FeyNN Labs: Project 2

Market Segment Analysis of Real Estate Market in India

Contributors: Pratyush Kumar Patnaik, Muskan Bharti, Ashish Anton Abraham, Mohd Saif Ali
GitHub Link: https://github.com/Pratyush-exe/FeyNN_Labs_proj_2

Problem Statement

Task is to analyse the Real Estate Market in India using Segmentation analysis and come up with a feasible strategy to enter the market, targeting the segments most likely to use their product in terms of Geographic, Demographic, Psychographic, Behavioral.

In this report we analyse the Real Estate Market in India using segments such as location, price, RERA verification, price, availability of water and electricity, furnishing, rating based on neighbourhood, schools, safety, cleanliness, locality and much more.

Data Collection

Data was scraped from the website https://www.magicbricks.com/ . This website was used as each house listing came with lots of features that are useful for analysing the Real Estate Market.

Link for scraping script:
https://github.com/Pratyush-exe/FeyNN_Labs_proj_2/tree/main/web_scraping_code

Data was scraped using BeautifulSoup and selenium, Python. First of all the Wikipedia's website https://en.wikipedia.org/wiki/List_of_cities_in_India_by_population was scraped to get the list of Indian cities. Each city from the list was searched in the magicbricks website automatically and the links for each apartment listing was collected. These links were opened and scraped automatically using selenium and the final data was transferred into a csv file.

Raw data generated:
https://github.com/Pratyush-exe/FeyNN_Labs_proj_2/blob/main/raw_magicBricks.csv

Each column explained below:

1. 'city' and 'developer' tells which city and developer the project belongs to
2. 'rera-id' tells if the apartment is rera verified and also provides it's ID
3. 'price' in RS, is the price of each house
4. 'water-availability', 'status-of-electricity', 'lift' and 'furnishing' tells us about water, electricity, lift and furniture's availability conditions respectively
5. 'bedrooms' and 'bathrooms' tells about the number of bedrooms and bathrooms available
6. 'status' tells about construction conditions of the house

7. ‘configuration’ tells about available BHK’s in the apartment
8. ‘recommended-for’ column tells about the category of people the house is recommended for
9. ‘neighborhood’, ‘road’, ‘safety’, ‘cleanliness’, ‘public-transport’, ‘parking’, ‘connectivity’, ‘traffic’, ‘school’, ‘restaurant’, ‘hospital’ and ‘market’ columns tells about the rating of all these factors from 5
10. ‘locality-rating’ is the overall rating of the locality from 5

Data Preprocessing

Steps taken to preprocess the raw data scraped:

1. ordinal encoded 'status'
2. created column 'verified_id' and deleted 'rera-id'
3. deleted column 'name'
4. cleaned column 'price'
5. created column 'capacity'(no.of people) using data from 'bedroom', 'bathroom' and 'recommended-for'
6. deleted 'lift' as it had only waste values
7. chose 2 major values from water and electricity; encoded them as 1 and 2 and set others to 0.
8. cleaned 'bedroom' and ‘furnishing’
9. deleted ‘tower-and-unit-details’ and created columns ‘towers’ and ‘units’ that stores data separately
10. deleted ‘recommended-for’ and created 5 new columns 'Retirees', 'Family', 'Couple', 'SingleProfessionals' and 'Students' each coding with 1 or 0, meaning recommended for or not recommended for respectively

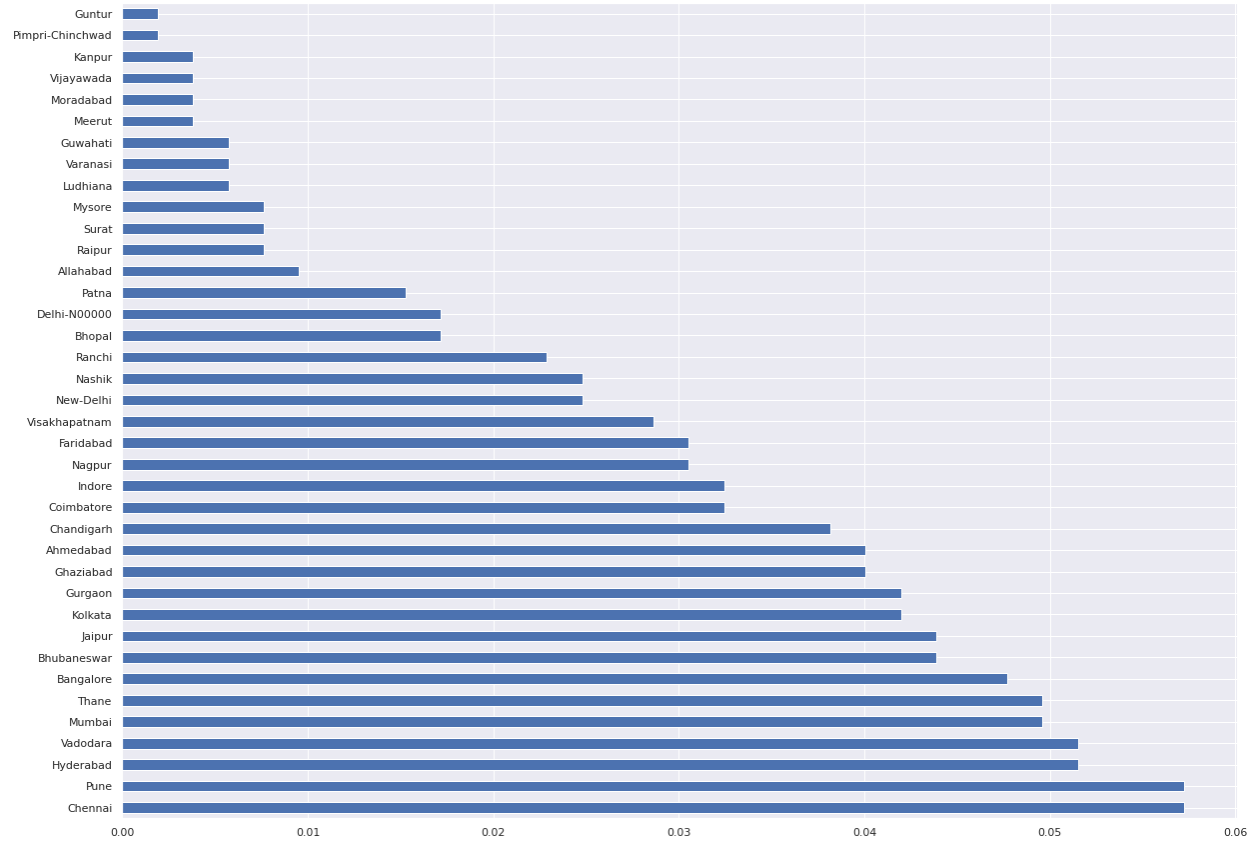
Final pre-processed data link:
https://github.com/Pratyush-exe/FeyNN_Labs_proj_2/blob/main/processed_magicBricks.csv

Exploratory Data Analysis

An Exploratory Data Analysis, or EDA is a thorough examination meant to uncover the underlying structure of a data set and is important for a company because it exposes trends, patterns, and relationships that are not readily apparent.

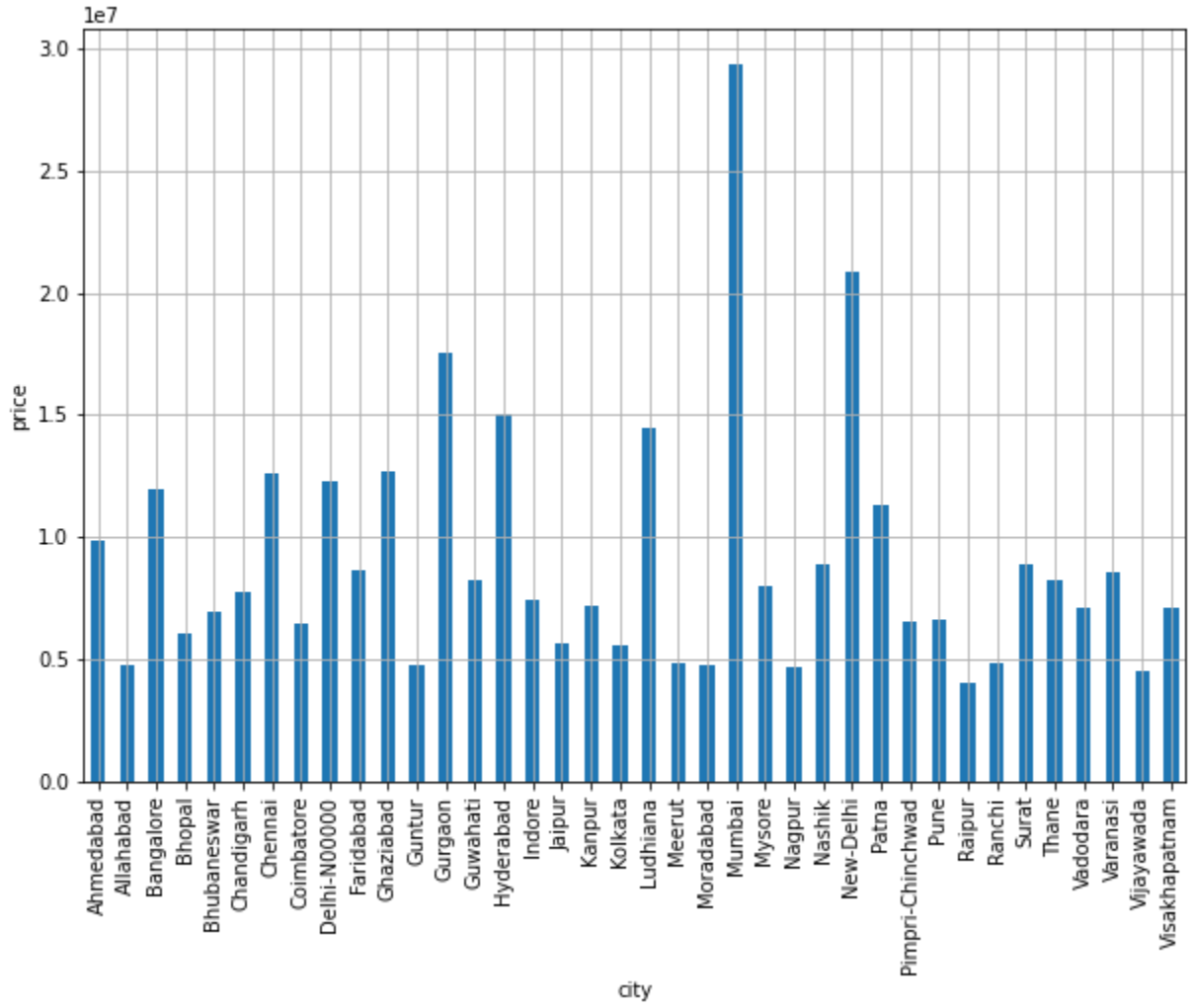
We analysed our dataset using *univariate* (analyze data over a single variable/column from a dataset), *bivariate* (analyze data by taking two variables/columns into consideration from a dataset) and *multivariate* (analyze data by taking more than two variables/columns into consideration from a dataset) analysis.

```
data.city.value_counts(normalize=True).plot.barh()
```



