Cluster and Cloud Computing Assignment 2
Big Data Analytics on the Cloud

An Analysis of Homelessness in Australia

Team 77

Hanchun Pan[1], Kaiyuan Cui[2], Runyu Yang[3], Yaotian Wang[4], & Zhenghan Zhang[5]

[1] Student ID: 1266219, Email: hanchunp@student.unimelb.edu.au
[2] Student ID: 1266180 , Email: kaicui@student.unimelb.edu.au
[3] Student ID: 1118665, Email: runyuy@student.unimelb.edu.au
[4] Student ID: 1503936, Email:yaotwang@student.unimelb.edu.au
[5] Student ID: 1136448, Email: zhenghanz1@student.unimelb.edu.au

Table of Contents

1. Introduction

1.1 Motivation

Homelessness in Australia has risen significantly over the past years. On any given night, over 120,000 people in Australia are experiencing homelessness. Yet, at the same time, Australia is one of the world's most wealthy nations, ranking 9th in GDP per capita. This discrepancy between Australia's economic success, high standards of living, and the prevalence of homelessness raises important questions. Why are so many people experiencing homelessness despite the country's prosperity? Using data from the Spatial Urban Data Observatory (SUDO), the Environment Protection Authority (EPA), and the Bureau of Meteorology (BoM), we aim to identify the factors that affect homelessness in Australia. Our analysis seeks to provide insights that could inform policies and decision-makers to address this critical issue.

1.2 Research Scenarios

In this project, our team focuses on the potential relationship between environment, income, demographic factors, crime, and homelessness. These factors are divided into three different scenarios for analysis and visualization:

Income, Demographic Factors, and Homelessness: in this scenario, we mainly focus on the relationships between an area's income level, population size and the number of homeless people. Income is an important indicator of an area's economic well-being. Low-income areas are often associated with high levels of economic stress. This phenomenon leads to higher levels of housing loss, which in turn increases the risk of homelessness. Population size also affects the demand for and supply of housing and social service systems in an area. Areas with large populations have increased housing needs and pressure on social service systems, leading to an increased risk of homelessness. Therefore, we are especially interested in how different combinations of average income levels and population size contribute to homelessness rates in various regions.

Environmental Factors and Homelessness: The second scenario analyzes and visualizes environmental data in relation to homelessness. Our aim is to observe whether environmental factors, such as weather conditions and pollution levels have an impact on the number of homeless people in different geographical locations. This analysis will help us identify any significant environmental influences on homelessness.

Crime and Homelessness: In this area, we wish to explore whether there is a positive relationship between the frequency of crime and the number of homeless people. In particular, we are interested in whether homelessness rises with the increase in crime rate. By analyzing crime data and homelessness, we aim to uncover any correlations that might inform policy decisions and intervention strategies.

By understanding the complex interplay between income, demographic factors, environmental conditions, and crime, we hope to support the development of targeted policies and interventions that address the root causes of homelessness and promote social stability.

1.3 Technologies

In this project, we use various advanced cloud computing and data analytic tools to analyze the potential factors influencing homelessness. Our tech stack includes Melbourne Research Cloud (MRC), Kubernetes, Fission and Elasticsearch.

Back-end deployed using Fission & Kubernetes: handles data processing and communication tasks, ensuring that data is transferred and processed smoothly from a variety of sources. Kubernetes provides powerful container orchestration capabilities to efficiently manage and scale our back-end services. Fission provides a serverless framework that allows us to rapidly develop and deploy APIs, simplifying integration and scaling. Fission seamlessly integrates with Kubernetes to provide an efficient Function-as-a-Service (FaaS) solution.

Front-end built with Jupyter Notebook: Jupyter Notebook provides a flexible environment for data exploration and visualization.

Elasticsearch Database: Elasticsearch provides powerful search and analytics capabilities to quickly process and retrieve large-scale datasets to support our analyses.

Melbourne Research Cloud: Leveraging the IaaS provided by Melbourne Research Cloud, we have access to high-performance computing resources and storage capacity to support our data processing and analytics needs.

By integrating these components, we aim to deliver a conceptual product that provides a scalable, efficient, web-based, integrated service capable of processing, analyzing and presenting large-scale data to provide valuable insights into understanding homelessness in Australia.

1.4 Team Roles

In our project teams, tasks are allocated according to each individual's specialization and area of expertise, ensuring a collaborative and efficient workflow.

- **Yaotian Wang** (K8s & MRC): Yaotian completed the initial setup of K8s & MRC for the group and added a fission function to handle harvesting data from EPA.
- **Hanchun Pan** (Fission & ElasticSearch): Hanchun is responsible for monitoring status of kubernetes cluster, proposing plans to streamline data processing, creating fission functions to pre-process and manipulate raw data sets obtained from external sources. He also worked with Kaiyuan to create API endpoints for exposing data to the front-end.
- **Kaiyuan Cui** (Fission & ElasticSearch): Kaiyuan's role was to create the fission functions and ElasticSearch indices necessary for harvesting data from the Bureau of Meteorology. He also worked with Hanchun to create API endpoints for exposing data to the front-end. Specifically, Kaiyuan configured multiple request parameters to enable flexible and efficient querying of data from ElasticSearch
- **Zhenghan Zhang** (Data-processing): Zhenghan works in data processing like data cleaning, transforming and preparing. This role is important to ensure the quality of data. Additionally, Zhenghan will assist Runyu in finding correlations between data, specifically in analyzing data correlations in three scenarios.

- **Runyu Yang** (Front-end): Runyu is in charge of developing an interactive front-end using Jupyter Notebook to create dynamic data visualizations for users. This is important for the presentation of data and realization of scenarios, as well as ensuring efficient data processing and API. In addition, Runyu also helps with database management and data upload.

