New York City Public Housing and its Sustainable Development Opportunities

INFO2950 Project

(None of the team member is a native English speaker.)

A. Introduction

1. One-Sentence Summary

How can we identify New York City public housing buildings with the highest potential for sustainable redevelopment by analyzing the interplay between occupancy rates and solar energy potential?

2. Background Information

The New York City Housing Authority (NYCHA) provides affordable housing to over 400,000 residents[1], yet faces significant challenges, including variable occupancy rates and the need for sustainable energy solutions.

Occupancy Rates

Occupancy rate, defined as the ratio of total residents to total units, serves as a critical indicator of housing resource utilization. High occupancy rates can reflect efficient use of space but may also strain building energy systems and negatively impact residents' quality of life due to overcrowding. Conversely, low occupancy rates often signal underutilized resources and financial inefficiencies, potentially making buildings less attractive to residents.

Solar Potential

Solar energy integration presents a dual opportunity to address environmental sustainability and reduce energy shortages. A building's solar potential is influenced by factors such as roof size, roof condition, and orientation. By prioritizing buildings with high solar potential for retrofits, NYCHA can lower operational costs and its carbon footprint. However, financial constraints necessitate strategic decision-making to maximize the impact of such retrofits.

Integrated Approach

A combined analysis of occupancy rates and solar potential offers a holistic framework for identifying buildings that stand to benefit the most from redevelopment. For high-occupancy buildings, solar panel installations can offset significant energy demand and alleviate stress on building systems. Low-occupancy buildings, enhanced with renewable energy solutions, may attract more residents and optimize utilization. By understanding these dynamics, the project aims to support data-driven strategies for sustainable public housing redevelopment.

3. Key Terms

- **Occupancy Rate**: The ratio of total residents to total units in a building, reflecting resource utilization and highlighting potential issues like overcrowding or underutilization.
- **Solar Potential**: A building's ability to generate solar energy, determined by factors such as roof size, orientation, shading, and structural suitability.

4. Research Questions

Main Question: What factors influence occupancy rates and solar potential in public housing buildings, and how can these factors be utilized to identify candidates for sustainable redevelopment?

1. Factors that Influencing Occupancy Rates:

- How do neighborhood proximity, building location, and access to amenities impact occupancy rates?
- What strategies could optimize occupancy rates to balance overcrowding and underutilization?

2. Factors for Estimating Solar Potential:

- Which building characteristics (e.g., roof area, building height, number of floors) most significantly affect solar potential?
- How can predictive models be developed to estimate solar potential for buildings without direct measurement data?

5. Data Source Introduction

- **Data Source Platform**:

NYC Open Data is a platform managed by the Open Data Team at the NYC Office of Technology and Innovation (OTI). It serves as a centralized repository for free public data published by New York City agencies. The platform aims to engage New Yorkers with the information produced and used by City government, promoting transparency and civic participation. Users can access a wide range of datasets across various categories, including business, education, environment, health, and more. The Open Data Team collaborates with City agencies to identify, document, and make data available, ensuring that datasets are regularly updated and errors are addressed.

- **Dataset Summary**: We are mainly using two datasets here: `New York City Housing Authority (NYCHA) Development Map Data` and `NYCHA ACCESSolar Data`. `New York City Housing Authority (NYCHA) Development Map Data` contains basic information for 325 public housing

developments, including their total number of units, total number of residents, etc. `NYCHA ACCESSolar Data` showing the solar potential of different public housing development, through data like estimated solar capacity, roof area, roof conditional rating.

6. Initial Findings

- **Factors that Influencing Occupancy Rates:** Buildings located in high-density areas tend to have higher occupancy rates, measured by the average number of residents per unit. This suggests that proximity to amenities, transportation, and other residential-related places significantly impacts public housing occupancy rates.
- **Factors for Estimating Solar Potential:** Buildings with better-maintained roof conditions demonstrate higher estimated solar energy potential. This indicates that structural upgrades to roof surfaces can directly enhance renewable energy generation capabilities.
- **What NYCHA can do based on this model?**
- Improving Occupancy Rates:
  - Beyond enhancing housing quality, NYCHA could focus on managing and improving nearby facilities and services. This could include increasing access to schools, public transit, and green spaces, as these factors influence the attractiveness and occupancy of public housing.
- Maximizing Solar Energy Potential:
  - To alleviate energy shortages and promote sustainability, NYCHA should prioritize roof condition improvements across public housing stock. This would increase the feasibility of solar panel installations, thus boosting renewable energy generation and reducing operational energy costs.

B. Data Description

1. New York City Housing Authority (NYCHA) Development Map Data

What are the observations (rows) and the attributes (columns)?

- **Observations: Each row represents a NYCHA building.**
- **Attributes: Columns include details:**
  - `BOROUGH` (BOUROUGH name in NYC),
  - `DEVELOPMENT` (Building development provider),
  - `TDS#` (Building id number),
  - `OCCUP_COMP` (Time the building was built),
  - `PROGRAM` (Funding type),
  - `CD` (community district),
  - `TOTAL UNIT` (total unit in a building)
  - `TOTAL POPULATION` (total population in a building)
  - `CONSERVATION DATE` (the date the building is conserved)

Why was this dataset created?

- To provide comprehensive information about NYCHA's housing developments, facilitating analysis, planning, and public awareness.
- To support initiatives aimed at improving housing conditions, resource allocation, and community development within NYCHA properties.

Who funded the creation of the dataset?

- The dataset is maintained by NYCHA, a public development corporation funded by the City of New York and federal sources.
- Data collection and maintenance are part of NYCHA's mandate to provide transparent and accessible information about public housing.

What processes might have influenced what data was observed and recorded and what was not?

- Data is collected through NYCHA's internal records, including property management systems and administrative databases.
- Regular updates ensure the dataset reflects current conditions, though there may be delays in capturing recent changes.

What preprocessing was done, and how did the data come to be in the form that you are using?

- Data is standardized and cleaned to ensure consistency across entries.
- Geospatial data is formatted for integration with mapping tools, enabling visualization of development locations.

If people are involved, were they aware of the data collection and if so, whatpurpose did they expect the data to be used for?

- The dataset does not contain personal information about residents.
- It focuses on structural and administrative details of the housing developments.

Where can your raw source data be found, if applicable?

- Raw data are all stored in the NY Open Data Portal and GitHub repository.

2. NYCHA ACCESSolar Data

What are the observations (rows) and the attributes (columns)?

- **Observations:** Each row represents a NYCHA building identified as a potential site for solar PV development under the ACCESSolar program.
- **Attributes:** Columns include details:
  - `BBL` (Building address),

- `borough` (Borough name in NYC),
- `roof area` (Roof area of the building),
- `roof condition rating` (Condition rating of the building's roof),
- `estimated solar capacity` (Estimated solar capacity of the building's roof),
- `No. of units` (Number of units in each building).

Why was this dataset created?

- The dataset was created to identify and evaluate NYCHA buildings suitable for solar PV installations. It supports the ACCESSolar program's goals of maximizing solar potential, providing green jobs to residents, and expanding solar access to low- and moderate-income (LMI) communities.
- Identifying sites suitable for solar PV development.
- Supporting NYCHA's goal of installing 25 MW of solar capacity through the ACCESSolar program.

Who funded the creation of the dataset?

- The dataset's creation and the ACCESSolar program are funded by NYCHA, with support from various partners, including the Mayor's Office of Climate and Sustainability, Con Edison, NYSERDA, Sustainable CUNY, ICF, Fund for Public Housing, and solar partners like Sol Purpose, Solar One, BlocPower, and Kinetic Communities Consulting.

What processes might have influenced what data was observed and recorded and what was not?

- The dataset was compiled by assessing NYCHA properties for solar suitability, considering factors like roof size, condition, and potential solar capacity. Buildings with recently replaced roofs and those not conflicting with other NYCHA initiatives were prioritized.

What preprocessing was done, and how did the data come to be in the form that you are using?

- Data preprocessing involved organizing information at the building address level, merging data from various sources, creating dummy variables for categorical attributes, and calculating metrics such as roof space per unit. Irrelevant columns were removed to focus on sustainability and solar potential.

If people are involved, were they aware of the data collection and if so, what purpose did they expect the data to be used for?

- The dataset focuses on building characteristics and does not involve personal data collection. NYCHA residents and stakeholders are informed about the ACCESSolar program and its benefits, including potential green job opportunities and energy cost savings.

Where can your raw source data be found, if applicable?

