EDA Project Stefan Berkenhoff

Analysis of the King County housing market for a specific client

The Client

(what was given)

Nicole Johnson, Buyer

Looks for a lively, central neighborhood, middle price range, right timing (within a year).

The Client

(additional assumptions)

Living alone, no kids.

Age 30 - 50

Pretty busy, no time for a garden or renovation

Still interested in the lively areas of a city and not the calm suburbs.

Not in a hurry, but kean following her dreams, hence the time frame of 1 year.

As a person with a clear vision for her life, she enjoys actionable and precise recommendations.

My hypothesis

based on the context given

Hypothesis Overview

1. Lively, central neighborhoods are high in demand and therefore will not be easy to find in middle price range / or just with compromises

2. Lively, central neighborhoods are close to the main city-center of Seattle.

3. Time of the year will affect prices to a great extent
(e.g. prices in summer 10% higher than winter) – buying at a specific time will save money

4. Houses close to the city center automatically have no garden. (Lot size is max 10% above living area for houses close to center)

Data understanding

“lively, central neighborhood” is not represented directly in the data

“middle price range” is not directly contained in the data

Detailed analysis of all attributes helped to get a good understanding of the data, domain and data quality

ID	Explanation	Task?	DT	DT trans?	Missing values?	Other remarks:
id	house id	drop	int	-	-	not unique due join
bedrooms	# of bedrooms in house	dt transf				
bathrooms	# of bathrooms in house	dt transf				
sqft_living	size of living area in squarefoot	-				
sqft_lot	size of hole property up to boundary	-				
floors	# of floors	dt transf				
waterfront	is object located at waterfront	fix				
view	# of views by interested ppl	fix				
condition	rating of the object's overall condition, based on based on King County grading system	-				
grade	overall grade given to the housing unit, based on King County grading system					
t_above	size of upper levels	-				
basement	size of basement	-				

Central Tendency / Overview by .describe()

```
df_age = df.groupby('house_id').agg({'yr_built': 'mean'})
df_age['age'] = df_age['yr_built'].apply(lambda x: 2015 - x)

display(
    df_age.describe()
)
```

	yr_built	age
count	21420.000	21420.000
mean	1971.093	43.907
std	29.387	29.387
min	1900.000	0.000
25%	1952.000	18.000
50%	1975.000	40.000
75%	1997.000	63.000
max	2015.000	115.000

Spread: Range, iqr, variance, std

+ Code + Markdown

```
age_range = df_age['age'].max() - df_age['age'].min()
age_iqr = df_age['age'].quantile(0.75) - df_age['age'].quantile(0.25)
age_variance = df_age['age'].var()
age_std = df_age['age'].std()

print(f" range: {age_range} \n iqr: {age_iqr}\n variance: {age_variance}\n std derivation: {age_std}")
```

Python

```
range: 115.0
iqr: 45.0
variance: 863.6040314477722
std derivation: 29.38714857964422
```

Univariate analysis for all the attributes helped to understand the domain better

Building a comprehensive overview over attributes, with tasks, remarks and explanation

Data cleaning

“middle price range” is not directly contained in the data

“lively, central neighborhood” is not represented directly in the data

Data cleaning encompasses fixing NaN Values, data errors and data type transformation

```
#Checking for missing values overall
df.isna().sum()
```

```
id                0
bedrooms          0
bathrooms         0
sqft_living       0
sqft_lot          0
floors            0
waterfront     2391
view             63
condition         0
grade             0
sqft_above        0
sqft_basement   452
yr_built          0
yr_renovated   3848
zipcode           0
lat               0
long              0
sqft_living15     0
sqft_lot15        0
date              0
price             0
id.1              0
dtype: int64
```

Missing values had to be fixed. Potentially impactful assumptions had to be made, e.g. missing ‘view’ data. Drop? set to?ç

Data cleaning for yr_renovated

```
display(df["yr_renovated"].unique())

df['yr_renovated'] = df.yr_renovated.apply(lambda x: x/10)
df = df.fillna({'yr_renovated': 0})
df = df.astype({'yr_renovated': 'int32'})

df["yr_renovated"].unique()
```

✓ 0.0s

```
array([    0., 20130.,    nan, 19730., 20100., 19910., 19790., 20010.,
       20120., 19860., 19900., 20030., 19620., 19920., 20060., 19400.,
       19550., 20070., 20140., 19890., 19820., 20050., 20000., 19540.,
       19960., 20150., 19830., 19600., 19720., 19970., 19940., 19450.,
       20040., 19700., 19950., 19990., 20080., 19840., 20110., 19980.,
       19880., 20090., 19670., 19690., 20020., 19770., 19870., 19650.,
       19640., 19580., 19680., 19850., 19630., 19800., 19740., 19810.,
       19500., 19560., 19570., 19930., 19750., 19460., 19480., 19780.,
       19760., 19340., 19590., 19530., 19440., 19510., 19710.])

rray([   0, 2013, 1973, 2010, 1991, 1979, 2001, 2012, 1986, 1990, 2003,
       1962, 1992, 2006, 1940, 1955, 2007, 2014, 1989, 1982, 2005, 2000,
       1954, 1996, 2015, 1983, 1960, 1972, 1997, 1994, 1945, 2004, 1970,
       1995, 1999, 2008, 1984, 2011, 1998, 1988, 2009, 1967, 1969, 2002,
       1977, 1987, 1965, 1964, 1958, 1968, 1985, 1963, 1980, 1974, 1981,
       1950, 1956, 1957, 1993, 1975, 1946, 1948, 1978, 1976, 1934, 1959,
       1953, 1944, 1951, 1971], dtype=int32)
```

Data improvement

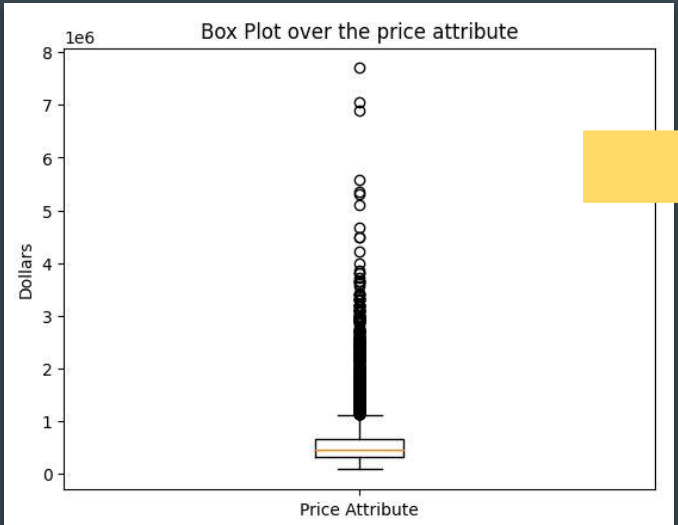
Tackling the challenges of

“middle price range” is not directly contained in the data

and

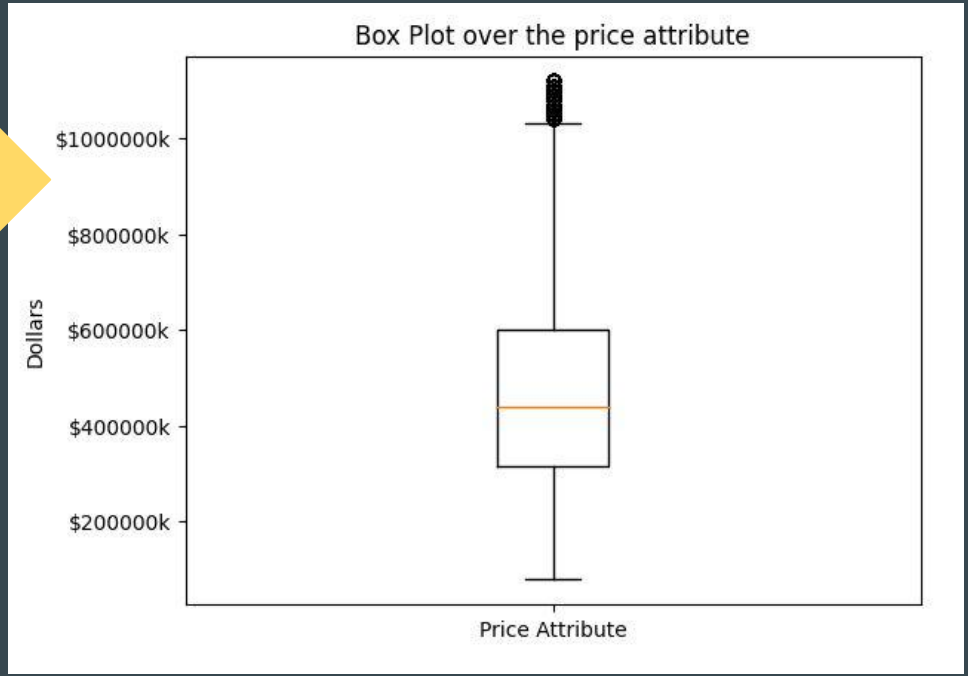
“lively, central neighborhood” is not represented directly in the data

Introducing price categories (I/II)



Max: $ 7 700 000
Median $ 450 550
Min: $ 78 000

# of upper outliers: 1158
# of lower outliers: 0



Max: $ 1 120 000
Median $ 439 000
Min: $ 78 000

Introducing price categories (II/II)

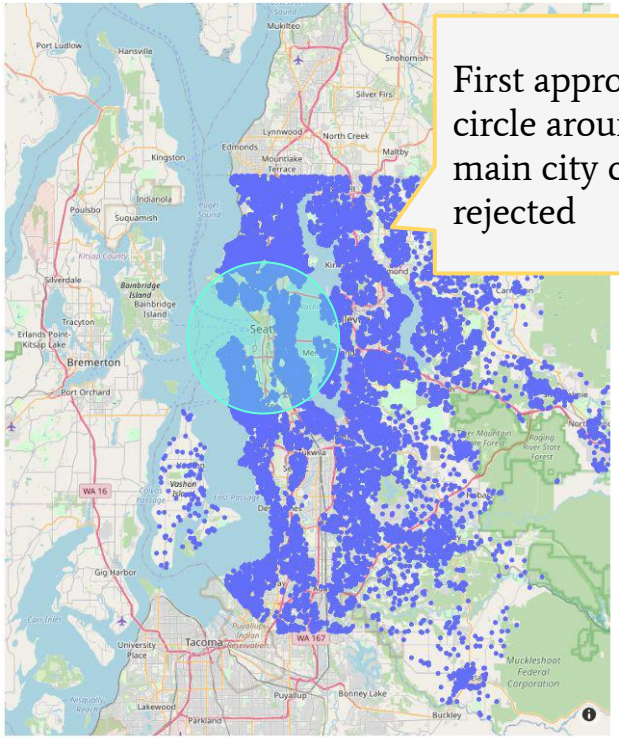
```
# Cutting data frame in 5 equal sized bins, renaming the labels to something speaking

temp_series = pd.cut(filtered_df_2['price'], bins=5, labels=['low', 'med-low', 'med', 'med-high', 'high'])
```

price_cat_total	min_price	max_price
low	78000.0	286308.0
med-low	286500.0	494500.0
med	494815.0	703011.0
med-high	703300.0	911100.0
high	912000.0	1120000.0

Improving my domain knowledge – researching information about central, lively neighborhoods

Locations of al objects provided



First approach, to circle around the main city centre, was rejected

Second approach resulted in a list of interesting zip codes after good old googling.

Researched List of Zip Codes having a lively, central neighborhood

1. **Capitol Hill** (Zip Code: 98102, 98112, 98122): Known for its hipster cu… of historic and modern architecture. Capitol Hill is centrally located an… restaurants, cafes, and entertainment options.
2. **Belltown** (Zip Code: 98121): This neighborhood is close to downtown … restaurants, bars, and live music venues. It's a hub of activity, particula…
3. **Fremont** (Zip Code: 98103): Often referred to as the "Center of the U… installations, unique shops, and lively events. It has a distinct and creat…
4. **Queen Anne** (Zip Code: 98109): With its stunning views of the city an… Needle and various cultural attractions), Queen Anne offers a mix of hi…
5. **South Lake Union** (Zip Code: 98109): This area has transformed in re… and is now a hub for innovation, offering a blend of residential and com… access.
6. **Ballard** (Zip Code: 98107): Known for its maritime history, Ballard com… …ong sense of community. It's also home to the famous Ballard Locks…

…neer Square (Zip Code: 98104): Seattle's historic district, Pioneer… …ques, and a variety of restaurants. It's a hub of cultural activity an…

…eenwood (Zip Code: 98103, 98117): Greenwood is a more residentia… …ng Greenwood Avenue with shops, cafes, and restaurants.

…allingford (Zip Code: 98103): This neighborhood is known for its tra… …s Works Park. It offers a quieter but still lively atmosphere.

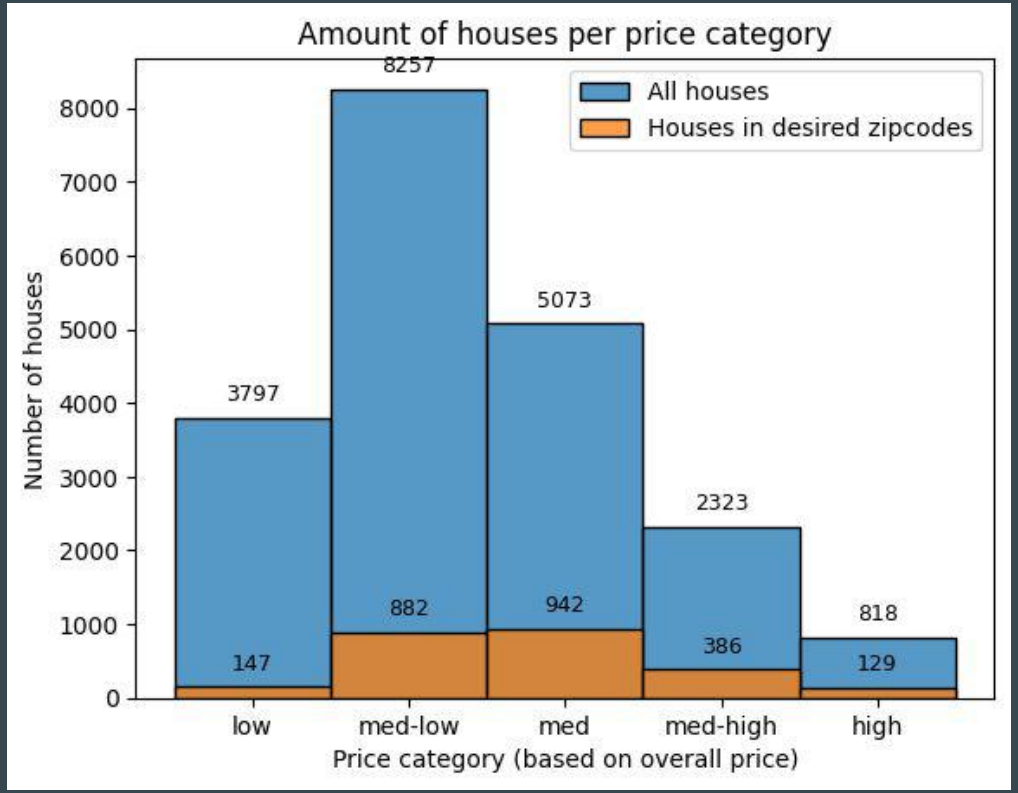
…lumbia City (Zip Code: 98118): Located in the Rainier Valley, Colum… …f restaurants, cafes, and shops. It often hosts community events.

98102, 98112, 98122, 98121, 98103, 98109, 98107, 98104, 98117, 98118

Checking the hypothesis

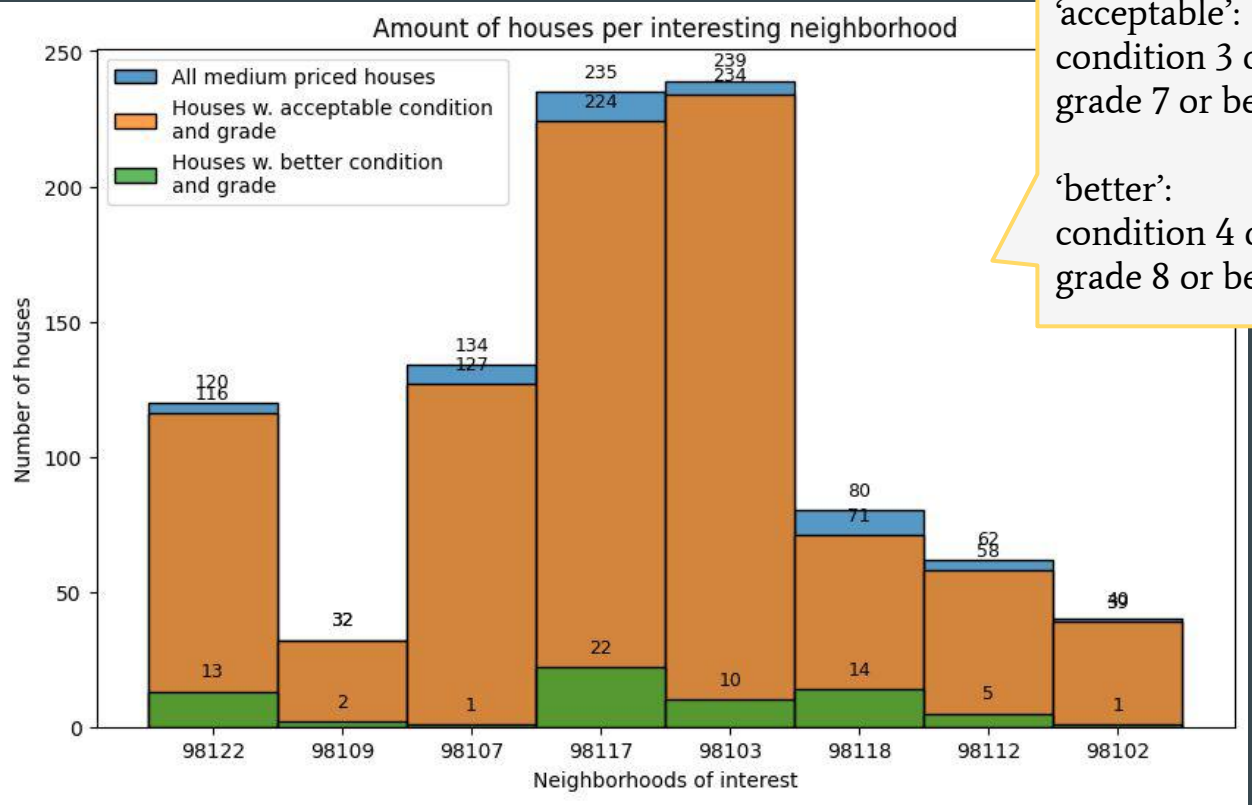
Are those lively, central neighborhoods even offering houses in the mid price range?

The do.


Amount of houses per price category
All houses
Houses in desired zipcodes
Number of houses
0
1000
2000
3000
4000
5000
6000
7000
8000
3797
8257
5073
2323
818
147
882
942
386
129
low
med-low
med
med-high
high
Price category (based on overall price)

But those houses must be in poor condition or of low quality?

No.



'acceptable':
condition 3 or better,
grade 7 or better

'better':
condition 4 or better,
grade 8 or better

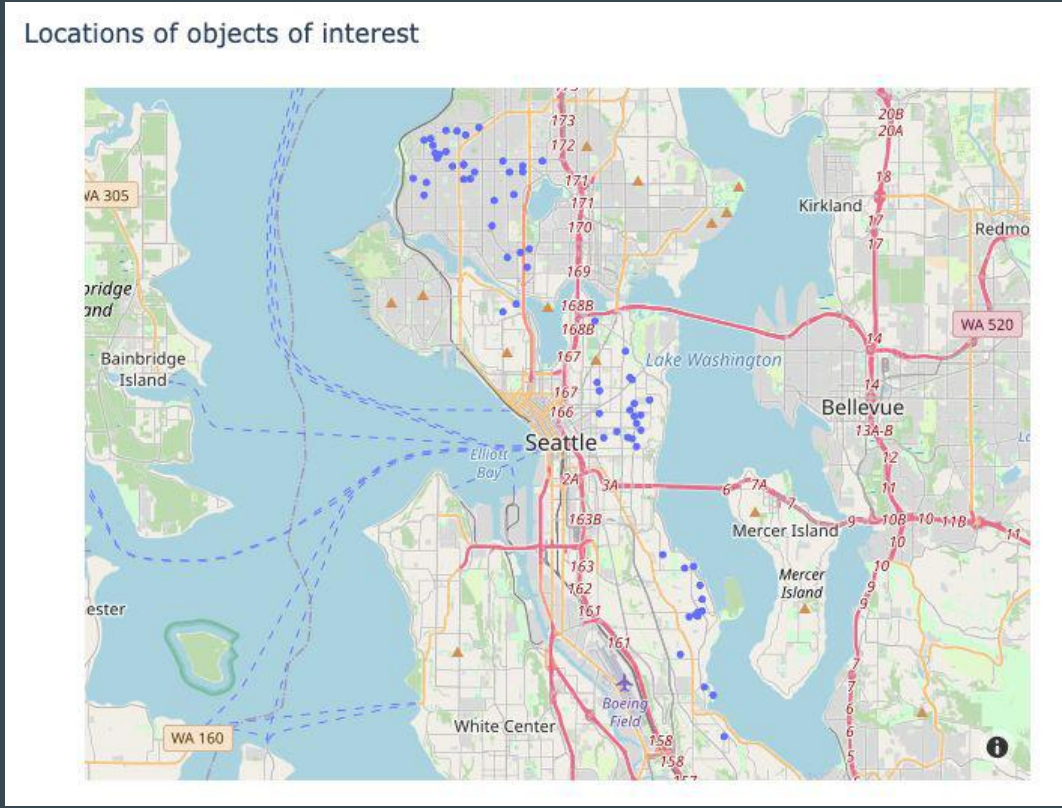
H1: rejected

~~Lively, central neighborhoods are high in demand and therefore will not be easy to find in middle price range / or just with compromises~~

There are **plenty of high quality houses** in the mid price range in lively, central neighborhoods

But are those interesting zip code areas still close to the main city center of Seattle?

Yes, to the main part



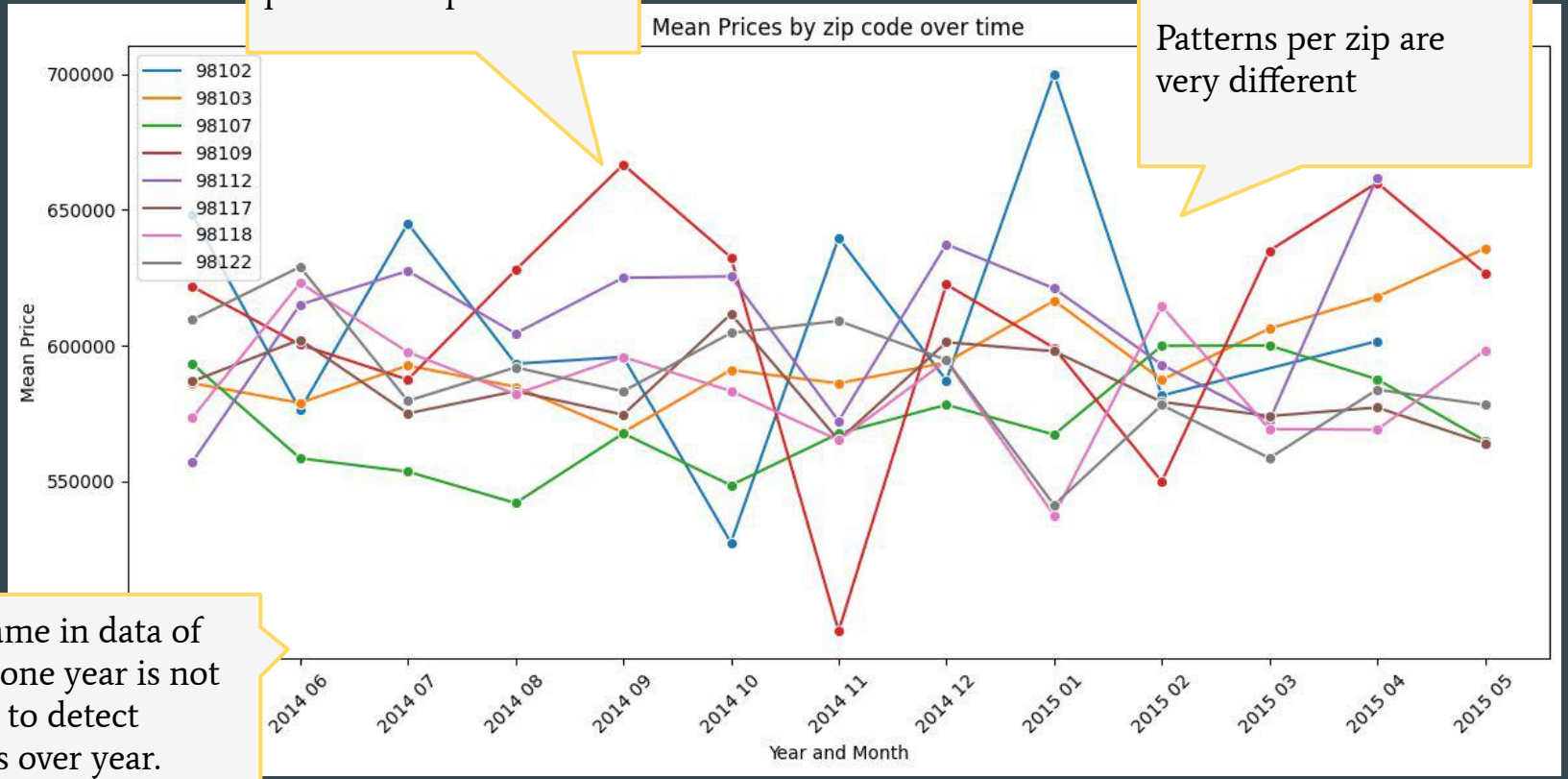
H2: confirmed

Lively, central neighborhoods are also close to the main city-center of Seattle.

Can I save money by choosing the right time in the year?

Not really.

Some neighborhoods are very volatile in the price development



Patterns per zip are very different

time frame in data of approx one year is not enough to detect patterns over year.

H3: rejected

~~Time of the year will affect prices to a great extent – buying at a specific time will save money.~~

If time of the year affects the prices could not be confirmed. Price development for each neighborhood is highly individual, a general heuristic cannot be derived.

But at least I don’t have to take care of a garden?

H4: rejected

~~Houses where sqft lot is more than 10% higher than sqft living are not in the city center.~~

Of the 68 houses in the final set only 4 had a lot size not being bigger than 10% of the living size. Therefore it can be assumed, that all houses have additional property to take care of – even close to the city center.

Summary and recommendation

Rec 1: Buy now

The housing market of Seattle seems to offer just enough houses in lively, central neighborhoods in above average condition and grad. Make use of that before the market changes and the offer is reduced.

Rainier Valley

Rec2: Buy in 98118 for more space

Here the houses offer - on average - the highest living area.