2. Melbourne Research Cloud

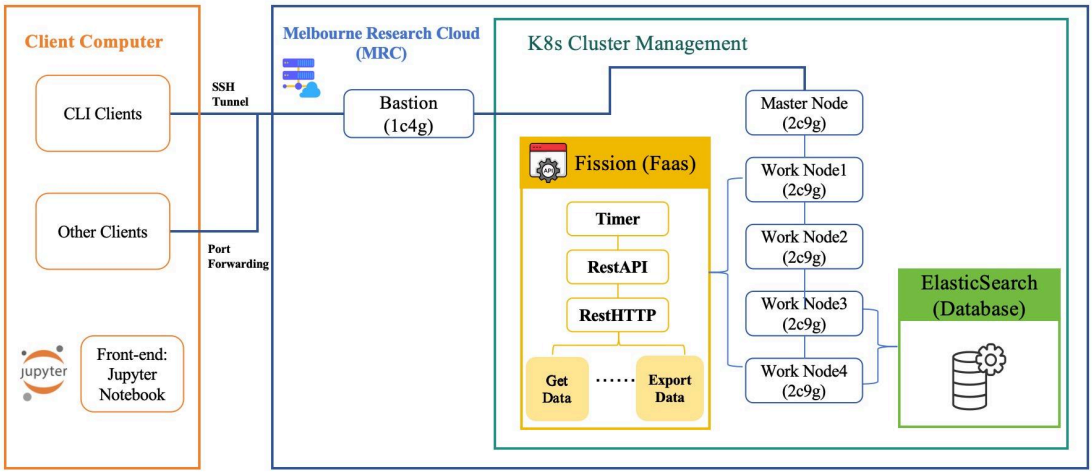


Figure 2.1.1 System Architecture

Melbourne Research Cloud(MRC) is an infrastructure-as-a-service platform which is provided by the University of Melbourne. It offers an allocation of nearly 20,000 virtual cores that are distributed across different virtual machine configurations. In this project, we are provided with 6 instance slots, 11 virtual CPUs and volume storage capacity of 700GB. The resource allocated is sufficient for our group to implement high-volume data retrieval and manipulation tasks.

2.1 Installation and Configuration

The diagram above illustrates how we deployed our project on the MRC. First, we created a new cluster consisting of three worker nodes (we added an additional one later) and a master node using the provided template. Next, we created the bastion and opened port 22 to allow secure access from the internet. The following graph is the snapshot of the running nodes from MRC.

We then began installing software on the server, starting with Elasticsearch. To ensure data integrity and avoid data loss, we instruct Kubernetes to maintain two instances of ElasticSearch. Following this, we installed Kibana and Fission. Kibana is used to examine the data, while Fission serves as our Function-as-a-Service implementation.

To collect data, we implemented a timer that automatically fetches data from the internet. Additionally, we developed a comprehensive REST API interface that allows us to upload data manually. For data visualization, we created a frontend using Jupyter Notebook, which displays graphs and results. The

frontend sends REST API requests to the bastion, which then triggers Fission. Fission runs the necessary functions and returns the results.

Instance Name	Image Name	IP Address	Flavour
elastic-2hezddnbseo7-node-5	fedora-coreos-37	192.168.10.155	uom.mse.2c9g
elastic-2hezddnbseo7-node-4	fedora-coreos-37	192.168.10.53	uom.mse.2c9g
elastic-2hezddnbseo7-node-3	fedora-coreos-37	192.168.10.72	uom.mse.2c9g
bastion	NeCTAR Ubuntu 22.04 LTS (Jammy) amd64 (with Docker)	**qh2-uom-internal** 172.26.130.41 **elastic** 192.168.10.47	uom.mse.1c4g
elastic-2hezddnbseo7-node-1	fedora-coreos-37	192.168.10.66	uom.mse.2c9g
elastic-2hezddnbseo7-master-0	fedora-coreos-37	192.168.10.59	uom.mse.2c9g

Figure 2.1.2 Instances on MRC

2.2 Pros and Cons:

There are many advantages to using MRC. Firstly, MRC is built, maintained, and used exclusively by UniMelb, reducing the risk of data leaks and other security issues compared to public clouds. Additionally, as students of the University of Melbourne, we have the opportunity to have access to MRC with no cost, whereas using a commercial cloud service could cost hundreds of dollars for implementation of the project. Furthermore, MRC is based on OpenStack, an open-source software, which enhances security as the community can inspect the code and identify potential bugs. MRC also excels in flexibility as researchers can easily adjust resources to meet the demand of the project, and be able to operate the virtual machines directly.

However, there are also several disadvantages to using MRC. The most significant drawback of MRC compared to other commercial cloud providers is the lack of documentation and support. Commercial cloud providers typically offer a simple and well-documented user interface. Once issues are detected, extensive documentation is available, and support is readily accessible. Unfortunately, MRC is not able to offer this level of support due to its cost-effectiveness. Moreover, commercial cloud providers are generally more stable. Conversely, during our development process of the project, we have encountered some issues related to MRC. Cloud providers like AWS also offer numerous managed services that simplify development and provide far more functionality compared to MRC.

3. Kubernetes

Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications. In this project, our implementation of Kubernetes is based on the concept of ‘infrastructure as code’ and ‘declarative application management’, where we use configuration files to manage nodes of the system.

Our Kubernetes cluster consists of 1 master node and 4 working nodes, where the master node is used to schedule the pods that are used for fission function deployment and ElasticSearch instances, and working nodes are used to implement the actual works of the applications. Kubernetes is able to expose containers

by assigning DNS names of service, which is particularly important for interactions among Fission functions , ElasticSearch databases and the client.

During development, we found that the biggest advantage of using Kubernetes is its high availability. When nodes or containers fail, Kubernetes automatically replaces or restarts them. Additionally, it manages resources efficiently, optimizing the utilization of CPU, memory, and other resources across the cluster.

However, Kubernetes is not an easy software to learn, it has a relatively steep learning curve. Although it is straightforward to follow the provided tutorial, attempting to implement custom configurations often requires extensive research and troubleshooting.

4. ElasticSearch

ElasticSearch is a NoSQL database that has the ability to scale horizontally and perform searches in real-time. In our project, we utilize this database architecture to store the data obtained from both external data sources in their original format and curated datasets obtained from fission functions.

