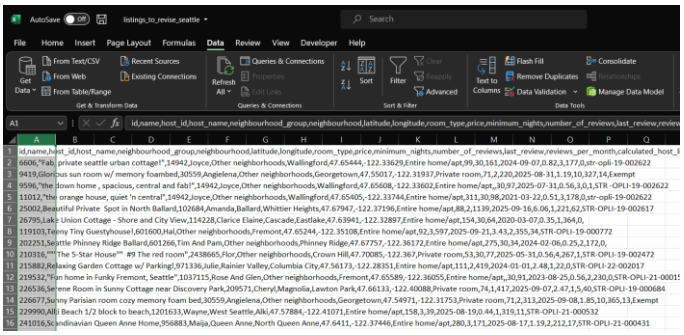
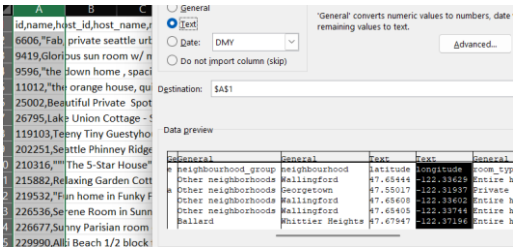
TABLEAU PROJECT

Airbnb Analysis

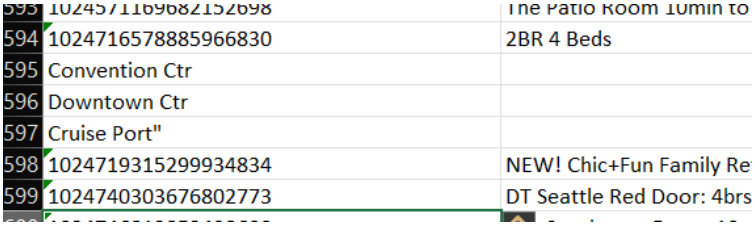
1. First step is cleaning data in Excel to prepare it for the visualization in Tableau.



1.1. It's also important to check the **data types** before proceeding with the Data transferring **to columns**, because some numerical ones can then look not as expected and we'll have much more additional work to do when updating it.

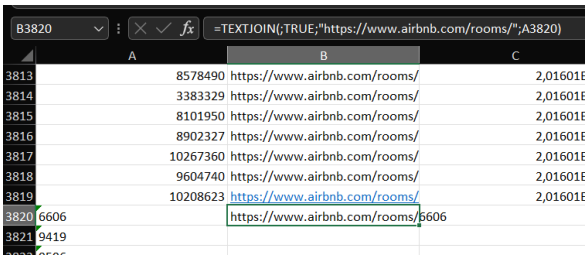


1.2. And to pay attention to the data after transfering.



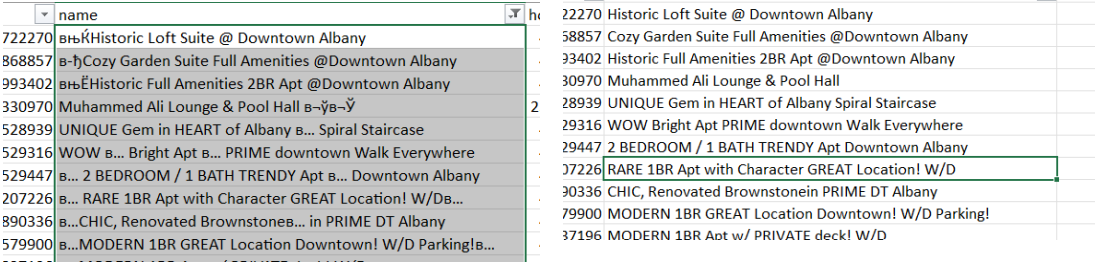
594	1024716578885966830 2BR 4 Beds
595	Convention Ctr
596	Downtown Ctr
597	Cruise Port"
598	1024719315299934834 NEW! Chic+Fun Family Re
599	1024740303676802773 DT Seattle Red Door: 4brs

1.3. Additionally, our main file (to which I want to add the updated data from few last years) contains url address for the pages. I can add it using TEXTJOIN here.



	A	B	C
3813	8578490	https://www.airbnb.com/rooms/	2,01601E
3814	3383329	https://www.airbnb.com/rooms/	2,01601E
3815	8101950	https://www.airbnb.com/rooms/	2,01601E
3816	8902327	https://www.airbnb.com/rooms/	2,01601E
3817	10267360	https://www.airbnb.com/rooms/	2,01601E
3818	9604740	https://www.airbnb.com/rooms/	2,01601E
3819	10208623	https://www.airbnb.com/rooms/	2,01601E
3820	6606	https://www.airbnb.com/rooms/6606	
3821	9419		

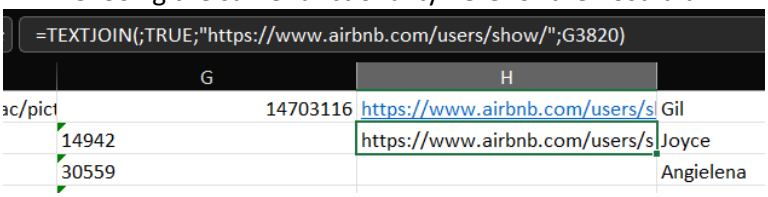
1.4. For some names we have weird signs in there. Needs to be cleared as well. Used FIND & REPLACE to do so. Looks way better now.



