**DECLARATION:** I understand that this is an **individual** assessment and that collaboration is not permitted. I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at <http://www.tcd.ie/calendar>. I understand that by returning this declaration with my work, I am agreeing with the above statement.

# Introduction

My novel visualization is an exploratory visualization of the geographical and demographic impact of homelessness in Ireland. I also provide a overview of the temporal and county-by-county shifts in the cost of rent for different types of accommodation in Ireland.

# Tools/Technologies

For this project. I used P5.js [1], D3.js [2] and Plotly [3]. My initial intention was to use P5js for some state management of the dashboard and to use D3js for graphs within that dashboard. However, I found D3js to be very tedious to work with (working with the browser DOM and CSS can be time-consuming and frustrating when trying to achieve a very specific minor change), so I eventually switched to Plotly, which was drastically easier to work with and I found I could achieve meaningful results in much less time. I also used Live Server [9] on VSCode to host the application locally.

# Data Sources

In terms of dataset types, all of the below datasets are tables (with items and attributes), with the exception of the Ireland TopoJSON Map, which is a geometry dataset (with positional information).

**Homelessness Report: [8]**

Each month, from January 2019 to January 2022, is provided by the CSO as separate datasets. For pre-processing, I appended the relevant yearly quarter as a column to each file so that this information could be used in the application (as other datasets I used do not have monthly data).

The *Region* value is a nominal (categorical) string value. All other values from the original dataset (e.g. *Total Adults, Male Adults, Female Adults*) are quantitative measurements.

**Homeless by Age, Gender: [4]**

This was the latest dataset I could find with an age breakdown, split by age. Since the year of this data is outside of the range of the other datasets (2019-2022), I made it clear in the heading of the population pyramid graph that this is 2016 data, to prevent the reader being misled. No pre-processing was required.

The *Age Group* value is quantitative and discrete (can be ordered), they are a series of age ranges (represented as strings). *Sex* is a categorical value. *Year* is quantitative and sequential, and *Value* is a quantitative measurement.

**Population by County [5]**

This dataset included population breakdowns by county, but only included 2016 and 2022 data (census years). In terms of pre-processing, I needed population estimates for 2019, 2020 and 2021, I use a linear approximation calculation in Excel to get these values, and saved them to the CSV.

The *Region* value is a nominal (categorical) string value. All other values (for each year) are quantitative measurements.

**Ireland TopoJSON Map: [6]**

This dataset includes county/boundary information for the Republic of Ireland, so that it could be drawn as a SVG (for my Choropleth map). The only preprocessing I needed to do to this file was to fix some county names to match my other datasets (some had Irish language county names).

This dataset includes positional vector values (for boundary coordinates) and nominal (categorical) string values for each county. I did not use any other information from this dataset. This dataset also does not include the equivalent region data for each county.

**RTB Average Rent: [7]**

From this dataset, I collected the RTB Average Monthly Rent for (i) the aggregated result for each County in Ireland (i.e. ignoring towns, postal codes) (ii) each individual property type (and the “All property types” aggregate), and (iii) collected data from 2019 Quarter 1 to 2022 Quarter 2 (to match with the homelessness dataset). All other filters (e.g. breakdown of number of bedrooms per property) were ignored.

*Property Type* and *Location* are nominal (categorical) string values. *Value* is the average cost for a given region/property type, is a currency amount, and is therefore a quantitative measurement.

# Idioms

## Choropleth - “Irish Adult Homelessness Choropleth - By Irish Statistical Region”

**Data:** County boundaries (arcs, vectors), and a continuous quantitative attribute in the form of the percentage of adults in given statistical region that are homeless.

**Encoding**: Counties (categorical) are encoded positionally in the graph. The percentage of adults in given statistical region that are homeless is encoded by saturation (white to blue, continuous). The legend provides a method to get a value from the saturation.

*Note: I will discuss in the criticisms section why it was not feasible to encode statistical regions in this graph.*

## Donut Charts - “Adult Homelessness in Ireland (by statistical region) in DATE”

These charts are visible by clicking on a county.

