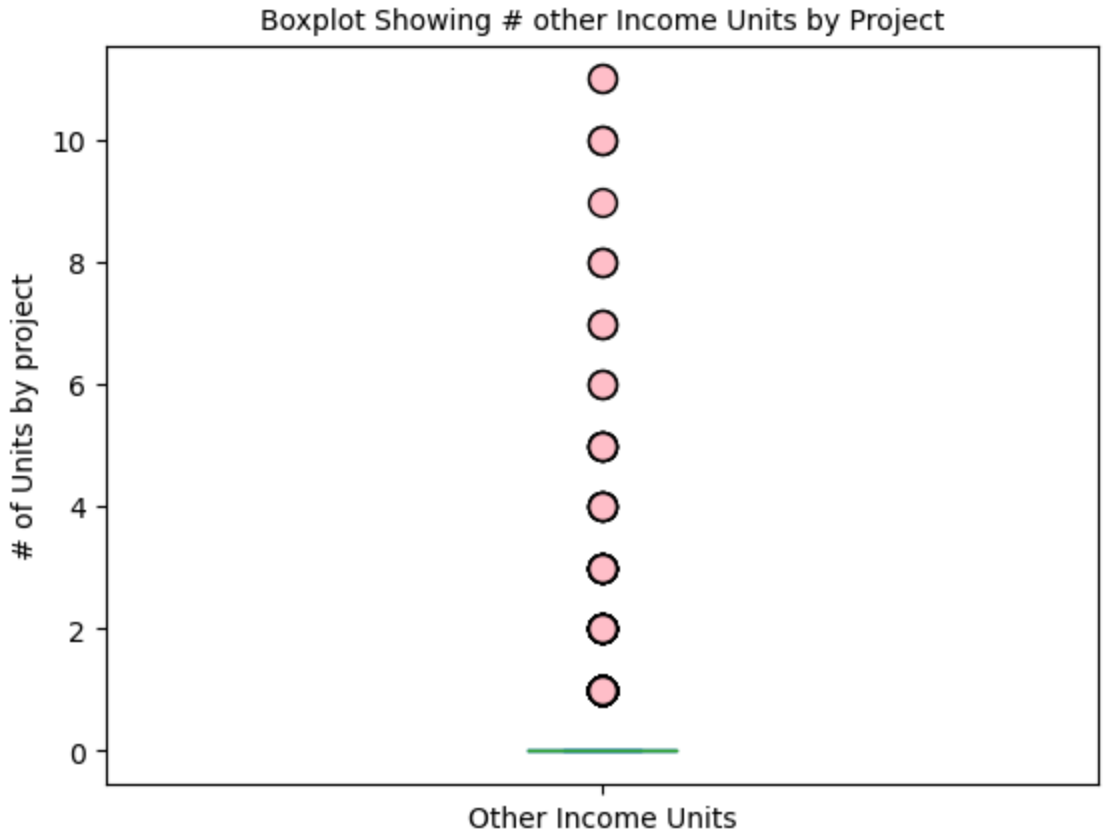
Some notable deductions:

- The majority of projects had less than 150 middle low-income unit

In [34]:
```
# Percent of Completed Middle Income Units
Middle = middle.sum()/total.sum()
round(Middle*100,2)
middle_per = round(Middle*100,2)
middle_per
```

Out[34]: 14.28

In [38]:
```
other.plot(kind = "box", flierprops=dict(marker='o', markersize=10, markerfacecolor ='pink'))
plt.ylabel("# of Units by project")
_=plt.title('Boxplot Showing # other Income Units by Project', fontsize = 10)
```




**This graph shows the distribution of completed other income units by project built in New York City from January 1, 2014 to December 30, 2025**

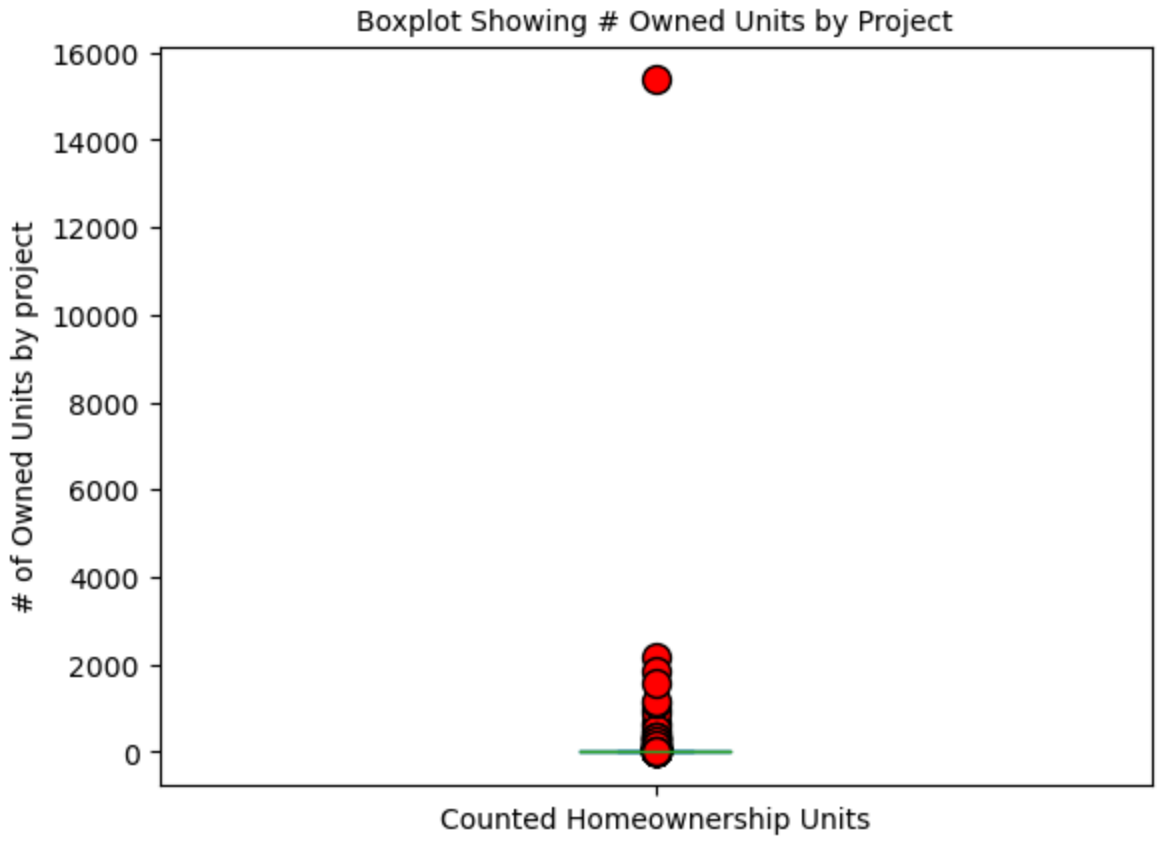
Some notable deductions:

- These units seem to be fairly distributed

In [41]:
```
# Percentage of Completed Other Income Units
Other = other.sum()/total.sum()
round(Other*100,2)
other_per = round(Other*100,2)
other_per
```

Out[41]:
```
0.47
```

In [45]:
```
owned.plot(kind = "box" , flierprops=dict(marker='o', markersize=10, markerfacecolor ='red'))
plt.ylabel("# of Owned Units by project")
_=plt.title('Boxplot Showing # Owned Units by Project', fontsize = 10)
```




**This graph shows the distribution of completed owned income units by project built in New York City from January 1, 2014 to December 30, 2025**