- Index Creation:

  An index in ElasticsSearch is similar to a database in traditional RDBMS. Each could contain many types of documents, which are basic units of information. For index creations, we set 3 shards and 1 replica for each index. This allows ElasticSearch to parallelize the storage and querying of data. Increasing the number of shards can improve the efficiency of indexing and querying as it balances the load effectively across the cluster. The addition of replicas ensures that if a node fails, the data can still be accessible from replica shards.


```
      }
    },
    "number_of_shards": "3",
    "provided_name": "bom",
    "creation_date": "1715942821486",
    "number_of_replicas": "1",
    "uuid": "cTW5EXlOSEuPV01A2kA4Sg",
    "version": {
      "created": "8050199"
    }
  }
```


Figure 4.1 Index Configuration

  We also specified mappings for some datasets to prevent type coercion and to simplify data manipulations using fission functions. This ensures that the data fields are accurately typed and indexed. By defining explicit mappings, we avoid potential issues with incorrect data types, such as dates being stored as ‘keywords’ which we have observed to be an issue during our development.

- Data Ingestion:

Our project frequently involves data ingestion to ElasticSearch as the foundation of the analysis. Instead of sending each document as a separate request, which can be time-consuming and resource-intensive, we use Elasticsearch's bulk API to batch multiple operations into a single request. This significantly reduces the overhead associated with HTTP connections, leading to faster and more efficient data ingestion. Specific cases of data ingestion will be discussed in the 'Data Processing' section.

- Unique Document ID:

  In order to prevent duplicate data and enable easier extraction of the data, we've assigned a unique identifier to each document before they were inserted into indices on ElasticSearch. This is usually done by combining attributes that uniquely identify each document, but can vary depending on the type of the data.

- Query Testing:

  Kibana offers a web interface for visualizing Elasticsearch data, accessible via port forwarding. In this setup, port 5601 on the local machine is forwarded to port 5601 on the Kubernetes cluster. Then, by creating data views for indexes, we are able to explore the query in the 'Discover' tab before deploying them in the production environment. This process allows us to validate that the data being queried is accurate and correctly indexed, and be able to identify slow-running queries.

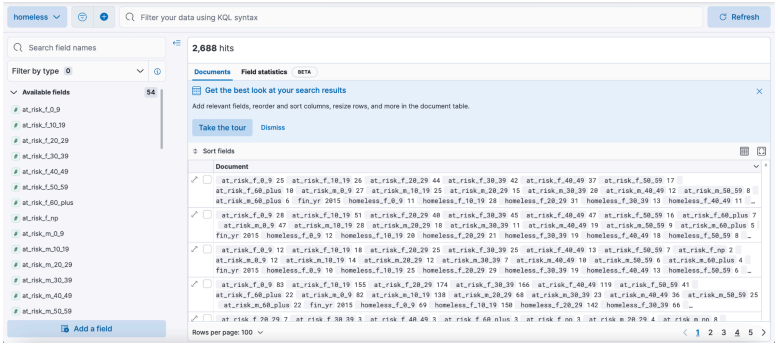


Figure 4.2 Kibana Discover Tab

5. Fission FaaS

Fission is a serverless framework that abstracts the complexities of deployment and management of functions on Kubernetes. In this research, we employ Fission as a Function-as-a-Service platform for data retrieval, preprocessing and transformation. It enables us to expose the processed data through ReSTful API endpoints which can then be utilized in frontend Jupyter Notebooks for data analysis.

5.1 Fission Function Deployment:

An environment needs to be created before deploying functions. In our project, we specifically utilized image ‘fission/python-env-3.9’ to create a Python 3.9 environment that supports the pandas module in version 2.2.2, a necessary component in our data processing functions.

Packaging allows additional dependencies that are not included in the base environment to be installed. In our deployment design, we organize all functions with the same dependencies into a single package. This reduces the number of packages that need to be created on Fission, thus effectively saving disk space which addresses the challenge we encountered during the development, which will be discussed in the later section.

With all pre-conditions satisfied, fission functions can be created individually or based upon a package. In our project, we employ HTTP requests and timers as triggers of the functions.

When a function is invoked, Fission schedules it to run in a Kubernetes pod. By employing pool manager as function execution approach, a set of pre-warmed pods are kept which are set up with necessary environment images, which minimizes the latency in executing functions.

To best adhere to ‘Infrastructure as code’ and ‘ Declarative Application Management’, upon creation of each environment, package or function, we use YAML files to store their configurations in the specs folder. This allows us to track changes over time by placing such files in the version-controlled repository. This also centralizes management, which eases the process of viewing, editing and validating configurations of functions.

5.2 Fission Functions High-level Overview:

- Data Retrieval and Ingestion:

  Fission has an event-driven architecture that allows users to configure their own triggers. For static datasets obtained from SUDO, we set up an API endpoint to upload the data in bulk to specified indexes in ElasticSearch. For dynamic datasets obtained from the Environment Protection Authority and Bureau of Meteorology, we set timer triggers to periodically invoke Fission functions for data harvesting. Before each document is saved to ElasticSearch, we check it for null/empty values and assign a unique identifier to it to prevent invalid or duplicate data in our database.

- Data Preprocessing:

  We use Fission functions to ensure the outgoing data is cleaned, validated and transformed to the desired formats before they are exposed via ReSTful API. We introduced ElasticSearch Python Client (elasticsearch-py) and Pandas module in this process. The Elasticsearch Python Client provides a convenient way to connect to ElasticSearch and ease the process of executing searches and aggregations. We use both ‘search’ and ‘scan’ operations for querying indexes of different sizes, and use specific queries to extract data in specified ranges. This allows us to easily filter the data to contain only areas of interest. Pandas is used for efficient data cleaning and manipulation. This includes handling Null values, operations by rows and data type conversions. We also perform join operations on datasets to satisfy our analysis scenarios in the frontend.

- ReSTful API Design:

Processed data is then exposed to the frontend via ReSTful API. We designed our API endpoints to be clear and resource-oriented. We also support query parameters which enables the frontend to flexibly obtain specific partitions of the data based on its needs. These optional parameters include the start and end times of the data, and limits to the size of the data that the API returns. Our response is in standard JSON format, which ensures consistency and compatibility with frontend analysis scenarios. Error handling mechanisms are also introduced which can provide error messages and HTTP status codes to reflect the status of the request.