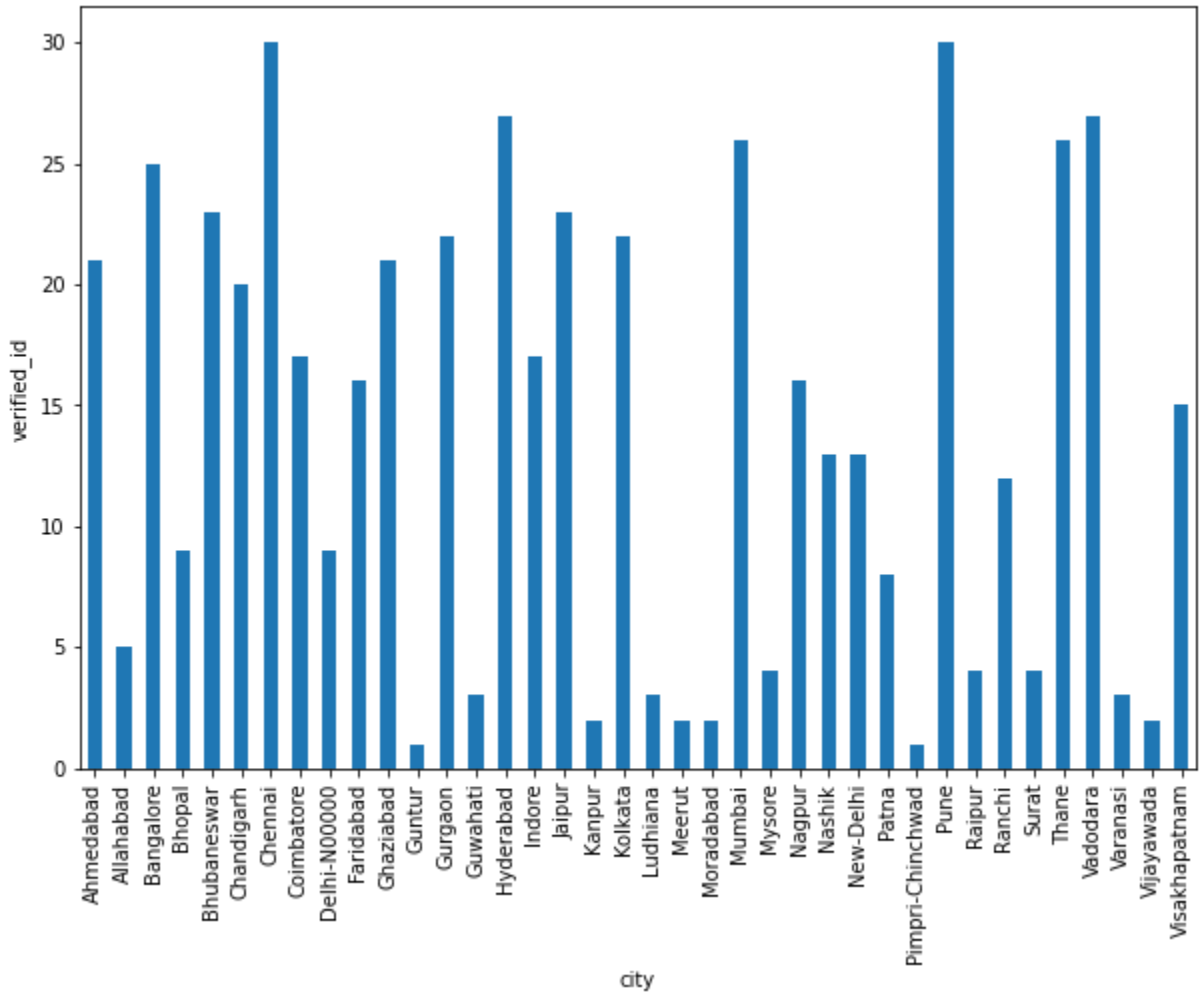
This Bar graph shows the diversity of the data geographically. We can see that we have the maximum number of data of cities *Chennai* and *Pune*; and minimum number of data for *Guntur* and *Pimpri-Chinchwad*. There are a total of 525 rows of data distributed among the cities shown in the graph.

```
upper_limit = df['price'].mean() + 3*df['price'].std()
lower_limit = df['price'].mean() - 3*df['price'].std()
df['price'] = np.where(
    df['price']>upper_limit, upper_limit,
    np.where(
        df['price']<lower_limit,
        lower_limit,
        df['price']))
df.groupby('city')['price'].mean().plot.bar()
plt.ylabel("price")
```



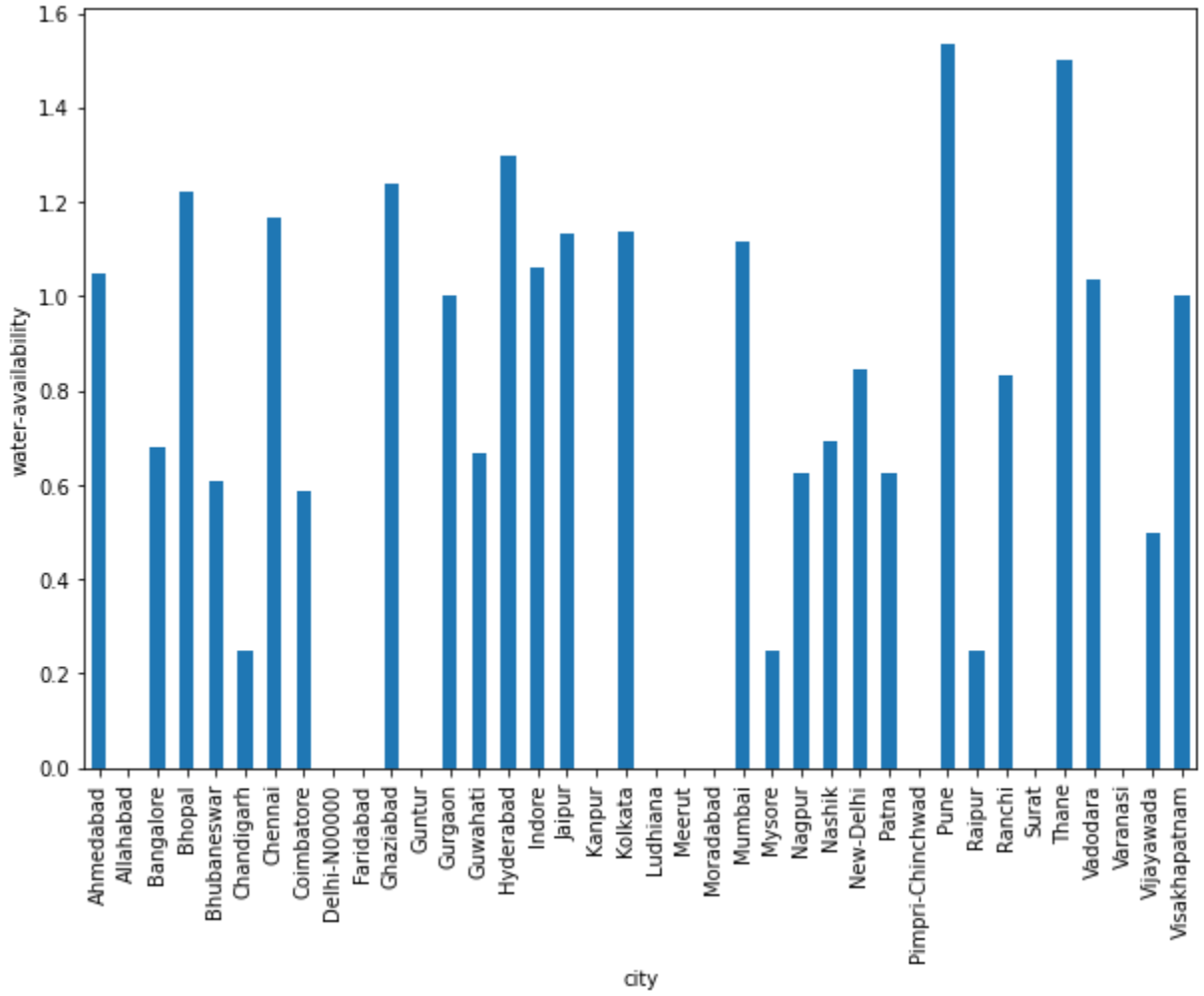
This Bar Chart shows the average or mean price of an apartment in each city from the dataset after removing meaningless outliers. Quick look at the graphs tells us that *Mumbai* is the most expensive city for buying or renting an apartment and it is followed by *New-Delhi*, *Gurgaon* or *Gurugram*, and *Hyderabad*.

```
plt.figure(figsize=(10,7))
df.groupby('city')['verified_id'].count().plot.bar()
plt.ylabel("verified_id")
plt.show()
```



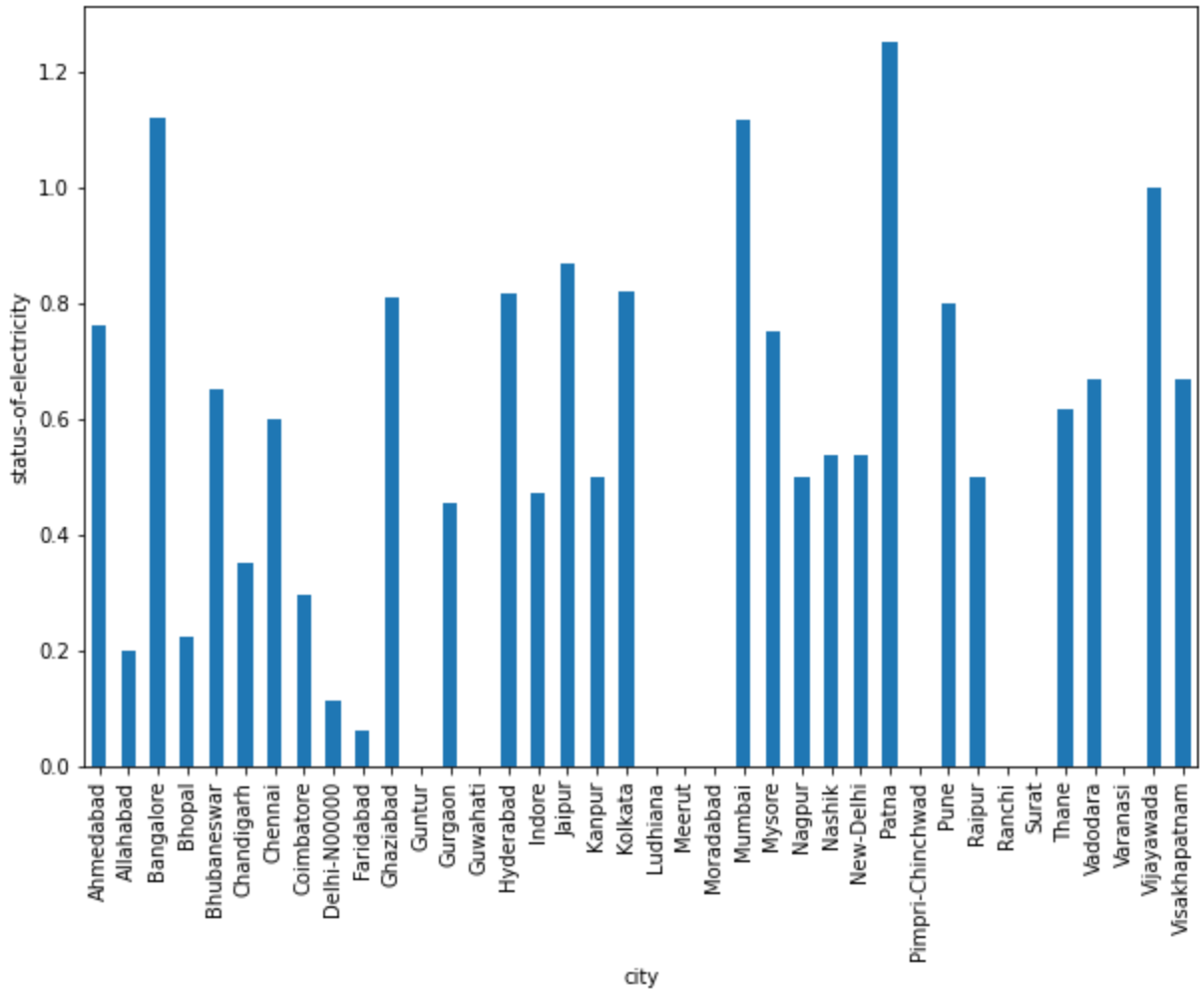
The above is the count plot of ‘verified-id’ and cities. This tells how many apartments in a particular city are verified by RERA. RERA stands for Real Estate Regulatory Authority came into existence as per the Real Estate (Regulation and Development) Act, 2016 which aims to protect the home purchasers and also boosts the real estate investments. RERA has a number of benefits for the buyer, the promoter, and the real estate agent. These include Standardisation of carpet area, Reducing the risk of insolvency of the builder, Advance payment, rights to the buyer in case of any defects, interest to be paid in case of default, buyer’s rights in case of false promises, Right to information, and Grievance Redressal.

```
plt.figure(figsize=(10,7))
df.groupby('city')['water-availability'].mean().plot.bar()
plt.ylabel("water-availability")
plt.show()
```



The mean value for water availability is higher for cities Pune, Thane, Hyderabad and Bhopal. This means these cities either have less 0 value, i.e. unavailable or garbage value, or more of 2 i.e. 24 hours water availability.

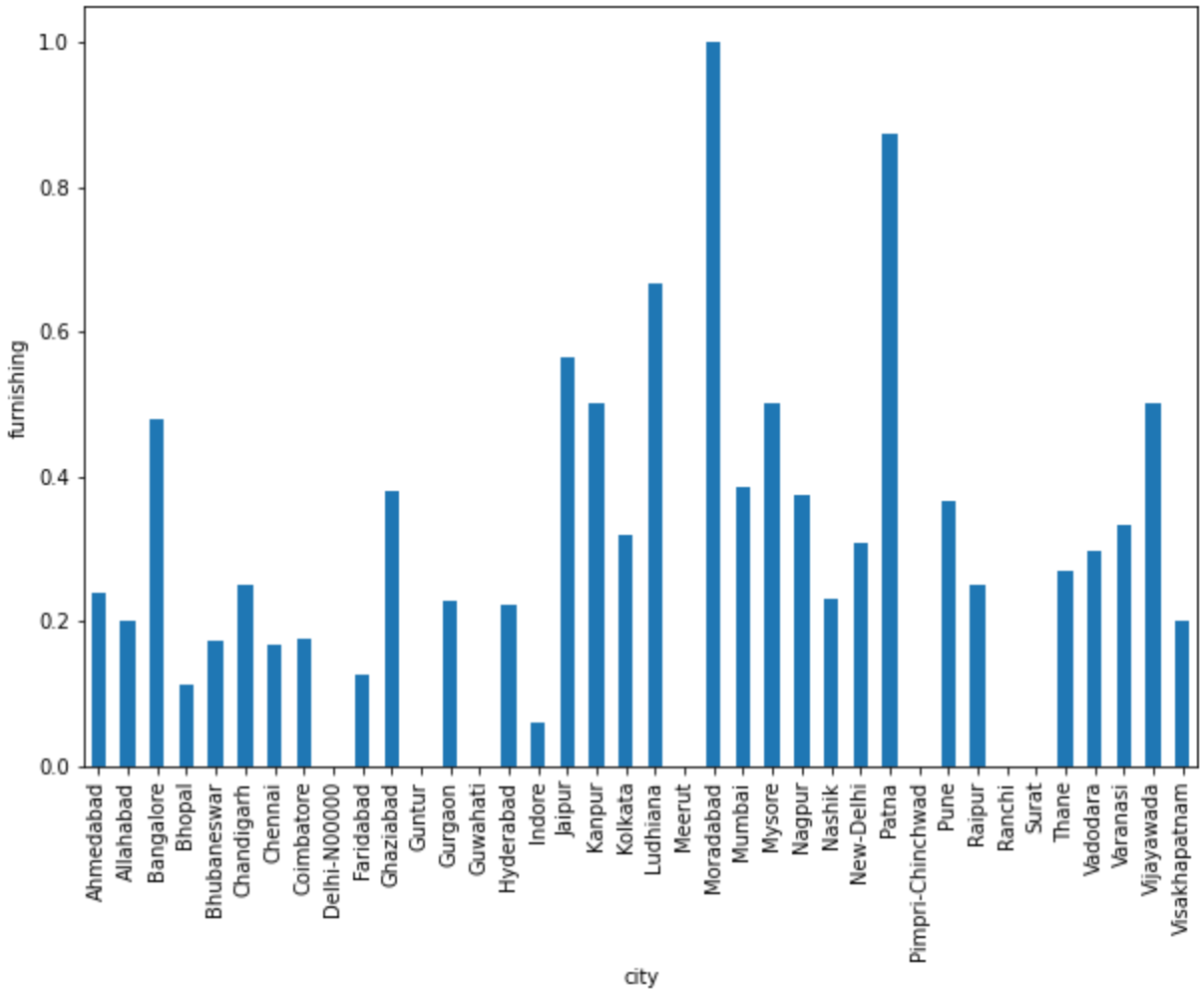
```
plt.figure(figsize=(10,7))
df.groupby('city')['status-of-electricity'].mean().plot.bar()
plt.ylabel("status-of-electricity")
plt.show()
```



In this Bar Plot of cities with mean value of status of electricity having values 0, 1, and 2, we can see that cities Patna, Bangalore, Mumbai and Vijayawada have higher mean values this implies that these cities either have less 0 value (No data or garbage value) and more of 2 and 1 values (No or Rare power cut and 24 hours available respectively).

```
plt.figure(figsize=(10,7))
df.groupby('city')['furnishing'].mean().plot.bar()
plt.ylabel("furnishing")
plt.show()
```

In the graph generated below from the above code we can see that Moradabad and Patna have the highest value of means. This means they have many values of 2 i.e. most of the data of apartments for these cities are Semi furnished.

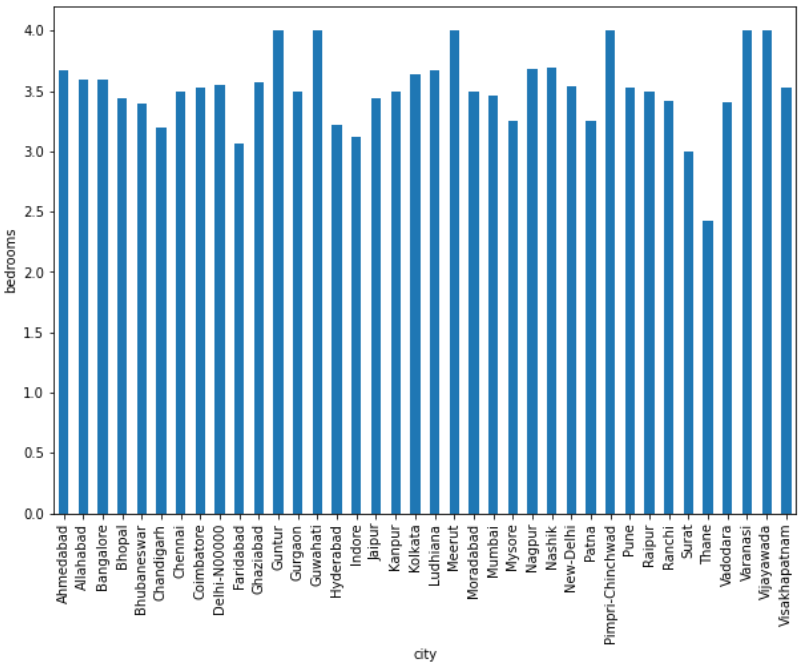


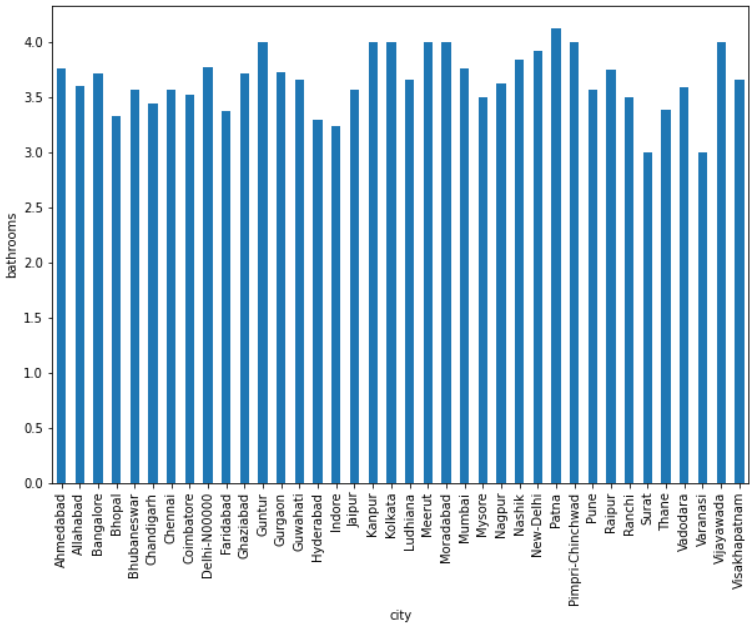

```
plt.figure(figsize=(10,7))
df.groupby('city')['bathrooms'].mean().plot.bar()
plt.ylabel("bathrooms")
plt.show()
```

```
plt.figure(figsize=(10,7))
df.groupby('city')['bedrooms'].mean().plot.bar()
plt.ylabel("bedrooms")
plt.show()
```

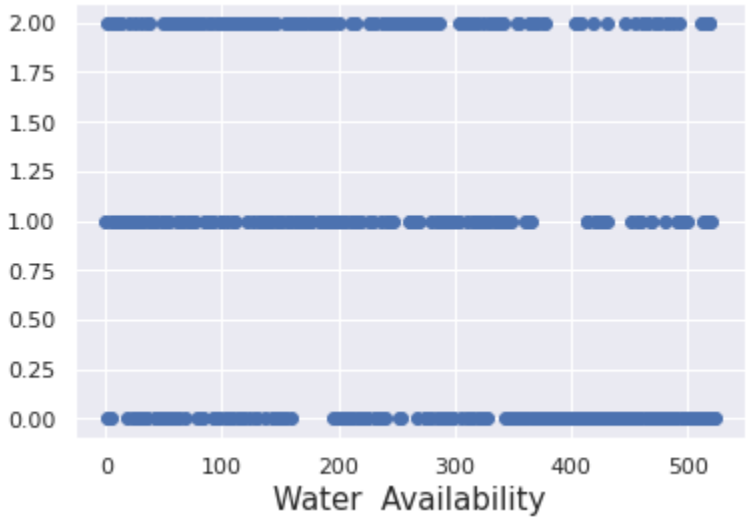
Plots for Mean values of number of bedrooms and bathrooms for cities:

We can see that the data for bedrooms and bathrooms for the apartments is almost evenly distributed within cities and the mean values of all the cities are almost the same.





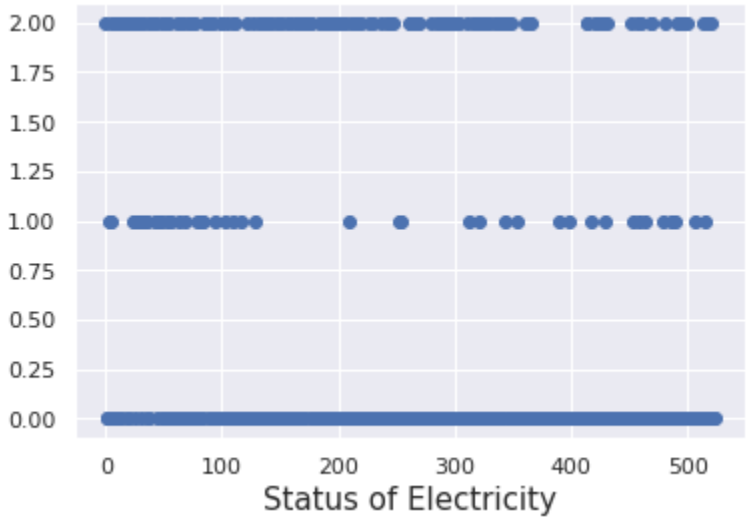

```
plt.scatter (data.index, data ['water-availability'])
plt.xlabel('Water  Availability', fontsize= 15)
plt.show()
```



This Scatter Plot shows water availability conditions for the apartments. The numbers 0, 1, and 2 are encoded as follows:

a. 1 and 2 means 24 hours available
b. 0 means unfixed or unavailable

```
plt.scatter (data.index, data ['status-of-electricity'])
plt.xlabel('Status of Electricity', fontsize= 15)
plt.show()
```

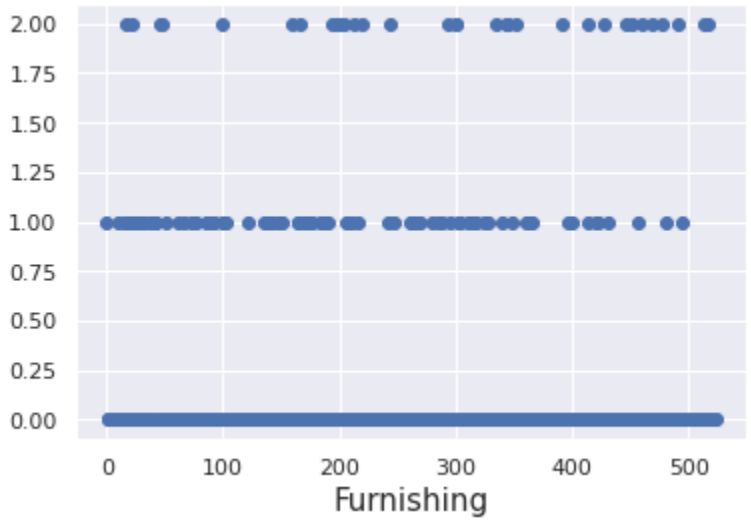


This Scatter Plot shows electricity availability conditions for the apartments. The numbers 0, 1, and 2 are encoded as follows:

a. 0, data unavailable
b. 1, 24 Hours available
c. No or Rare Power Cut

We see that the data points are populated for the value of 0, this means the data for status of electricity either contains garbage value or no value for the majority of apartments.

```
plt.scatter (data.index, data ['furnishing'])
plt.xlabel('Furnishing', fontsize= 15)
plt.show()
```

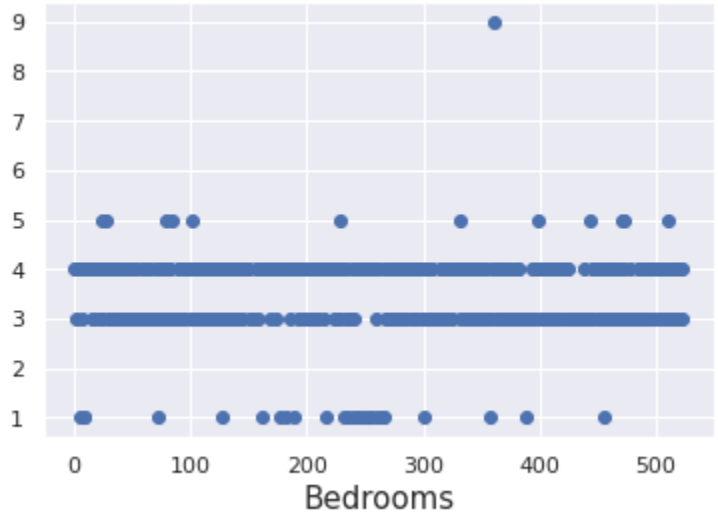


This Scatter Plot shows furniture availability conditions for the apartments. The numbers 0, 1, and 2 are encoded as follows:

a. 0, data unavailable or other information, or garbage value
b. 1, Unfurnished
c. 2, Semi-furnished

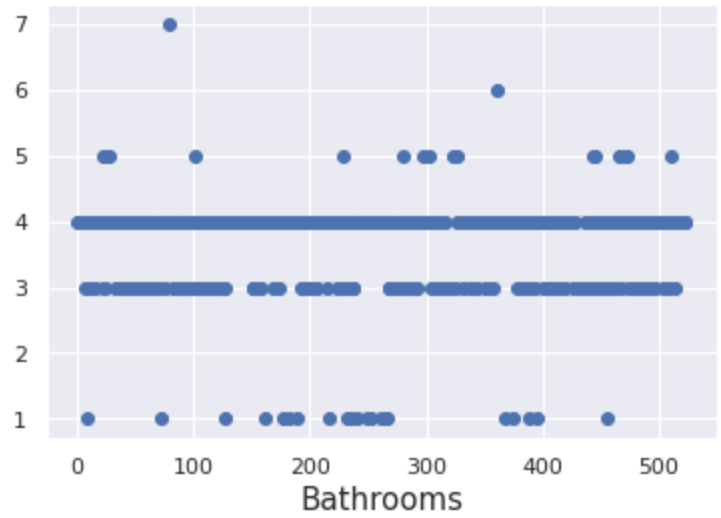
Here also we can see that much of the data is garbage, no value or unavailable information. Apart from 0, we can see that data contains apartments with no furnishing than with semi-furnishing.

```
plt.scatter (data.index, data ['bedrooms'])
plt.xlabel('Bedrooms', fontsize= 15)
plt.show()
```



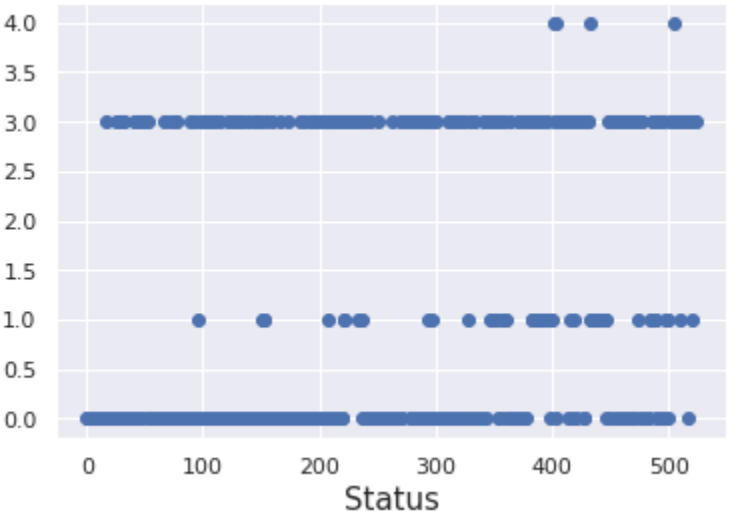
This Scatter Plot is a plot between the column ‘bedrooms’ and the count of each unique type. ‘bedroom’ column contains the number of bedrooms in the apartment. We can see that the data points lie in 1, 3, 4, 5 and 9. From the graph we can also see that in the dataset there are many data points for 3, 4 and some for 5, 1 but a single point or an outlier at 9. This tells us that the large number of data are for 3 and 4 bedroom apartments.

```
plt.scatter (data.index, data ['bathrooms'])
plt.xlabel('Bathrooms', fontsize= 15)
plt.show()
```



This Scatter Plot is between the unique values of the ‘bathrooms’ column and their value counts. In the graph we can see that a large number of points lie in 4 and 3 this means that most of our dataset has apartments of 3 or 4 bathrooms, some of 1 and 5 bathrooms and very less (outliers) 6 and 9 bathrooms.

```
plt.scatter (data.index, data ['status'])
plt.xlabel('Status', fontsize= 15)
plt.show()
```



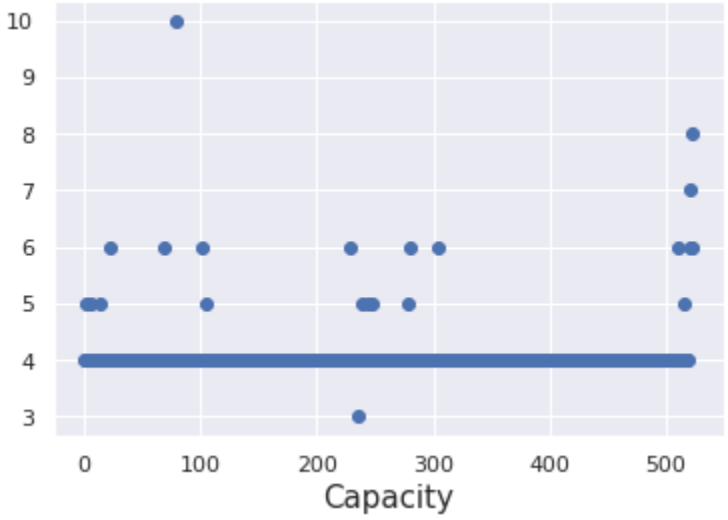
This Plot is again between the unique values of the ‘status’ column and their value counts. The status column consists of values 0, 1, 2, 3, 4 which decode to:

a. 'Under Construction': 0
b. 'New Property': 1
c. 'Ready to Move': 3

d. 'Resale': 4

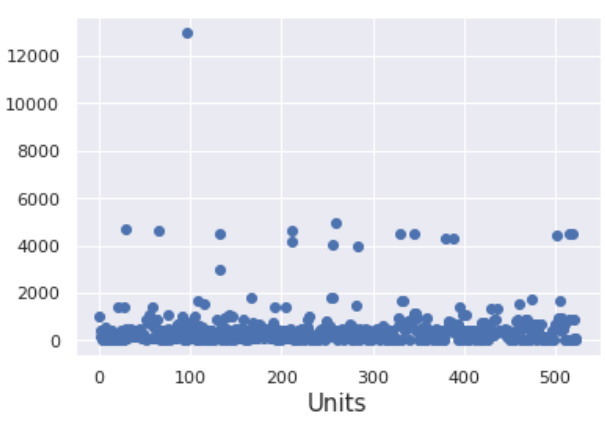
We see there are many points lying within 0 and 3, which means most of the apartments are either ‘Under Construction’ or ‘Ready to Move’. Also there are very few data points for the apartment that are under ‘Resale’.

```
plt.scatter (data.index, data ['capacity'])
plt.xlabel('Capacity', fontsize= 15)
plt.show()
```

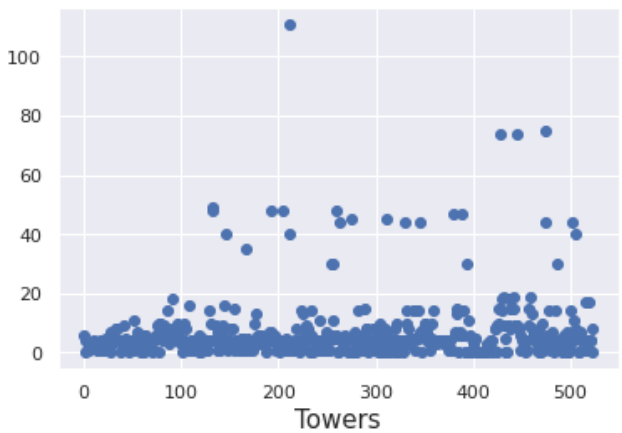


This Scatter Plot is between unique values of ‘capacity’ column and the value of counts of the same. As discussed earlier, ‘capacity’ column tells us about the number of persons that can live in the apartment. We see that the data points are highly populated for ‘4’. This means that a large chunk of apartments in our dataset is for 4 people living houses. We can also see an outlier at 10.

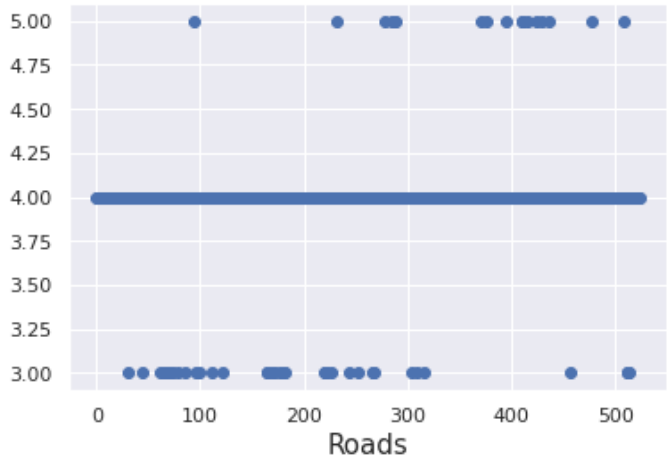
```
plt.scatter (data.index, data ['towers'])
plt.xlabel('Towers', fontsize= 15)
plt.show()
plt.scatter (data.index, data ['units'])
plt.xlabel('Units', fontsize= 15)
plt.show()
```

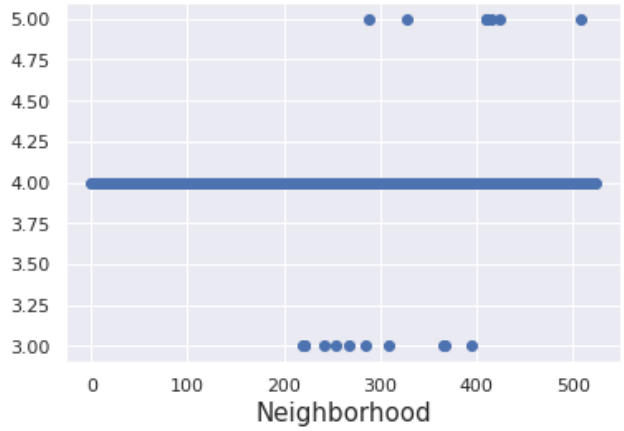


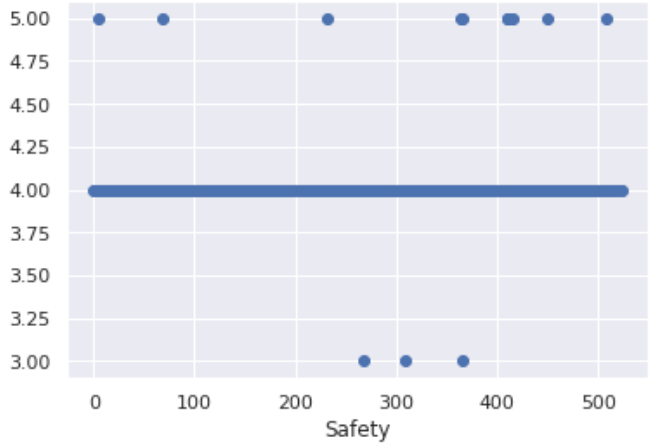
rs

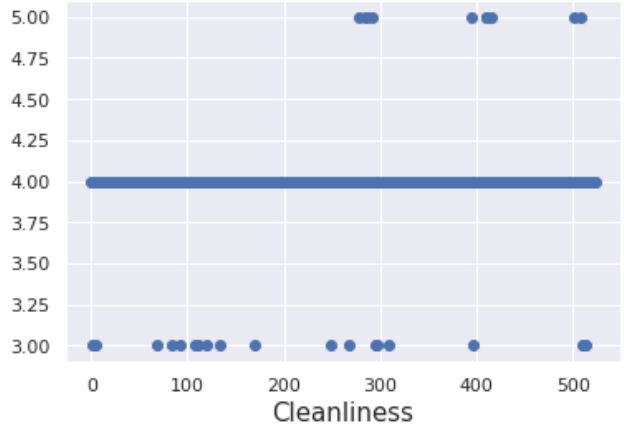


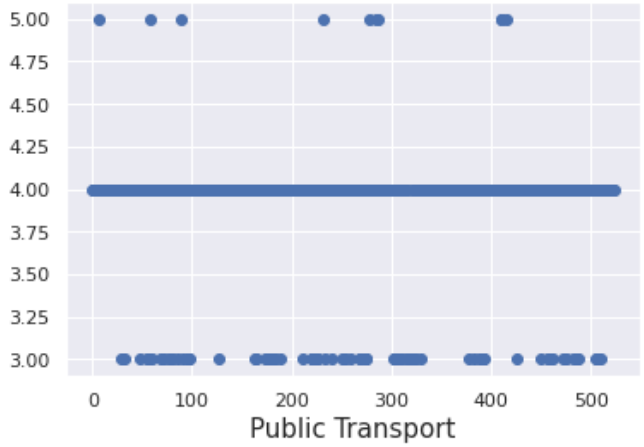
Graphs on rating columns (values are out of 5):

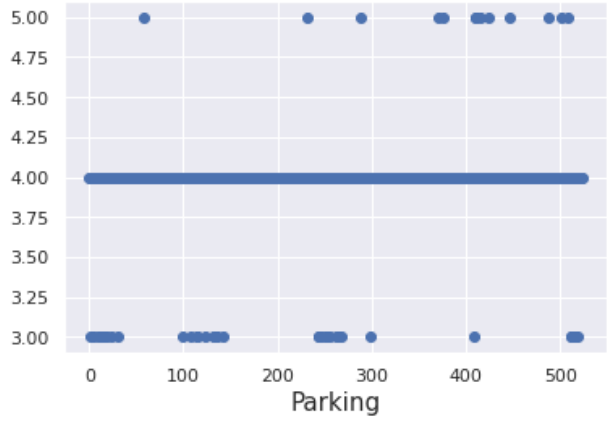


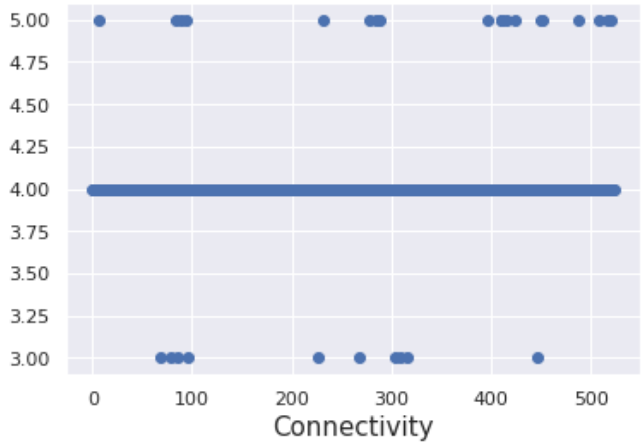


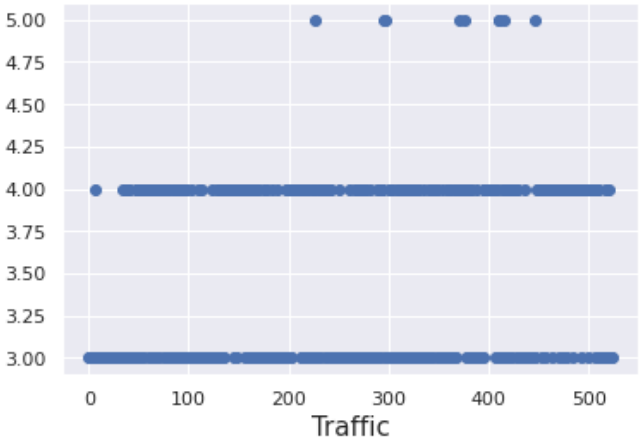


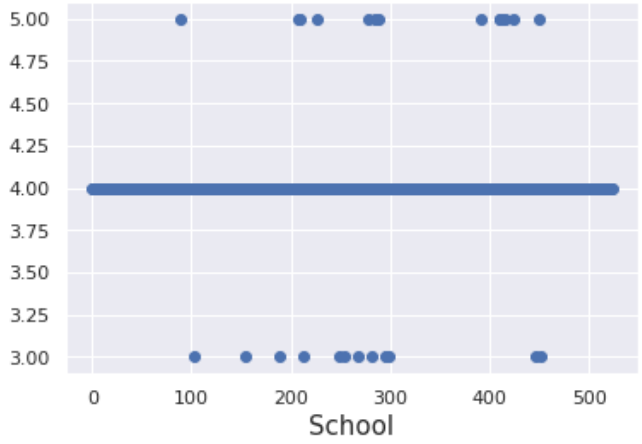


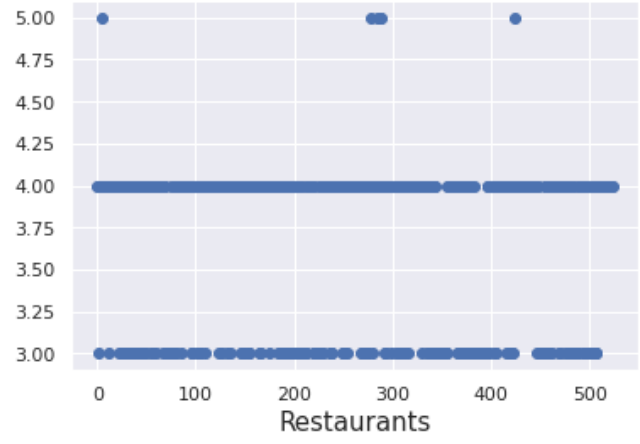


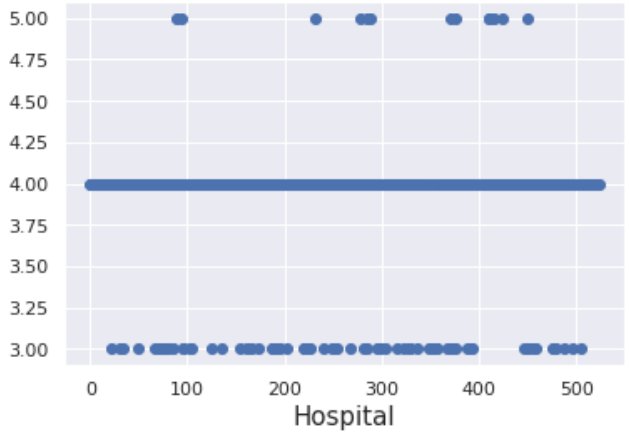


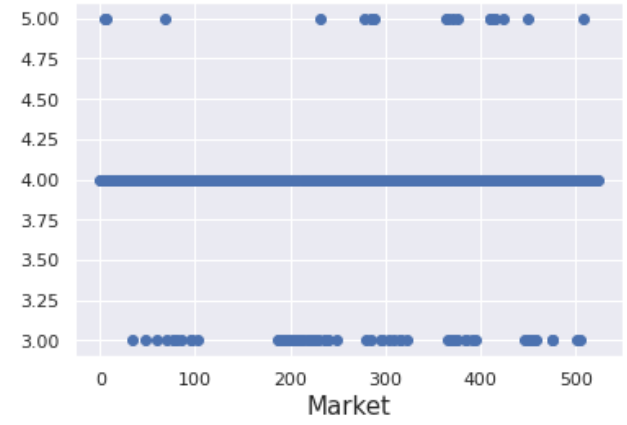








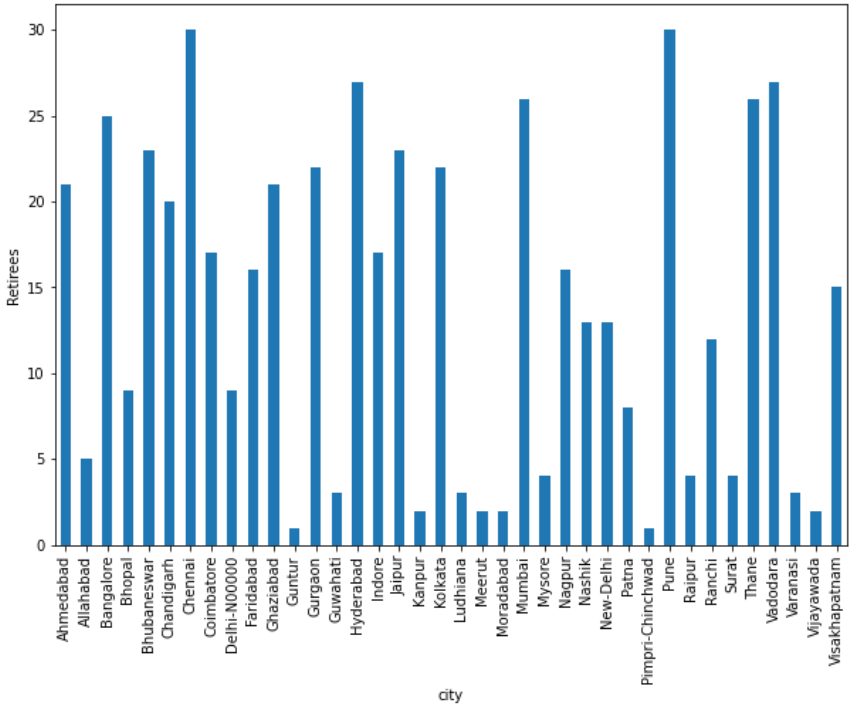




For all these graphs most of the data points lie within 4 and none go lesser than 3, this means the overall rating for the apartments in our dataset for the factors ‘Roads’, ‘Neighbourhood’, ‘Safely’, ‘Cleanliness’, ‘Public Transport’, ‘Parking’, ‘Connectivity’, ‘Traffic’, ‘School’, ‘Restaurants’, ‘Hospitals’ and ‘Market’ are good.

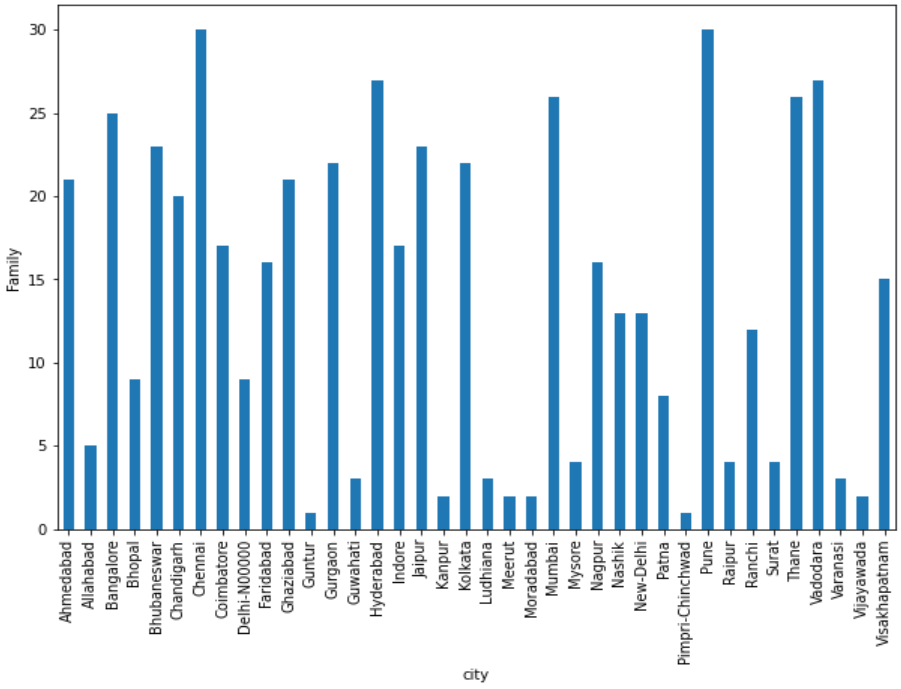
Code and plots for count plot for ‘Retirees’, ‘Family’, ‘Couple’, ‘SingleProfessionals’, ‘Couples’ and cities:

```
plt.figure(figsize=(10,7))
df.groupby('city')['Retirees'].count().plot.bar()
plt.ylabel("Retirees")
plt.show()
```



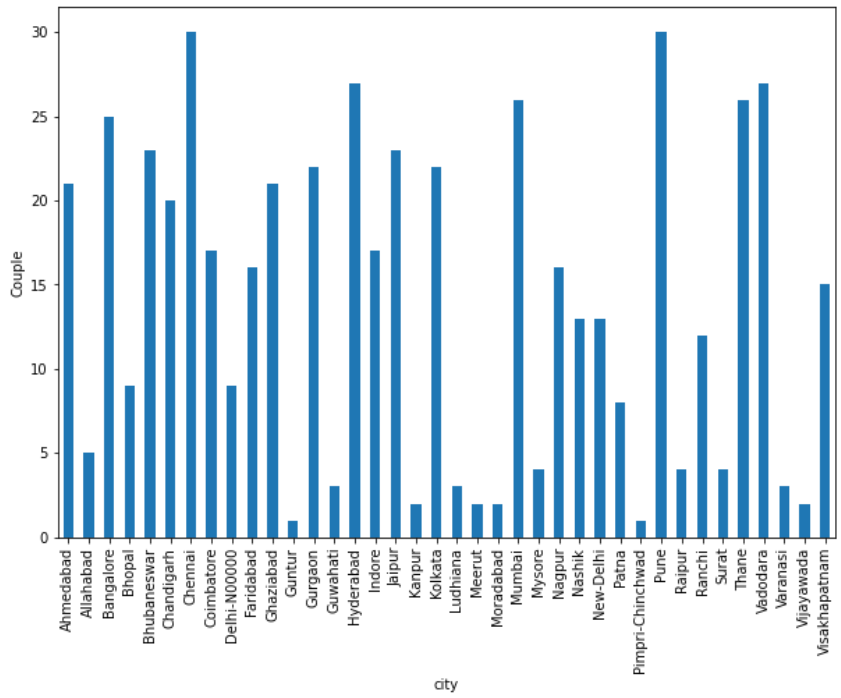
In this Bar Plot we can see that cities like Pune, Chennai, Hyderabad, Mumbai, Thane, and Vadodara have apartments that are recommended to retirees. And cities like Guntur, Ludhiana, Raipur, Kampur are cities that are less recommended to retirees.

```
plt.figure(figsize=(10,7))
df.groupby('city')['Family'].count().plot.bar()
plt.ylabel("Family")
```



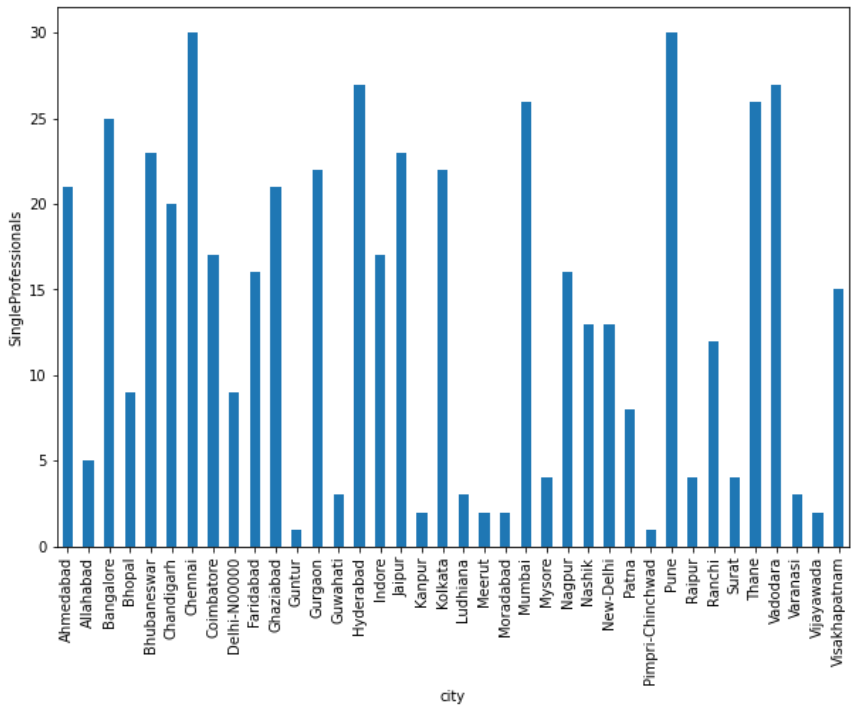
In this Bar plot we see that cities Pune, Chennai, Hyderabad are more recommended to families than to cities like Guntur, Kanpur, Surat.

```
plt.figure(figsize=(10,7))
df.groupby('city')['Couple'].count().plot.bar()
plt.ylabel("Couple")
plt.show()
```



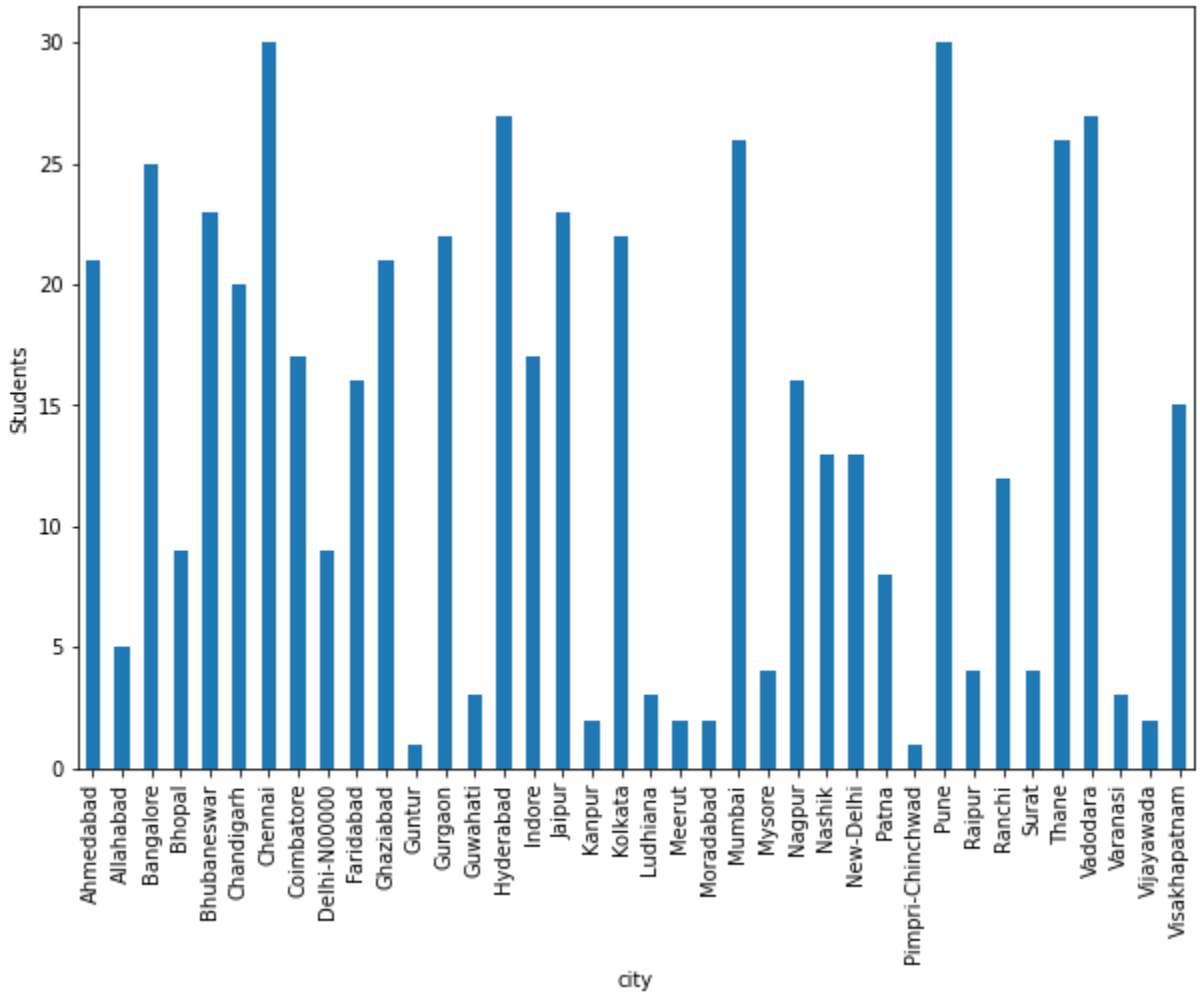
Chennai, Pune, Hyderabad, Thane and Varodera have more apartments that are recommended to couples.

```
plt.figure(figsize=(10,7))
df.groupby('city')['SingleProfessionals'].count().plot.bar()
plt.ylabel("SingleProfessionals")
plt.show()
```



Cities like Chennai, Pune Thane, Vadodara are more recommended to Single Professionals.

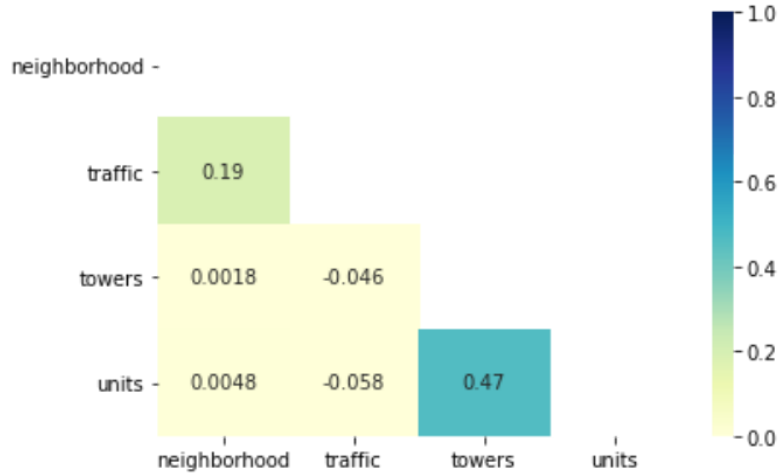
```
plt.figure(figsize=(10,7))
df.groupby('city')['Students'].count().plot.bar()
plt.ylabel("Students")
plt.show()
```



Cities like Chennai, Mumbai, Pune, Vadodara are more recommended to students.

Note: Graphs and results for the plots' Retirees', 'Family', 'Couple', 'SingleProfessionals', 'Couples' may differ due to availability or lack of data. Drawing conclusions based on these are hard and might not be accurate.

```
mask = np.triu(np.ones_like(df[feature1].corr()))
dataplot = sns.heatmap(df[feature1].corr(), vmin=0, vmax=1, cmap="YlGnBu",
annot=True, mask=mask)
plt.show()
```



Correlation between traffic and neighbourhood 0.19 that is related to each other and towers,units have negligible correlation between neighborhood and traffic.

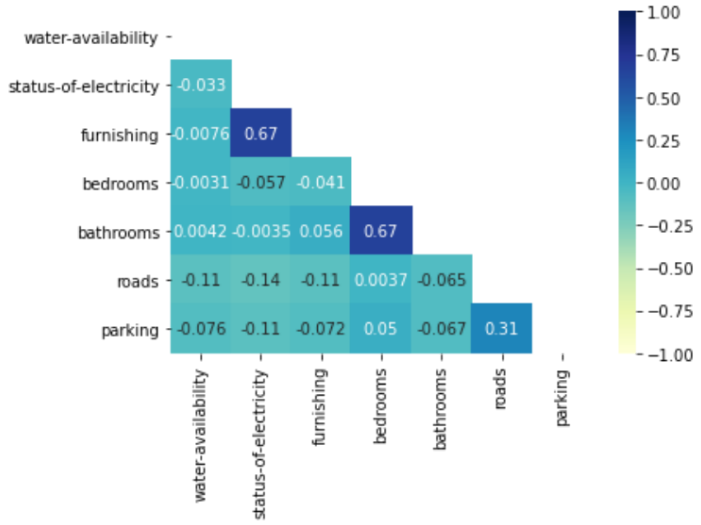
```
mask = np.triu(np.ones_like(df[feature2].corr()))
dataplot = sns.heatmap(df[feature2].corr(), vmin=0, vmax=1, cmap="YlGnBu",
annot=True, mask=mask)
plt.show()
```

This correlation matrix shows that most retirees would like to leave with family. And couple retirees and couple families have a better correlation.

```
mask = np.triu(np.ones_like(df[feature3].corr()))
dataplot = sns.heatmap(df[feature3].corr(), vmin=0, vmax=1, cmap="YlGnBu",
annot=True, mask=mask)
plt.show()
```

Cleanliness and safety have some correlation that means people at that place are good by hygiene and nature. And safety and cleanliness have a very low impact on locality rating.

```
mask = np.triu(np.ones_like(df[feature4].corr()))
dataplot = sns.heatmap(df[feature4].corr(), vmin=-1, vmax=1, map="YlGnBu",
annot=True, mask=mask)
plt.show()
```



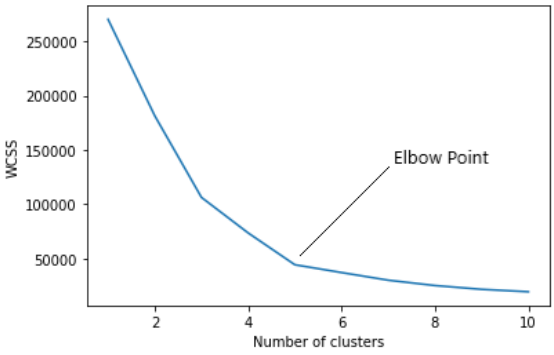
This correlation matrix shows that furnishing and status of electricity are almost dependent on each other and similarly bathrooms and bedrooms. Water availability, electricity and furnishing are almost not correlated with roads and parking.

Segment Extraction

K means is one of the most popular Unsupervised Machine Learning Algorithms Used for Solving Classification Problems. K Means segregates the unlabeled data into various groups, called clusters, based on having similar features, common patterns.

Suppose we have N number of Unlabeled Multivariate Datasets of various features like water-availability, price, city etc. from our dataset. The technique to segregate Datasets into various groups, on the basis of having similar features and characteristics, is called Clustering. The groups being Formed are known as Clusters. Clustering is being used in Unsupervised Learning Algorithms in Machine Learning as it can segregate multivariate data into various groups, without any supervisor, on the basis of a common pattern hidden inside the datasets.

In the Elbow method, we are actually varying the number of clusters (K) from 1 – 10. For each value of K, we are calculating WCSS (Within-Cluster Sum of Square). WCSS is the sum of squared distance between each point and the centroid in a cluster. When we plot the WCSS with the K value, the plot looks like an Elbow.



As the number of clusters increases, the WCSS value will start to decrease. WCSS value is largest when K = 1. When we analyze the graph we can see that the graph will rapidly change at a point and thus creating an elbow shape. From this point, the graph starts to move almost parallel to the X-axis. The K value corresponding to this point is the optimal K value or an optimal number of clusters.

Analysing Market Segments

Geographic Segmentation: Geographic segmentation divides a target market by location so marketers can better serve customers in a particular area. This type of market segmentation is based on the geographic units themselves (countries, states, cities, etc.), but also on various geographic factors, such as climate, cultural preferences, populations, and more. Geographic segmentation involves segmenting your audience based on the region they live or work in. This can be done in any number of ways: grouping customers by the country they live in, or smaller geographical divisions, from region to city, and right down to postal code.

Geographic segmentation might be the simplest form of market segmentation to get your head around, but there are still plenty of ways it can be used that companies never think about. The size of the area you target should change depending on your needs as a business. Generally speaking, the larger the business the bigger the areas you'll be targeting. After all, with a wider potential audience, targeting each postcode individually simply won't be cost-effective.

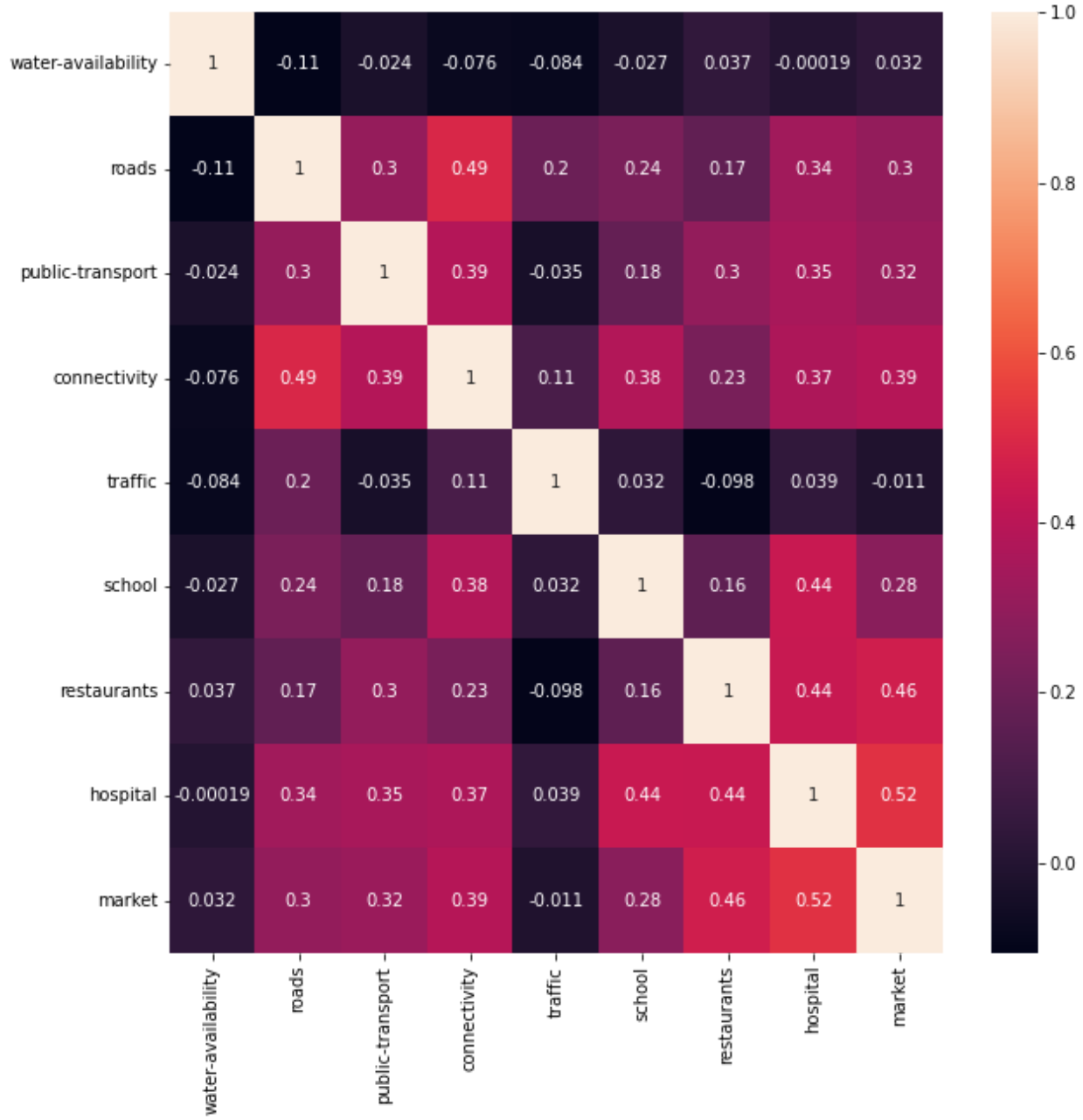
Our dataset contains cities of India and geographical factors that our dataset contains are 'water-availability', 'roads', 'public-transport', 'connectivity', 'traffic', 'school', 'restaurants', 'hospital' and 'market'. Let's try to create a heatmap for correlation matrix for Geographic Segmentation columns.

```
Geographic = ['water-availability', 'roads', 'public-transport',
'connectivity', 'traffic', 'school', 'restaurants', 'hospital', 'market']

x = df[Geographic]

# Normalizing values
for column in x.columns:
    x[column] = x[column]  / x[column].abs().max()

plt.figure(figsize=(15,15))
sns.heatmap(x.corr(), annot=True)
plt.show()
```



Demographic Segmentation: Demographic segmentation is a market segmentation technique where an organization's target market is segmented based on demographic variables such as age, gender, education, income, etc. It helps organizations understand who their customers are so that their needs can be addressed more effectively.

Instead of reaching an entire market, companies can use demographic segmentation to focus their time and resources on those segments that have customers who are most likely to make purchases, and are therefore most valuable to them.

There are several different variables by which demographic segmentation is done:

a. Age: Age is one of the most important variables used within demographic segmentation as consumers' preferences and needs differ significantly based on the age group they fall under. When an organization wants to target young adults or teenagers, digital marketing campaigns may prove to be most effective as they appeal to this age group. However, older adults often prefer traditional marketing methods, such as television and magazine advertisements.

b. Income: Income levels have a significant effect on consumer purchasing decisions. Those with higher-income levels may prefer high-end and luxury products. Conversely, individuals with lower income levels may prefer to get products at the best deal and are likely to choose inexpensive products/services.

c. Religion, Race, and Ethnicity: Racial and ethnic preferences may reflect differences in sentiments. This will affect the kind of marketing campaigns that will appeal to customers. Additionally, religion can also have a significant impact on customer preferences, so it is important to be aware of the religious categorization of your target market.

d. Gender: Individuals may identify with different areas of the gender spectrum, like feminine or masculine, and this will have a significant effect on their preferences and purchasing decisions. By understanding which gender your product or service appeals to, you can tailor your marketing campaigns accordingly to meet the needs of your consumers better.

