Databases and Advanced Data Techniques Mid-Term Report

Exploring the New Housing Prices in Canada

1. Introduction

In this report, I will explore the changes over time in the builders' selling prices of new residential houses in Canada from January 1981 to October 2023.
I will summarize statistics from over 60,000 rows of data and provide average, minimum and maximum price points across different months, years, provinces and cities.

2. Dataset Overview

2.1 Data Source

I chose to investigate the New Housing Price Index - Monthly dataset, maintained by the Government of Canada.
The data is publicly available[1] and published by Statistics Canada, under the Open Government License - Canada[2].

2.2 Data License

The Open Government License - Canada states that:

“you are free to copy, modify, publish, translate, adapt, distribute or otherwise use the Information in any medium, mode or format for any lawful purpose.”

Therefore the data is considered to be open, and I am permitted to explore, modify and publish it for the midterm report purposes.

---

[1] New Housing Price Index - Monthly, (Updated on 2023-12-18): https://open.canada.ca/data/en/dataset/324befd1-893b-42e6-bece-6d30af3dd9f1
[2] Open Government License - Canada, (Updated on 2022-12-02): https://open.canada.ca/en/open-government-licence-canada

2.3 Data Reliability

The data is provided by an official government entity and was first made public in 2018. Given the license permissions and the accessibility from a government website, the public had over 5 years to dispute and scrutinize inaccuracies or discrepancies. Therefore I consider this data source reliable and accurate to thre best of my knowledge.

3. Dataset Analysis

For a deep assessment of the quality and content of the database, I used a framework called Rapids cuDF[3] and stored my progress in a Jupyter Lab document (please refer to item 9.1 from the appendix).

3.1 Dataset Quality

The dataset includes the information required for my task in a relatively clean manner. There are several issues that are immediately noticeable:

- **Inconsistency:** the missing values in the dataset are sometimes presented as “None” and sometimes as “NaN”.
- **Unclear Labels:** some of the dataset values are ambiguous, and are not explicitly explaining what they represent. For example, it is unclear what kind of data “VECTOR” stores or what is the meaning of having a “STATUS” E.
- **Merged Entities:** some distinct entities like city and province are merged together in the same field, for example “Victoria, British Columbia”.
- **Functional Dependency:** the non-key column of “price” is functionally dependant on the non-key column of “New housing price indexes”.

[1]:
```
%load_ext cudf.pandas
import pandas as pd

data = pd.read_csv("18100205.csv")
data.tail(3)
```

[1]:

	REF_DATE	GEO	DGUID	New housing price indexes	UOM	UOM_ID	SCALAR_FACTOR	SCALAR_ID	VECTOR	COORDINATE	VALUE	STATUS	SYMBOL	TERM
61677	2023-10	Victoria, British Columbia	2011S0503935	Total (house and land)	Index, 201612=100	347	units	0	v111955559	40.1	120.7	None	NaN	
61678	2023-10	Victoria, British Columbia	2011S0503935	House only	Index, 201612=100	347	units	0	v111955560	40.2	126.5	None	NaN	
61679	2023-10	Victoria, British Columbia	2011S0503935	Land only	Index, 201612=100	347	units	0	v111955561	40.3	109.8	E	NaN	

Figure 1. Initial reading of the raw dataset.

[3] Rapids cuDF documentation: https://docs.rapids.ai/api/cudf/stable/

3.2 Dataset Level of Detail

At first glance, the data seems to include many details beyond the requirements of my task. However, a deeper look reveals that 7 of the 15 columns store the same value in all their fields. Other columns, as mentioned in 3.1, store ambiguous details that may or may not be meaningful for my specific task.
Most importantly, the city name and province name are included, as well as different pricing for different types of properties. For the purpose of my report, the level of detail is satisfactory. However, it could have been better if the square footage of the properties was included, as well as the number of rooms and or listing versus selling price.

```
[2]: redundant_cols = []

     for i in list(data.columns):
         n_unique = len(data[i].unique())
         if n_unique == 1:
             redundant_cols.append(i)
             print("all fields in {} are: {}".format(i, data[i][0]))

     all fields in UOM are: Index, 201612=100
     all fields in UOM_ID are: 347
     all fields in SCALAR_FACTOR are: units
     all fields in SCALAR_ID are: 0
     all fields in SYMBOL are: <NA>
     all fields in TERMINATED are: <NA>
     all fields in DECIMALS are: 1
```

Figure 2. Columns that store a single value in all their fields.

3.3 Dataset Documentation

Statistics Canada provides very minimal documentation, but lists contact information for personal inquiries. I've reached out to obtain more information about the dataset and within 4 days, Statistics canada replied with detailed answers, additional resources, symbol mapping and even links to public data quality reports[4].