Example API endpoints:

Bureau of Meteorology Data Retrieval

Retrieves the data harvested from BOM, taking start and end times as optional parameters

Endpoint: /get-bom-data

Parameters:(GET) Query Parameters:

- start: yyyyMMddHHmmss
- end: yyyyMMddHHmmss

Data Upload

Uploads any data to ES, given an index name and the data

Endpoint: /post-data

Parameters:(POST) request body:

```
{
        'index_name': 'string',
        'data': {}
}
```

5.3 Fission Functions Robustness

We've employed multiple methods to ensure that our fission functions run smoothly and without errors. Functions were rigorously tested locally before they were deployed on Fission. This was achieved using local environment variables and test data. Local testing also addresses the challenge we encountered on the limited disk space which will be discussed in the later section. We've also implemented robust error handling in case of any empty or malformed data. Any potential error is properly catched and processed accordingly. After the functions were deployed, they were tested with a series of unit tests to ensure that they work as expected. Additionally, as an additional safety measure, we've introduced the mechanism to rerun requests in the frontend in case a network error occurs. Through these measures, we ensure that all errors are properly handled, regardless of where they occur.

```
(base) runyuyang@ravpn-200-student-10-8-2-52 test % python end2end.py
.......
----------------------------------------------------------------------
Ran 7 tests in 67.189s

OK
(base) runyuyang@ravpn-200-student-10-8-2-52 test % python end2end.py
.......
----------------------------------------------------------------------
Ran 7 tests in 74.383s

OK
```

Figure 5.3.1 Passing Unit Tests

6. Data Collection

6.1 Bureau of Meteorology

To extract data from the Bureau of Meteorology (BoM), we first need to obtain the list of relevant weather stations, as BoM only allows querying data with specific site IDs. For example:

http://reg.bom.gov.au/fwo/IDV60901/IDV60901.95936.json

We were able to find a dataset of all weather stations at https://reg.bom.gov.au/climate/data/lists_by_element/stations.txt

Bureau of Meteorology product IDCJMC0014. Produced: 20 May 2024

Site	Dist	Site name	Start	End	Lat	Lon	Source	STA	Height (m)	Bar_ht	WMO
001000	01	KARUNJIE	1940	1983	-16.2919	127.1956	.....	WA	320.0	..	..
001001	01	OOMBULGURRI	1914	2012	-15.1806	127.8456	GPS	WA	2.0	..	..
001002	01	BEVERLEY SP	1959	1967	-16.5825	125.4828	.....	WA	..	..	..
001003	01	PAGO MISSION	1908	1940	-14.1331	126.7158	.....	WA	5.0	24.4	..
001004	01	KUNMUNYA	1915	1948	-15.4167	124.7167	.....	WA	47.0	..	..
001005	01	WYNDHAM PORT	1886	1995	-15.4644	128.1000	.....	WA	20.0	..	..
001006	01	WYNDHAM AERO	1951	..	-15.5100	128.1503	GPS	WA	3.8	4.3	95214
001007	01	TROUGHTON ISLAND	1956	..	-13.7542	126.1485	GPS	WA	6.0	8.0	94102
001008	01	MOUNT ELIZABETH OLD SITE	1959	1978	-16.3017	126.1825	.....	WA	640.0	..	..
001009	01	KURI BAY	1961	2012	-15.4875	124.5222	GPS	WA	12.0	17.0	..
001010	01	THEDA	1965	..	-14.7885	126.4963	GPS	WA	210.0	..	..
001011	01	PANTA DOWNS	1966	1969	-16.0497	124.9500	.....	WA	..	..	..
001012	01	MITCHELL PLATEAU	1968	2002	-14.7925	125.8258	GPS	WA	315.0	269.0	..
001013	01	WYNDHAM	1968	..	-15.4869	128.1236	GPS	WA	11.0	..	..
001014	01	EMMA GORGE	1998	..	-15.9083	128.1286	.....	WA	130.0	..	..
001015	01	KING RIVER PUMPING STN	1923	1931	-15.6000	128.0833	.....	WA	..	..	..
001016	01	CARSON RIVER STATION	1970	1997	-14.4861	126.7664	.....	WA	59.0	..	..
001017	01	NULLA NULLA	1923	1926	-15.5000	127.8333	.....	WA	..	..	..
001018	01	MOUNT ELIZABETH	1973	..	-16.4181	126.1025	GPS	WA	546.0	547.0	94211

Figure 6.1.1 Data from Bureau of Meteorology

However, this dataset contains many missing values and outdated stations. We performed an initial filtering of the data that extracts all stations currently in use and contains no missing data. As we were unable to find a direct mapping between stations and their corresponding Area IDs, we decided to limit our data harvesting to only stations within capital cities, for which we found that their Area IDs are usually made up in the following way:

Area ID = ID + *first letter of state name* + 60901

With this in mind, we were able to form queries for all weather stations by extracting their WMOs and state codes:

url = f'http://reg.bom.gov.au/fwo/ID{state[0]}60901/ID{state[0}}60901.{station_wmo}.json'

We then used a trial-and-error method to filter out the stations that contained no data. The related data of the remaining stations was then saved to an index on ElasticSearch for later use.

In order to continuously harvest the latest weather data, we deployed a Fission function that queries the previously saved information of each weather station from ElasticSearch and uses them to pull data from the Bureau of Meteorology. The obtained data from each weather station is then combined and cleaned before they are saved to another index on ElasticSearch. This Fission function is run every 12 hours to ensure we have the latest data, while not consuming too much computational resources. However, this may result in duplicate data, as the data may not be updated yet at the time of harvest. To address this issue, we've assigned a unique ID to each document, created from a combination of the name of the site that produced the data, and the timestamp of the data. Additionally, we've also found that the automatic mapping of the harvested data on ElasticSearch is sometimes inaccurate. For example, the timestamps of the data were being recognised as keywords, due to their unconventional format of 'yyyyMMddHHmmss'. We chose to manually adjust the mappings for easier querying of the data in the retrieval process.