**Task**:

**Data**: (i) All Regions (nominal, categorical); (ii) the percentage of homeless adults in given statistical region (quantitative measurement), (iii) the total count of homeless adults in a given statistical region (quantitative measurement); (iv) The user’s selected Region (nominal categorical)

**Encoding**: I created two Donut Charts, one encodes (i) (i.e. per-capita by region) and the other encodes (ii). Region is encoded in both graphs with hue (same hues are used for both). (ii) is compared, region by region, which is encoded by angle on Donut 1. The actual value for (ii) (i.e. percentage, which is the same value used for saturation on the Choropleth) is encoded by text (by hovering over a slice). The users’ selected region (i.e. the region of county they clicked on) is encoded positionally in the charts – its slice Is separated from the other regions on the graph. The “Total (Adult) donut also encodes (iii) with text (slice labels).

## Line Graph – RTB Average Monthly Rent (Single County, or compare two counties)

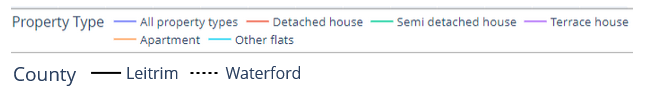
There are two graphs encoded here – for a single county, and then also comparing two counties (achieved through dropdown).

**Task:** Lookup the average rent of a given property type (for the selected county) over time. Discover trends in this information (typically an upward-trend). Select another county, compare the average rent (for each property type) between two counties. Identify outliers in rent price (Dublin). Filter by property type/region (see note below).

**Data**: County (nominal, categorical), Property Type (nominal, categorical), Prices (quantitative, continuous), time (temporal, continuous interval, by yearly quarter).

**Encoding:** Rent cost is encoded on the x-yxis, time/year/quarter is encoded on the x-axis. For a graph that has only one county, the county is encoded by text (in the title). For a graph that is comparing two counties, the county is encoded both by text and by texture (i.e. solid line vs dashed line). Each property type is encoded by hue.

**Note**: In Plotly, I tried to generate two legends that separated the County encoding (line vs dashed line) and the Property Type (hue).



I was able to generate these legends, which I feel is a lot more visually intuitive than the default legend generated, i.e.:



However, the problem is that since this is not functionality native to Plotly and I implemented it with a workaround, I then lost the Plotly feature to click on a legend entry to filter that line from the graph. I opted to leave my version of the legend in-place in the visualization; but a version of these graphs with the default legend (that allows for filtering) can be easily generated with the Jupiter notebook included.

## Stacked Bar Chart – “Family Composition of Homeless People, By Region”

This graph has two versions: total count, and percentage. With the total count version, the height of the full stacked bar, for a given region, is the total count of homeless (at the specified time). For the percentage version, the stacks are the full height of the bar chart, and it shows the ratio of the Dependants vs Adults (in families) vs Adults (without families), per region.

**Task:** Lookup region. Identify the total homeless count for that region and compare the demographic distribution of that region. Identify outliers for total homelessness count (Dublin). Change the time interval of data presented on the visualisation. If in Percentage mode, lookup and compare the demographic distribution of each region directly, regardless of total count. Identify regions that have outlier values for homelessness (e.g. the NorthEast had only 1 recorded homeless dependant in January 2021).

**Data**: Time (temporal, continuous interval, by month), Number of dependants by region (quantitative measurement), Number of homeless Adults (in a family) by region (quantitative measurement), Number of homeless Adults without families (quantitative measurement), Region (categorical)

**Encoding:** Time is encoded through animation, Dependants/Adults count is encoded by the y-axis value (for each category) and text (for total count of all three categories), and are categorized by hue (hue for each stack: orange, blue, green). Further detail about each stack is encoded by text (on hover of a stack). Region is encoded on the x-axis. For the percentage graph, the percent value is encoded by y-axis and by text.

## Population Pyramid – “Homeless Population Pyramid (Census 2016, Static)

**Task:** Lookup a given age-group / gender. Compare age groups for prevalence in homelessness. Compare sexes (on an age-by-age basis) for prevalence of homelessness. Discover the overall distribution/trends of age groups in homelessness.

**Data:** Sex (categorical), Age group (quantitative, discrete, ordinal), Count (quantitative, continuous).

**Encoding:** Sex is encoded by hue (pink/blue) and by position (left for female, right for male) i.e. it is partitioned. This double encoding makes the role of each side of the chart immediately obvious. Age Group is encoded on the y-axis and is sorted. Count is encoded on both the x-axis and by text.