This information cannot be found in the official documentation, however, it is available through private inquiries. Please refer to item 9.2 from the appendix.

3.4 Dataset Interrelation

The dataset is quite tangled as many columns depend on one another, and many records contain repeating information. In addition, multiple scopes of geographical data are stored in one column (canada-wide records, provincial records and municipal records all appear under the same GEO column), which is not ideal for my task.

---

[4] New Housing Price Index Detailed information for November 2023, (Updated on 2023-12-15): https://www23.statcan.gc.ca/imdb/p2SV.pl?Function=getSurvey&Id=1531535

To avoid dependencies and repetition, I've transformed the Data Frame such that instead of storing 3 records about one geographical location on a specific date, as shown in Figure 3.

DATE	GEO	TYPE	VALUE
x1_date	x1_location	House and Land	a1
x1_date	x1_location	House	b1
x1_date	x1_location	Land	c1
x2_date	x2_location	House and Land	a2
...	...	...	...

Figure 3. Pseudo-structure of the raw dataset, before the transformation.

The transformed Data Frame merged the type and value columns, showcasing the price for each type of property on the same record, as shown in Figure 4.
This allows reducing each data point into a single record rather than three, and finding an easier way to uniquely identify each record based on the location at a given data (which shortly will be distilled into an ID bumber).

DATE	GEO	House and Land Value	House Value	Land Value
1x_date	1x_location	a1	b1	c1
x2_date	x2_location	a2	b2	c2
...	...	...	...	...

Figure 4. Pseudo-structure of the transformed dataset.

3.5 Dataset Discoverability

The dataset lists a range of keywords and subjects that may provide the webpage a higher discoverability rating. However a simple Google search of the term "New housing price index, monthly" does not present this webpage in the top entries.
This dataset is easy to find when searching directly in the Government site, however, it is not as discoverable for external search engines.

4. Motivations

4.1 Motivations

I have several reasons for choosing this dataset for my report:

- **Personal Interest:** I am Canadian, and I am curious to investigate the state of my country's housing market across different provinces, cities and years.
- **Lack of Comparison:** The dataset includes entries on a county level, province level and city level, all stacked one after the other for each month at a time. This type of order prevents users from comparing any entity to itself across different months of the year, as well as two entities across a continuous period of time.
  For example, users who view the housing pricing in British Columbia in July 2022 must scroll through hundreds of other entries before they can read the housing pricing for British Columbia in August 2022. I am aiming to solve this issue by re-ordering the data to minimize redundancy.
- **Lack of Summarization:** When viewing enormous amounts of data, it's difficult to understand the average, minimum or maximum values of the records it holds. I am aiming to solve this by providing custom summarization of the data across different dates and locations for the end user.

4.2 Questions

I have several questions regarding the content of the dataset:

1. What are the average values of properties in each Canadian city for the past 5 years?
2. Which province has the least expensive new property prices in the past 5 years?
3. Which province has the most expensive new property house and land prices in the past 5 years?
4. What are the top 3 cities that experienced the highest rate of new property price change in the past 5 years?
5. What are the top 3 cities that experienced the least rate of new property price change in the past 5 years?

5. Data Modelling

5.1 Selecting a Subset of the Data

5.1.1 City, Province, Country Scopes

In my model, I chose to focus on a subset of the data, describing the scope of cities. That way, data regarding the other two scopes can be easily concluded; a combination of all the cities in a province represents provincial data, and a combination of all the provinces represents country-wide data.

5.1.2 Unnecessary Columns

I chose to remove columns that either store the same value in all their fields, or do not provide meaningful information regarding the task at hand.

5.1.2.1 Redundant Columns

As shown in section 3.2, there are 7 columns that store a single value that repeats across all rows: UOM, UOM_ID, SCALAR_FACTOR, SCALAR_ID, SYMBOL, TERMINATED, and DECIMALS.

5.1.2.2 Ambiguous Columns

Columns with ambiguous labels and changing values, as shown in section 3.1, require a deeper analysis to ensure no valuable information is removed from my model. I reached out to Statistics Canada to clarify what these columns store, and the following information was obtained:

- **DGUID:** stores a unique, location-based identification, refers to other datasets of Statistics Canada.
- **VECTOR:** stores a unique variable length reference code that describes a time series of data points. It allows users to easily compare records[5] from different datasets by searching for a "vector" reference.
- **COORDINATE:** stores a concatenation of the member ID values, helps identify unique time series of data points.
- **STATUS:** represents the data quality and stores a number of table symbols[6] that represent different levels of certainty. The symbol "E" for example, represents data that should be used with caution. There are two status symbols that I am concerned with: "F"

[5] Search by vector - Statistics Canada: https://www150.statcan.gc.ca/t1/tbl1/en/sbv.action
[6] Standard Table Symbols - Statistics Canada: https://www.statcan.gc.ca/en/concepts/definitions/guide-symbol

that refers to data too unreliable to be published and “...” that refers to data that is not applicable. Any record with this status must be removed from my dataset from a reliability standpoint.

```
data = pd.read_csv("18100205.csv")
data['STATUS'].value_counts()

E     17226
..    10746
x        80
Name: STATUS, dtype: int64
```

Figure 5. Occurrences of different STATUS symbols.

As the first three columns help with searching across other Statistics Canada datasets, they are irrelevant for the purpose of my report. In addition, since no symbols that present a substantial accuracy concern were found in the STATUS column, as shown in figure 5, I chose to remove it as well.

5.2 Data-Specific Modeling Complexity

There is a complexity associated with the provincial jurisdiction of the city Ottawa-Gatineau. Some parts of the city are under the provincial jurisdiction of Ontario and some are under the provincial jurisdiction of Quebec. It creates a situation where one city affects the cardinality relationship of the entire table, such that many cities are located in many provinces.

```
[7]: data[data['GEO']=='Ottawa-Gatineau, Quebec part, Ontario/Quebec'].iloc[0:1]
```

[7]:

	REF_DATE	GEO	DGUID	New housing price indexes	UOM	UOM_ID	SCALAR_FACTOR	SCALAR_ID	VECTOR	COORDINATE	VALUE	STATUS	SYMBOL
45	1981-01	Ottawa-Gatineau, Quebec part, Ontario/Quebec	2011S050324505	Total (house and land)	Index, 201612=100	347	units	0	v111955487	16.1	None	..	NaN

```
[8]: data[data['GEO']=='Ottawa-Gatineau, Ontario part, Ontario/Quebec'].iloc[0:1]
```

[8]:

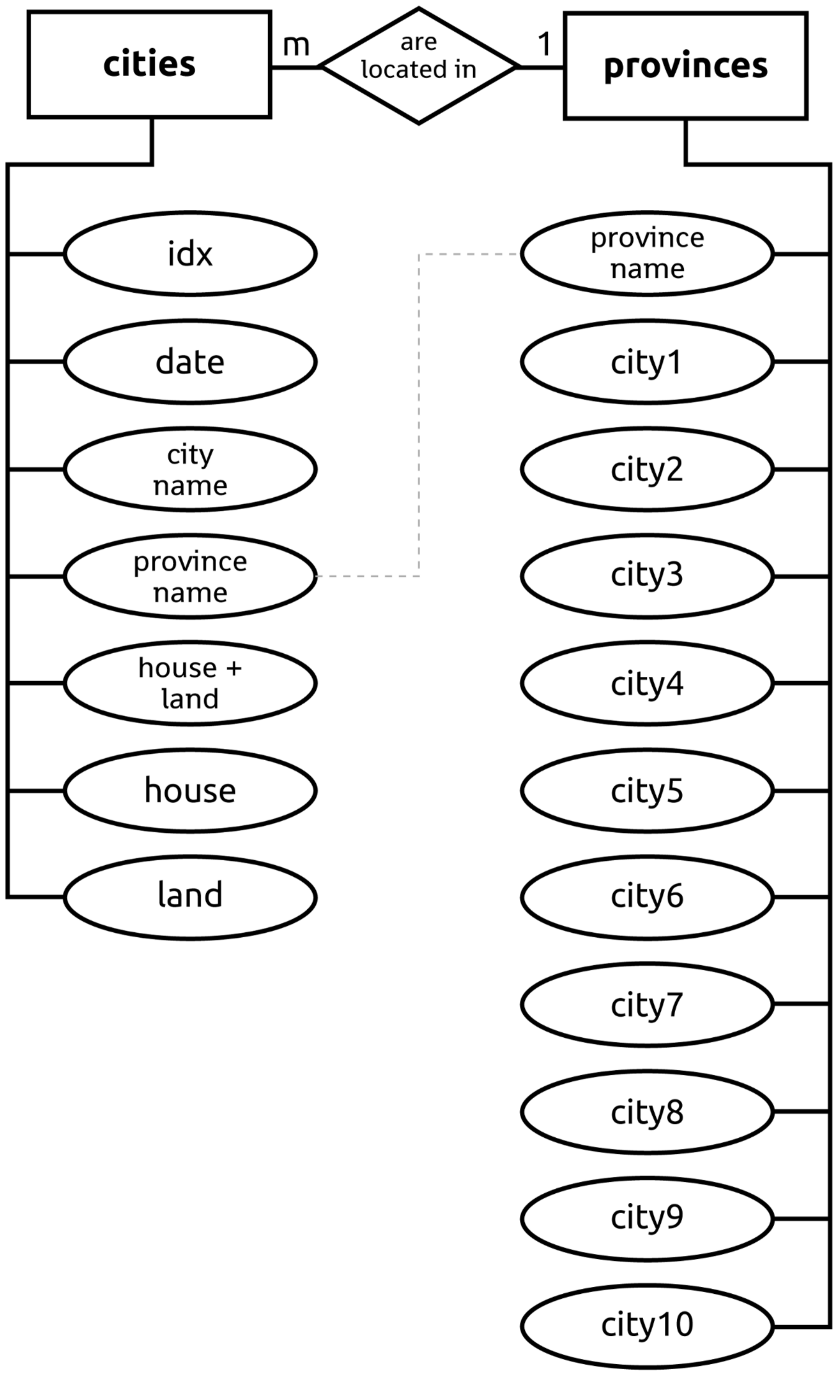
	REF_DATE	GEO	DGUID	New housing price indexes	UOM	UOM_ID	SCALAR_FACTOR	SCALAR_ID	VECTOR	COORDINATE	VALUE	STATUS	SYMBOL
51	1981-01	Ottawa-Gatineau, Ontario part, Ontario/Quebec	2011S050335505	Total (house and land)	Index, 201612=100	347	units	0	v111955493	18.1	32.6	None	NaN

Figure 6. The Modelling Complexity of Ottawa-Gatineau.

I chose to address this issue differently from Statistics Canada. Instead of creating unique province instances of “Ontario part” and “Quebec part”, I chose to create unique city instances of “Ottawa-Gatineau Ontario part”, and “Ottawa-Gatineau Quebec” part.
That way, the proper cardinality of many cities are located in one province is preserved, and each city instance will be stored in the appropriate province - Ontario or Quebec.

5.3 Entity Relationship Model

Based on section 5.1 and 5.2, my Entity Relationship Model can be represented with the following diagram. Where the cardinality lists multiple cities for one province, and the primary key in the province table serves as the foreign key in the cities table.



5.3 Dataset Tables

The database will include two tables to represent two distinct entities: cities and provinces.

5.3.1 Cities Table

The cities table will store columns of: unique indices, dates, cities, provinces, house and land price, house price, land price.
The primary key in the cities table is an incrementing index value that uniquely identifies a geographic location at a given date.
The province name serves as a foreign key that references the provinces table.

5.3.2 Provinces Table

The provinces table will store columns of: province name, as well as a list of cities it has. At this iteration of the project, the provinces table will only provide information that can be extracted from the cities table. However, in future iteration, an array of new columns can be added, describing different aspects of the province. For example: population, economic growth, birth rate, etc.
The primary key in the provinces table is the province name that identifies a geographic location. The province name also serves as a foreign key in the cities table.

```
[16]: cities['province'][cities['date']=='1981-01-01'].value_counts()

[16]: Ontario                      10
      Quebec                        5
      British Columbia              3
      Alberta                       2
      Saskatchewan                  2
      Manitoba                      1
      New Brunswick                 1
      Newfoundland and Labrador     1
      Nova Scotia                   1
      Prince Edward Island          1
      Name: province, dtype: int64
```

Figure 7. Finding how many city columns are required in the provinces table.

5.4 Normal Forms

5.4.1 Cities Table Normalization

The cities table was normalized according to the requirements of 3NF, with the following considerations:

- cities complies the **First Normal Form**, as the table is a relation that stores a single scalar value. No arrays, lists or complex data structures are stored in the fields, only single decimal numbers and variable length strings. Each records describes a unique datapoint and therefore 1NF is honored.
- cities also complies with the **Seconds Normal Form**, as the non-key attributes of date, city name, house and land price, house price and land price are irreductibly dependant on the primary key index. The only dependant replation is between the city name and

the province name, however, since the province name is a foreign key - it does not violate the requiremenents of 2NF.
- cities complied with the **Third Normal Form**, as there is no transitive dependance implied by any of the non-key columns.
- Lastly, cities does not comply with the **Fourth Normal Form**, as even though there are no functional dependencies between columns, the structure of the data includes a large amount of repetition. Records with the same dates, or city and province names repeat many times. The reason why I did not proceed with normalizing this table further is because it would require interfaces and techniques that go beyond the scope of the Coursera lab environment or the mid-term report instructions.

5.4.2 Provinces Table Normalization

The provinces table was normalized according to the requirements of 4NF, with the following considerations:
- Each province name is uniquely identifiable, and each record stores scalar information about each province, which complies with 1NF.
- There is no functional or transitive dependency between the non-keys of the table, and therefore it complies with 2NF and 3NF.
- Lastly, there is no structural repetition or multi-valued dependencies in the table. Each province name and each city name repeaats precisely once in the table. Therefore it complies with 4NF.

5.4.3 Ideal Normalization

Ideally, I would have set each city as an entity of its own, storing it in a separate table such that:

Victoria Table

date	house + land price	house price	land price
2023-01-01	120	115	110
2023-02-01	125	120	115

Provinces Table

city	province
Vancouver	British Columbia

However, such structure would have resulted in 27 tables of individual cities and 1 table of provinces, which would have been very difficult to model in an E/R Diagram or within the MYSQL terminal of the mid-term Coursera working environment.
Therefore I decided to persue a setisfactory normalization rather then an ideal one.

6. Database Development

6.1 Database Structure

6.1.1 Create The Database

To create my database, I first checked the names of the existing databases in the schema. As they were named in a snake case convention, I chose to implement it in my database as well, naming it canadian_housing.

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)

mysql> CREATE DATABASE canadian_housing;
Query OK, 1 row affected (0.02 sec)

mysql> USE canadian_housing;
Database changed
```

Figure 8. Creating the canadian_housing Database.

6.1.2 Create The Provinces Table

The first table in the canadian_housing dataset maps the names of provinces to all the cities they contain.

```
mysql> CREATE TABLE provinces (
    -> province_name VARCHAR(100) NOT NULL,
    -> city1 VARCHAR(100),
    -> city2 VARCHAR(100),
    -> city3 VARCHAR(100),
    -> city4 VARCHAR(100),
    -> city5 VARCHAR(100),
    -> city6 VARCHAR(100),
    -> city7 VARCHAR(100),
    -> city8 VARCHAR(100),
    -> city9 VARCHAR(100),
    -> city10 VARCHAR(100),
    -> PRIMARY KEY (province_name)
    -> );
Query OK, 0 rows affected (0.15 sec)
```

Figure 9. Creating the provinces table.

6.1.3 Create The Cities Table

The cities table stores information about the new housing prices in different cities. The primary key was set to an incrementing integer, as the names of cities were not unique, and repeat for every month of every year.

```
mysql> create TABLE cities (
    -> idx INT NOT NULL,
    -> date DATE NOT NULL,
    -> city_name VARCHAR(100) NOT NULL,
    -> province_name VARCHAR(100) NOT NULL,
    -> house_and_land_price DECIMAL,
    -> house_price DECIMAL,
    -> land_price DECIMAL,
    -> PRIMARY KEY (idx),
    -> FOREIGN KEY (province_name)
    -> REFERENCES provinces(province_name)
    -> );
Query OK, 0 rows affected (0.17 sec)
```

Figure 10. Creating the cities table.

6.2 Data Insertion

To insert data into the housing_price database, I first transformed the original dataset in a Python Jupyter Lab environment, exported it to CSV and loaded all the table records using the following commands:

```
mysql> LOAD DATA INFILE '/home/coder/project/data/provinces.csv' REPLACE INTO TABLE provinces FIELDS TERMINATED
 BY ',' OPTIONALLY ENCLOSED BY '"' ESCAPED BY '"';
Query OK, 12 rows affected (0.02 sec)
Records: 12  Deleted: 0  Skipped: 0  Warnings: 0

mysql> LOAD DATA INFILE '/home/coder/project/data/cities.csv' REPLACE INTO TABLE cities FIELDS TERMINATED BY ',
' OPTIONALLY ENCLOSED BY '"' ESCAPED BY '"';
Query OK, 13878 rows affected, 29645 warnings (0.92 sec)
Records: 13878  Deleted: 0  Skipped: 0  Warnings: 29645
```

Figure 11. Inserting data into cities and provinces.

6.3 User Privileges

As my web application is intended to be publicly available, I’ve limited the access of users to parts of the database that should not concern them.

6.3.1 Creating a User

I’ve created a user with SELECT privileges, as the only reason why the database should be accessed from the browser application level is to read the data. Any insertion or structural privileges should not be granted to the end-user.

```
mysql> CREATE USER viewer
    -> IDENTIFIED BY '7f0j@Ht';
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> GRANT SELECT ON cities TO viewer;
Query OK, 0 rows affected (0.03 sec)
```

Figures 12, 13. Create public user and set their privileges.

6.3.2 Setting Database User in Node.js

An additional step in creating a user, is specifying it in the application. Since I intend to display my database in a Node.js web application (please refer to section 7), I'll need to add the following lines of code to restrict user access to my database.

```
const db = mysql.createConnection({
    host: "localhost",
    user: "viewer",
    password: "7f0j@Ht",
    database: "canadian_housing"
});
```

Figure 14. Specify the public username when connecting to the database.

6.3.3 Form Input Limitations

In addition to restricting the privileges, I chose to limit the form input fields to a set of predefined values. This measure helps prevent malicious or accidental use of the form input fields prior to querying the database.

6.4 Reflection of Data

I think my database reflects the data in a more user-friendly way than the raw version. Any redundant and ambiguous columns were removed, records with high level of repetition were merged and the data wes distilled to only present details that are meaningful to the end user. What worked well was taking data that was designed for professional statisticians, and adapting it to suit individuals with little to no experience in the realm of data science. This simplification allowed to easily identify price changes between different cities and compare the pricing of each type of property across a specific timestamp and location.
What didn't work well is the instability to achieve the ideal normalization that I've planed and combine it with the requirements of the mid-term report (mostly, the complexity of designing an ideal E/R Diagram for 28 tables).

6.4 Answers to Queries

1. What are the average values of properties in each Canadian city for the past 5 years?

```
mysql> SELECT city_name, province_name, AVG(house_and_land_price) AS house_and_land, AVG(house_price) AS house, AVG(land_price) AS land FROM
cities WHERE date>'2018-01-01' GROUP BY city_name, province_name LIMIT 8;
+------------------------------------+---------------------------+---------------+----------+----------+
| city_name                          | province_name             | house_and_land| house    | land     |
+------------------------------------+---------------------------+---------------+----------+----------+
| St. John's                         | Newfoundland and Labrador |      101.2464 | 101.7681 | 100.7826 |
| Charlottetown                      | Prince Edward Island      |      111.0435 | 112.9565 | 102.1449 |
| Halifax                            | Nova Scotia               |      111.3913 | 114.1159 | 102.3623 |
| Saint John Fredericton and Moncton | New Brunswick             |      109.2464 | 109.6957 | 101.7826 |
| Québec                             | Quebec                    |      111.7826 | 114.8986 | 104.5942 |
| Sherbrooke                         | Quebec                    |      105.6667 | 105.8696 | 103.3913 |
| Trois-Rivières                     | Quebec                    |      106.4348 | 107.4783 | 102.5362 |
| Montréal                           | Quebec                    |      127.1304 | 131.9565 | 113.9420 |
+------------------------------------+---------------------------+---------------+----------+----------+
8 rows in set (0.01 sec)
```

Figure 15. Average value of city properties in the past 5 years.

2. Which province has the least expensive new property prices in the past 5 years? Saskatchewan (please refer to figure 16).
3. Which province has the most expensive new property house and land prices in the past 5 years? Manitoba.

```
mysql> SELECT province_name, AVG(house_and_land_price) AS house_and_land, AVG(house_price) AS house, AVG(land_price) AS land FROM cities WHER
E date>'2018-01-01' GROUP BY province_name ORDER BY house_and_land DESC;
+---------------------------+----------------+----------+----------+
| province_name             | house_and_land | house    | land     |
+---------------------------+----------------+----------+----------+
| Manitoba                  |       119.9130 | 126.3043 | 105.8986 |
| Ontario                   |       119.4928 | 121.2290 | 113.5014 |
| British Columbia          |       113.5411 | 114.8986 | 109.3285 |
| Quebec                    |       112.2145 | 114.5942 | 105.5275 |
| Nova Scotia               |       111.3913 | 114.1159 | 102.3623 |
| Prince Edward Island      |       111.0435 | 112.9565 | 102.1449 |
| New Brunswick             |       109.2464 | 109.6957 | 101.7826 |
| Alberta                   |       105.1014 | 106.1884 | 102.1522 |
| Newfoundland and Labrador |       101.2464 | 101.7681 | 100.7826 |
| Saskatchewan              |        99.8261 | 100.0362 | 100.0290 |
+---------------------------+----------------+----------+----------+
10 rows in set (0.02 sec)
```

Figure 16. Most and least expensive provincial housing prices in the past 5 years.

4. What are the top 3 cities that experienced the highest rate of new property price change in the past 5 years? The Ontario part of Ottawa-Gatineau, Kitchener-Cambridge-Waterloo, and Montréal.

```
mysql> SELECT city_name, province_name, MAX(house_and_land_price) - MIN(house_and_land_price) AS house_and_land_change, MAX(house_price) - MI
N(house_price) AS house_change, MAX(land_price) - MIN(land_price) AS land_change FROM cities WHERE date>'2018-01-01' GROUP BY city_name, prov
ince_name ORDER BY house_and_land_change DESC LIMIT 3;
+-----------------------------+---------------+-----------------------+--------------+-------------+
| city_name                   | province_name | house_and_land_change | house_change | land_change |
+-----------------------------+---------------+-----------------------+--------------+-------------+
| Ottawa-Gatineau Ontario part | Ontario      |                    71 |           85 |          34 |
| Kitchener-Cambridge-Waterloo | Ontario      |                    56 |           52 |          58 |
| Montréal                    | Quebec        |                    51 |           59 |          26 |
+-----------------------------+---------------+-----------------------+--------------+-------------+
3 rows in set (0.00 sec)
```

Figure 17. Top 3 cities with the most unstable housing prices.

5. What are the top 3 cities that experienced the least rate of new property price change in the past 5 years? St. John’s, Trois-Rivières, and Regina.

```
mysql> SELECT city_name, province_name, MAX(house_and_land_price) - MIN(house_and_land_price) AS house_and_land_change, MAX(house_price) - MI
N(house_price) AS house_change, MAX(land_price) - MIN(land_price) AS land_change FROM cities WHERE date>'2018-01-01' GROUP BY city_name, prov
ince_name ORDER BY house_and_land_change ASC LIMIT 3;
+----------------+------------------------+-----------------------+--------------+-------------+
| city_name      | province_name          | house_and_land_change | house_change | land_change |
+----------------+------------------------+-----------------------+--------------+-------------+
| St. John's     | Newfoundland and Labrador |                 10 |           14 |           2 |
| Trois-Rivières | Quebec                 |                    10 |           12 |           4 |
| Regina         | Saskatchewan           |                    11 |           14 |           4 |
+----------------+------------------------+-----------------------+--------------+-------------+
3 rows in set (0.02 sec)
```

Figure 18. Top 3 cities with the most stable housing prices.

7. Web Application Development

I developed a web application that allows users to interact with my database and customize queries. Users may search for average, minimum and maximum pricing values for specific provinces and periods of time, and the app will display the requested data.

7.1 Main Page

The following screenshot is taken as soon as the app opens in the browser, prior to any user interactions. The form fields were pre-set to return the average data from the latest year across all provinces.

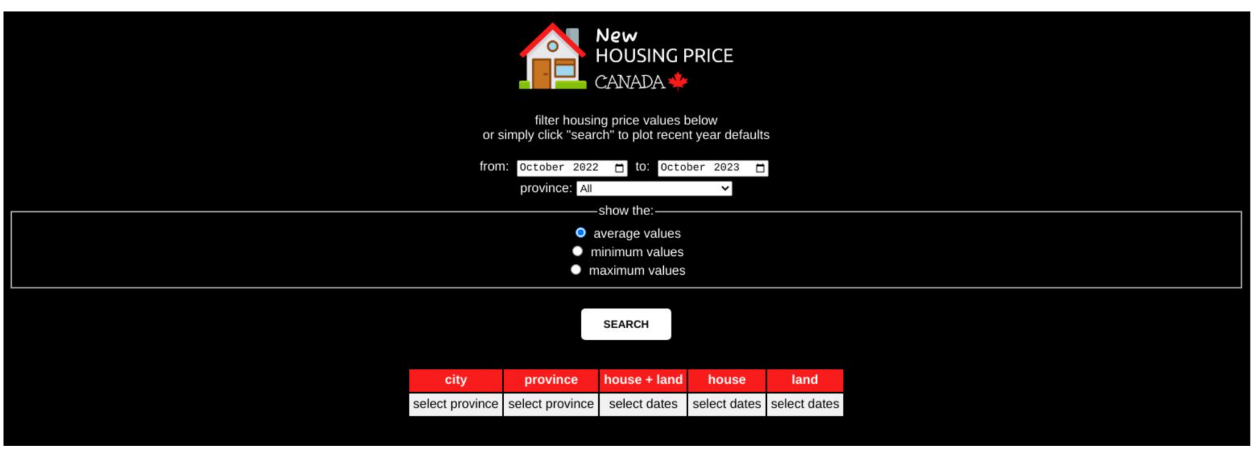


Figure 19. Web application main page, prior to querying from the database.

7.2 Provincial Search

When a specific province is searched, the web application only returns data about its cities.

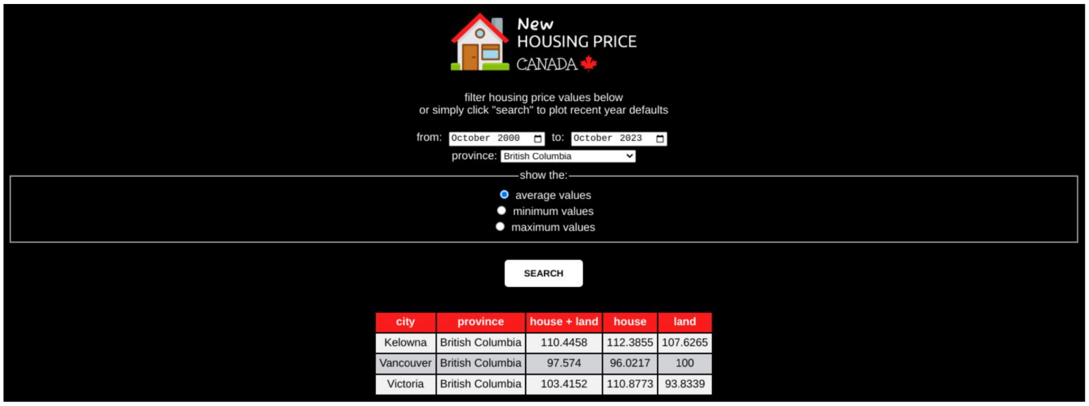


Figure 20. Province-specific search for average prices in British Columbia cities.

7.3 Canada-Wide Search

Canada-wide searches that span across all provinces return a large collection of rows, which may be difficult to read and analyze. To solve this problem, I've implemented several styling techniques that allow better readability:

- **Contrast:** I've added high levels of contrast between the colour elements of the app. I chose distinct colours like red, yellow, black and white to better separate elements from one another.
- **Even Row Constraints:** every even row of data was set to a slightly darker background colour which visually separates one record from the next.
- **Hover Row Highlight:** when users hover above a row with the mouse - the background colour of the row turns bright yellow which easily separates the record from the rest.

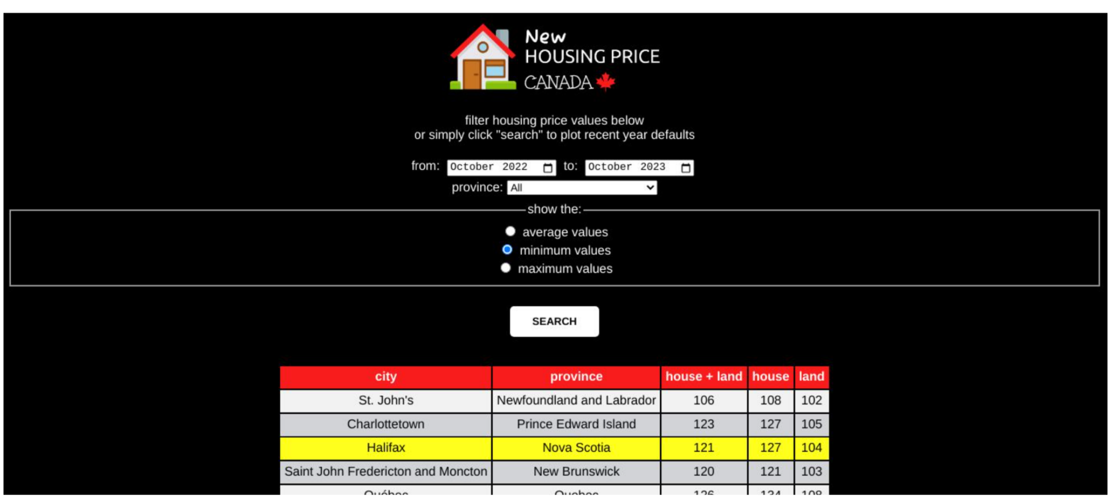


Figure 21. Canada-wide search for minimum housing prices of different cities the last year.

8. References

[1] New Housing Price Index - Monthly, (Updated on 2023-12-18): https://open.canada.ca/data/en/dataset/324befd1-893b-42e6-bece-6d30af3dd9f1

[2] Open Government License - Canada, (Updated on 2022-12-02): https://open.canada.ca/en/open-government-licence-canada

[3] RAPIDS cuDF documentation: https://docs.rapids.ai/api/cudf/stable/

[4] New Housing Price Index Detailed information for November 2023, (Updated on 2023-12-15): https://www23.statcan.gc.ca/imdb/p2SV.pl?Function=getSurvey&Id=1531535

[5] Search by vector - Statistics Canada: https://www150.statcan.gc.ca/t1/tbl1/en/sbv.action

[6] Standard Table Symbols - Statistics Canada: https://www.statcan.gc.ca/en/concepts/definitions/guide-symbol

[7] Web App Logo Icon was provided by Flaticon under the Flaticon license, for personal and commercial use with attribution: https://www.flaticon.com/free-icon/house_619032

9. Appendix

There are 2 additional files, that accommodate my report:

9.1 data_preprocessing.html

A detailed Jupyter Lab document, exported to an HTML format, that demonstrated the data transformation steps and the full workflow specified in section 3.

9.2 email_correspondence.pdf

A document that includes my email to Statistics Canada, as well as their reply. Please note, the names and email addresses were redacted to avoid revealing the identities of the people involved.