6.2 Environment Protection Authority

In this project, we aim to determine whether there is a correlation between weather and homelessness. To achieve this, we decided to download data from the Environment Protection Authority Victoria (EPA). Since we need weather data from across Victoria, we chose to use the API called "All Air Monitoring Sites with Scientific Parameters." This API provides station names, types of stations, and their readings.

To ensure we always have the most up-to-date data, we implemented an hourly timer in Fission. This timer triggers Fission to fetch data using the API and save it to Elasticsearch. Since not every station updates its data hourly, this could result in duplicate entries. To prevent this, we created a unique ID for each reading, consisting of the station name and the timestamp of the reading. Due to the complexity of the returned data, we also developed a detailed mapping system to manage the data correctly. This mapping is located in the database folder.

6.3 Spatial Urban Data Observatory

Datasets from SUDO are static. This means that there is no need to set a timer trigger to periodically obtain the data, and Sudo does not seem to have an API available for data retrieval. Therefore, in our project, our approach was to manually download the datasets from SUDO, and then upload them via one of our API endpoints that uses the bulk upload API of ElasticSearch to store the data into the previously created indices. The data will then be extracted from the database through different fission functions via ElasticSearch Python Client for integration and transformation as described in the previous sections.

7. Challenges

7.1 Disk Space

We encountered no space error after several updates of the package on fission.

Figure 7.1.1 Package Update Error

We initially suspected that it was caused by a large number of packages created on Fission. Subsequently, we deleted all old packages and environments that are no longer used, but the problem persists. After posting an inquiry on Ed, the teaching team helped us to identify the problem that the node itself is not automatically clearing disk space used by previously generated packages. This means that each update of a fission package will permanently increase the usage of ephemeral storage by approximately 2 to 3 percentage points. We attempted to drain the node, but it did not solve the issue. Therefore the teaching team assists us in resetting the node and adding another node in the cluster, However, it does not fundamentally solve the problem. By strictly monitoring disk usage, we notice that the ephemeral storage of the malfunctioning node is still being consumed very quickly.

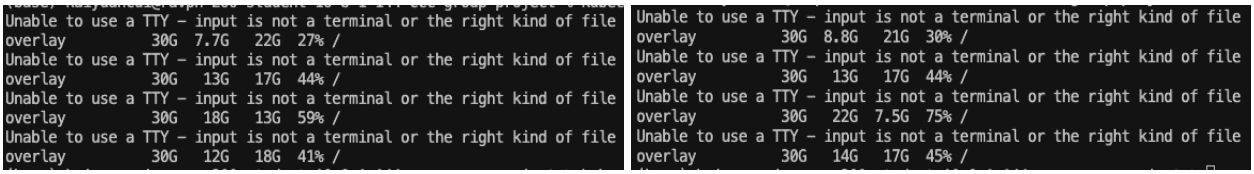


Figure 7.2.1 : Comparsion of Disk Space Usage

We were advised that this is an unsolved problem on fission* and the best practice is to keep monitoring disk usage, update one package at a time and use the disk space efficiently. Therefore, as mentioned in the Fission Function Deployment section, we reduce the total number of packages used by putting all functions that are functionally similar and share the same dependencies in the same package. In addition, we reduce the frequency of updating packages by testing locally in a rigorous manner by using port forwarding. This ensures that we can finish the development of our system without exceeding the disk space limit.

7.2 Size of Datasets

The huge size of the datasets proves to be another major challenge for us. For example, we utilized the entirety of a geodata dataset of around 180 megabytes to create map-based visualizations of our analyses.

This causes problems when we expose it to the front-end via our API. The queries would often take minutes, and sometimes even fail due to timeouts, significantly slowing our development and preventing us from a smooth demo. To address this, we decided to instead pull one small chunk of the data at a time, and then combine them in the front-end when all of the data is loaded. This is achieved by adding two optional parameters in the API endpoint, *limit* and *from-last*. *limit* restricts the total size of the data that is retrieved from ElasticSearch, while *from-last* controls how many documents from the last one to start the query. The same is done for other API endpoints responsible for retrieving large datasets. This eliminated any errors regarding transmitting large amounts of data through HTTP requests. However, one downside of our solution is the need to make HTTP requests multiple times, which resulted in a longer total retrieval time due to the added startup time from every request made.

8. Scenarios Analysis and Results

8.1 Overall Homelessness Data Distribution

Nowadays, homelessness has raised significant concerns among the public. The homeless dataset by LGA is our base dataset. Analyzing homelessness data can help us better understand the current situation in each state and narrow down the scope of the study area. Figure 8.1 illustrates the distribution of homelessness data across different states, including the At-Risk population, the Homeless population, and the population not counted by the state.

Firstly, the X-axis of the chart represents the states and territories in order from left to right: Australian Capital Territory, New South Wales, Northern Territory, Other Territories, Queensland, South Australia, Tasmania, Victoria and Western Australia.

As can be seen from the graph, the Australian Capital Territory has relatively low levels of homelessness, people at risk and people not counted by the state government. This suggests homelessness is a relatively minor problem in this region and that the total number of homeless is the lowest of all the states. This shows the relative success of the ACT in dealing with the problem of homelessness.

The high numbers of homeless and at-risk populations in New South Wales and Queensland reflect the severe state of the homelessness problem in these regions. The total number of homeless-related people in these two states is also relatively high, which indicates that more policies and resources are needed to address homelessness.

The highest number of homeless and at-risk populations in Victoria (VIC), and the highest combined total of all states, suggest that homelessness is the most serious in the region. This demonstrates the severity of the problem of homelessness in Victoria and the need to prioritize more policies and resource allocation to address it.