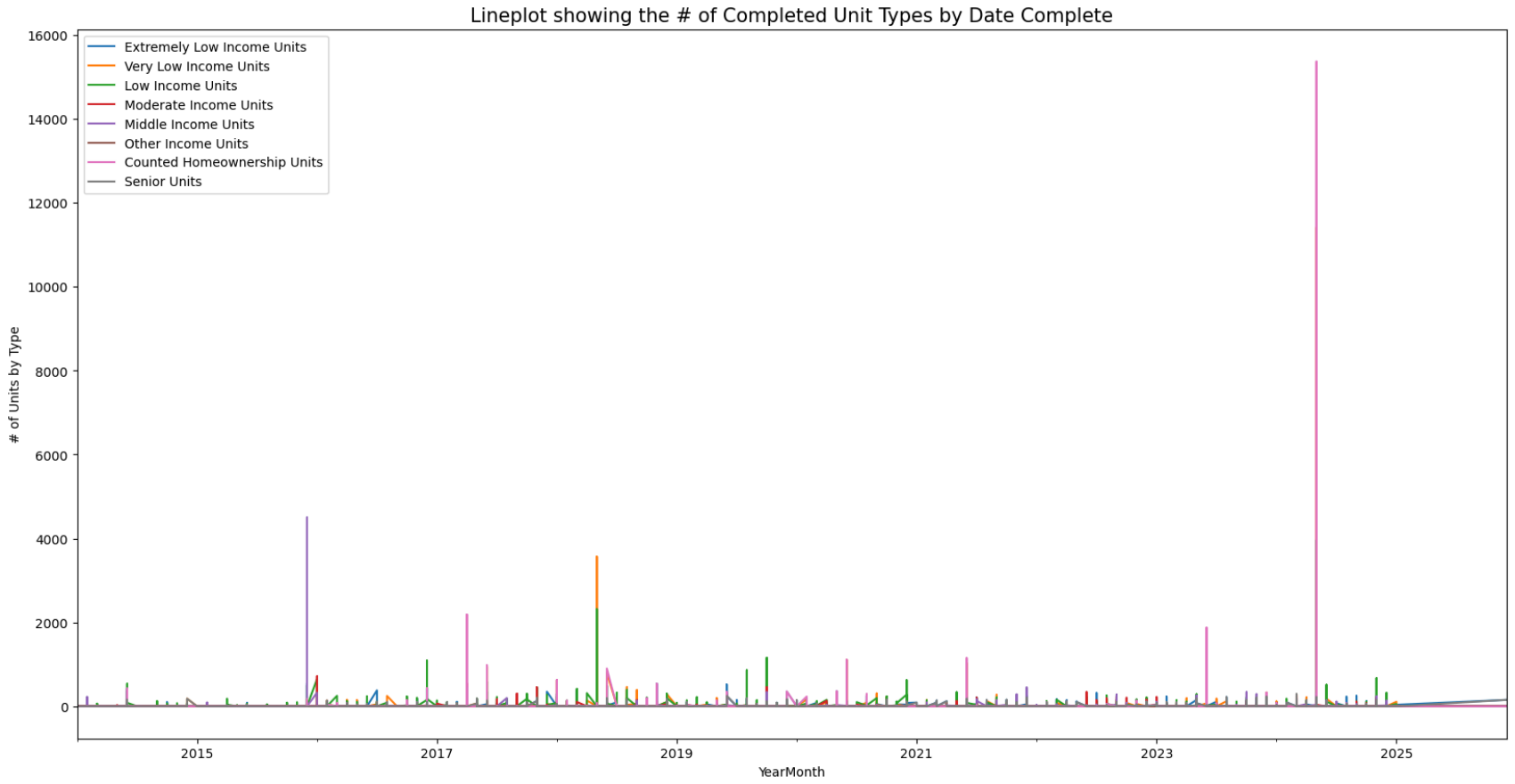
Some notable deductions:

- The majority of projects had less than 2000 owned unit
- One project had 15,000+ units owned, which is an obvious outlier

In [48]:
```
# Percentage of Completed Owned Units
Owned = owned.sum()/total.sum()
round(Owned*100,2)
owned_per = round(Owned*100,2)
owned_per
```

Out[48]: 18.93

In [52]:
```
projects = complete_projects.drop(["Project ID", "Total Units", "All Counted Units", "Counted Rental Units"], axis =
projects['YearMonth'] = projects['Project Completion Date'].dt.to_period('M')
projects.drop(["Project Completion Date"], axis = 1).plot(x="YearMonth",  figsize = (20, 10)).legend()
plt.ylabel("# of Units by Type")
_=plt.title('Lineplot showing the # of Completed Unit Types by Date Complete', fontsize = 15)
```




**This line graph shows the number of completed units by type built in New York City from January 1, 2014 to December 30, 2025**

Some key points shown in the graph are:

