EDA with King County House prices

finding a house to buy for Thomas Hansen

Goal of this notebook

Find top houses to buy for Thomas Hansen

Dataset: King County house prices

Stakeholder Information:

- 5 kids
- no money
- wants nice (social) neighborhood
- Timing?
- Location?

Content

Exploratory data analysis (EDA)

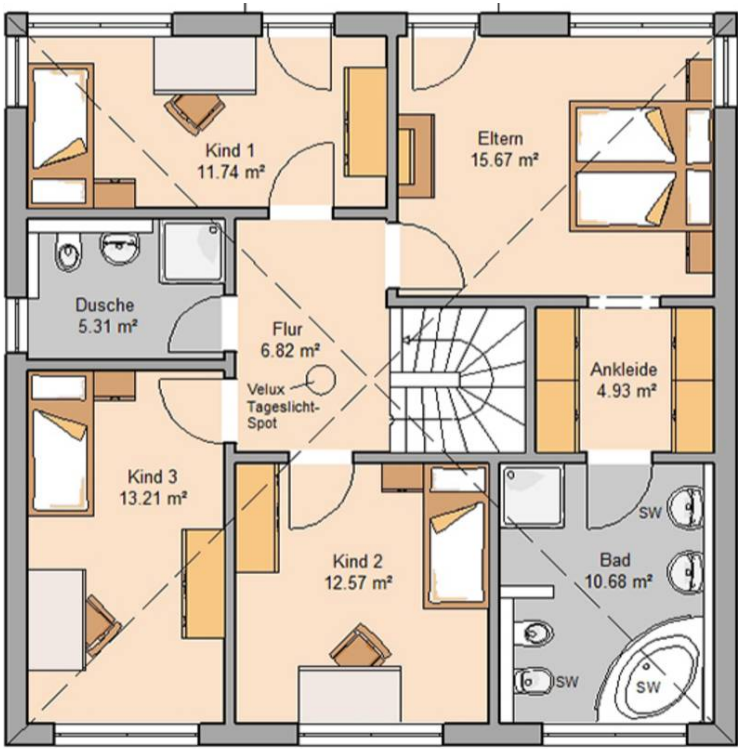
1. Get an overview of the data
2. Filter data and create new values regarding Thomas Hansens needs
3. Create a scoring system
4. Show results in order of highest to lowest score