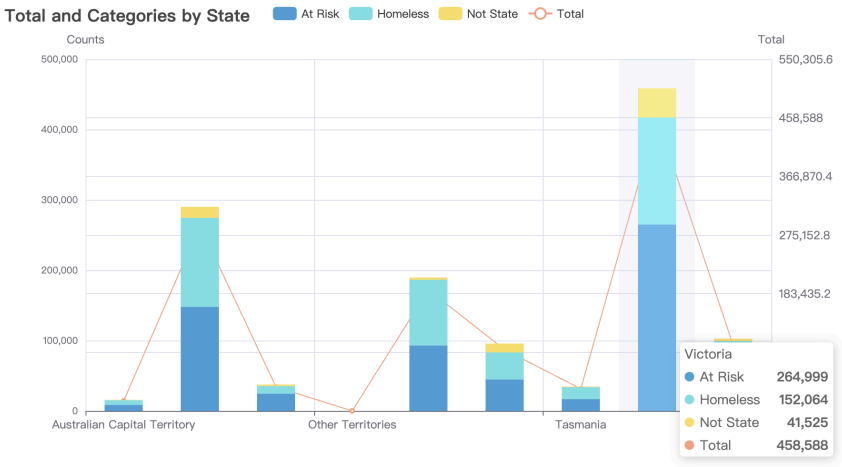
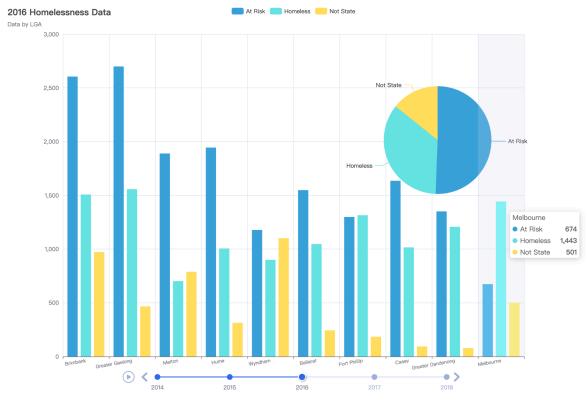


Figure 8.1.1 Distribution of homelessness data across different states

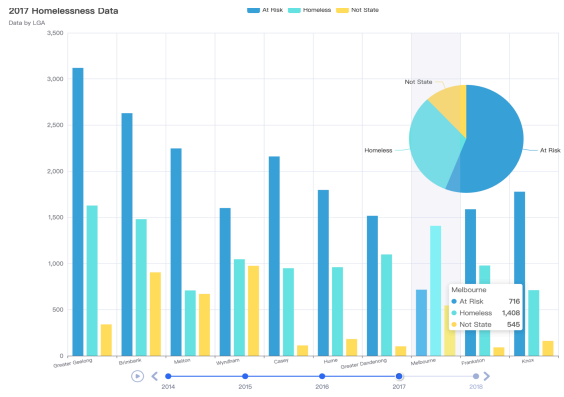
8.2 Analysis of Victorian Homelessness Data

The above analysis of state homelessness data reveals Victoria has the most severe problem. Therefore, in order to further research the issue of homelessness in Victoria, we have focussed on how the data has changed in the city of Melbourne. Figures 8.2.1 and 8.2.2 present data for the top ten Local Government Areas (LGAs) with the highest homeless populations in Victoria in 2016 and 2017.

Melbourne ranked 10th in terms of the number of homeless people in 2016, while in 2017 Melbourne increased to 8th position. This suggests that homelessness in Melbourne has increased over the years. Other areas show relatively small changes, with Greater Geelong and Brimbank having consistently higher homelessness numbers in both years, and Port Phillip and Casey having relatively low homelessness numbers.



Figures 8.2.1 2016 Homelessness Data



Figures 8.2.2 2017 Homelessness Data

8.3 Scenario 1: Victoria Income, Population and Homeless Analysis

8.3.1 Insights Derived from Map of Income Data and Homeless Data

Figure 8.3.1 illustrates the relationship between the number of homeless people and average personal income in each of Victoria's Local Government Areas (LGAs). This map shows the average personal income of the different regions using gradient colours, and markers represent the number of homeless people, with the shade of the marker colour being proportional to the number of homeless people.

As can be seen from the graph, the homeless population is concentrated in and around Melbourne, which are also the areas with high average personal incomes. Specifically, several LGAs in and around Melbourne's city centre such as Greater Dandenong, Brimbank and Hume have both higher homeless populations and higher average personal incomes.

It can be noticed that despite some areas having relatively high average personal incomes, homelessness is still a significant problem in these areas. This may be due to the fact that economically developed regions attract more people, leading to an increase in the demand for housing and cost of living, and then exacerbating the problem of homelessness.

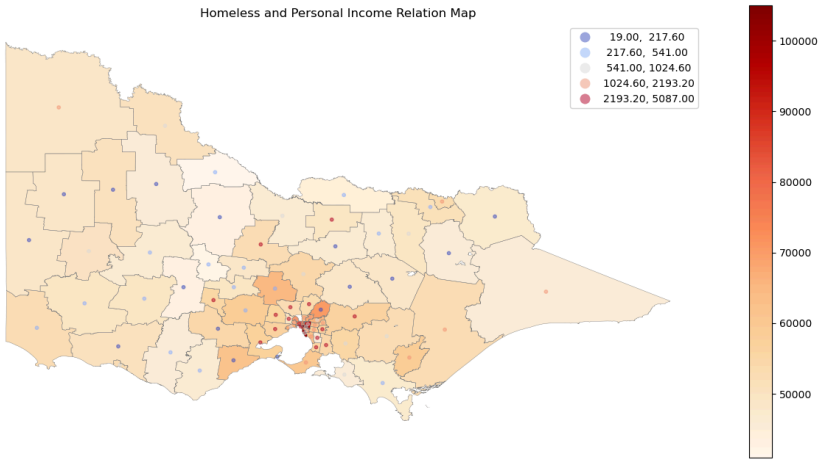


Figure 8.3.1 Homelessness and Personal Income Relation Map

8.3.2 Insights Derived from Map of Income, population and homeless

Figure 8.3.2 illustrates the distribution of income, population and homeless population across local government areas (LGAs) in Victoria. It provides a more intuitive view of the characteristics of the distribution of the homeless population across different income and population regions.

The map uses a colour progression to represent average personal incomes in different areas, ranging from light yellow to dark red to represent a change in income from low to high. The red circle represents the size of the homeless population, which is proportional to the ratio of the standardized ERP population (per 25 people) to the total homeless population in the LGA.