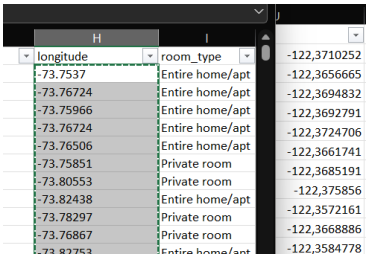
	name
722270	вњЌHistoric Loft Suite @ Downtown Albany
868857	в-ђCozy Garden Suite Full Amenities @Downtown Albany
993402	вњЁHistoric Full Amenities 2BR Apt @Downtown Albany
330970	Muhammed Ali Lounge & Pool Hall в¬ўв¬Ў
528939	UNIQUE Gem in HEART of Albany в... Spiral Staircase
529316	WOW в... Bright Apt в... PRIME downtown Walk Everywhere
529447	в... 2 BEDROOM / 1 BATH TRENDY Apt в... Downtown Albany
207226	в... RARE 1BR Apt with Character GREAT Location! W/Dв...
890336	в...CHIC, Renovated Brownstoneв... in PRIME DT Albany
579900	в...MODERN 1BR GREAT Location Downtown! W/D Parking!в...

22270	Historic Loft Suite @ Downtown Albany
58857	Cozy Garden Suite Full Amenities @Downtown Albany
93402	Historic Full Amenities 2BR Apt @Downtown Albany
30970	Muhammed Ali Lounge & Pool Hall
28939	UNIQUE Gem in HEART of Albany Spiral Staircase
29316	WOW Bright Apt PRIME downtown Walk Everywhere
29447	2 BEDROOM / 1 BATH TRENDY Apt Downtown Albany
07226	RARE 1BR Apt with Character GREAT Location! W/D
90336	CHIC, Renovated Brownstonein PRIME DT Albany
79900	MODERN 1BR GREAT Location Downtown! W/D Parking!
37196	MODERN 1BR Apt w/ PRIVATE deck! W/D

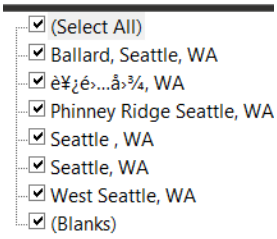
1.5. Using the same functionality here for the host id url.



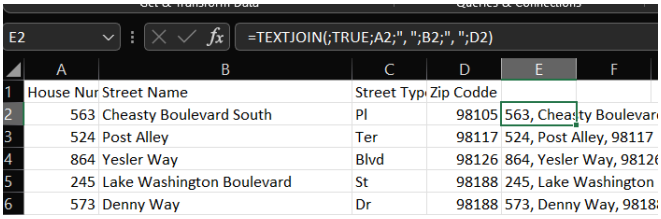
1.6. Also needed to replace **longitude & latitude** values "." with "," to match.



1.7. Location description that needs to be cleaned as well.



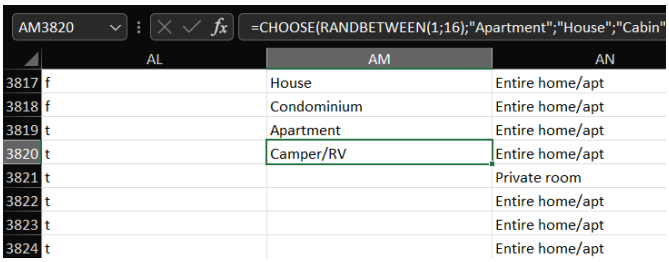
1.8. After I combined the new, fresh UP to date file with listings from AIRBNB, I went checking on the base file and saw, that data still has some columns that were not present in downloaded listing. I decided to add missing addresses. As Im just working on Seattle and mainly need to visualize the data I possess (after I clean it), it doesn't have to be exactly accurate. So, why not to use Python here, I though.



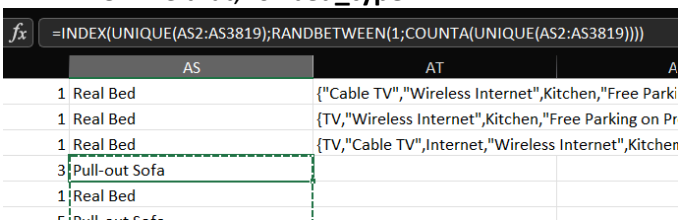
1.8.1. Since we already have all necessary libraries in, and just commony known data from available sources, here how the final code looks like.

```
cwiczenia > python > random_streets.py > ...
import random
import csv

street_names = ['1st Avenue (Seattle)', 'Alaskan Way',  'The Ave',  'Broadway (Seattle)',   'Cant
street_types = ['St', 'Ave', 'Blvd', 'Way','Dr', 'Pl', 'Ln', 'Rd', 'Ct', 'Ter']
zip_codes = ['98101',   '98102',    '98103',    '98104',    '98105',    '98106',    '98107', '981

rows = 6996
data = [['House Number', 'Street Name', 'Street Type', 'Zip Codde']]

for _ in range(rows):
    house_num = random.randint(1, 999)
    street_name = random.choice(street_names)
    street_type = random.choice(street_types)
    zip_code = random.choice(zip_codes)
    data.append([house_num, street_name, street_type, zip_code])

with open('seattle_addresses.csv', 'w', newline='') as f:
    writer = csv.writer(f)
    writer.writerows(data)

print('CSV file "seattle_addresses" generated with 6996 rows!')
```

2. Some position don't have accomodation type while others do? Not a problem! **CHOOSE+RANDBETWEEN+UNIQUE(**earlier) – and we have it handled!

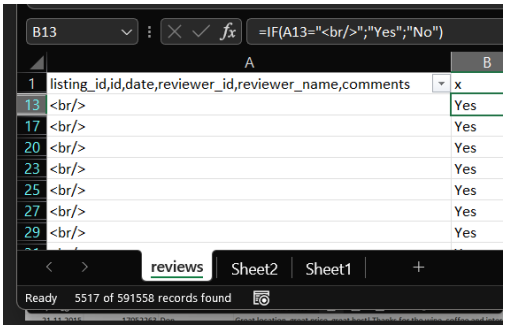


2.1. Or like that, for **bed_type.**



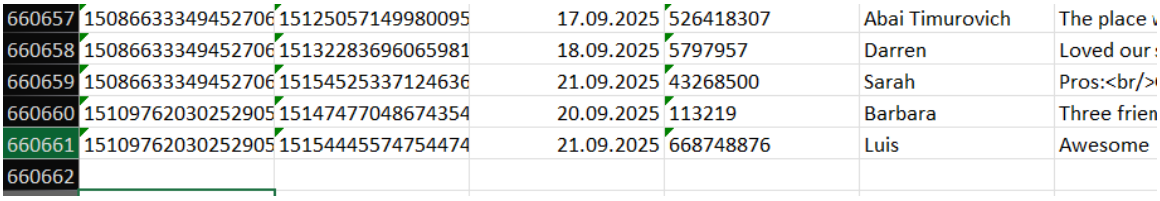
2.2. The **UP TO DATE** reviews file has pretty many positions though… Let's clean them and see what we have.

2.2.1. Apparently won't need those…



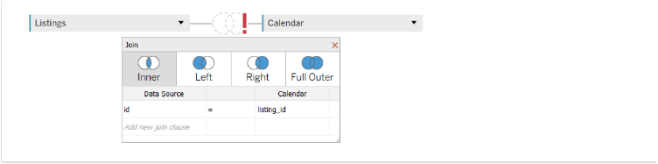
2.2.2. Some data (numeric) was not read well by Excel after using **Data to columns.** Good that I had a backup file.

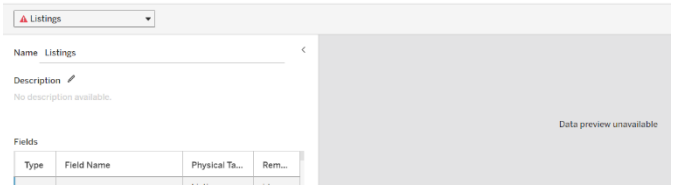
2.2.3. Uf, cleaned and transferred over to the base dataset. 660k reviews in total.



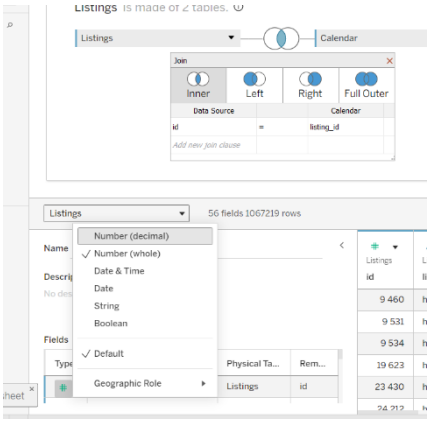
2.3. Well, for the **calendar** data (appointments) there are some that I can leave aside this time, I guess…

2.4. Now as it's all clean I can go and start with **Tableau.** Trying to remember to keep the data type same for JOINS.

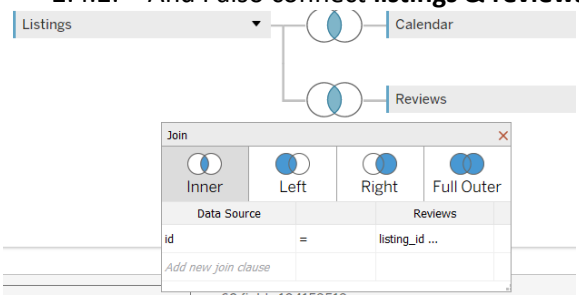




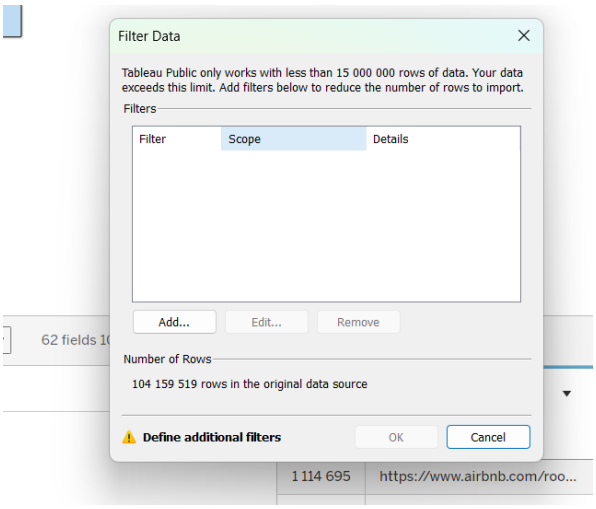
2.4.1. Aaand issue fixed.



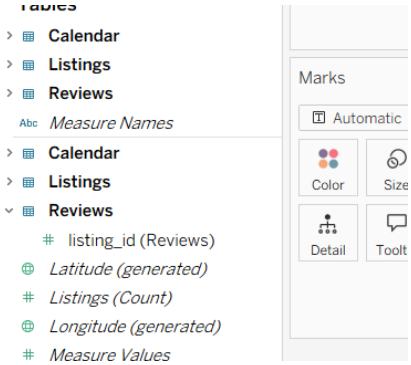
2.4.2. And I also connect **listings & reviews** by listings id.



2.4.3. Wellll, that I was not expecting… Well, at least not having so many rows. Will need to reduce them significantly.

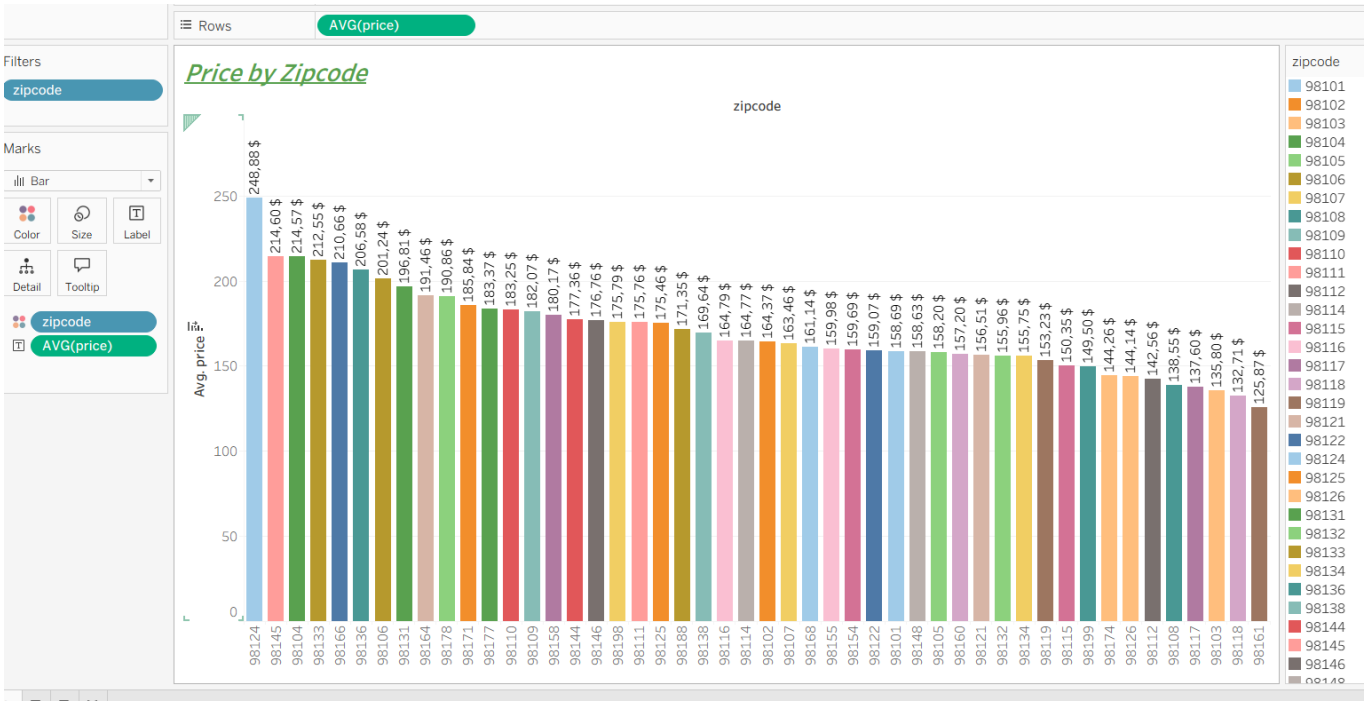


2.4.4. It took really a while, but I manage to reduce it down to Tableau limits – from 104 mill rows to around 15 mill. It works now and that's great. And this all will be about verifying based on the

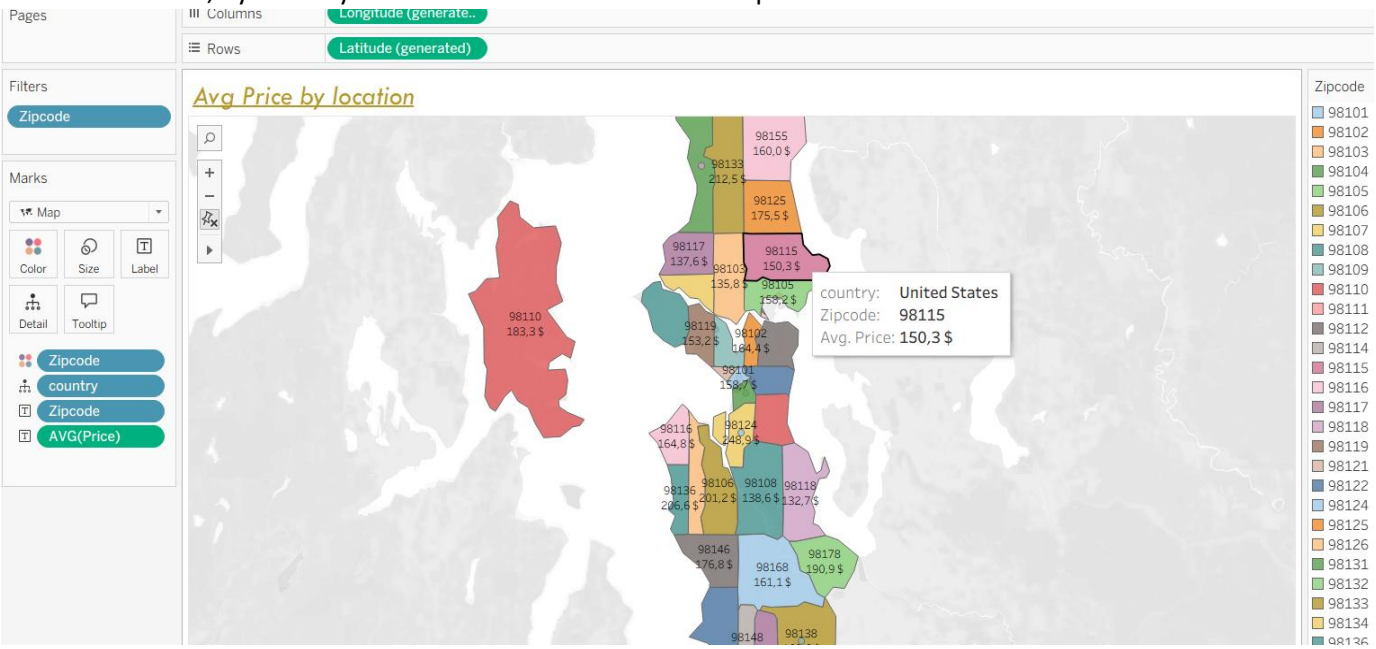


available data – where is the best to buy home and start Airnbn business.

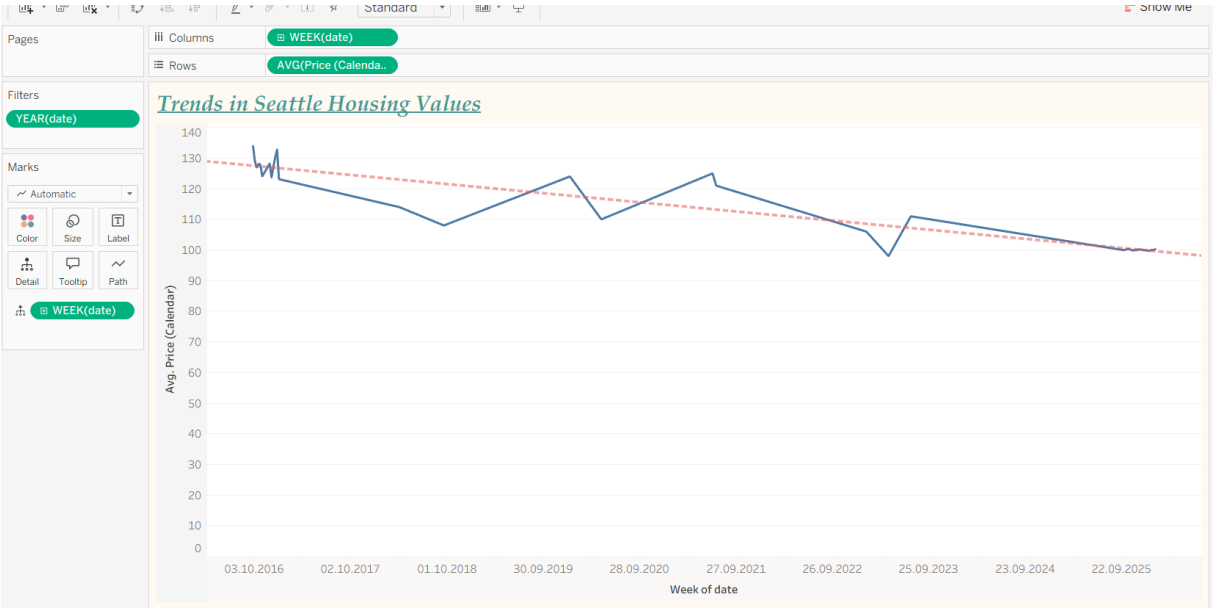
3. We will start from the Zip code and the average price for the accommodation. Let's also color it and sort. And with that it's the first visualization, it's done.



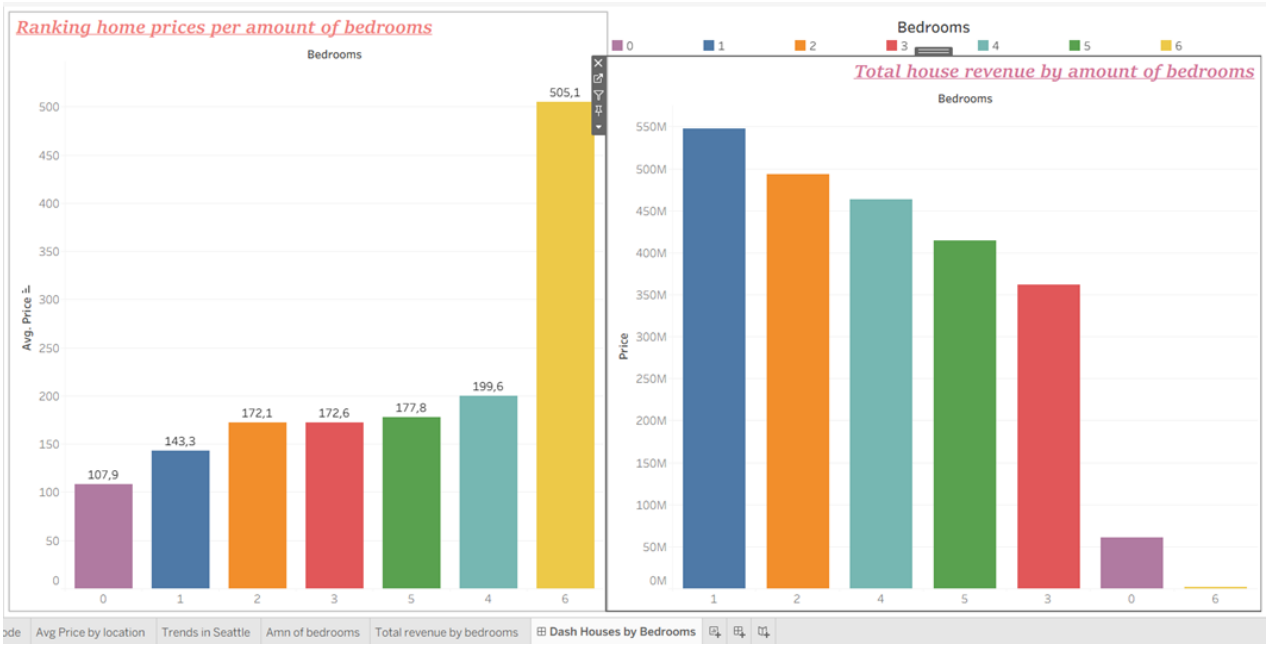
4. The next sheet – a **Map** style. The colors here match the colors by zip codes from the previous sheet, by the way. The data used here matches the previous visualization.



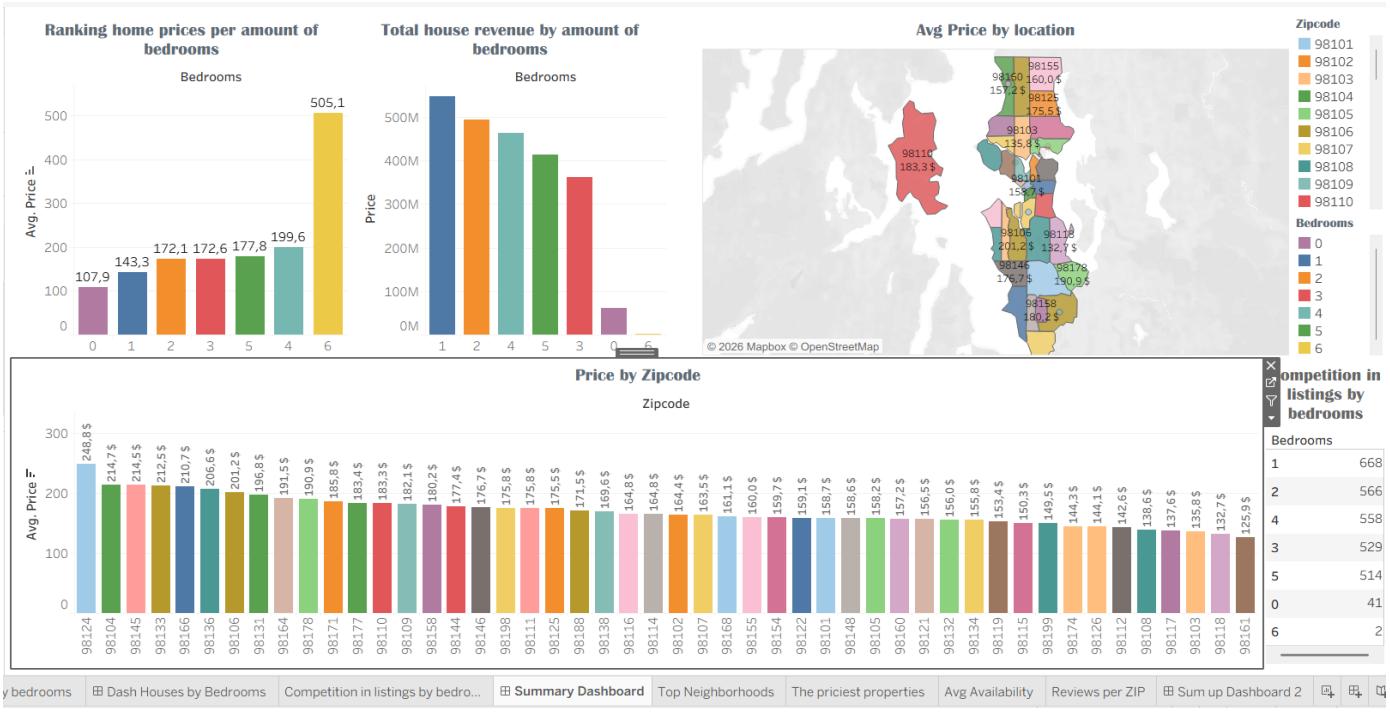
5. The next one will be showing the visualization by years and prices. Prices fluctuated early on, declined through 2019, peaked around 2021 amid post-pandemic demand, and have since shown a gradual cooling trend—ending near 100 in 2025—with a subtle downward trajectory highlighted by the trend line. This reflects real-world patterns of tech-driven growth followed by recent market stabilization due to higher interest rates and increased inventory.



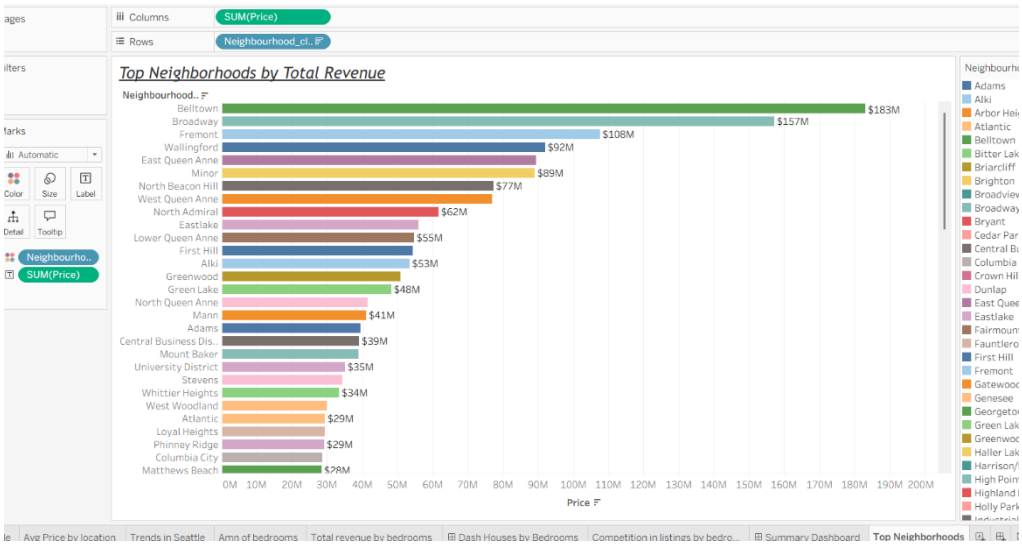
6. Next step – adding two worksheets where **Avg prices** & **Total revenue** for houses shown by amount of **Bedrooms**. Basically the same data used, so not to waste much time – one visual with created first Dashboard. Totally, 1 bedroom-houses are bringing the most efficiency by Avg price and total revenue. So, the user doesn't have to by 2-3, or even 6 bedrooms house to rent it out.



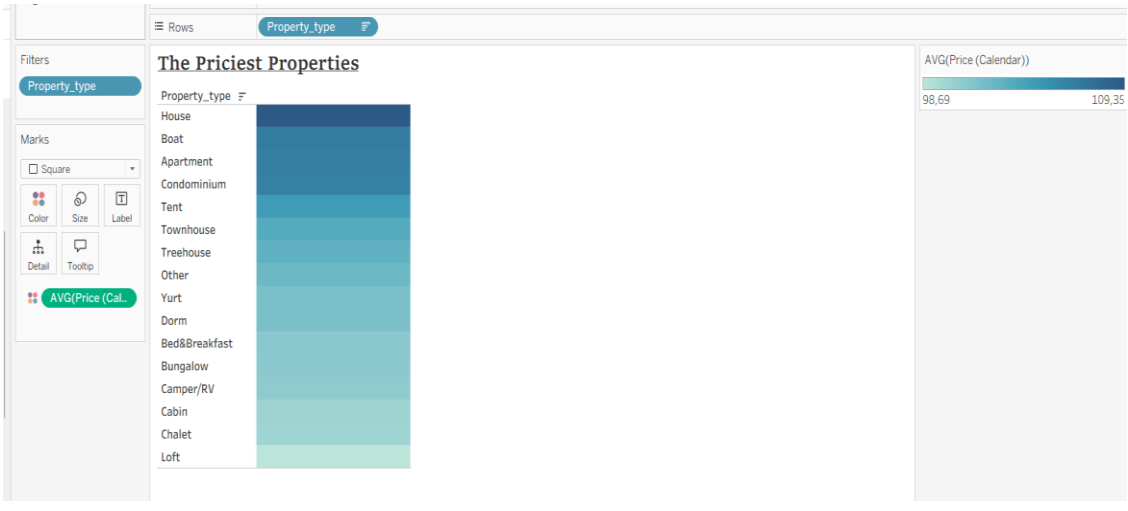
7. And with that is how the final **Dashboard** looks like!



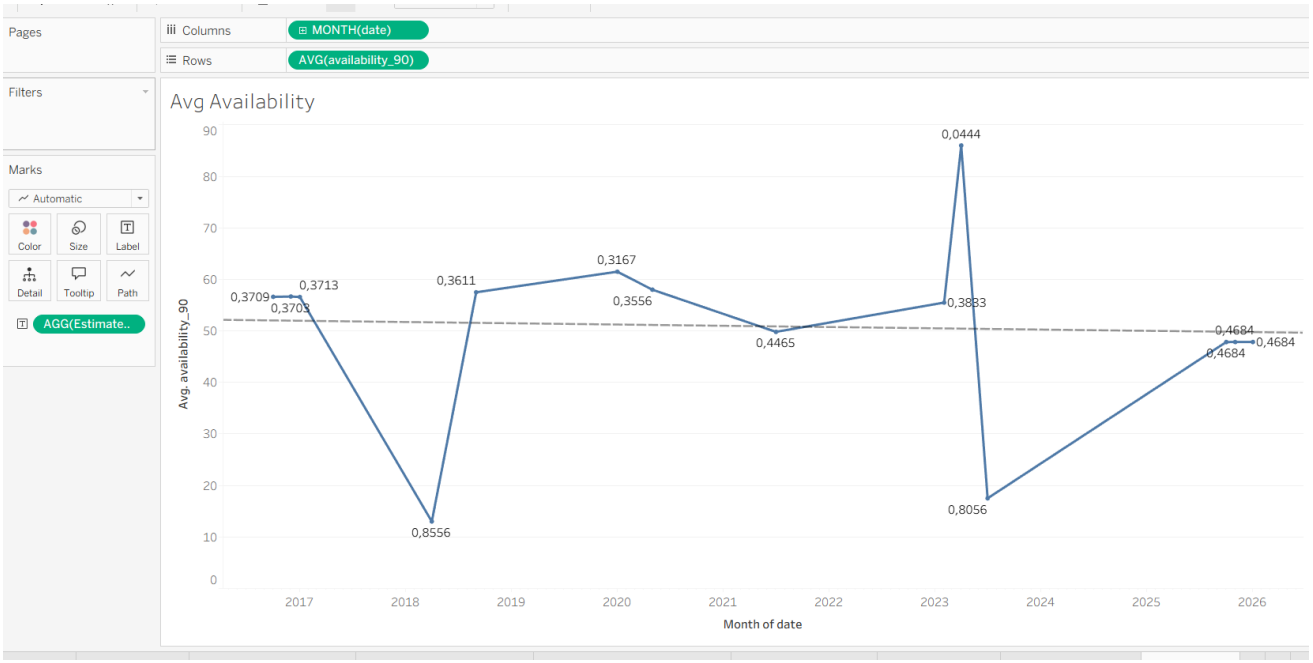
8. Additional sheet with **Top Neighborhoods**.



9. By that one now, we can see that the **Priciest** property (avg Prices) are Houses.



10. With this worksheet we can see **Avg Availabillity Over time.** Here the lowest value means that there were the highest Occupancy at that period of time.



11. And just like that – I finished the second **Dashboard** for this project!