- The majority of projects had under 3000 completed units regardless of type

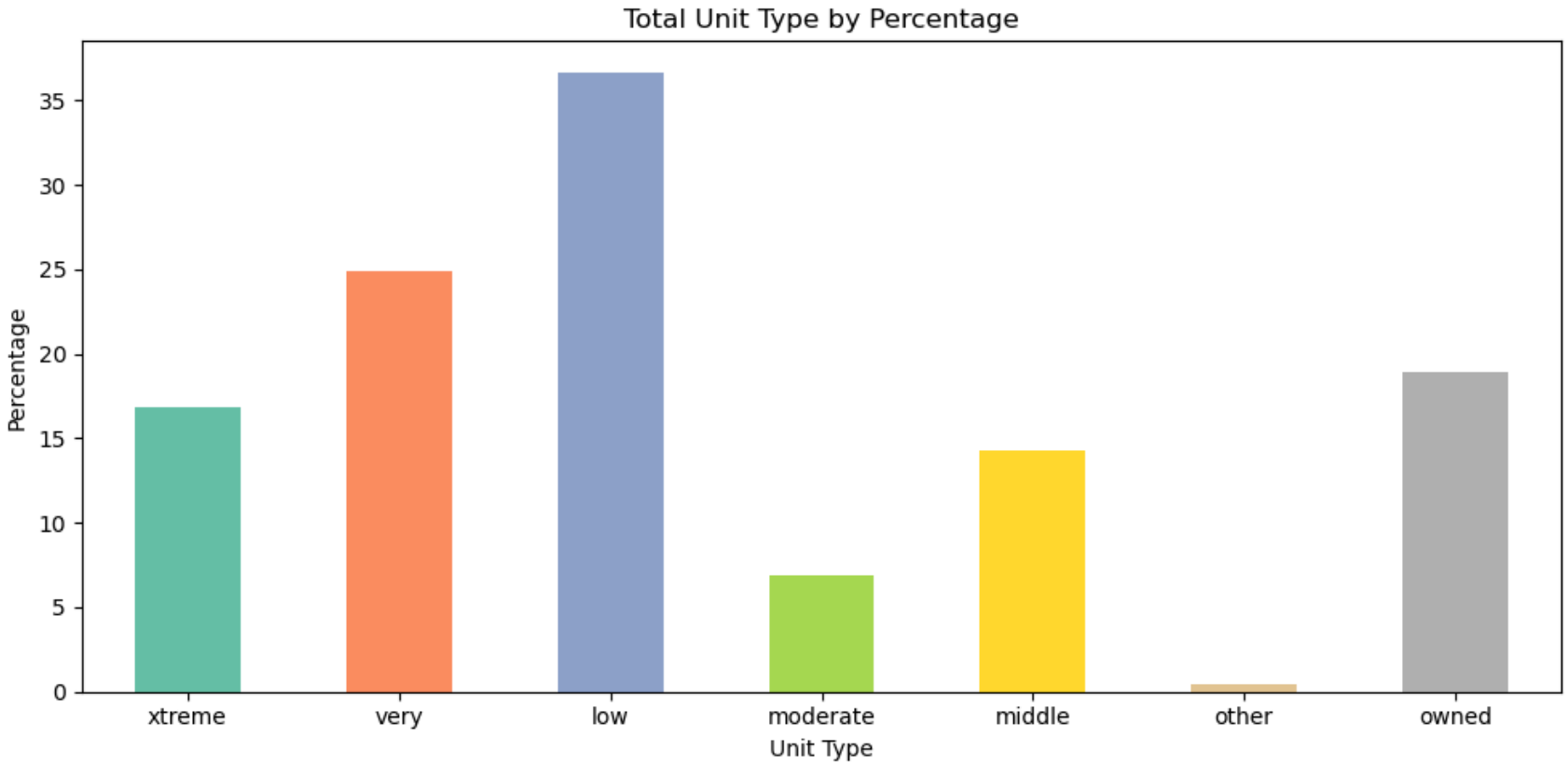
- At least 4 projects had over 2000 units completed which makes them outliers in the data
- The most units completed for a single project type fall under the home-owner category, completed after 2024

In [83]:
```
data = {
    'Unit Type': ["xtreme", "very", "low", "moderate", "middle", "other", "owned"],
    'Percentages': [16.84, 24.88, 36.67, 6.85, 14.28, 0.47, 18.93]}

df = pd.DataFrame(data)
colors = plt.cm.Set2(np.linspace(0, 1, len(data['Unit Type'])))

df.plot(kind='bar', x='Unit Type', y='Percentages', color=colors, legend=False, figsize = (10, 5))

plt.title("Total Unit Type by Percentage")
plt.ylabel("Percentage")
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
plt.savefig("total.png", dpi=300, bbox_inches='tight')
```




```
<Figure size 640x480 with 0 Axes>
```

**This bar graph shows the percentage of completed units by type built in New York City from January 1, 2014 to December 30, 2025**

Some keys points shown on the chart are:

- Low income units were the most built built in New York City during the 10 year period
- Other income units units were the least built in the period
- Units falling under the xtreme, very and low uncome categories account for the bulk of units builts 78.39%
- Only 19% of units completed are owned

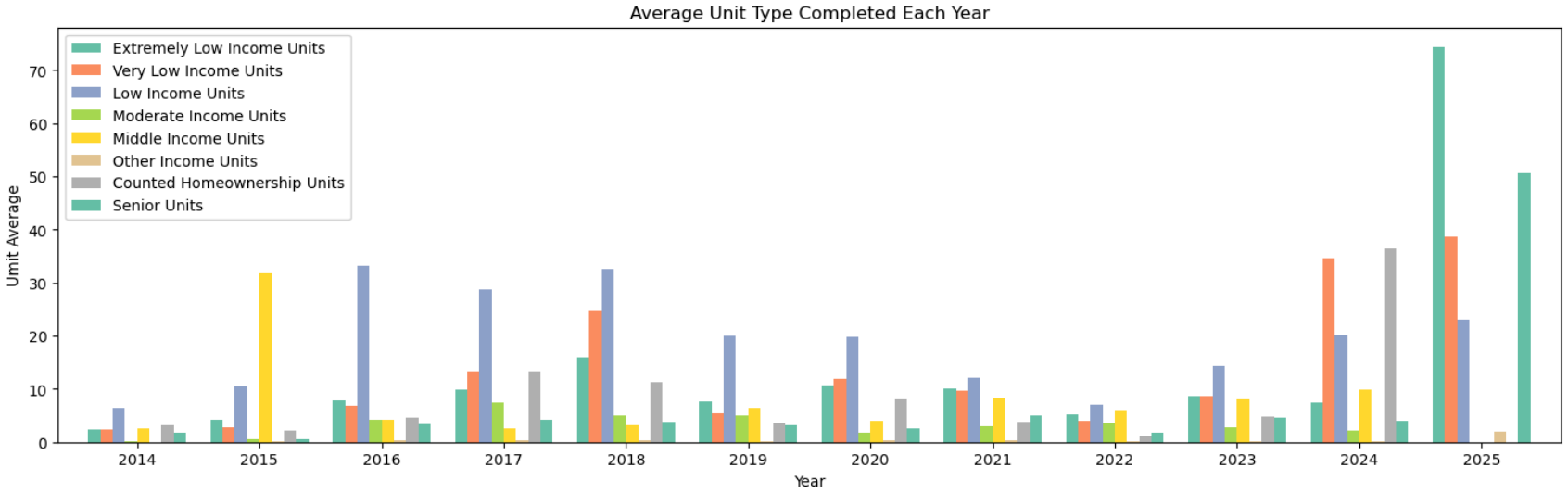
**Average Unit Type Completed Completed By Year**

In [63]:
```
# Extract year from date
projects["Year"] = projects['Project Completion Date'].dt.year
# Group by year and get average of numeric columns
year_units_avg = projects.groupby("Year").mean(numeric_only=True).reset_index()
year_units_avg
```

Out[63]:

| | Year | Extremely Low Income Units | Very Low Income Units | Low Income Units | Moderate Income Units | Middle Income Units | Other Income Units | Counted Homeownership Units | Senior Units |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 2014 | 2.428571 | 2.443769 | 6.528875 | 0.212766 | 2.653495 | 0.042553 | 3.188450 | 1.808511 |
| **1** | 2015 | 4.285714 | 2.809524 | 10.401361 | 0.571429 | 31.700680 | 0.149660 | 2.251701 | 0.517007 |
| **2** | 2016 | 7.941441 | 6.819820 | 33.184685 | 4.283784 | 4.171171 | 0.256757 | 4.572072 | 3.351351 |
| **3** | 2017 | 9.892405 | 13.382911 | 28.775316 | 7.522152 | 2.677215 | 0.405063 | 13.227848 | 4.120253 |
| **4** | 2018 | 15.977707 | 24.738854 | 32.627389 | 5.003185 | 3.121019 | 0.321656 | 11.232484 | 3.729299 |
| **5** | 2019 | 7.550117 | 5.375291 | 20.053613 | 5.069930 | 6.526807 | 0.240093 | 3.589744 | 3.086247 |
| **6** | 2020 | 10.588235 | 11.823529 | 19.852941 | 1.860294 | 3.963235 | 0.327206 | 8.106618 | 2.500000 |
| **7** | 2021 | 10.034985 | 9.769679 | 12.145773 | 3.069971 | 8.172012 | 0.297376 | 3.749271 | 5.040816 |
| **8** | 2022 | 5.297189 | 3.967871 | 7.056225 | 3.500000 | 5.965863 | 0.112450 | 1.068273 | 1.863454 |
| **9** | 2023 | 8.575592 | 8.714026 | 14.431694 | 2.723133 | 8.151184 | 0.214936 | 4.759563 | 4.515483 |
| **10** | 2024 | 7.468303 | 34.697342 | 20.159509 | 2.130879 | 9.860941 | 0.198364 | 36.425358 | 3.916155 |
| **11** | 2025 | 74.333333 | 38.666667 | 23.000000 | 0.000000 | 0.000000 | 2.000000 | 0.000000 | 50.666667 |