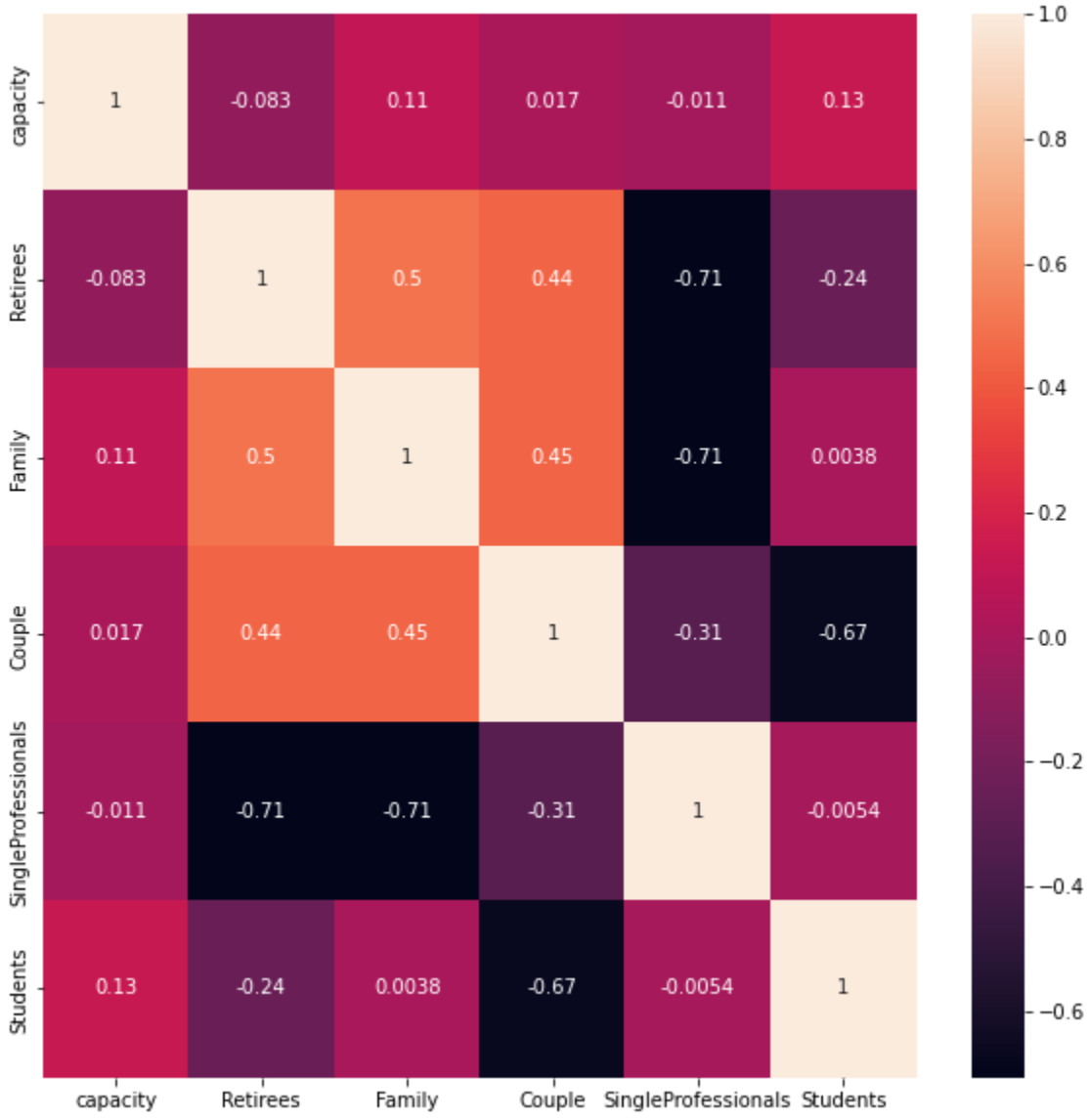
Demographic factors from our dataset contain 'capacity', 'Retirees', 'Family', 'Couple', 'SingleProfessionals' and 'Students' columns. Let's try to create a heatmap for correlation matrix for Demographic Segmentation columns.

```
Demographic = ['capacity', 'Retirees', 'Family', 'Couple',
'SingleProfessionals', 'Students']

x = df[Demographic]

# Normalizing values
for column in x.columns:
    x[column] = x[column]  / x[column].abs().max()
```

```
plt.figure(figsize=(15,15))
sns.heatmap(x.corr(), annot=True)
plt.show()
```



Psychographic Segmentation: Psychographic segmentation is the research methodology used for studying consumers and dividing them into groups using psychological characteristics including personality, lifestyle, social status, activities, interests, opinions, and attitudes.

Psychographic marketing enables you to engage with multiple target audiences in the ways that will make the biggest impact for each one. This approach saves time and

money on approaches that might fall flat and makes it easier to relate to the groups you care about.

We can use psychographics for market segmentation to understand:

a. How consumers really perceive your products and services
b. What consumers really want—and why
c. Gaps or pain points with your current products or services
d. Opportunities for future engagement
e. How to better communicate with your target audience

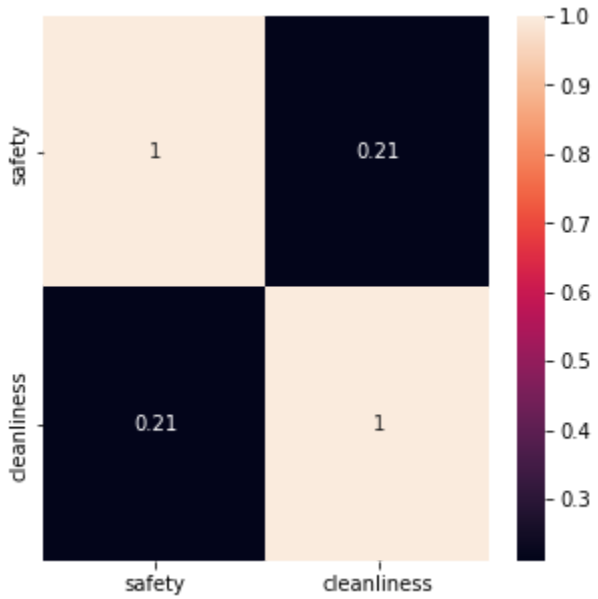
Psychographic factors from our dataset include 'safety' and 'cleanliness'. Let's try to create a heatmap for correlation matrix for Psychographic Segmentation columns.

```
Psychographic = ['safety', 'cleanliness']

x = df[Psychographic ]

# Normalizing values
for column in x.columns:
    x[column] = x[column]  / x[column].abs().max()

plt.figure(figsize=(15,15))
sns.heatmap(x.corr(), annot=True)
plt.show()
```



Behavioral Segmentation: Behavioral segmentation refers to a process in marketing which divides customers into segments depending on their behavior patterns when interacting with a particular business or website.

These segments could include grouping customers by:

a. Their attitude toward your product, brand or service;
b. Their use of your product or service,
c. Their overall knowledge of your brand and your brand's products,
d. Their purchasing tendencies, such as buying on special occasions like birthdays or holidays only, etc.

Behavioral segmentation offers marketers and business owners a more complete understanding of their audience, thus enabling them to tailor products or services to specific customer needs.

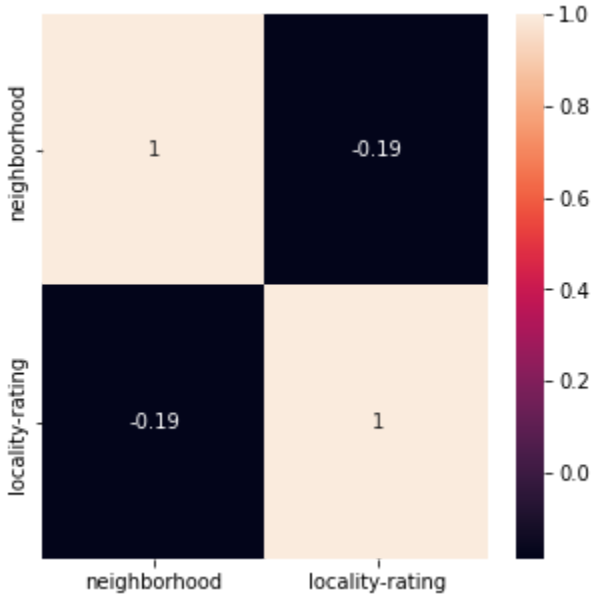
Behavioral factors from our dataset include 'neighborhood' and 'locality-rating'. Let's try to create a heatmap for correlation matrix for Behavioral Segmentation columns.

```
Behavioral = ['neighborhood', 'locality-rating']

x = df[Behavioral  ]

# Normalizing values
for column in x.columns:
    x[column] = x[column]  / x[column].abs().max()

plt.figure(figsize=(15,15))
sns.heatmap(x.corr(), annot=True)
plt.show()
```



Customizing the Market Mix

The *marketing mix* refers to the set of actions, or tactics, that a company uses to promote its brand or product in the market. The 4Ps make up a typical marketing mix - Price, Product, Promotion and Place**.**

a. **Price***:* refers to the value that is put for a product. It depends on costs of production, segment targeted, ability of the market to pay, supply - demand and a host of other direct and indirect factors. There can be several types of pricing strategies, each tied in with an overall business plan

b. **Product***:* refers to the item actually being sold. The product must deliver a minimum level of performance; otherwise even the best work on the other elements of the marketing mix won't do any good.

c. **Place***:* refers to the point of sale. In every industry, catching the eye of the consumer and making it easy for her to buy it is the main aim of a good distribution or 'place' strategy. Retailers pay a premium for the right location. In fact, the mantra of a successful retail business is 'location, location, location'.

d. **Promotion***:* this refers to all the activities undertaken to make the product or service known to the user and trade. This can include advertising, word of mouth, press reports, incentives, commissions and awards to the trade. It can also include consumer schemes, direct marketing, contests and prizes.

All the elements of the marketing mix influence each other. They make up the business plan for a company and handle it right, and can give it great success. The marketing mix needs a lot of understanding, market research and consultation with several people, from users to trade to manufacturing and several others.

Potential Sales in Early Market

Purchasing a “Dream Home” is one of those life accomplishments that tops nearly everyone’s bucket list.

The majority of the customers have a family. For such folks there are a variety of reasons, including market and schooling. Whether you prefer a modernized urban loft or a sprawling suburban home with a white picket fence, most of us hope to find a home that feels like it was made specifically for our family. Here is where our app comes in to

assist such people to find a property at the best-fixed price according to the area and several other factors.

There are some couples who lack the financial means to acquire a home since property prices are too high. As a result, they are left with two options: taking out a house loan or renting the property. Both solutions are equally advantageous. Potential Profit: Because the product caters to the majority of society's demographics, it has a large consumer base.

Suitable Early Market Strategy

Now we try to analyse which location in India is most suitable to create the early market in accordance with the Innovation Adoption Life Cycle.

The technology Adoption Life Cycle is a sociological model that describes the adoption or acceptance of a new product or innovation, according to the **demographic** and **psychological** characteristics of defined adopter groups. The process of adoption over time is typically illustrated as a classical normal distribution or "*bell curve*". The model indicates that the first group of people to use a new product is called "*innovators*", followed by "*early adopters*". Next come the early majority and late majority, and the last group to eventually adopt a product are called "*Laggards*" or "*phobics*."

We defined demographic and psychological characteristics of our dataset they were:

a. Demographic: *'capacity'*, *'Retirees'*, *'Family'*, *'Couple'*, *'SingleProfessionals'*, *'Students'*.
b. Psychological: *'safety'*, *'cleanliness'*

Cities that have lower cost of living and are recommended to different groups of people like family, students, etc with good safety for the people would be the best place for starting the business.