# Criticism

If I were to start this project over, I think I would have worked only with Dublin data (assuming that a TopoJSON dataset for Dublin boundaries is available). I think that the fact that homelessness is a more significant issue in Dublin (per region capita) than any other county/region meant that it would likely be more useful/interesting to apply the same idioms and visualizations with the Dublin postcodes/towns, as opposed to national statistical regions/counties.

The biggest issue (in my opinion) with my Choropleth map is that the quantitative value for each county is not immediately clear – while there is a legend that shows the scale (0% - 0.362% of regions population), it would have been ideal to also have a text descriptor over each region to show the percentage value, too. While I made attempts to implement this, I found D3 to be very difficult to work with (if I had the chance to start over, I definitely would have done this graph in PlotlyJS). I introduced the Donut Charts (on click) for this reason. Despite the fact that this was a workaround, I feel that the benefit of this is that the immediate view of the dashboard is not overbearing – if the user wants more specific information on this value, they are able to click on the region/county, and the granular data is visible. This lessens the immediate mental load of the visualization.

Another issue with the Choropleth graph is that the statistical regions (SouthEast, West, etc…) do not have boundaries around them. Instead, the only boundaries on the graph are for each county. This is just a consequence of the Ireland TopoJSON dataset – it does not have data for these boundaries that I could use. Maybe I could have manually created boundaries using the boundaries of each county, but I think that this would have been much more time-consuming than it would be worth. The county sub-dashboard informs the user of which region the county belongs to, however.

During development of the animation (on the Choropleth map and the Family Composition graphs), I began to worry that it was not an effective-enough method to convey temporal information, in this case. If the user wants to compare 2016 data with 2022 data, there is an element of memory required, which is not ideal. However, I made three changes to diminish the impact of this – (i) the y-axis scale of the Family Composition graph is kept the same across all time spans – this is drastically better than the scale adjusting each time, as an increase in the number of homeless is much more visually obvious, and there is no additional mental load of thinking about the range of values (ii) I added a text value to the top of each stacked bar chart (total count of 3 stacks), so that the true quantitative value was immediately clear, and (iii) the user can select specific times on the slider (at the top of the screen) and immediately switch between them which allows for very immediate comparison. Maybe an even more ideal approach would be if the user could select a time, then (with a dropdown) select a second timespan, and for each region on the x-axis, the two stacked bar-charts (for both times) would be side-by-side. I think I would do it this way if I were to start-over.

Another issue with the animation is that it is not fluid – ideally, the change of color over time for a region would be smooth, to lessen mental load. This was primarily the consequence of some overarching project mistakes I made early on (e.g. trying to combine P5 with D3), it was my initial intention for the animation to be smooth but it eventually became too hard to implement. Again, this would have been much easier to achieve if I had done the project entirely in Plotly.JS.

One final issue is a bug in my application. If you change the slider value while the county sub-dashboard is closed, then you will not be able to open that sub-dashboard again (until you refresh). I couldn’t get to the bottom of this bug, but there is a quick workaround to see the Donut Charts for other timespans – change the slider while this sub-dashboard is open. A consequence of this is that the donut charts will *look like* they are meant to be animated – this was not my intention, these were meant to be discrete graphs. Regardless, this workaround means that there is no visualization lost as a consequence of this bug.

# References

[1] <https://p5js.org/>

[2] <https://d3js.org/>

[3] <https://plotly.com/>

[4] <https://data.cso.ie/table/E5003>

[5] <https://www.cso.ie/en/releasesandpublications/ep/p-cpr/censusofpopulation2022-preliminaryresults/geographicchanges/>

[6] <https://raw.githubusercontent.com/deldersveld/topojson/master/countries/ireland/ireland-counties.json>

[7] <https://data.cso.ie/table/RIQ02>

[8] <https://data.gov.ie/dataset/homelessness-report-june-2022?package_type=dataset> **Note: This link is just one month’s report. The full list of Homelessness Report datasets used (from Jan 2019 to June 2022) can be found here:** <https://pastebin.com/5ffRGgcW>

[9] https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer