Rigs Question 1: Some viles required for preprocessing have been excluded as they require more thatn 100 mb space

What are the trends regarding student housing across the city, by district, e.g. what % of the rental housing is taken up by students for each district and how has this changed over time?

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import re
```

In [2]:
```
boston_wards = {
    1: "East Boston",
    2: "Charlestown",
    3: "North End, West End, Financial District, Chinatown",
    4: "Fenway, South End, Back Bay",
    5: "Back Bay, Beacon Hill",
    6: "South Boston",
    7: "South Boston, Dorchester",
    8: "South End, Roxbury",
    9: "South End, Roxbury",
    10: "Mission Hill",
    11: "Roxbury, Jamaica Plain",
    12: "Roxbury",
    13: "Dorchester, Savin Hill",
    14: "North Dorchester, Mattapan",
    15: "Dorchester, Meeting Hill House",
    16: "Dorchester (Neponset Cedar Grove)",
    17: "Dorchester (Lower Mills)",
    18: "Hyde Park, Mattapan",
    19: "Jamaica Plain, Roslindale",
    20: "West Roxbury",
    21: "Allston, Brighton",
    22: "Unknown or Newer Areas"
}
```

In [3]:
```
# Load data, ready for cleaning and drop obvious not needed columsn
df = pd.read_csv('StudentAddresses(UP).csv', low_memory=False)
df = df.drop(columns=['6d. unit #'])
df = df.drop(columns=['9. at-home or not-at-home'])
df['full_address'] = df['6a. street #'].astype(str) + ' ' + \
                     df['6b. street name'] + ' ' + \
                     df['6c. street suffix'] + ', ' + \
                     df['6e. zip'].astype(str)
df['full_address'] = df['6a. street #'].astype(str) + ' ' + \
```

```
                    df['6b. street name'] + ' ' + \
                    df['6c. street suffix']



# dropping the NaN values:
df_cleaned = df.dropna(subset=['full_address']).reset_index(drop=True)

# Filtered out data ready for district assingment:

book = df_cleaned
book['street_number'] = pd.to_numeric(book['6a. street #'], errors='coerce')
book['street_name'] = (book['6b. street name'] + ' ' + book['6c. street suff
book.head(100)
```

Out[3]:

	6a. street #	6b. street name	6c. street suffix	6e. zip	7. undergraduate (u) or graduate (g)	8. full-time (ft) or part-time (pt)	9. 5 or more undergrads/unit (y/n)	unive
0	10	Higgins	ST	2134	U	FT	NaN	Emma Co
1	10	Higgins	ST	2134	U	FT	NaN	Emma Co
2	1189	Commonwealth	AVE	2134	U	FT	NaN	Emma Co
3	12	Glenville	AVE	2134	U	FT	NaN	Emma Co
4	12	Glenville	AVE	2134	U	FT	NaN	Emma Co
...	...	...	...	...	...	...	...	
95	58	Queensberry	ST	2215	U	FT	NaN	Emma Co
96	58	Queensbury	ST	2215	U	FT	NaN	Emma Co
97	660	Washington	ST	2111	U	FT	NaN	Emma Co
98	660	Washington	ST	2111	U	FT	NaN	Emma Co
99	69	Park	DR	2215	U	FT	NaN	Emma Co

100 rows × 12 columns

Ward assignment

In [4]:
```
## Exmapnstion of location to get full set for comparison

lookup_df = pd.read_csv('../location.csv')
lookup_df.head()
count = 0

expanded_rows = []  # sotre the new row
error_rows = [] # store the error rows

for _, row in lookup_df.iterrows():
    if row['IS_RANGE'] == 1:
        # integer conversion of rows range
        try:
            range_from = int(float(row['RANGE_FROM']))
            range_to = int(float(row['RANGE_TO']))
        except ValueError:
            # print(row["RANGE_FROM"], row["RANGE_TO"])
            count += 1
            error_rows.append(row)
            continue  # invalid range indexing

        # enumerate through the entire range and create rows for each number
        for num in range(range_from, range_to + 1):
            new_row = row.copy()
            new_row['STREET_NUMBER'] = str(num)
            new_row['FULL_ADDRESS'] = f"{num} {row['FULL_STREET_NAME']}"
            expanded_rows.append(new_row)
    else:
        # if not a range, just append the row as-is
        expanded_rows.append(row)

# Convert the expanded rows back into a DataFrame
expanded_lookup_df = pd.DataFrame(expanded_rows).reset_index(drop=True)
print("The error count is: ", count)

# Standardize street number and full street name to create a properly format
expanded_lookup_df['actual_address'] = expanded_lookup_df['STREET_NUMBER'].a

expanded_lookup_df.head(100)


## PS: there are a total of 2591 range errors since they are not indexed wit
## This error_list will also be checked to see if they are in the error list
```

```
/var/folders/vt/h6zk5t3106dgbrm4lvpjm5x40000gn/T/ipykernel_78572/1595117873.
py:3: DtypeWarning: Columns (6,7,15,23,24) have mixed types. Specify dtype o
ption on import or set low_memory=False.
  lookup_df = pd.read_csv('../location.csv')
The error count is:  2591
```

Out[4]:

	SAM_ADDRESS_ID	BUILDING_ID	RELATIONSHIP_TYPE	FULL_ADDRESS	STREET_
0	1	100778	1	6 A St	
1	1	100778	1	7 A St	
2	1	100778	1	8 A St	
3	1	100778	1	9 A St	
4	1	100778	1	10 A St	
...	...	...	...	...	
95	56	175978	1	82 A St	
96	57	175978	2	80 A St 5	
97	58	175978	2	80 A St 6	
98	59	180953	1	84 A St	
99	59	180953	1	85 A St	

100 rows × 30 columns

In [5]:
```
book.head()
```

Out[5]:

	6a. street #	6b. street name	6c. street suffix	6e. zip	7. undergraduate (u) or graduate (g)	8. full-time (ft) or part-time (pt)	9. 5 or more undergrads/unit (y/n)	univers
0	10	Higgins	ST	2134	U	FT	NaN	Emman Colle
1	10	Higgins	ST	2134	U	FT	NaN	Emman Colle
2	1189	Commonwealth	AVE	2134	U	FT	NaN	Emman Colle
3	12	Glenville	AVE	2134	U	FT	NaN	Emman Colle
4	12	Glenville	AVE	2134	U	FT	NaN	Emman Colle

In [8]:
```
book['real_address'] = (
    book['6a. street #'].astype(str)  # Convert street number to string
    + book['6b. street name'].fillna('').astype(str)  # Ensure street name i
    + book['6c. street suffix'].fillna('').astype(str)  # Ensure street suff
)

# Normalize by removing spaces, full stops, and converting to lowercase
book['real_address'] = (
    book['real_address']
    .str.replace(r'\s+', '', regex=True)  # Remove all spaces
    .str.replace(r'\.$', '', regex=True)  # Remove trailing full stops
    .str.lower()  # Convert to lowercase
)

print(book[['6a. street #', '6b. street name', '6c. street suffix', 'real_ad
```

```
  6a. street # 6b. street name 6c. street suffix         real_address
0           10         Higgins               ST          10higginsst
1           10         Higgins               ST          10higginsst
2         1189    Commonwealth              AVE  1189commonwealthave
3           12       Glenville              AVE       12glenvilleave
4           12       Glenville              AVE       12glenvilleave
5           12        Saunders               ST         12saundersst
6         1251    Commonwealth              AVE  1251commonwealthave
7           17        Highgate               ST         17highgatest
8           28          Linden               ST           28lindenst
9           28           Quint              AVE           28quintave
```

In [9]:
```
book['real_address'] = book['real_address'].astype(str).str.strip().str.lowe
# Create actual_address by concatenating street_number and full_street_name
# Ensure we include the actual street name (STREET_BODY)
expanded_lookup_df['actual_address'] = (
```

```
    expanded_lookup_df['STREET_NUMBER'].astype(str)  # Convert street number
    + expanded_lookup_df['STREET_PREFIX'].fillna('').str.replace(r'\s+', '',
    + expanded_lookup_df['STREET_BODY'].fillna('').str.replace(r'\s+', '', r
    + expanded_lookup_df['STREET_FULL_SUFFIX'].fillna('').str.replace(r'\s+'
).str.lower()  # Convert everything to lowercase
expanded_lookup_df['actual2_address'] = (
    expanded_lookup_df['STREET_NUMBER'].astype(str)  # Convert street number
    + expanded_lookup_df['STREET_PREFIX'].fillna('').str.replace(r'\s+', '',
    + expanded_lookup_df['STREET_BODY'].fillna('').str.replace(r'\s+', '', r
    + expanded_lookup_df['STREET_SUFFIX_ABBR'].fillna('').str.replace(r'\s+'
).str.lower()  # Convert everything to lowercase
expanded_lookup_df.head(100)
```

Out[9]:

	SAM_ADDRESS_ID	BUILDING_ID	RELATIONSHIP_TYPE	FULL_ADDRESS	STREET_
0	1	100778	1	6 A St	
1	1	100778	1	7 A St	
2	1	100778	1	8 A St	
3	1	100778	1	9 A St	
4	1	100778	1	10 A St	
...	...	...	...	...	
95	56	175978	1	82 A St	
96	57	175978	2	80 A St 5	
97	58	175978	2	80 A St 6	
98	59	180953	1	84 A St	
99	59	180953	1	85 A St	

100 rows × 31 columns

In [11]:
```
address_to_ward = expanded_lookup_df.set_index('actual_address')['WARD'].to_
address2_to_ward = expanded_lookup_df.set_index('actual2_address')['WARD'].t
book['real_address'] = book['real_address'].str.rstrip('.')
# Fast mapping using dictionary lookup (O(1) time complexity per lookup)
print(list(address_to_ward.keys())[:20])  # Print first 20 keys
print("Expected Key:", book['real_address'].iloc[0])  # Print first address
```

```
['6astreet', '7astreet', '8astreet', '9astreet', '10astreet', '15astreet',
'172astreet', '173astreet', '174astreet', '176astreet', '177astreet', '178as
treet', '21astreet', '232astreet', '249astreet', '250astreet', '251astreet',
'252astreet', '253astreet', '254astreet']
Expected Key: 10higginsst
```

In [12]:
```
book['ward'] = book['real_address'].map(address_to_ward)
# Second mapping: Only update where ward is still NaN
book['ward'] = book['ward'].fillna(book['real_address'].map(address2_to_ward
book.head()
```

Out[12]:

	6a. street #	6b. street name	6c. street suffix	6e. zip	7. undergraduate (u) or graduate (g)	8. full-time (ft) or part-time (pt)	9. 5 or more undergrads/unit (y/n)	univers
0	10	Higgins	ST	2134	U	FT	NaN	Emman Colle
1	10	Higgins	ST	2134	U	FT	NaN	Emman Colle
2	1189	Commonwealth	AVE	2134	U	FT	NaN	Emman Colle
3	12	Glenville	AVE	2134	U	FT	NaN	Emman Colle
4	12	Glenville	AVE	2134	U	FT	NaN	Emman Colle

In [13]:
```
book.to_csv("book_with_wards.csv", index=False)
expanded_lookup_df.to_csv("expanded_lookup.csv", index=False)
book.head(100)
```

Out[13]:

	6a. street #	6b. street name	6c. street suffix	6e. zip	7. undergraduate (u) or graduate (g)	8. full-time (ft) or part-time (pt)	9. 5 or more undergrads/unit (y/n)	unive
0	10	Higgins	ST	2134	U	FT	NaN	Emma Co
1	10	Higgins	ST	2134	U	FT	NaN	Emma Co
2	1189	Commonwealth	AVE	2134	U	FT	NaN	Emma Co
3	12	Glenville	AVE	2134	U	FT	NaN	Emma Co
4	12	Glenville	AVE	2134	U	FT	NaN	Emma Co
...	...	...	...	...	...	...	...	
95	58	Queensberry	ST	2215	U	FT	NaN	Emma Co
96	58	Queensbury	ST	2215	U	FT	NaN	Emma Co
97	660	Washington	ST	2111	U	FT	NaN	Emma Co
98	660	Washington	ST	2111	U	FT	NaN	Emma Co
99	69	Park	DR	2215	U	FT	NaN	Emma Co

100 rows × 14 columns

Further Processing

In [14]:
```
# Drop rows where 'ward' is NaN (missing ward assignments)
book = book.dropna(subset=['ward']).reset_index(drop=True)
book['ward_name'] = book['ward'].astype(int).map(boston_wards)
book.to_csv("book_with_wards.csv", index=False)
expanded_lookup_df.to_csv("expanded_lookup.csv", index=False)
book.head()
```

Out[14]:

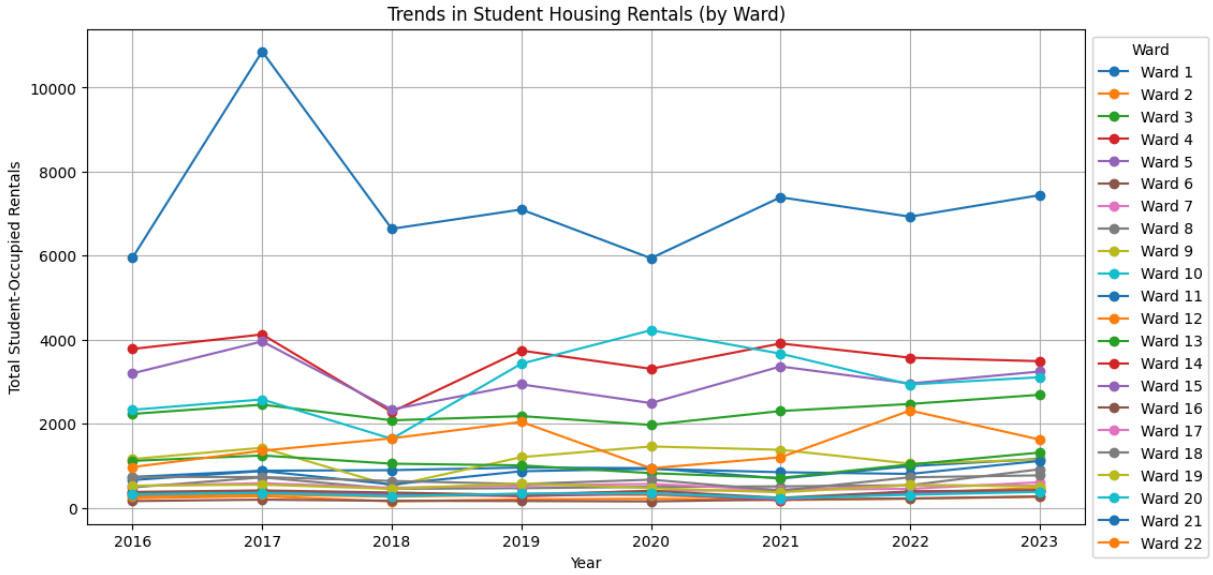
	6a. street #	6b. street name	6c. street suffix	6e. zip	7. undergraduate (u) or graduate (g)	8. full-time (ft) or part-time (pt)	9. 5 or more undergrads/unit (y/n)	univers
0	10	Higgins	ST	2134	U	FT	NaN	Emman Colle
1	10	Higgins	ST	2134	U	FT	NaN	Emman Colle
2	1189	Commonwealth	AVE	2134	U	FT	NaN	Emman Colle
3	12	Glenville	AVE	2134	U	FT	NaN	Emman Colle
4	12	Glenville	AVE	2134	U	FT	NaN	Emman Colle

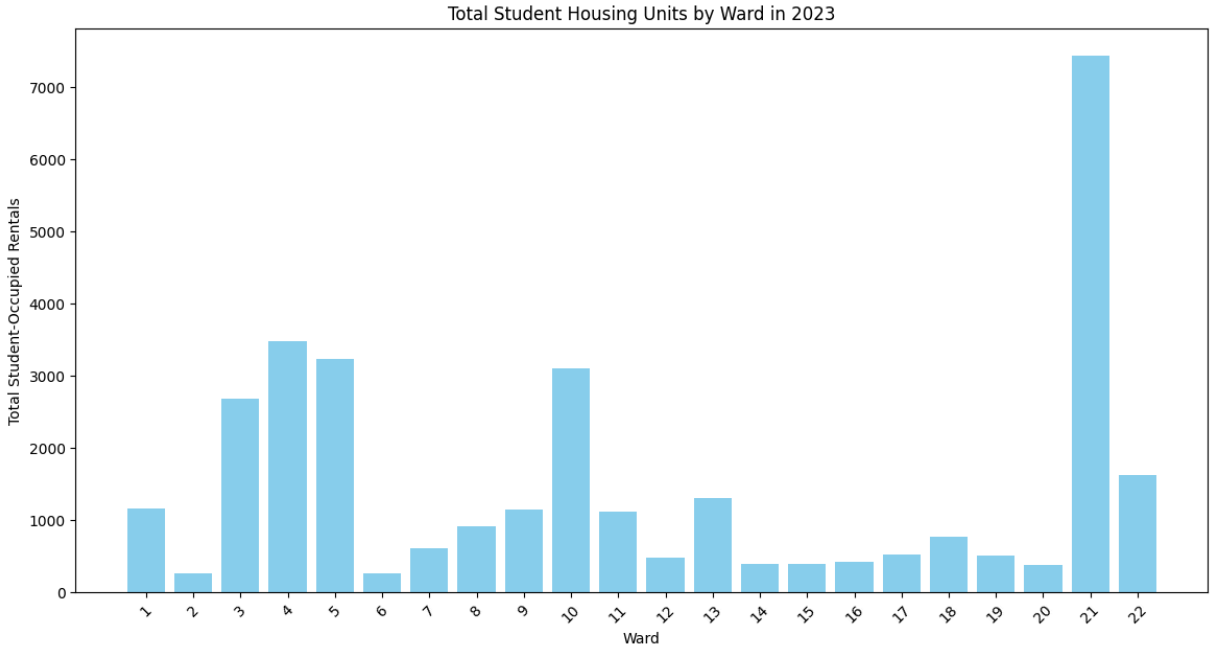
First job is to cluster the people together of the same district: (Given Book is ready:)

In [26]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Ensure ward and year are properly formatted
book['ward'] = book['ward'].astype(int)
book['year'] = book['year'].astype(str).str.extract(r'(\d{4})').astype(int)

# Calculate student housing trends
student_housing_trends = book.groupby(['ward', 'year']).size().reset_index(r

# Display DataFrame
print(student_housing_trends.head())

# 🔹 **Visualization: Line Plot for Student Housing Trends (Total Units)**
plt.figure(figsize=(12, 6))
for ward in student_housing_trends['ward'].unique():
    subset = student_housing_trends[student_housing_trends['ward'] == ward]
    plt.plot(subset['year'], subset['student_units'], marker='o', label=f'Wa
plt.xlabel("Year")
plt.ylabel("Total Student-Occupied Rentals")
plt.title("Trends in Student Housing Rentals (by Ward)")
plt.legend(title="Ward", bbox_to_anchor=(1, 1))
plt.grid(True)
plt.show()

# 🔹 **Visualization: Bar Chart for Latest Year Student Housing by Ward**
plt.figure(figsize=(14, 7))
latest_year = student_housing_trends['year'].max()
```

```
latest_data = student_housing_trends[student_housing_trends['year'] == lates
plt.bar(latest_data['ward'].astype(str), latest_data['student_units'], color

plt.xlabel("Ward")
plt.ylabel("Total Student-Occupied Rentals")
plt.title(f"Total Student Housing Units by Ward in {latest_year}")
plt.xticks(rotation=45)
plt.show()
```

	ward	year	student_units
0	1	2016	734
1	1	2017	883
2	1	2018	896
3	1	2019	958
4	1	2020	944





In [ ]:
```
# Normalize full_address and actual_address: lowercase + remove all spaces
```

We are going to first start by converting our address into