In [67]:
```
_=year_units_avg.plot(kind='bar', x = "Year", width = .8, color=colors, legend=False, figsize = (18, 5)).legend()
_=plt.title("Average Unit Type Completed Each Year")
_=plt.ylabel("Umit Average")
_=plt.xticks(rotation = 0)
```




**The bar graph displays the average unit types completed each year**

- Low-income units make up the largest portion, at approximately 36%.
- Very low-income units follow at around 25%.
- Owned units represent about 19%.
- Extreme low-income units make up about 17%.
- Middle-income units are at 14%.
- Moderate-income units account for roughly 7%.
- Other unit types make up a very small portion, just under 1%.

**Overall, the chart highlights that the majority of units fall within the low- and very low-income categories, indicating a significant focus on lower cost housing.**

In [69]:
```
# Data Summary
projects.drop(["Project Completion Date", "Year"], axis = 1).describe()
```

Out[69]:

| | Extremely Low Income Units | Very Low Income Units | Low Income Units | Moderate Income Units | Middle Income Units | Other Income Units | Counted Homeownership Units | Senior Units |
|---|---|---|---|---|---|---|---|---|
| **count** | 3911.000000 | 3911.000000 | 3911.000000 | 3911.000000 | 3911.000000 | 3911.000000 | 3911.000000 | 3911.000000 |
| **mean** | 8.211966 | 12.130913 | 17.876502 | 3.341089 | 6.963436 | 0.228330 | 9.227819 | 3.348249 |
| **std** | 36.071743 | 196.941309 | 94.876869 | 26.751088 | 76.357427 | 0.746878 | 254.783852 | 20.061183 |
| **min** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **50%** | 0.000000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **75%** | 0.000000 | 0.000000 | 2.000000 | 0.000000 | 3.000000 | 0.000000 | 1.000000 | 0.000000 |
| **max** | 545.000000 | 11413.000000 | 3959.000000 | 716.000000 | 4505.000000 | 11.000000 | 15372.000000 | 291.000000 |

In [71]:
```
# Create New Dataframe for
projects2 = projects.drop(["Project Name", "Program Group", "Project Start Date", "Project Completion Date", "Extende
projects2
```

Out[71]:

| | Prevailing Wage Status | Extremely Low Income Units | Very Low Income Units | Low Income Units | Moderate Income Units | Middle Income Units | Other Income Units | Counted Homeownership Units | Senior Units |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Non Prevailing Wage | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **1** | Non Prevailing Wage | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **2** | Non Prevailing Wage | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **3** | Non Prevailing Wage | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| **4** | Non Prevailing Wage | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **3906** | Non Prevailing Wage | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| **3907** | Non Prevailing Wage | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| **3908** | Non Prevailing Wage | 35 | 108 | 54 | 0 | 0 | 4 | 0 | 0 |
| **3909** | Non Prevailing Wage | 36 | 8 | 15 | 0 | 0 | 1 | 0 | 0 |
| **3910** | Prevailing Wage | 152 | 0 | 0 | 0 | 0 | 1 | 0 | 152 |

3911 rows × 9 columns

In [73]:
```
print(projects2.columns.tolist())
projects2 = pd.get_dummies(projects2, columns = ["Prevailing Wage Status"], drop_first = True, dtype=int)
projects2
```

```
['Prevailing Wage Status', 'Extremely Low Income Units', 'Very Low Income Units', 'Low Income Units', 'Moderate Incom
e Units', 'Middle Income Units', 'Other Income Units', 'Counted Homeownership Units', 'Senior Units']
```

