

# Overview

This project originated on the curiosity of newly hired grads looking for more information about the changing scope of real estate in relation to their own earning power in the near future: how are things changing? Are prices hovering at an attainable range, or fluctuating outside the realm of possibility? Many young people, emerging from school, would like the ability to evaluate the current climate of real estate before pushing towards a home in any city in the Surrey. Our team has gathered various data surrounding the city of Surrey to investigate what's changing about the city, and whether those changes bode well for potential buyers with modest salaries.

## Introduction

Our visualizations are meant to shed some light on the underlying patterns in real estate for various neighborhoods found in Surrey. By examining data provided by both Surrey and the Greater Vancouver area, we can hopefully gain some insights to when someone might have luck finding a home, and how much it will end up costing them proportionally to their finances.

Young jobseekers and new graduates are the primary audience for our visualizations. These people are typically between 22 and 27, with a post-secondary education, looking to someday purchase a home in the north, south, or central regions of Surrey. With their youth and inexperience in the workforce, introductory salaries ranging from \$40,000 to \$65,000 CAD annually is expected -- certainly not in the territory of hedge fund managers or venture capitalists. Though lifestyles in this age bracket are subject to change, it is assumed that the number of children for this primary audience will be between 0 and 3.

For the secondary audience, realtors and investors aged 30 to 50 with an interest in Surrey properties for the purpose of profit and trends before residence. While the information is not of the highest fidelity, there is no reason that more educated audiences won't be able to read and enjoy the graphs.

## Problem and Domain Questions

- Is now a good time to purchase a home?
  - If so, which region and which type of house to purchase?
  - If not, when is a good time?
- How well is the median salary of earners in the city doing in comparison to the price of property?
  - Is the median salary changing?
  - Is the price of property changing?
  - If either are changing, will property remain realistically affordable for the average worker in the Greater Vancouver area?

## The Data

All of our data comes from official open data that have been released by the city of Surrey, where possible (income data was gathered from the Greater Vancouver region). All of our data utilizes time as a reference so that viewers can see trends. Other dimensions used include values such as active listings, population, House Price Index, and median income.

Since the data that Surrey provides are extensive and detailed, we had to clean up the data. Originally, the HPI and income are from two separate sources. We combined them together so that these two pieces of data can be compared together across 3 or 4 columns. Furthermore, both Listings and population had more specific regions such as Guildford and Fleetwood but we ended up compressing these regions together into North (Delta), Central, and South (Whiterock).

Cleaning happened in 2 ways: either the incoming data had its cells pruned and extracted (population, income), or the data was manually entered to a CSV sheet from a PDF (housing price index). With the variety of data formats, sometimes manual data entry was necessary, as it's very difficult to automate the scanning of long PDFs to find 3 values.

# Visualization Design

Overall, there are 3 total sets of data we've chosen to examine in our graphs:

- Number of available property listings across North Surrey (Delta), Central Surrey, and South Surrey (Whiterock)
- Changes in population year over year in these 3 Surrey regions
- Total median income for Greater Vancouver vs. Surrey Housing Price Index (HPI)

Originally, the first set of data was to be mapped to one graph: 3 different types of listings across 3 regions, which we decided to map to a line graph, categorizing both region and listing type with colour. This decision was made to reveal the *trend* across listings: are more listings popping up yearly, or are they in a decline? In order to also expose the details of the active listings, we've also opted for a dotted plot on the lines that reveal the specific data points to make the graph more readable.

Unfortunately, the first iteration graph was too crowded: 9 lines of different colours do not effectively sit on the same graph. *Filtering* was implemented to reduce the noise between lines, but it introduced issues with memory. By hiding other lines, it was difficult to recall what values the other lines had, so the decision was made to split the graph into 3 using small multiple views. Each graph measured only 1 type of property listing (apartment, townhouse, detached home), reducing the number of colours and lines per graph to 3. Having the data split across these smaller views made the information much easier to consume, as the visual noise was reduced.

A similar situation arose for the population graph. By using data from 2004 to 2014, we wanted to explore *trends* again in the 3 regions of Surrey. For each of the 3 regions, how have the overall populations changed? Will the trend in population appear similar at all to the number of active listings, or reveal a correlation? This solution also leverages details on demand via hovers, exposing explicit values on mouse hover over each point on the lines.

For the HPI-Income graph, we wanted to expose the difference between Housing Price Index and Income, as pairs. With this in mind, a paired bar graph was implemented. By placing the values next to each other, examining the difference between the values is simple. Time is used on the x-axis to also display the longer term behaviour of both values. This ended up exposing the startling changes in the housing price index, compared to the relatively stagnant median annual income.

Through the graphs, the North (blue), Central (green), and South (orange) region data is mapped to the same colours. Scale is maintained as a constant between the active listing graphs, in order to expose the differences between each listing type.

Interactions in our visualization are limited to mouse over events. By hovering over any particular value (circular line point or bar graph element), you can view its value while brushing the value itself to a different colour.

## Further Work

We managed to implement the base features showing our visualizations in a static way with some minor interactions. What would have been the next step would be some more advanced brushing and more comprehensive comparison tools between values. While there is continuity between some of our visualizations, currently there is little interaction between the data - it would be quite beneficial to be able to select a data point and compare it to other values across the other graphs on the page.

Being able to unify the graphs further through *time* would also be beneficial: brushing columns in each of the graphs based on a year selection and retrieving statistics from those years in a “summary” section.

## Conclusion

The visualizations our team managed to produce some interesting trends. Active listings tend to pulse in the summer, with most listings closing by the beginning of the next year. In terms of the most popular type of listing by far turned out to be detached homes in Central Surrey, capping out at almost 1400 in the summer of 2010. If anyone was attempting to buy a home, they'd have the widest range of choice just after May in all 3 regions, no matter what type of property is up for sale.

Population over the past 10 years in Surrey has seen a steady, almost linear increase. There is a small spike in 2007 across the 3 regions of interest, but other than that the stream of new residents is fairly calm.

Finally, for Housing Price Index and median annual income, the data has shown that although Greater Vancouver's median income is overall increasing, it is not increasing in a magnitude significant enough when compared to the Housing Price Index. HPI managed to jump up by \$75,000 CAD in just 8 years, with median annual income increasing only by ~\$13,000. If this trend continues, many new graduates entering the workforce will not likely be able to save for a home, as the price simply continues to climb at a rate that yearly wages cannot keep up with.

# Bibliography and Resources

## Line Chart Tutorial

<http://code.tutsplus.com/tutorials/building-a-multi-line-chart-using-d3js--cms-22935>

## Sorting Date Objects

<https://onpub.com/how-to-sort-an-array-of-dates-with-javascript-s7-a109>

## Sorting Objects by Date Key

<http://stackoverflow.com/questions/19430561/how-to-sort-a-javascript-array-of-objects-by-date>

## Population data source

<http://www.surrey.ca/business-economic-development/1418.aspx>

## jQuery

<https://jquery.com/download/>

## Income data source

[https://www.surrey.ca/files/Income\\_\(Total\)\\_Demographic\\_Profile.pdf](https://www.surrey.ca/files/Income_(Total)_Demographic_Profile.pdf)

## HPI data source

<http://www.rebgv.org/home-price-index>

## Listings data source

<https://data.surrey.ca/group/land-use-and-development>

## Converting Numbers to Comma-separated Strings

<http://stackoverflow.com/questions/2901102/how-to-print-a-number-with-commas-as-thousands-separators-in-javascript>

## Hovering tooltip library (d3-tip)

<https://github.com/Caged/d3-tip>

# Contribution Report - Jeremy

More details are available on our **github commit logs**:  
(<https://github.com/jeremy-jtlo/iat355/commits/master>)

## Summary

Handled code refactoring and new architecture, small multiple graphs, page layout, CSS, helper function creation, and general fixes to display issues. The new architecture relies on a `$( document ).ready()`, calling 3 general functions after the ready statement finishes. The full codebase is available in our [github repo](#).

### **getDataRange(), dateSortAsc(), sortObjByDate(), numberWithCommas()**

Helper functions used in the deeper sections of code. Useful for pulling data from individual columns, sorting objects, and formatting strings for output.

### **listing()**

Handles the drawing of the 3 active listing graphs. Has 4 notable helper functions:

- **fixListingRow()**
  - Cleans data from the CSV file, returning Date() objects and numbers rather than strings
- **returnDataSet()**
  - Reads an input data set and returns a list of entries matching a desired user criteria, based on the listing type and the region.
- **drawListAxes()**
  - Draws the x and y axis on some target visualization based on given x/y scales and data set
- **visAppend()**
  - Appends a <path> and <circle> objects to the desired visualization, based on the specified data set and scale parameters. Classes for the drawn lines and circles are added here for CSS styling later on.

For the actual drawing of the graphs, 3 separate Draw() functions were called, each leveraging the helper functions to help reduce duplicate code.

In addition to the above structures, Jeremy's duties included bug fixes, support, and the implementation of desired features like adding data points to the line graphs or scale correction. Data hovering and d3 tooltips were handled in Ernie's work, along with the foundation work required to draw the population and HPI-income graphs.

# Contribution Report - Ernie

Throughout the entire project, I was primarily responsible for gathering data and cleaning that data as well as assisting with the coding. For gathering data, I pulled open data from Surrey's official website and I cleaned them up manually by both combining and excluding data. For my contribution in the programming, I followed Jeremy's code architecture to build both the "Population" visualization and the "HPI vs Income" visualization.

For both **pairedBars()** and **population()**, each visualization contains these functions:

- **fixDataRow()**
  - Helps convert incoming CSV data into strings and Date() objects
- **returnDataSet()**
  - Returns the criteria that is desired for that specific piece of data, in the case of these two visualizations, it is the region.
- **drawListAxes()**
  - Draws the x and y axis based on xscale and yscale, including the ticks
- **visAppend()**
  - Appends the lines and the circles for the population visualization and appends the bars for the HPI vs Income visualization.

In addition to building these two visualizations, I also built the hover interaction to all of our visualizations. I searched online and found the d3-tip library and implemented it into all of our visualizations. The css styling for the tooltip is done by Jeremy.

- **var tip = d3.tip()**
  - Initializes the tooltip by referencing the library
- **var svg = d3.select("body").append("svg")**
  - Creates a svg on the body, separate from the visualizations
- **svg.call(tip);**
  - Invoking tip in context of visualization

In the append functions, I used `.selectAll()` to select all elements for the tip to highlight over. In the case of the bar chart, I used `target_vis.selectAll("rect")` including the following attributes:

- **.on('mouseover', tip.show)**
  - When mouse hovers over a piece of data such as a circle or a bar, show the tip at that location
- **.on('mouseout', tip.hide)**
  - When the mouse leaves that piece of data, hide the tip

Some of my other contributions are bug fixing, general coding support, and testing.