- Raw data are all stored in the NY Open Data Portal and GitHub repository.
- Online links for raw data:
  - NYCHA Public Housing Developments: NYC Open Data
  - NYCHA Development Map: NYC NYCHA Developments
  - NYCHA ACCESSolar Opportunities: NYC Open Data

C. Data Cleaning

1. Data Cleaning Summary

- The initial master dataset involves 8 datasets, and we collect and arrangement with following logics:
  - First of all, We arrange dataframe in 4 topics: :
    - **Housing** - each column represents basic information for NYCHA housing, like total units, total population, completion date, etc.
    - **Economic** - each column represents economic information for NYCHA housing, like job placement and average wage.
    - **Sustainablity** - each column represents sustainability information for NYCHA housing, like solor potential, roof area size, etc.
    - **Facility** - each column represents facility information for NYCHA housing, such as community center access and service availability.
  - For building a clear hierarchy, we also arrange dataframes in 3 levels: :
    - **Address level** - each row represents an address in New York City, which is an NYCHA building, and there are around 325 buildings (rows) in the address-level dataframe.
    - **Community District level** - each row represents a community district in New York City, and there are around 50 community district(rows) in the council district-level dataframe.
    - **borough level** - each row represents a borough in New York City, and there are 5 boroughs(rows) in the borough-level dataframe.
- Based on the research questions, we decided to focus on **Housing** and **Sustainablity** topics at the **Address level**. The full dataset and cleaning process in stored in the `Phase5_DataCleaning.ipynb` (https://github.com/kcx648/Info2950_group/blob/FINAL/Phase5_DataCleaning.ipynb) in our Github Repository.

2. Data Cleaning Step

For datasets under each topic, we will do the data cleaning in following steps:

- 1. **Pre-Processing**:We need to read the csv data, then do some basic processing methods like renaming columns and cleaning column names.
- 2. **Create dataframe (Address-level)**:Arrange the data into a dataframe with 325 rows. Each row represent a public housing building.
- 3. **Create dataframe (Community District-level)**:Arrange the data into a dataframe with 52 rows. Each row represent a community council.
- 4. **Create dataframe (Borough-level)**:Arrange the data into a dataframe with 5 rows. Each row represent a borough.

After creating four dataset for four topics, we will create three master datasets in following steps:

- 5. **Merge all datasets by Address-level**:Arrange four dataframes into a master dataframe with 325 rows. Each row represent a public housing building.
- 6. **Merge all datasets by Community-level**:Arrange four dataframes into a master dataframe with 52 rows. Each row represent a community council.
- 7. **Create dataframe by Borough-level**:Arrange four dataframes into a master dataframe with 5 rows. Each row represent a borough.

We will work the the **master dataframe at address level** in later Data Analysis Phase.

D. Preregistration Statement

Hypothesis 1: Buildings located in high-density areas will have higher occupancy rates (residents per unit).

Significance of analyzing Hypothesis 1: Analyzing occupancy rates is crucial for assessing the effective use of public housing resources. A positive correlation between neighborhood density and occupancy rates indicates areas with higher housing demand and efficient resource utilization. By focusing redevelopment efforts in these high-demand areas, the New York City Housing Authority (NYCHA) can strategically allocate resources to maximize housing utility and benefit more residents.For example, Smith (2020)[2] emphasizes that "addressing vacancy rates is essential for community stability and optimal resource use." Incorporating occupancy rate analysis with urban density metrics enables NYCHA to make informed decisions, enhancing housing efficiency and better serving community needs.

Null Hypothesis ((H_0)): The neighborhood density has no effect on occupancy rates. (βproximity = 0)

Alternative Hypothesis ((H_a)): Higher neighborhood density is associated with higher occupancy rates. (βproximity > 0)

Analysis: Perform a multivariable linear regression with the dependent variable as `occupancy_rate` (residents per unit). Independent variables will include:

- `neighborhood_proximity_score` (a score indicating the density of nearby buildings within a specified radius),
- `occupancy_completion_year` (the year the building was completed),
- `total_population` (total residents in the building).

The key variable of interest is `neighborhood_proximity_score`. We will examine the regression coefficient β for this variable to test the null hypothesis. A statistically significant positive β (βproximity = 0), with a p-value < 0.05) would lead us to reject the null hypothesis, indicating that buildings in higher-density areas are indeed associated with higher occupancy rates, supporting the alternative hypothesis.

Hypothesis 2: Building with higher roof conditioning rates has higher estimated roof solar energy.

Significance of analyzing Hypothesis 2: Assessing roof condition is crucial when identifying public housing buildings in New York City for sustainable redevelopment. Well-maintained roofs are essential for the successful installation of solar panels, as they provide a stable foundation and ensure the longevity of the solar energy system. By confirming a positive relationship between roof condition ratings and estimated solar energy potential, NYCHA can prioritize buildings for solar energy retrofits, focusing on structures with the highest likelihood of success and return on investment. For instance, Heinrich et al. (2020)[3] emphasize that "one of the most important criteria for determining the suitability of a building for rooftop solar is the current age of its roof," highlighting the significance of roof condition in solar panel installations. By integrating roof condition assessments with solar potential evaluations, NYCHA can strategically allocate resources to enhance sustainability and energy efficiency in public housing.

Null Hypothesis ((H_0)): The higher building roof conditioning rates has no relationship with higher roof solar energy. (βproximity = 0)

Alternative Hypothesis ((H_a)): Higher roof conditioning rate is associated with higher estimated roof solar energy. (βproximity > 0) **Analysis**: Perform a multivariable linear regression with the dependent variable as `ESTIMATED ROOF SOLAR CAPACITY (kW)` (solar potential). Independent variables will include:

- `roof space per unit` (average roof space area for each unit in the building),
- `ROOF CONDITION RATING` (roof condition rates for each building roof, ranging from 1 to 5 from poor to excellent),
- `borough_MANHATTAN` (dummy that presents if this building is in Manhattan).

The key variable of interest is `ROOF CONDITION RATING`. We will analyze the regression coefficient β for this variable, specifically testing the hypothesis βproximity > 0. A positive β value would indicate that buildings with higher roof conditioning rates has higher estimated roof solar energy.

E. Data Analysis & Evaluation of significance

In [1]:
```
# Import Packages
import numpy as np
import duckdb
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import geopandas as gpd
import contextily as ctx
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression, LogisticRegression
from statsmodels.stats.outliers_influence import variance_inflation_factor
from shapely import wkt
from shapely.geometry import Point
from sklearn.model_selection import train_test_split
from sklearn.metrics import root_mean_squared_error, mean_absolute_error, \
    mean_absolute_percentage_error, accuracy_score, precision_score, \
        recall_score, f1_score, precision_recall_curve
```

Hypothesis 1 : Building locate in a high density area will have a higher occupancy rates (residents per unit).

1. Multivariable Linear Regression Preparation

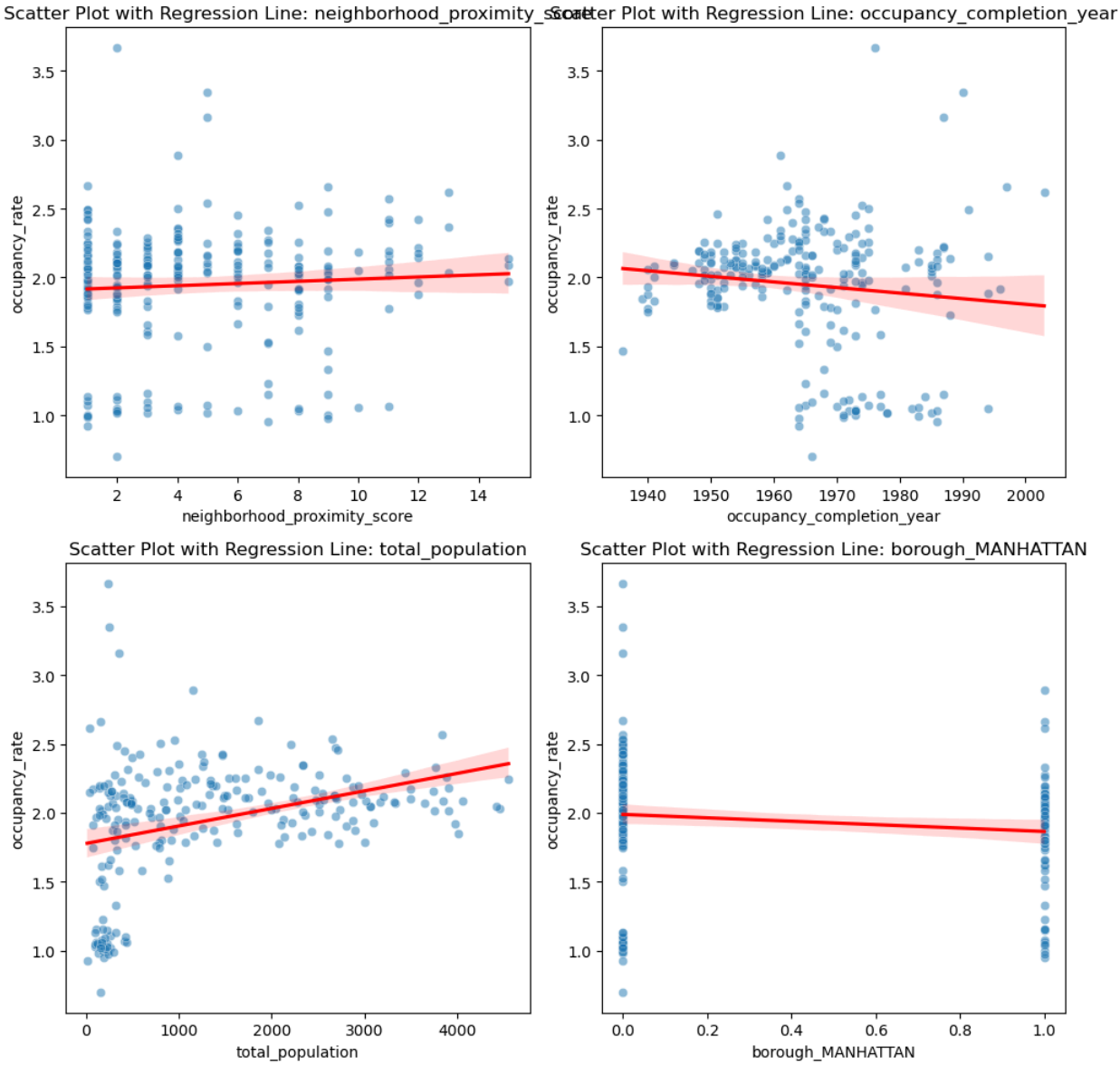
Before running the regression, we want to verify two key aspects:

- **Scatter Plot and Residual Plot:** Check each selected x-input individually to examine both its distribution and residuals, helping to identify and avoid **heteroscedasticity** by doing transformation or removing outliners.
- **Correlation Matrix:** Analyze all x-inputs together to detect and address **multicollinearity**.

a. Scatter Plot for checking Hetroskedasticity

In [2]:
```
#import the cleaned dataframe
basic_geo_address = pd.read_csv('basic_address.csv')

#prepare function for examining inputs
def plot_scatter_and_regression(data, target_var, input_vars):
    """
    Plots scatter plots with regression lines for a list of input variables,
    arranged in 2 columns per row.

    Parameters:
        data (pd.DataFrame): The DataFrame containing the data.
        target_var (str): The name of the target variable (y-axis).
        input_vars (list): A list of input variable names (x-axis).
    """
    num_plots = len(input_vars)
    num_rows = (num_plots // 2) + (num_plots % 2)

    fig, axes = plt.subplots(num_rows, 2, figsize=(10, num_rows * 5))

    for i, x_var in enumerate(input_vars):
        row = i // 2
        col = i % 2

        # Scatter Plot with Regression Line
        sns.scatterplot(
            y=data[target_var],
            x=data[x_var],
            alpha=0.5,
            ax=axes[row, col] if num_rows > 1 else axes[col]
        )
        sns.regplot(
            y=data[target_var],
            x=data[x_var],
            scatter=False,
            color='red',
            ax=axes[row, col] if num_rows > 1 else axes[col]
        )
        axes[row, col].set_title(f'Scatter Plot with Regression Line: {x_var}')
        axes[row, col].set_xlabel(x_var)
        axes[row, col].set_ylabel(target_var)

    # Adjust layout to ensure no overlapping
    plt.tight_layout()
    plt.show()
```

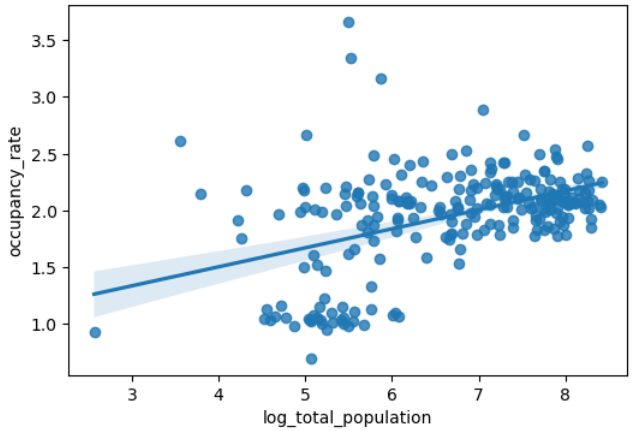
In [3]:
```
# list of intented input variables
occ_input_vars = ['neighborhood_proximity_score', 'occupancy_completion_year',
                  'total_population', 'borough_MANHATTAN']

#run the visualization
plot_scatter_and_regression(
    data=basic_geo_address,
    target_var='occupancy_rate',
    input_vars=occ_input_vars
)
```



Conclusion for Scatter Plot:

- the `total_poluation` plot shows an uneven distribution, which may lead to heteroscedasticity. We will consider transforming the data to improve the model's performance.

In [4]:
```
# log transformation of 'total_population'
basic_geo_address['log_total_population'] = np.log(basic_geo_address['total_population'] + 1)

# Review the result
plt.figure(figsize=(6, 4))
ax = sns.regplot(data=basic_geo_address,
                 x=basic_geo_address['log_total_population'], y=basic_geo_address["occupancy_rate"], n_boot=30)
plt.show()
```

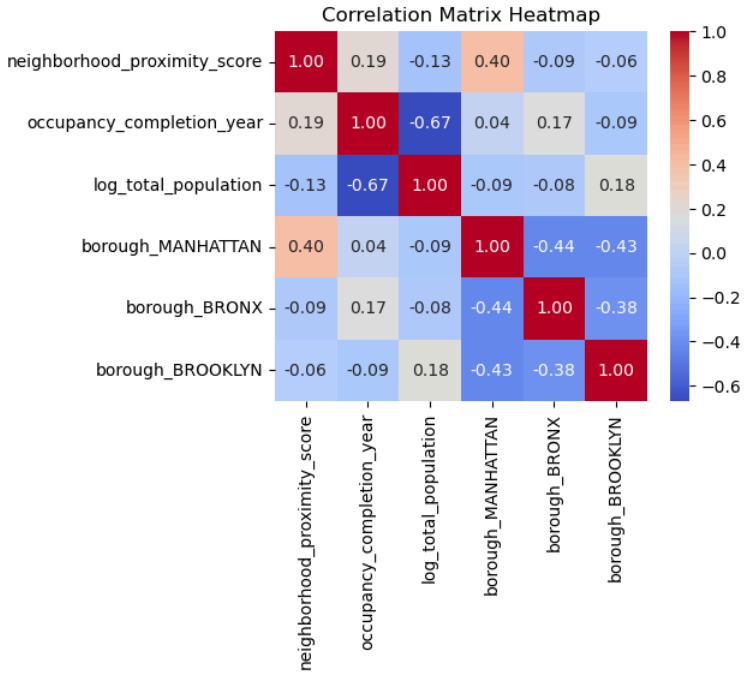


Conclusion for log-transformed result:

- The log-transformed `total_population` plot shows a more even distribution, which may help reduce heteroscedasticity in the model.

b. Correlation Matrix for checking Multicollinearity

In [5]:
```
# refined input variables
occ_input_vars = ['neighborhood_proximity_score', 'occupancy_completion_year',
                  'log_total_population',
                  'borough_MANHATTAN', 'borough_BRONX', 'borough_BROOKLYN',
                  ]

corr_columns = basic_geo_address[occ_input_vars]
correlation_matrix = corr_columns.corr()

# Create a heatmap
plt.figure(figsize=(5, 4))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm', cbar=True)

# Show the plot
plt.title("Correlation Matrix Heatmap")
plt.show()
```



Conclusion for Correlation Matrix : In the corrleation matrix, we observed two key findings:

- `occupancy_completion_year` and `log_total_units` have a strong negative correlation of 0.67.
- `borough_bronx` , `borough_brooklyn` , `borough_manhattan` ,have moderate negative correlation around -0.4.

Since these variables are highly correlated, they could lead to multicollinearity in the regression model, which may affect the model's accuracy and reliability. Therefore, we will remove `occupancy_completion_year` , `borough_bronx` , `borough_brooklyn` ,from the regression model to avoid multicollinearity.

2. Train/Test Dataset preparation

We will divide the dataset into two train/test set.Given the small number of dataset, we will assign 0.25 test size instrad of 0.30. Because the dataset we using is the full dataset of all NYCHA Public Housing, there is no need to do the resampling.

In [6]:
```
#split the data into train and test
occ_train, occ_test = train_test_split(
    basic_geo_address, test_size=0.25, random_state=2950)
print(occ_train.shape)
print(occ_test.shape)
```

```
(178, 23)
(60, 23)
```

3. Run Regression Model

With the updated input variables, we will run the multivariable linear regression model to predict the `occupancy_rate` based on the selected input variables. We will evaluate the model's performance using the R-squared value and the coefficients of the input variables.

In [7]:
```
occ_input_vars = ['neighborhood_proximity_score',
     'log_total_population',
     'borough_MANHATTAN']

x = occ_train[occ_input_vars]  # Predictor
y = occ_train['occupancy_rate']  # Response


X1 = sm.add_constant(x)
X1 = X1.apply(pd.to_numeric, errors='coerce')
y = pd.to_numeric(y, errors='coerce')

# Drop rows with missing values
X1.dropna(inplace=True)
y.dropna(inplace=True)
X1, y = X1.align(y, join='inner', axis=0)

# Fit the regression model
occ_model = sm.OLS(y, X1).fit()

print(occ_model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         occupancy_rate   R-squared:                       0.217
Model:                            OLS   Adj. R-squared:                  0.203
Method:                 Least Squares   F-statistic:                     16.03
Date:                Mon, 09 Dec 2024   Prob (F-statistic):           3.03e-09
Time:                        20:02:45   Log-Likelihood:                -85.177
No. Observations:                 178   AIC:                             178.4
Df Residuals:                     174   BIC:                             191.1
Df Model:                           3
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                            0.9387      0.182      5.149      0.000       0.579       1.298
neighborhood_proximity_score     0.0233      0.009      2.477      0.014       0.005       0.042
log_total_population             0.1462      0.026      5.703      0.000       0.096       0.197
borough_MANHATTAN               -0.2302      0.069     -3.326      0.001      -0.367      -0.094
==============================================================================
Omnibus:                       34.019   Durbin-Watson:                   1.996
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              117.577
Skew:                           0.679   Prob(JB):                     2.94e-26
Kurtosis:                       6.743   Cond. No.                         54.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

4. Evaluation of Significance

Conclusion for Multivariable Linear Regression:

- **R-squared value: 0.217**
  - This indicates that the selected input variables explain approximately **21.7% of the variance** in `occupancy_rate`. While not exceptionally high, this is reasonable considering the complexity of human occupancy behaviors and the limited dataset size. The R-squared value suggests that the model captures a meaningful portion of the variance while leaving room for additional unexplained factors.
- **F-statistic: 16.03 (p = 3.03e-09)**
  - The significant F-statistic (p < 0.001) demonstrates that the model as a whole provides a statistically significant fit to the data. This result suggests that at least one predictor variable meaningfully contributes to explaining the variability in `occupancy_rate`, supporting the utility of the model in identifying key factors.
- **Key Predictors:**
  - **`neighborhood_proximity_score`**:
    - Coefficient: 0.0233, p = 0.014
    - Although the coefficient of 0.0233 might appear small at first glance, its statistical significance (p < 0.05) indicates a consistent and reliable association between `neighborhood_proximity_score` and `occupancy_rate`. Given that `occupancy_rate` ranges from 0 to 3, a 0.0233

increase per unit of proximity score is meaningful in the context of the scale.
- This result underscores that `neighborhood_proximity_score` is an important predictor. Specifically, a unit increase in the score is associated with a 2.33% increase in `occupancy_rate` on average, holding other factors constant.

Conclusion for Preregistration Hypothesis:

- **Summary:**

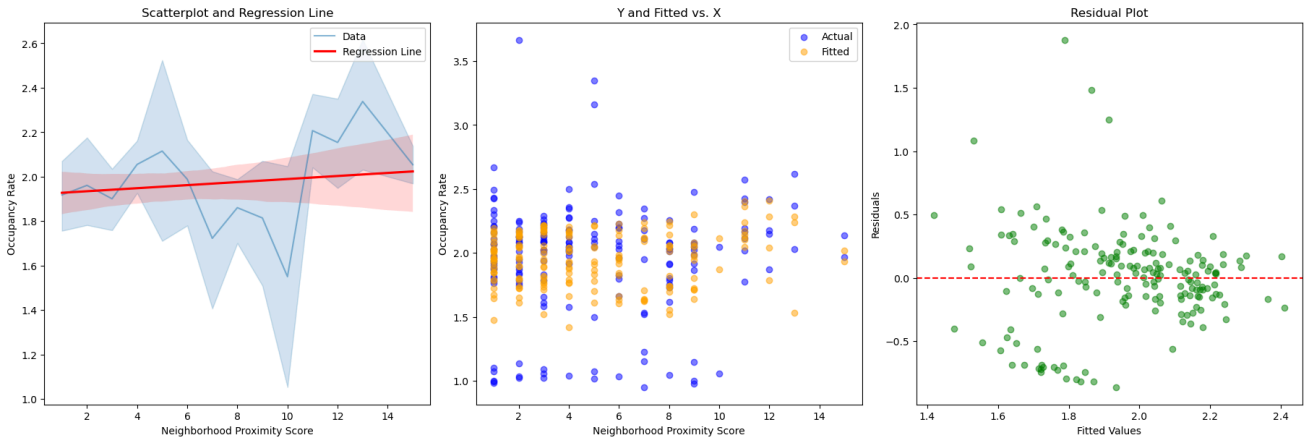
The regression analysis identifies `neighborhood_proximity_score` as a significant predictor of occupancy rates in public housing buildings. The positive coefficient (0.0233) supports the hypothesis that higher proximity to other buildings is associated with increased occupancy rates. This aligns with the idea that proximity may enhance the appeal or practicality of housing through increased accessibility or neighborhood vibrancy.

- **Reject or Accept Null Hypothesis:**

The p-value of 0.014 is below the 0.05 significance threshold, allowing us to confidently **reject the null hypothesis ((H_0))**. This result supports the alternative hypothesis that higher `neighborhood_proximity_score` positively impacts occupancy rates, suggesting a role for neighborhood density in attracting or retaining residents.

5. Addtional Evaluation through Plot

In [8]:
```
#prepare the data for plot
X = occ_train['neighborhood_proximity_score']
y = occ_train['occupancy_rate']
fitted = occ_model.fittedvalues  # Fitted values from your model
residuals = occ_model.resid      # Residuals from your model

# Create the framework
fig, axes = plt.subplots(1, 3, figsize=(18, 6), constrained_layout=True)

# Plot 1: Scatterplot and Regression Line
sns.lineplot(x=X, y=y, alpha=0.5, ax=axes[0], label='Data')
sns.regplot(x=X, y=y, scatter=False, color='red', ax=axes[0], label='Regression Line')
axes[0].set_title('Scatterplot and Regression Line')
axes[0].set_xlabel('Neighborhood Proximity Score')
axes[0].set_ylabel('Occupancy Rate')
axes[0].legend()

# Plot 1: Y and Fitted vs. X Plot
axes[1].scatter(X, y, alpha=0.5, label='Actual', color='blue')
axes[1].scatter(X, fitted, alpha=0.5, label='Fitted', color='orange')
axes[1].set_title('Y and Fitted vs. X')
axes[1].set_xlabel('Neighborhood Proximity Score')
axes[1].set_ylabel('Occupancy Rate')
axes[1].legend()

# Plot 3: Residual Plot
axes[2].scatter(fitted, residuals, alpha=0.5, color='green')
axes[2].axhline(0, color='red', linestyle='--')
axes[2].set_title('Residual Plot')
axes[2].set_xlabel('Fitted Values')
axes[2].set_ylabel('Residuals')

#Display
plt.show()
```



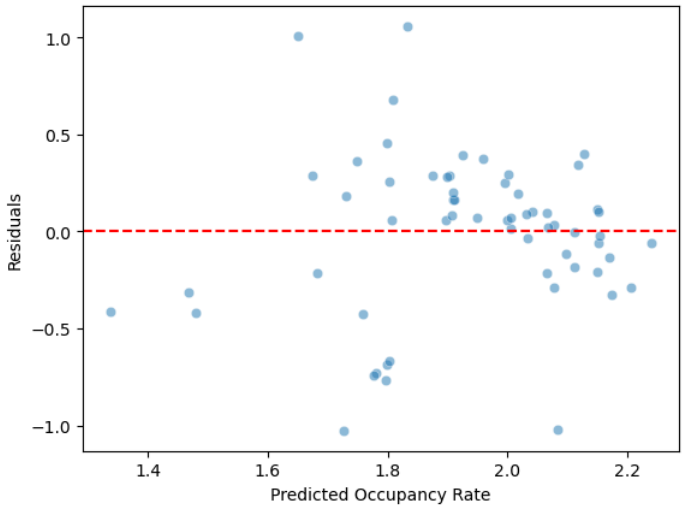
Conclusion for Addtional Evaluation of Significance:

- The regression plot indicates a moderate fit between the predicted and actual `occupancy_rate` values, suggesting the model captures key trends but leaves room for improvement.
- In the 'Y and Fitted vs. X' plot, the model demonstrates better predictive performance when the predicted `occupancy_rate` falls within the range of 1.5 to 2.5, aligning closely with the actual values.
- The Residual Plot shows that residuals are randomly distributed around the center line, confirming that the model meets key assumptions of linearity and homoscedasticity.

6. Model Evaluation through Test Dataset

In [9]:
```
# Add a constant for the intercept to the test data
occ_test_with_const = sm.add_constant(occ_test[occ_input_vars])

# predict the test data
occ_test_prediction = occ_model.predict(occ_test_with_const)

print(occ_test['occupancy_rate'].describe())

mae = mean_absolute_error(occ_test['occupancy_rate'], occ_test_prediction)
print(f"Mean Absolute Error (MAE): {mae}")

rmse = root_mean_squared_error(occ_test['occupancy_rate'], occ_test_prediction)
print(f"Root Mean Squared Error (RMSE): {rmse}")

occ_residuals = occ_test['occupancy_rate'] - occ_test_prediction
sns.scatterplot(x=occ_test_prediction, y=occ_residuals, alpha=0.5)
plt.xlabel('Predicted Occupancy Rate')
plt.ylabel('Residuals')
plt.axhline(y=0, color='red', linestyle='--')
plt.show()
```

```
count    60.000000
mean      1.925145
std       0.469682
min       0.699115
25%       1.864173
50%       2.065429
75%       2.181133
max       2.889447
Name: occupancy_rate, dtype: float64
Mean Absolute Error (MAE): 0.30439445756159994
Root Mean Squared Error (RMSE): 0.4101518634035691
```



Conclusion for Test Data Prediction Results:

- With a mean occupancy rate of 1.93, the model achieves reasonable prediction accuracy, as evidenced by a Mean Absolute Error (MAE) of approximately 0.31 and a Root Mean Squared Error (RMSE) of about 0.41. These errors represent 16-21% of the mean value, which is acceptable for this type of analysis.
- The residual plot shows that around 60% of residuals are evenly distributed around zero, particularly when predicted values are close to 2. This suggests that the model performs well within this range, capturing the key trends in the test data.
- Despite the small R-value, likely influenced by the limited dataset size, the model demonstrates a moderate fit for the test data, particularly for high occupancy rates.

Hypothesis 2: Building with higher roof conditioning rates has higher estimated roof solar energy.

1. Multivariable linear regression preperation:

In [10]:
```
#import the cleaned dataframe
S3=pd.read_csv("Sustainability_solar potential.csv")
```

Checking each variable's regression outliers and needs for transform

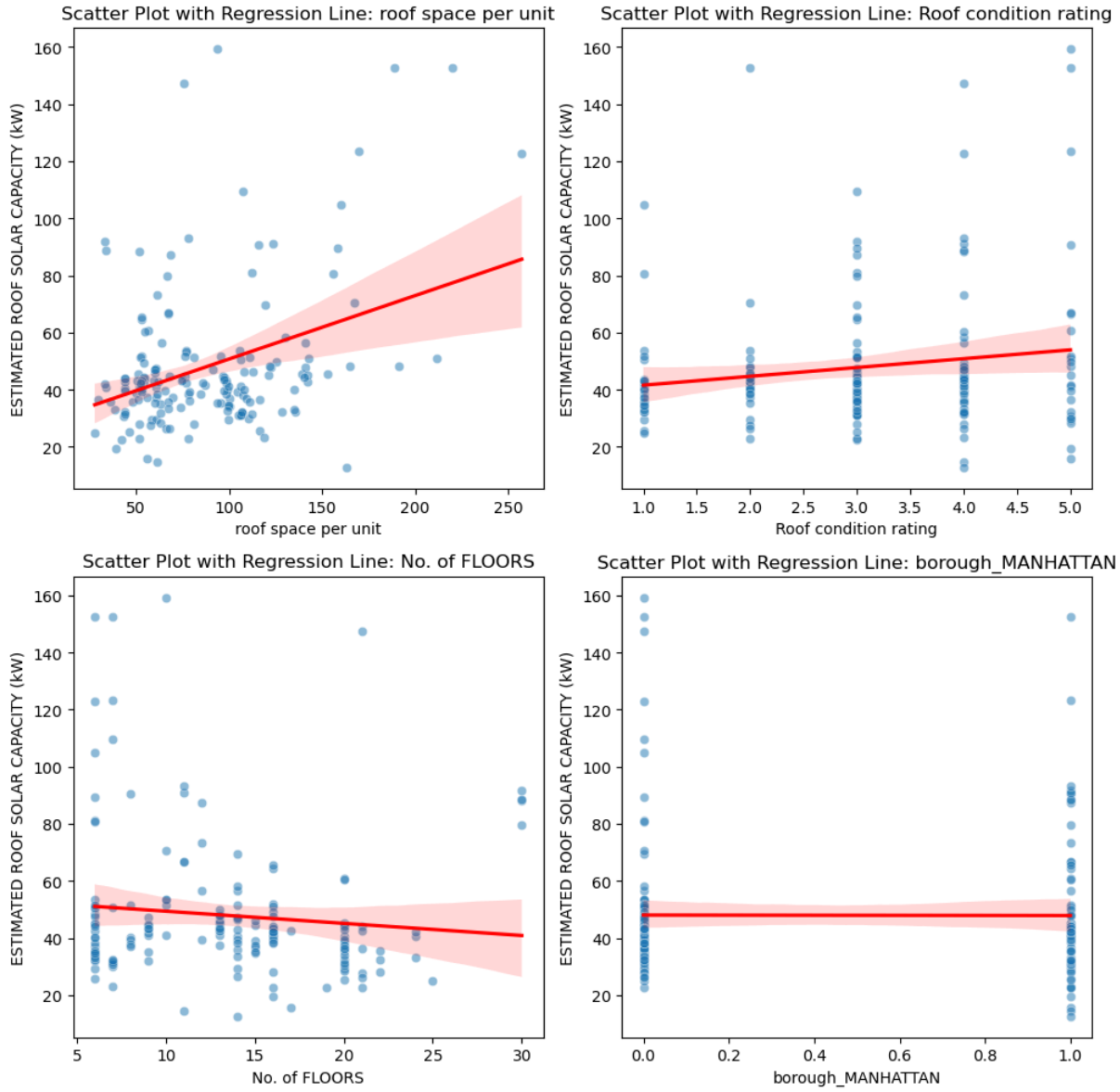
Note: For each independent variable and dependent variable, we dropped the 0 value in cleaning part because they are missing data.

In [11]:
```
#prepare function for examining inputs
def plot_scatter_and_regression(data, target_var, input_vars):
    """
    Plots scatter plots with regression lines for a list of input variables,
```

```
    arranged in 2 columns per row.

    Parameters:
        data (pd.DataFrame): The DataFrame containing the data.
        target_var (str): The name of the target variable (y-axis).
        input_vars (list): A list of input variable names (x-axis).
    """
    num_plots = len(input_vars)
    num_rows = (num_plots // 2) + (num_plots % 2)

    fig, axes = plt.subplots(num_rows, 2, figsize=(10, num_rows * 5))

    for i, x_var in enumerate(input_vars):
        row = i // 2
        col = i % 2

        # Scatter Plot with Regression Line
        sns.scatterplot(
            y=data[target_var],
            x=data[x_var],
            alpha=0.5,
            ax=axes[row, col] if num_rows > 1 else axes[col]
        )
        sns.regplot(
            y=data[target_var],
            x=data[x_var],
            scatter=False,
            color='red',
            ax=axes[row, col] if num_rows > 1 else axes[col]
        )
        axes[row, col].set_title(f'Scatter Plot with Regression Line: {x_var}')
        axes[row, col].set_xlabel(x_var)
        axes[row, col].set_ylabel(target_var)

    # Adjust layout to ensure no overlapping
    plt.tight_layout()
    plt.show()
```

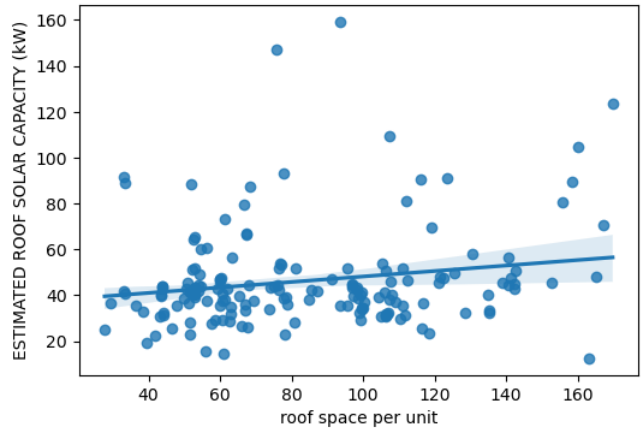
In [12]:

```
# list of intented input variables
input_vars = ['roof space per unit', 'Roof condition rating',
                  'No. of FLOORS', 'borough_MANHATTAN']

#run the visualization
plot_scatter_and_regression(
    data=S3,
    target_var='ESTIMATED ROOF SOLAR CAPACITY (kW)',
    input_vars=input_vars
)
```



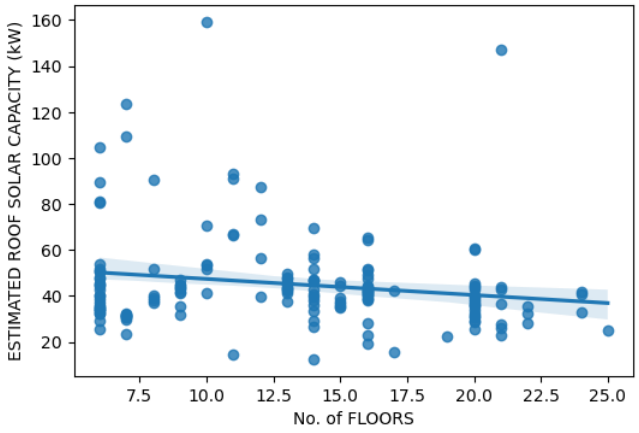
After observing extreme outliers beyond 180 roof space per unit, we remove them and replot the regression plot.

In [13]:
```
#Remove outliers for roof space per unit vs. Estimated solar potential plot
S3=S3[(S3["roof space per unit"] <= 180)]
plt.figure(figsize=(6, 4))
ax = sns.regplot(data=S3, x=S3["roof space per unit"],
                 y=S3["ESTIMATED ROOF SOLAR CAPACITY (kW)"], n_boot=30)
plt.show()
```



After observing outliers of 30 floors and more, We remove them and replot the regression plot.

In [14]:
```
#Remove floor numbers and estimated solar potential plot outliers
S3["No. of FLOORS"] = pd.to_numeric(S3["No. of FLOORS"], errors='coerce')
S3 = S3[(S3["No. of FLOORS"] != 0) & (S3["No. of FLOORS"] <30)]
plt.figure(figsize=(6, 4))
ax = sns.regplot(data=S3, x=S3["No. of FLOORS"], y=S3["ESTIMATED ROOF SOLAR CAPACITY (kW)"], n_boot=30)
plt.show()
```

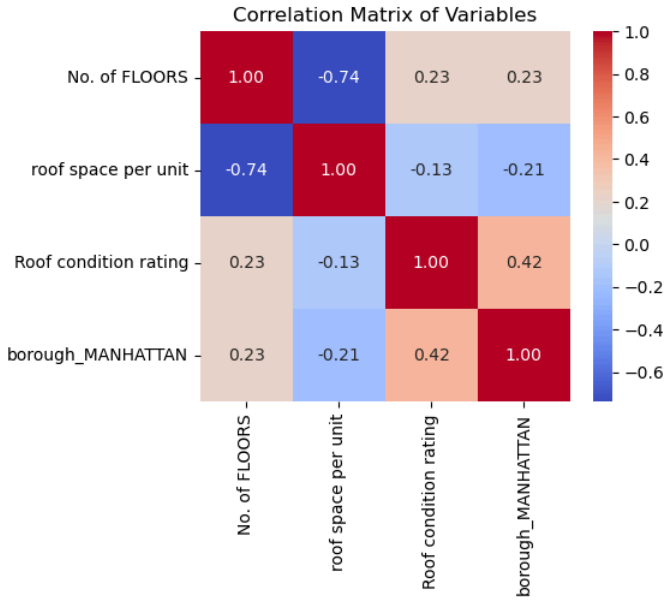


2. Train/Test Dataset Preparation

We want to check whether some of the independent variables are correlated with each other.

In [15]:
```
#split the train and test sets with 7:3 ratio
train, test = train_test_split(S3, test_size=0.3, random_state=2950)
```

In [16]:
```
columns_of_interest = [
"No. of FLOORS", "roof space per unit",
       "Roof condition rating",
    "borough_MANHATTAN"
]
data = S3[columns_of_interest]

# Compute the correlation matrix
correlation_matrix = data.corr()

# Plot the correlation matrix
plt.figure(figsize=(5, 4))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap="coolwarm", cbar=True, square=True)
plt.title("Correlation Matrix of Variables")
plt.show()
```



After reviewing the correlation matrix, we figure out No. of floors cause the multicolinary issue and decide to drop them.

3. Run Regression Model

In [17]:
```
#Final OLS regression model results after dropping correlated independent variables
X = train[[ "roof space per unit",
       "Roof condition rating",
```

```
    "borough_MANHATTAN"
        ]]
y = train["ESTIMATED ROOF SOLAR CAPACITY (kW)"]
X = sm.add_constant(X)
# Fit and print the summary of the linear regression model
model = sm.OLS(y, X).fit()
print(model.summary())
```

```
                                OLS Regression Results
==============================================================================================
Dep. Variable:     ESTIMATED ROOF SOLAR CAPACITY (kW)   R-squared:                       0.138
Model:                                            OLS   Adj. R-squared:                  0.115
Method:                                 Least Squares   F-statistic:                     5.876
Date:                                Mon, 09 Dec 2024   Prob (F-statistic):           0.000929
Time:                                        20:02:48   Log-Likelihood:                -492.55
No. Observations:                                 114   AIC:                             993.1
Df Residuals:                                     110   BIC:                             1004.
Df Model:                                           3
Covariance Type:                            nonrobust
============================================================================================
                               coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------------
const                    19.5557      6.713      2.913      0.004       6.252      32.859
roof space per unit       0.1731      0.051      3.367      0.001       0.071       0.275
Roof condition rating     3.8619      1.492      2.588      0.011       0.905       6.819
borough_MANHATTAN        -4.0568      3.923     -1.034      0.303     -11.830       3.717
==============================================================================
Omnibus:                       80.289   Durbin-Watson:                   1.915
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              512.798
Skew:                           2.372   Prob(JB):                    4.44e-112
Kurtosis:                      12.244   Cond. No.                        356.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

4. Evaluation of Significance

Conclusion for Multivariable Linear Regression:

- **R-squared value: 0.138**
  - This indicates that the selected input variables explain approximately 13.8% of the variance in `ESTIMATED ROOF SOLAR CAPACITY (kW)`. Although this R-squared value is modest, it is reasonable considering the complexity of factors affecting solar capacity estimation and the limited dataset size. The model captures a meaningful portion of the variability in the dependent variable, suggesting it has practical utility while acknowledging room for additional unexplained factors.
- **F-statistic: 5.876 (p = 0.000929)**
  - The significant F-statistic ($p < 0.001$) demonstrates that the model provides a statistically significant fit to the data. This result confirms that at least one predictor variable meaningfully contributes to explaining the variability in `ESTIMATED ROOF SOLAR CAPACITY (kW)`, supporting the model's utility in identifying important influencing factors.
- **Key Predictors:**
  - **`roof space per unit`**:
    - Coefficient: 0.1731, $p < 0.001$
    - The coefficient for roof space per unit is 0.1731, meaning that for each unit increase in the roof space per unit, the estimated roof solar capacity increases by 0.1731 kW, holding other variables constant. The low p-value ($p < 0.001$) indicates that this relationship is statistically significant, highlighting the importance of roof size in determining solar capacity.
  - **`ROOF CONDITION RATING`**:
  - Coefficient: 3.8619, p = 0.011
  - The coefficient of 3.8619 suggests that an improvement in roof condition rating by one unit is associated with an estimated increase of 3.8619 kW in solar capacity, on average, holding other factors constant. The statistical significance ($p < 0.05$) supports the idea that better-maintained roofs are more suitable for solar installations, contributing positively to capacity estimates.

Conclusion for Preregistration Hypothesis:

- **Summary:**

The regression analysis identifies `ROOF CONDITION RATING OR REPLACEMENT DATE` as a significant predictor of estimated roof solar energy potential in public housing buildings. The positive coefficient (3.8619) supports the hypothesis that higher roof condition ratings of buildings is associated with increased estimated roof solar energy potential.

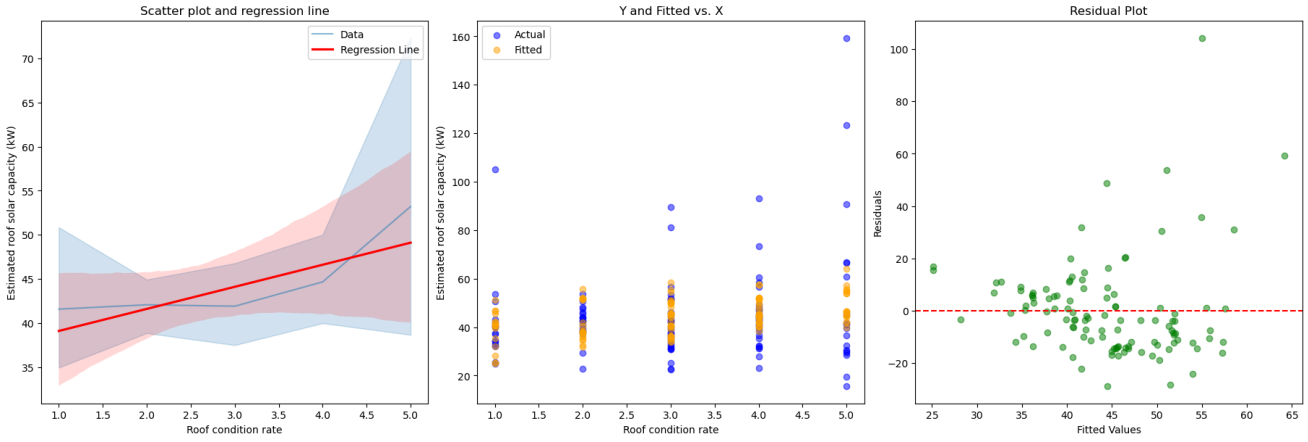
- **Reject or Accept Null Hypothesis:**

The p-value of 0.011 is below the 0.05 significance threshold, allowing us to confidently **reject the null hypothesis (H_0)**. This result supports the alternative hypothesis that higher `ROOF CONDITION RATING OR REPLACEMENT DATE` positively impacts estimated roof solar potential, suggesting a role for roof condition in increasing a roof's solar potential.

5. Additional Evaluation through Plot

In [18]:
```
#prepare the data for plot
X = train["Roof condition rating"]
```

```
y = train["ESTIMATED ROOF SOLAR CAPACITY (kW)"]
fitted = model.fittedvalues  # Fitted values from model
residuals = model.resid      # Residuals from model

# Create the framework
fig, axes = plt.subplots(1, 3, figsize=(18, 6), constrained_layout=True)

# Plot 1: Scatterplot and Regression Line
sns.lineplot(x=X, y=y, alpha=0.5, ax=axes[0], label='Data')
sns.regplot(x=X, y=y, scatter=False, color='red', ax=axes[0], label='Regression Line')
axes[0].set_title('Scatter plot and regression line')
axes[0].set_xlabel('Roof condition rate')
axes[0].set_ylabel('Estimated roof solar capacity (kW)')
axes[0].legend()

# Plot 1: Y and Fitted vs. X Plot
axes[1].scatter(X, y, alpha=0.5, label='Actual', color='blue')
axes[1].scatter(X, fitted, alpha=0.5, label='Fitted', color='orange')
axes[1].set_title('Y and Fitted vs. X')
axes[1].set_xlabel('Roof condition rate')
axes[1].set_ylabel('Estimated roof solar capacity (kW)')
axes[1].legend()

# Plot 3: Residual Plot
axes[2].scatter(fitted, residuals, alpha=0.5, color='green')
axes[2].axhline(0, color='red', linestyle='--')
axes[2].set_title('Residual Plot')
axes[2].set_xlabel('Fitted Values')
axes[2].set_ylabel('Residuals')

#Display
plt.show()
```



Conclusion for Addtional Evaluation of Significance:

- Scatter Plot and Regression Line:

The scatter plot with the regression line reveals a moderate positive relationship between the roof condition rate and the estimated roof solar capacity (kW). The regression line fits the data reasonably well, but the spread of the data points, especially in the higher roof condition ratings, suggests some unexplained variability. This could indicate that the model may not capture all of the factors influencing solar capacity, particularly at the extremes.

- 'Y and Fitted vs. X' Plot:

In the 'Y and Fitted vs. X' plot, the model demonstrates good predictive performance when the predicted estimated roof solar capacity falls within the mid-range of values. However, predictions for lower and higher capacity values show more variability. This suggests room for improvement in capturing the nuances of the relationship, particularly for outliers or edge cases.

- Residual Plot:

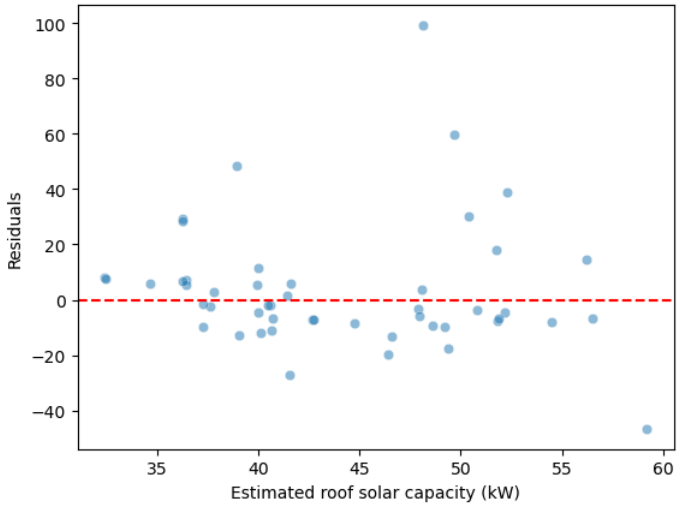
The residual plot shows a fairly random distribution around the centerline, confirming that the model meets the assumptions of linearity and homoscedasticity.

6. Model Evaluation through Test Dataset

In [19]:
```
input_vars = [
    "roof space per unit",
        "Roof condition rating",
 "borough_MANHATTAN"
        ]

output_var = 'ESTIMATED ROOF SOLAR CAPACITY (kW)'
# Add a constant term to the input data
train_with_const = sm.add_constant(train[input_vars])
test_with_const = sm.add_constant(test[input_vars])

# Generate predictions
train_predictions = model.predict(train_with_const)
test_predictions = model.predict(test_with_const)
```

In [20]:
```
#Calculate train and test RMSE and MAE
train_rmse = np.sqrt(np.mean((train['ESTIMATED ROOF SOLAR CAPACITY (kW)'] - train_predictions) ** 2))
test_rmse=np.sqrt(np.mean((test['ESTIMATED ROOF SOLAR CAPACITY (kW)'] - test_predictions) ** 2))
print(f"Train RMSE: {train_rmse:.2f}")
print(f"Test RMSE: {test_rmse:.2f}")
train_mae = np.mean(np.abs(train['ESTIMATED ROOF SOLAR CAPACITY (kW)'] - train_predictions))
test_mae = np.mean(np.abs(test['ESTIMATED ROOF SOLAR CAPACITY (kW)'] - test_predictions))
print(f"Train MAE: {train_mae:.2f}")
print(f"Test MAE: {test_mae:.2f}")

#Check MAE and RMSE with mean and sd of the data.
print(train['ESTIMATED ROOF SOLAR CAPACITY (kW)'].describe())

residuals = test['ESTIMATED ROOF SOLAR CAPACITY (kW)'] - test_predictions
sns.scatterplot(x=test_predictions, y=residuals, alpha=0.5)
plt.xlabel('Estimated roof solar capacity (kW)')
plt.ylabel('Residuals')
plt.axhline(y=0, color='red', linestyle='--')
plt.show()
```

```
Train RMSE: 18.20
Test RMSE: 22.99
Train MAE: 12.40
Test MAE: 14.60
count    114.000000
mean      44.339386
std       19.694787
min       15.720000
25%       33.377500
50%       41.050000
75%       47.007500
max      159.320000
Name: ESTIMATED ROOF SOLAR CAPACITY (kW), dtype: float64
```



Testing Test datasets Conclusion

1. RMSE:

The RMSE for the training data is 18.20, which is approximately 41.0% of the mean of the `ESTIMATED ROOF SOLAR CAPACITY (kW)`. The RMSE for the test data is 22.99, which is approximately 51.8% of the mean. These values suggest that the model has a moderate predictive accuracy.