Out[73]:

| | Extremely Low Income Units | Very Low Income Units | Low Income Units | Moderate Income Units | Middle Income Units | Other Income Units | Counted Homeownership Units | Senior Units | Prevailing Wage Status_Prevailing Wage |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **2** | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **3** | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **3906** | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 |
| **3907** | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| **3908** | 35 | 108 | 54 | 0 | 0 | 4 | 0 | 0 | 0 |
| **3909** | 36 | 8 | 15 | 0 | 0 | 1 | 0 | 0 | 0 |
| **3910** | 152 | 0 | 0 | 0 | 0 | 1 | 0 | 152 | 1 |

3911 rows × 9 columns

In [75]:
```
projects2.columns = projects2.columns.str.replace(' ', '_')
projects2
```

Out[75]:

| | Extremely_Low_Income_Units | Very_Low_Income_Units | Low_Income_Units | Moderate_Income_Units | Middle_Income_Units | O |
|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 1 | |
| **1** | 0 | 0 | 0 | 0 | 1 | |
| **2** | 0 | 0 | 0 | 0 | 1 | |
| **3** | 0 | 0 | 1 | 0 | 0 | |
| **4** | 0 | 0 | 0 | 0 | 1 | |
| **...** | ... | ... | ... | ... | ... | |
| **3906** | 0 | 0 | 0 | 0 | 2 | |
| **3907** | 0 | 0 | 1 | 0 | 0 | |
| **3908** | 35 | 108 | 54 | 0 | 0 | |
| **3909** | 36 | 8 | 15 | 0 | 0 | |
| **3910** | 152 | 0 | 0 | 0 | 0 | |

3911 rows × 9 columns

In [77]:
```
model = smf.logit("Prevailing_Wage_Status_Prevailing_Wage ~ Extremely_Low_Income_Units + Very_Low_Income_Units + Low_
model.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.065639
         Iterations 13
```

Out[77]:

Logit Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Prevailing_Wage_Status_Prevailing_Wage | **No. Observations:** | 3911 |
| **Model:** | Logit | **Df Residuals:** | 3902 |
| **Method:** | MLE | **Df Model:** | 8 |
| **Date:** | Mon, 14 Apr 2025 | **Pseudo R-squ.:** | 0.3290 |
| **Time:** | 14:26:49 | **Log-Likelihood:** | -256.71 |
| **converged:** | True | **LL-Null:** | -382.57 |
| **Covariance Type:** | nonrobust | **LLR p-value:** | 7.430e-50 |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **Intercept** | -4.6378 | 0.172 | -26.925 | 0.000 | -4.975 | -4.300 |
| **Extremely_Low_Income_Units** | 0.0132 | 0.002 | 5.640 | 0.000 | 0.009 | 0.018 |
| **Very_Low_Income_Units** | -0.0151 | 0.006 | -2.405 | 0.016 | -0.027 | -0.003 |
| **Low_Income_Units** | 0.0034 | 0.001 | 2.602 | 0.009 | 0.001 | 0.006 |
| **Moderate_Income_Units** | -0.2176 | 0.119 | -1.835 | 0.066 | -0.450 | 0.015 |
| **Middle_Income_Units** | -4.81e-05 | 0.001 | -0.043 | 0.966 | -0.002 | 0.002 |
| **Other_Income_Units** | 0.3322 | 0.149 | 2.223 | 0.026 | 0.039 | 0.625 |
| **Counted_Homeownership_Units** | -0.0047 | 0.004 | -1.305 | 0.192 | -0.012 | 0.002 |
| **Senior_Units** | 0.0223 | 0.003 | 7.243 | 0.000 | 0.016 | 0.028 |

In [79]:
```
model.pred_table()
```

Out[79]:
```
array([[3822.,   11.],
       [  56.,   22.]])
```

In [126…
```
x = projects2.drop("Prevailing_Wage_Status_Prevailing_Wage", axis = 1)
y = projects2["Prevailing_Wage_Status_Prevailing_Wage"]
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size= .2)
```

In [ ]:
```
tree = DecisionTreeClassifier(max_depth = 5)
tree.fit(x_train,y_train)
```

In [ ]:
```

```