98118 - Rainier Valley - offers on average the biggest houses

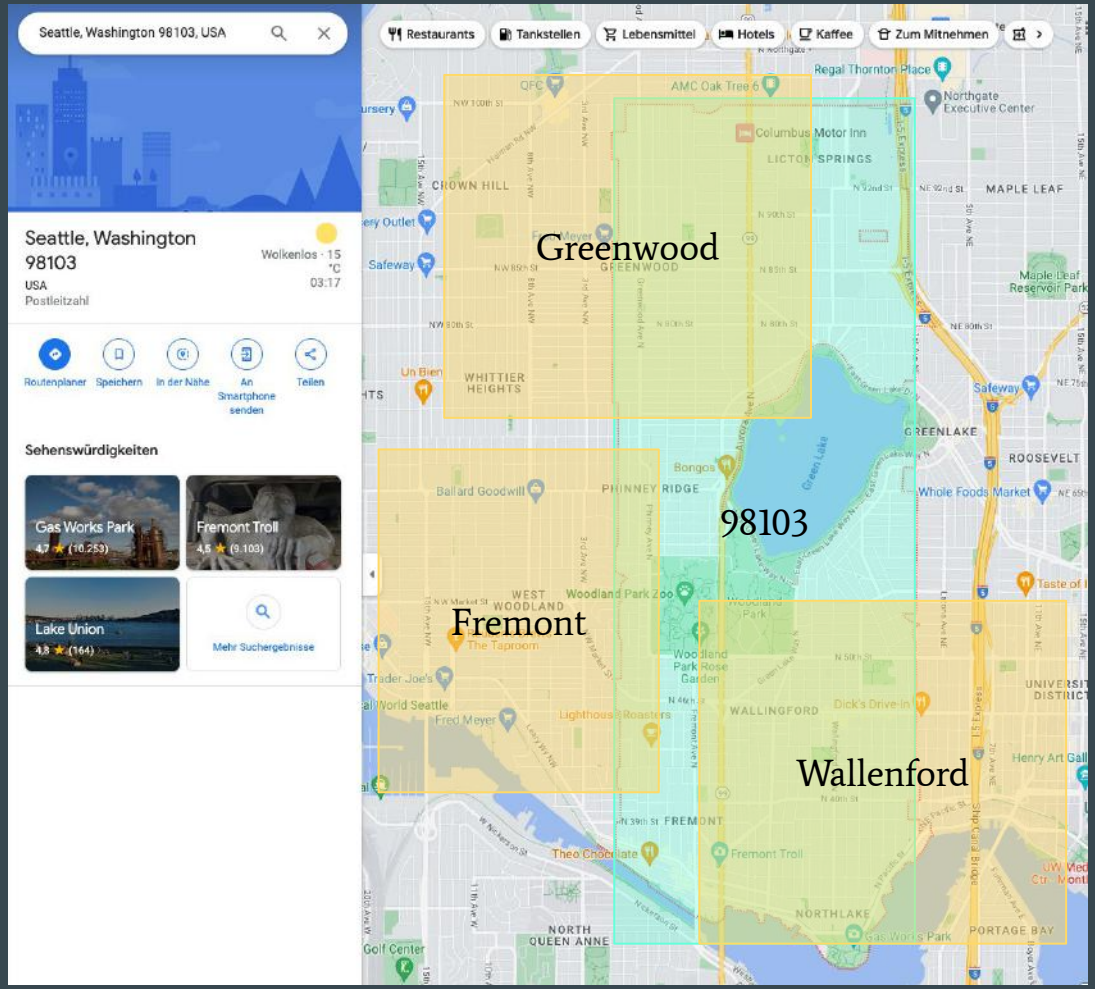
Capitol Hill

Rec3: Buy in 98112 to be extra close + no garden

Critical assessment and next steps

finding neighborhoods by zipcode

- neighborhoods and zip codes do not match very well
- Next Step: Use lat / long info to get precise neighborhood name.




```python
def get_neighborhood_name(latitude, longitude):
    api_key = G_API
    url = "https://maps.googleapis.com/maps/api/geocode/json?latlng={},{}&key={}".format(latitude, longitude,
    response = requests.get(url)
    if response.status_code == 200:
        answer = json.loads(response.content)
        pprint.pprint(answer)
        neighborhood = answer["results"][0]["address_components"][2]["long_name"]
        return neighborhood
    else:
        return None

neighborhood = get_neighborhood_name(47.606209,
print(neighborhood)
```

```
{'plus_code': {'compound_code': 'JM49+F5M Seattle,
               'global_code': '84VVJM49+F5M'},
 'results': [{'address_components': [{'long_name':
                                      'short_name':
                                      'types': ['st
                                     {'long_name':
                                      'short_name': '5th Ave',
                                      'types': ['route']},
                                     {'long_name': 'Downtown Seattle',
                                      'short_name': 'Downtown Seattle',
                                      'types': ['neighborhood', 'political']},
                                     {'long_name': 'Seattle',
                                      'short_name': 'Seattle',
                                      'types': ['locality', 'political']},
                                     {'long_name': 'King County',
                                      'short_name': 'King County',
                                      'types': ['administrative_area_level_2',
                                                'political']},
                                     {'long_name': 'Washington',
                                      'short_name': 'WA',
                                      'types': ['administrative_area_level_1',
                                                'political']},
                                     {'long_name': 'United States',
                                      'short_name': 'US',
                                      'types': ['country', 'political']},
...
                'place_id': 'ChIJCzYy5IS16lQRQrfeQ5K5Oxw',
                'types': ['country', 'political']}],
 'status': 'OK'}
Downtown Seattle
```

Google Maps Geocode API will return neighborhood name based on long / lat

Critical assessment and next steps

Relations?

- I didn't find / made use of any (interesting) relations between attributes.
- Maybe redoing it with another client persona would increase the need for that.

thank you all.