As can be seen from the graph, the city centre of Melbourne and its surrounding areas have higher average personal incomes, and those are also the places that have a higher number of homeless people. In addition, it also shows that some remote areas of Victoria have lower numbers of homeless people. The

relatively low average personal income in these areas may indicate that the cost of living is also lower. This allows people in these areas to afford basic housing costs, and thus it reduces the risk of homelessness.

The circle markers clearly show the distribution of the homeless population across the different LGAs. By comparing these markers to income levels, it can be observed that a higher income area does not mean homelessness is less of a problem. In fact, economically developed areas may aggravate the problem of homelessness due to the fact these areas attract more people and have a higher cost of living.

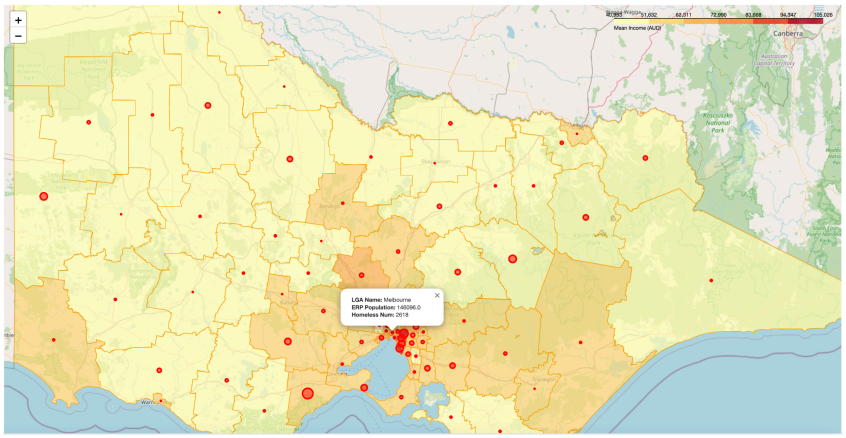


Figure 8.3.2 Average personal incomes in different area

8.3.3 Insights Derived from Income, homeless, population PLS Analysis

Figure 8.3.3 presents the results of analyses based on Partial Least Squares (PLS) regression for income, homelessness and population data in Victoria. The left figure shows a scatter plot of the PLS principal components, and the right figure shows its feature weights.

The scatterplot on the left shows the relationship between the two principal components, each point represents a Local Government Area (LGA) and the colors indicate the total homeless population. The gradual transition in color from purple to yellow represents a change in the homeless population from low to high. Some LGAs have higher homeless populations in areas with high values of Principal Component 1 and Principal Component 2, and lower homeless populations in areas with low values.

The feature weight graph on the right shows the weights of each feature on two principal components. The weight of each feature shows how much the feature contributes to the principal components. Mean income (mean_aud) has a higher weight on principal component 1, which suggests income level has a significant impact on homelessness. Internal migration (net_internal_migration) has a higher weight on principal component 2, which implies population migration is also an important influence.

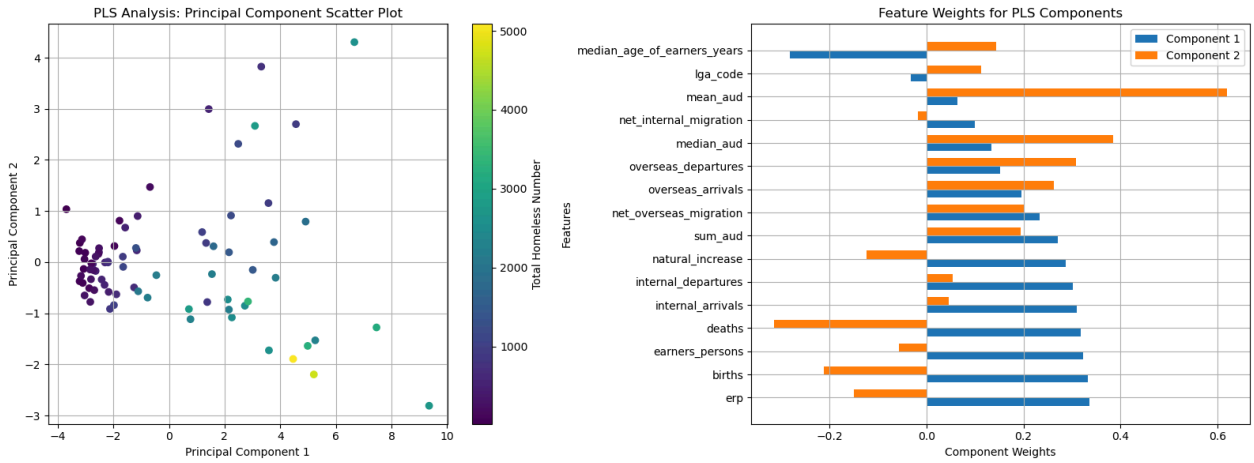


Figure 8.3.3 Partial Least Squares (PLS) regression for income, homelessness and population

8.4 Scenario 1 Summary

Analysis of Victorian income, population and homelessness data shows the homeless population is concentrated in and around Melbourne, even though these areas have higher average incomes. This means that economically well-developed areas have a greater problem of homelessness due to higher housing needs and cost of living. This finding emphasizes that high income does not automatically solve the problem of homelessness; instead, these areas may require more policy interventions and investment of resources.

In addition, some remote areas also have lower homeless populations although they have lower incomes. This might be because of the lower cost of living , which allows residents to afford living necessities , and hence reducing the risk of homelessness.

The PLS analysis found the main factors affecting homelessness problems include income levels and population migration. Average income has a significant impact on homelessness and higher-income areas need more policy support to reduce housing pressure. The high weight of internal migration suggests that population migration is also an important influencing factor, which may be related to the level of economic development and employment opportunities between regions.

8.5 Scenario 2: Environment

8.5.1 Real-Time Data and Homeless Analysis

Figure 8.5.1 shows the real-time air and body temperature monitoring data for the Melbourne (Olympic Park) site. The top graph shows air temperature and the bottom graph shows body temperature (apparent_t) for a time period ranging from 17 May 2024 to 20 May 2024.