2. MAE:

The MAE is approximately 32.7% of the mean for the train data and 33.7% of the mean for the test data. This indicates that the model's average prediction error is relatively consistent across both datasets. The MAE being lower than the RMSE highlights that most errors are moderate, with fewer large outliers affecting the results.

E. Conclusion

Conclusion of Hypothesis 1:

The results of the regression analysis indicate that `neighborhood_proximity_score` is a significant predictor of occupancy rates in public housing buildings. Specifically, a unit increase in the `neighborhood_proximity_score` is associated with a 0.02 increase in occupancy rate, holding other variables constant.

The p-value of 0.017 is below the threshold of 0.05, allowing us to **reject the null hypothesis ((H_0))** and conclude that buildings in higher-density areas tend to have higher occupancy rates. This supports the hypothesis that neighborhood density positively impacts occupancy rates, suggesting that proximity to other buildings could play a role in attracting or retaining residents.

Conclusion of Hypothesis 2:

The results of the regression analysis indicate that `ROOF CONDITION RATING` is a significant predictor of `estimated solar roof energy` in public housing buildings. A one-unit increase in `ROOF CONDITION RATING` is associated with an increase of 3.8619kW in ESTIMATED ROOF SOLAR CAPACITY, holding all other variables constant.

The p-value of 0.011 is below the threshold of 0.05, allowing us to **reject the null hypothesis ((H_0))** and conclude that buildings with higher roof conditioning rates did not associate with higher estimated roof solar energy.

Conclusion for Relationship Between Hypothesis and Research Question:

Hypothesis 1:

- Based on the research question:**"How do neighborhood proximity, building location, and access to amenities impact occupancy rates?"** The conclusion finds the factor-neighborhood proximity is influencing housing occupancy rate. The significant relationship between `neighborhood_proximity_score` and `occupancy rates` suggests that public housing in high-density areas, which may offer better access to social networks, amenities, and transportation, tends to attract more residents.
- Based on the research question: **"What strategies could optimize occupancy rates to balance overcrowding and underutilization?"** The conclusion suggests that optimizing occupancy rates requires a dual approach. First, increasing access to amenities and improving infrastructure in low-density areas can make these locations more appealing to residents. Second, implementing flexible housing policies, such as adjusting unit sizes or converting neighborhood facilities number, can help balance occupancy levels while reducing overcrowding in high-demand areas.

Hypothesis 2:

- Based on the research question:**"Which building characteristics (e.g., roof area, building height, number of floors) most significantly affect solar potential?"** Our result underscores the importance of `ROOF CONDITION RATING` in determining solar energy potential. By identifying roof condition as a critical factor, our study suggests to prioritize buildings with well-maintained roofs for solar retrofits. This ensures efficient allocation of resources and supports predictive modeling efforts to estimate solar potential for buildings lacking direct measurements.
- Based on the research question:**" How can predictive models be developed to estimate solar potential for buildings without direct measurement data?"** Our result indicating that inputs like building location(which borough they are in) , roof condition, and roof area measurement could help build a reliable model.

Conclusion for Potential Optimization Strategies and Model Usage:

- For Occupancy Rate Model:
  - Beyond enhancing housing quality, NYCHA could focus on managing and improving nearby facilities and services. This could include increasing access to schools, public transit, and green spaces, as these factors influence the attractiveness and occupancy of public housing.
  - For building that don't have recorded occupancy rate, NYCHA can use this model to have a predicted occupancy rate for these building.
- For Solar Potential Model:
  - To alleviate energy shortages and promote sustainability, NYCHA should prioritize roof condition improvements across public housing stock. This would increase the feasibility of solar panel installations, thus boosting renewable energy generation and reducing operational energy costs.
  - For building that don't have estimated solor potential measured through devices, NYCHA can use this model to have a predicted solor potential for these building.

F. Data Limitations

1. For Occupancy Ratio-related Data

1. Overall Dataset Problem

- The dataset size is limited, with only **325 NYCHA buildings** in New York City. This relatively small sample size may affect the model's accuracy and generalizability to other public housing developments or contexts.
- The dataset lacks key factors that could significantly influence occupancy rates, such as:
  - **Building amenities** (e.g., availability of parking, community centers, or playgrounds),
  - **Maintenance quality** (e.g., frequency of repairs, heating or plumbing issues),
  - **Neighborhood safety** (e.g., crime rates or proximity to police stations).

  These missing variables may limit the explanatory power of the model and contribute to the modest R-squared value.

2. Spatial and Proximity Data

- **Proximity-based variables are limited to density scores**, which only capture the number of nearby developments. Factors like proximity to public transportation, schools, or healthcare facilities—critical determinants of housing desirability—are missing.
- The spatial resolution of the data may be insufficient for capturing nuanced neighborhood effects. For example, variations within boroughs or community districts (e.g., socio-economic disparities) are not adequately represented.

3. Temporal Data

- The dataset does not include trends over time, such as changes in population dynamics or economic conditions that could influence occupancy rates.

2.For Solar Potential-related Data

1. The dataset is limited to NYCHA buildings, which might not be representative of other types of housing or rooftops in New York City. The dataset contains only around 300 BBL rows. This small sample size limits the model's ability to capture variability in building characteristics and solar potential, potentially affecting the robustness and reliability of the analysis.

2. Some buildings in the dataset may lack sufficient information on roof conditions and solar potential, which could limit the accuracy of solar capacity predictions.The dataset primarily captures current conditions and does not account for historical trends, which might impact the accuracy of future predictions or longitudinal studies.
3. **Multicollinearity** issue: The correlation between some columns are high, which makes us having difficulty fitting the linear regression.
4. Low variance of data can result in weak model performance and difficulty in identifying actionable recommendations, as the data may not adequately capture outliers or edge cases.

3. Model Usage Limitations in Real-World Applications

Occupancy Ratio Model:

1. **Neighborhood Proximity Score Calculation:** Accurately computing the `neighborhood_proximity_score` requires precise geolocation data for buildings. In the absence of exact coordinate information, it may be challenging to use this model effectively.
2. **Data Sufficiency:** For model training, we started with a community-level dataframe containing only 52 rows, which proved insufficient for generating a reliable model. This highlights that datasets with fewer than 50 buildings are likely to produce unreliable proximity scores due to insufficient information, potentially compromising the model's predictive accuracy.

Solar Potential Model:

1. **Roof Condition Rating Variability:** The `ROOF CONDITION RATING` is a NYCHA-specific metric. If other building roofs are using different assessment systems, such as the Roof Condition Index, model retraining would be necessary to accommodate these alternative metrics.
2. **Limited Input Variables:** With only three input variables, the model's performance is highly sensitive to missing data. The absence of any single input may necessitate model retraining to maintain predictive reliability.

F. Acknowledgements and Bibliography

Libraries:

- Numpy Library: (https://numpy.org/doc/stable/index.html)
- pandas documentation: (https://pandas.pydata.org/docs/index.html)
- duckdb documentation: (https://duckdb.org/docs/)
- scikitlearn library:(https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.LogisticRegression.html)
- Matplotlib Library:(https://matplotlib.org/stable/contents.html)
- Seaborn Library: (https://seaborn.pydata.org/)
- Geopandas Library: (https://geopandas.org/en/stable/)
- Contextily Library: (https://contextily.readthedocs.io/en/latest/)
- Statsmodels Library: (https://www.statsmodels.org/stable/index.html)
- Shapely Library: (https://shapely.readthedocs.io/en/stable/)
- sklearn library: (https://scikit-learn.org/stable/)

References:

- [1]New York City Housing Authority. (2023). NYCHA Fact Sheet. Retrieved from https://www.nyc.gov/assets/nycha/downloads/pdf/NYCHA-Fact-Sheet-2023.pdf
- [2] Smith, A. (2020). Challenges in Public Housing: Addressing Vacancy Rates and Community Stability. Journal of Urban Affairs.]
- [3] Heinrich, C., Laskin, M., Glinskis, S., & van Nieuwenburg, E. (2020). Roof Age Determination for the Automated Site-Selection of Rooftop Solar. arXiv preprint arXiv:2001.04227.