No money + 5 kids

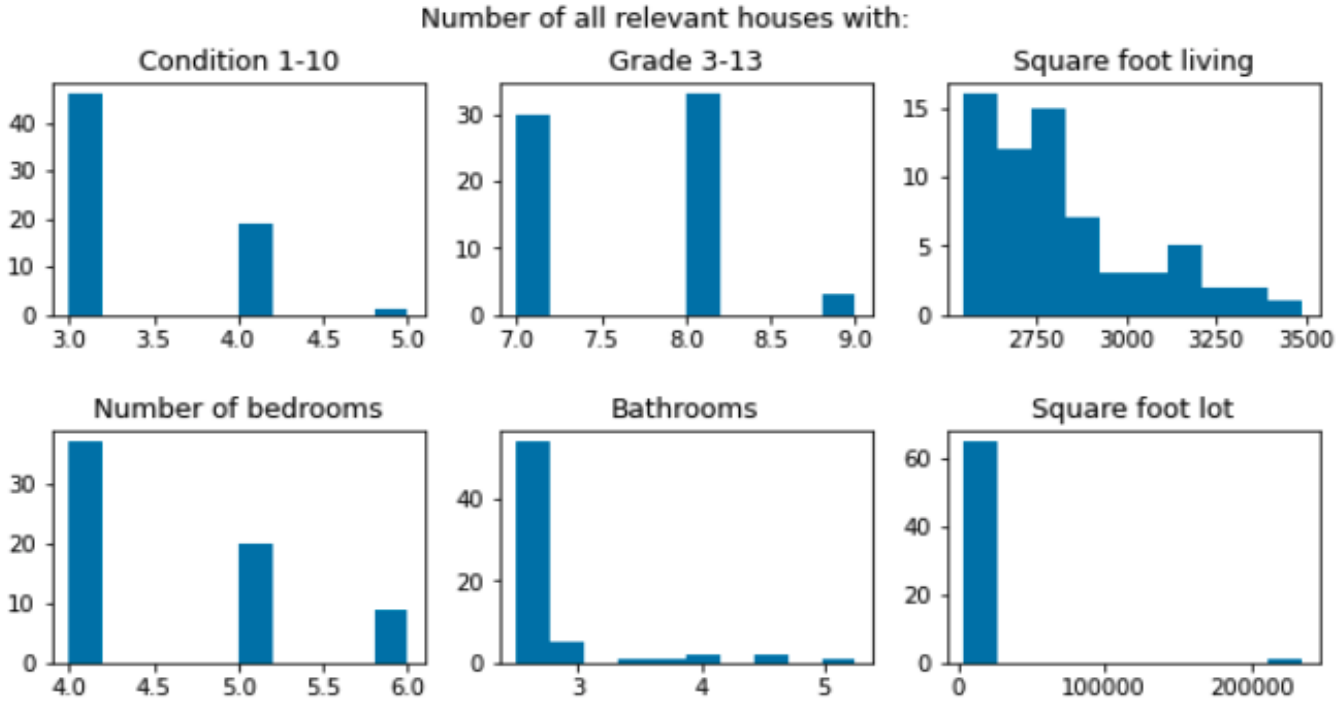
Filtering existing data

- **Price - cheapest 25%**: **<= 322k US$**
- **Bedrooms - biggest 25%**: **>= 4 bedrooms**
- **Bathrooms - biggest 25%**: **>= 2.5 bathrooms**
- **Sqft_living - biggest 25%**: **>= 2550 sqr feet**

→ Only 66 of 21,597 houses left



House comparison

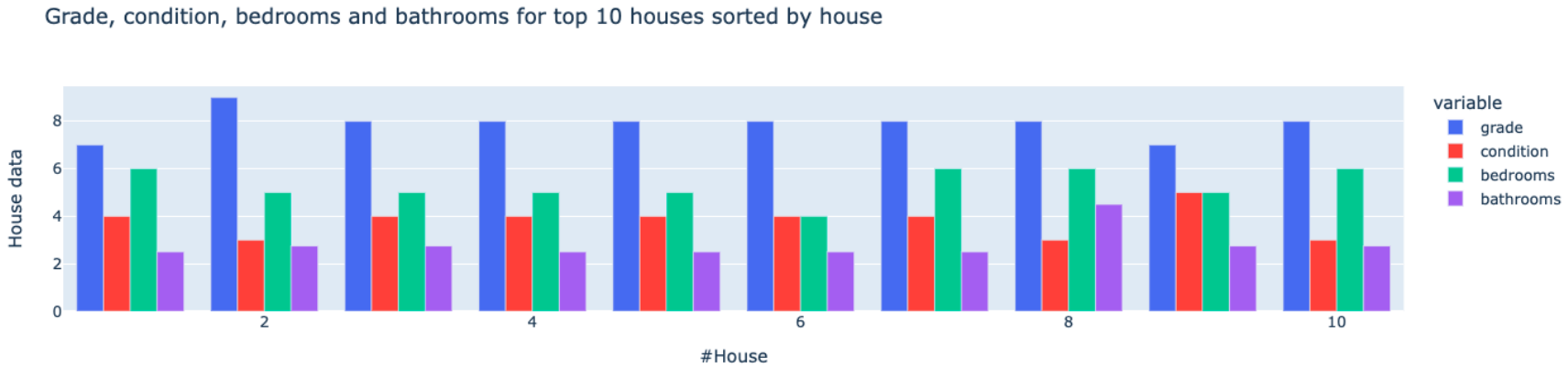


Comparison between top10 houses

Interestingly our #1 has the lowest grade within the top10.

This shows, our individual scoring is different compared to KingHouse grading.

(But it has 6 bedrooms, one for each kid + one for the parents. Sounds great!)



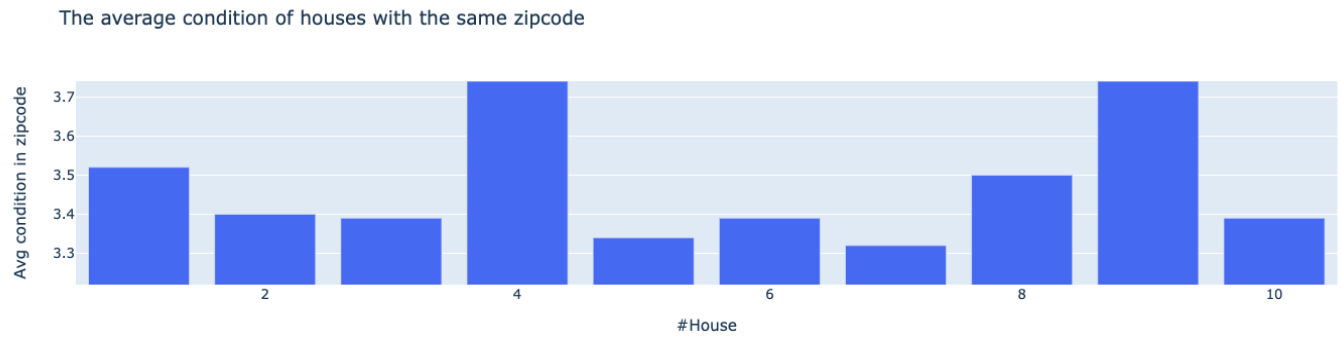
Nice (social) neighborhood

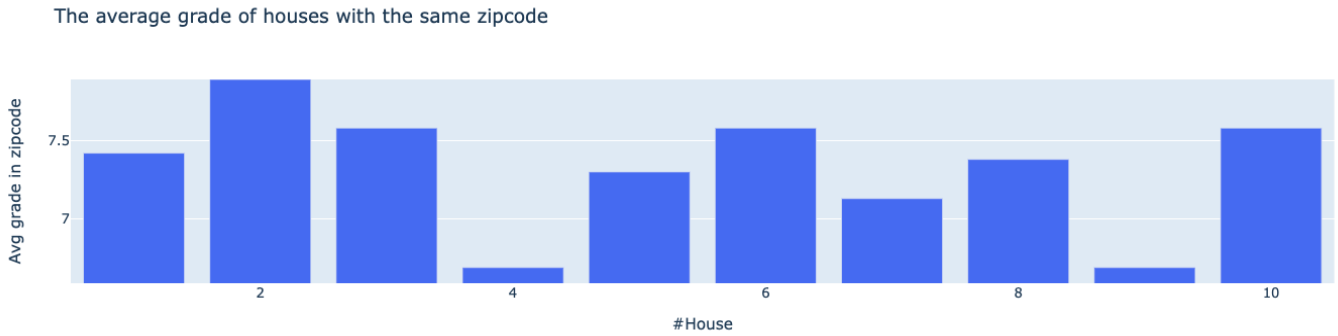
Defining new values: average condition and average grade in the neighborhood (same zip code)

average condition

and

average grade within the same zipcode

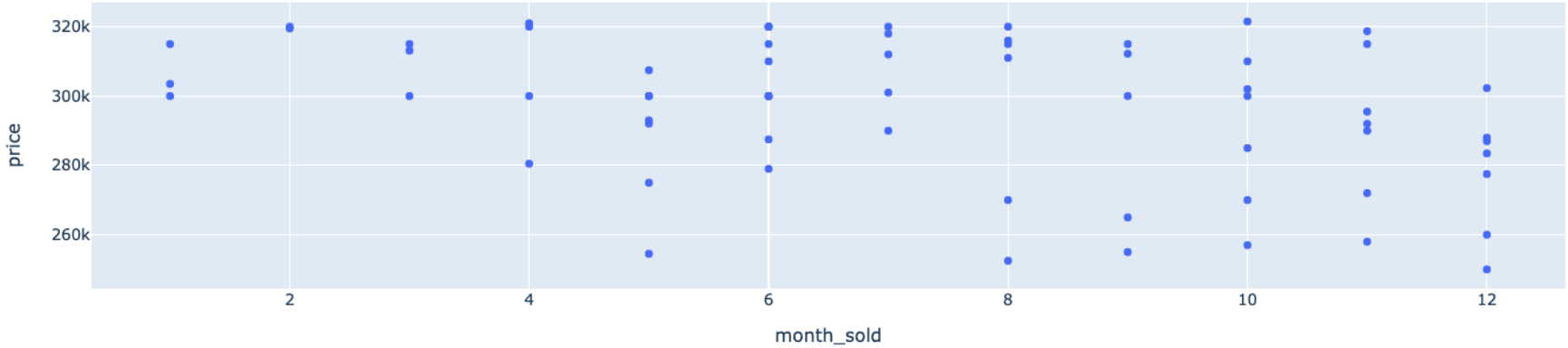




Timing?

Checking, when cheapest houses were sold

Thomas Hansen seems unsure about the timing to buy a house. Regarding our 66 houses, we show, when the cheapest houses were sold. It appears, **May** and **August-December** could be good timing.



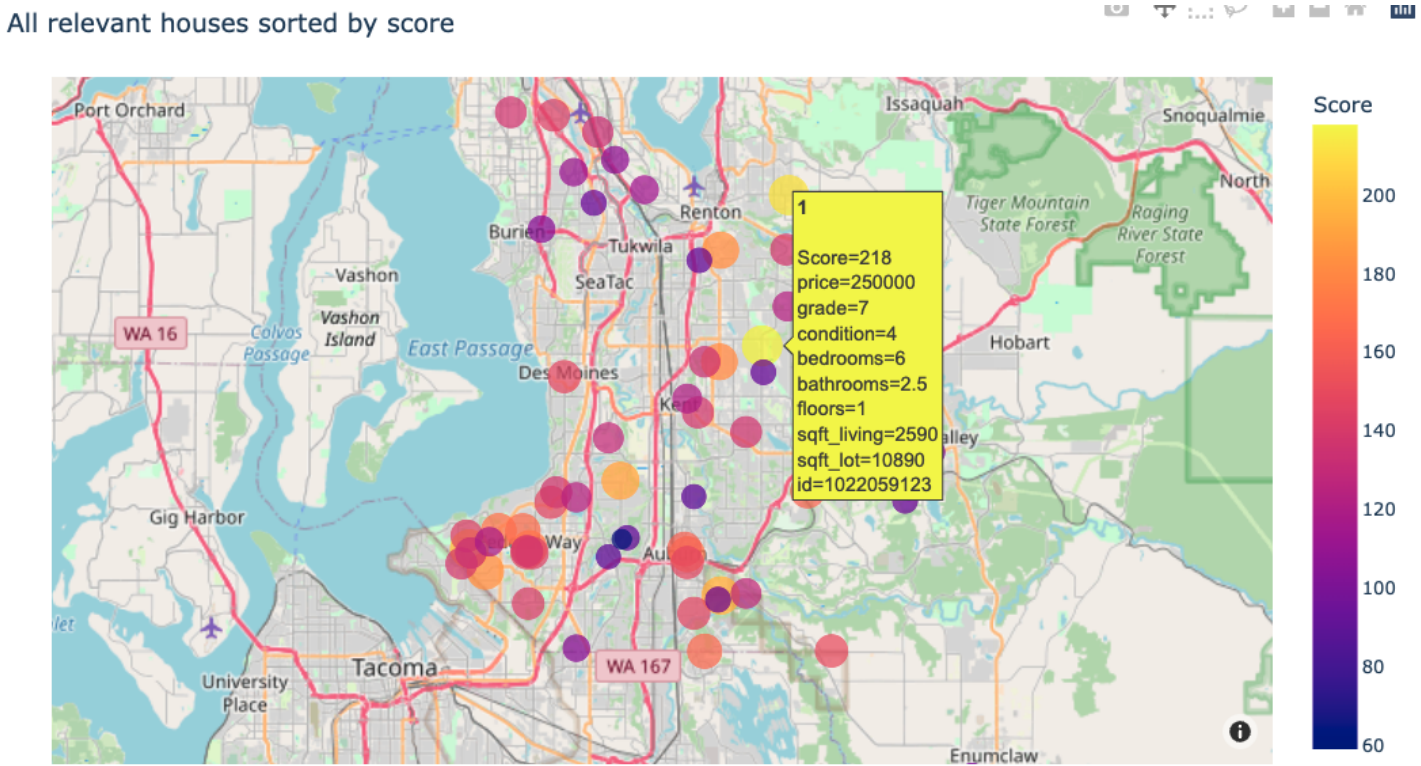
Building an individual scoring system

1. *housescore*: best overall values (in this priority order: bedrooms, bathrooms, price, condition, grade)
2. *pricescore*: cheapest price
3. *conditionscore*: highest condition of this house and the neighbourhood houses
4. *gradescore*: highest grade of this house and the neighbourhood houses

Final Score: We divide by the qty of rows (here: 66) and the qty of different scores (here: 4). **This results in a score between 0 and 1 (with 1 being the perfect score).**

	House	Score	housescore	bedrooms	bathrooms	price	pricescore	grade	grade_mean	gradescore	condition	condition_mean	conditionscore	sqft_living	sqft_lot	floors	zipcode
22	1	0.83	65	6	2.50	250000.0	66	7	7.42	25	4	3.52	62	2590	10890	1.0	98042
40	2	0.80	57	5	2.75	257000.0	62	9	7.89	66	3	3.40	27	2930	10148	2.0	98059
51	3	0.76	55	5	2.75	302300.0	28	8	7.58	59	4	3.39	59	3130	9450	1.0	98023
7	4	0.74	42	5	2.50	270000.0	57	8	6.69	32	4	3.74	64	2630	8470	1.5	98002
44	5	0.72	50	5	2.50	290000.0	46	8	7.30	40	4	3.34	54	2780	9652	1.0	98001
50	6	0.69	26	4	2.50	287500.0	48	8	7.58	52	4	3.39	57	2570	9000	1.0	98023
14	7	0.69	62	6	2.50	300000.0	31	8	7.13	36	4	3.32	53	2590	11250	1.0	98055
27	8	0.68	63	6	4.50	290000.0	45	8	7.38	41	3	3.50	31	2810	11214	1.0	98031
47	9	0.65	41	5	2.75	265000.0	59	7	6.69	5	5	3.74	66	2920	5250	1.5	98002
1	10	0.64	66	6	2.75	315000.0	19	8	7.58	61	3	3.39	24	2940	7350	1.0	98023

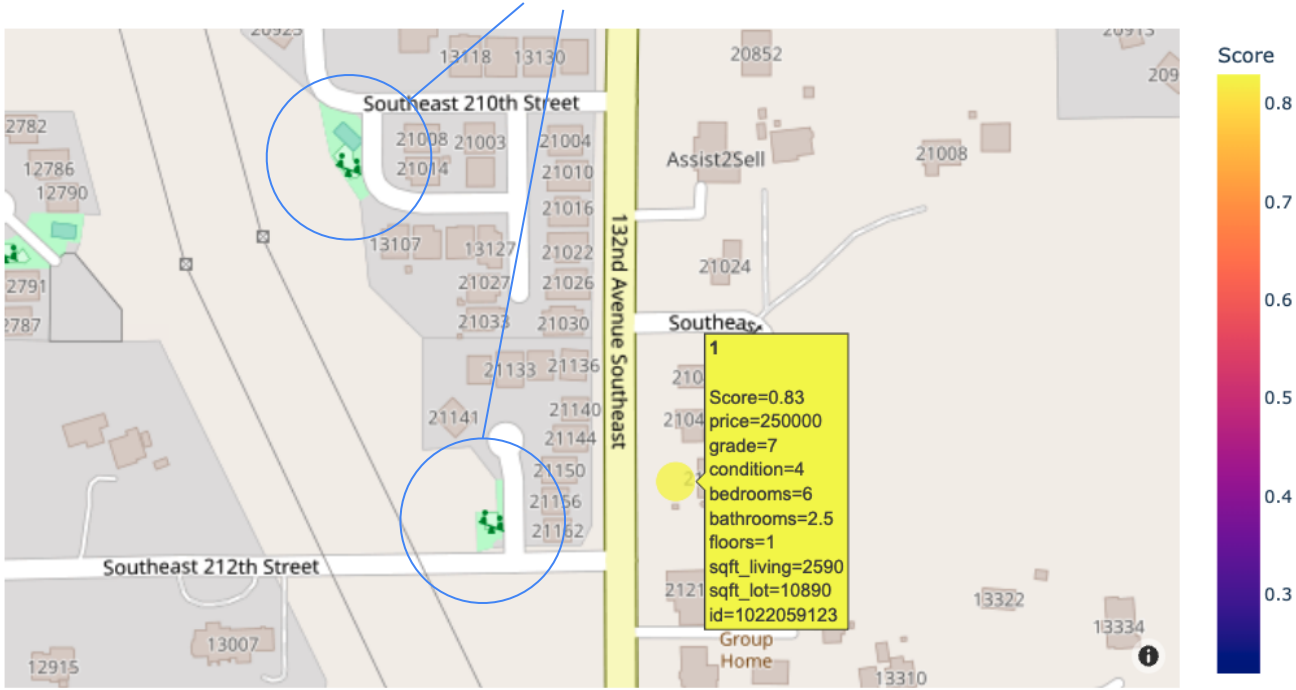
Result: Relevant houses in streetmap



Zooming in on #1

Playgrounds for the kids - yaay!

All relevant houses sorted by score



Summary, Bonus and Outlook

Summary

- analysis and filters according to Thomas Hansen’s needs
- creation of a scoring system
- plotting the results on a map, incl. relevant hover data

Recommendation: use the map (sorted by score) and have a look at the area of the house

Outlook

- the scoring system could be used for other stakeholders. A Python script could be created, which asks for input (e.g. # of bedrooms, price limit, ...) and automatically plots output
- additional values could be included within the scoring system, such as: sqft_living, sqft_lot, floors