As can be seen from the graphs, both air temperature and body temperature fluctuate significantly over this time period. The air temperature reached its highest point in the morning of 17 May 2024, which is close to 16°C, and then gradually decreased and reached its lowest point in the evening. The body temperature has a similar trend, but the overall temperature was slightly lower.

The fluctuations in air temperature and body temperature reveal a large temperature difference between day and night. Higher temperatures during the day resulted in correspondingly higher body-sense temperatures, while lower temperatures at night resulted in lower body-sense temperatures. Cold temperatures at night may pose a threat to the health and safety of homeless people, and they will face a greater risk of exposure to cold.

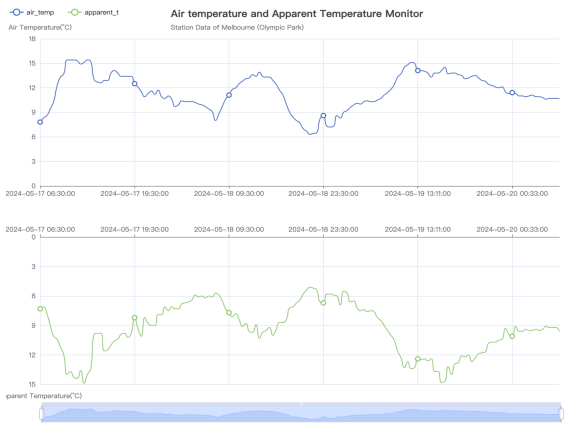


Figure 8.5.1 Temperature monitor

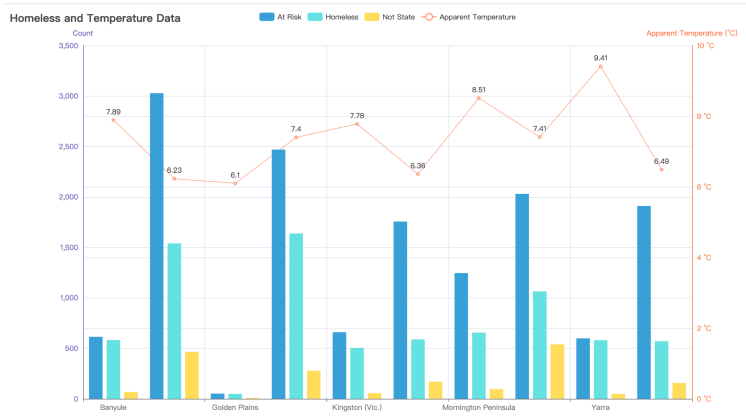


Figure 8.5.2 Homeless and Temperature Data

8.5.2 Analysis of BoM and Homeless Data

Figure 8.5.2 illustrates the relationship between homelessness population data and body temperature for different Local Government Areas (LGAs) in Victoria. The bars represent the population at risk, the homeless population and the population not counted by the state government (Not State) in each LGA, and the line graph represents the apparent temperature(body temperature).

The chart shows that areas with lower body temperatures such as Golden Plains and Mornington Peninsula have relatively low homeless populations. This suggests that temperature may have an impact on the distribution of the homeless population. Homeless people in those areas with lower temperatures face greater survival challenges, then they may choose to move to warmer areas to avoid the harsh cold.

In addition, locations with high body temperatures, such as Yarra and Kingston, have higher levels of homelessness. This may be because of the more pleasant climate in these areas, which attracts a greater inflow of people, however, it also creates housing pressures and an increase in the cost of living, which aggravates the problem of homelessness.

8.5.3 Analysis of Air Quality and Homeless Data

Figure 8.5.3 shows the relationship between the total homeless population and the mean of PM2.5 value in different local government areas (LGAs) in Victoria. The size of the bubbles in the graph represents the total homeless population, the colours refer to different LGAs, the horizontal axis is the PM2.5 mean and the vertical axis is the total homeless population.

As can be seen from the graph, most of the homeless population is concentrated in those areas with lower PM2.5 values. This suggests places with better air quality may attract more homeless people. This may be because these areas are more pleasant to live in and suitable for the homeless.

Conversely, locations with higher PM2.5 values have smaller numbers of homeless people, especially in areas with very high PM2.5 values, which are almost non-existent. This could be because areas with poor air quality pose a health threat that makes homeless people unwilling to stay in these places.

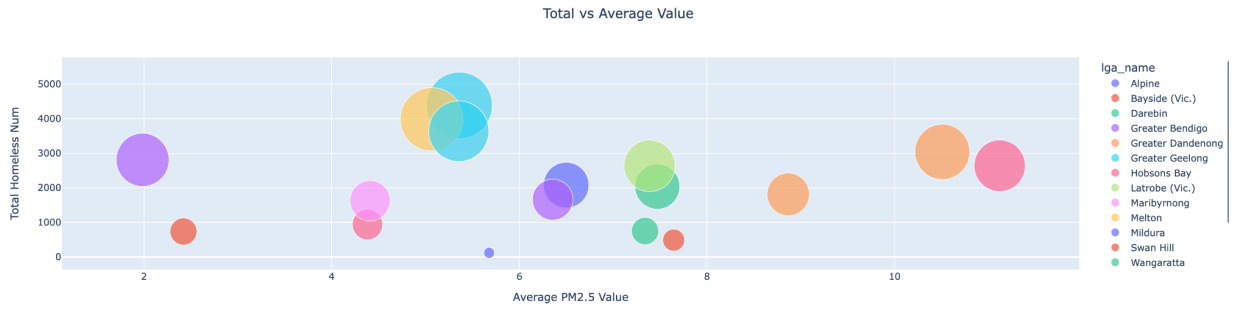


Figure 8.5.3 total homeless population and the mean of PM2.5 value in different LGAs

8.6 Scenario 2 Summary

The environmental analyses above give the impact of temperature and air quality on the distribution of the homeless population. Air temperature and body temperature data reveal a large difference between day and night temperatures, and cold temperatures at night pose a threat to the health and safety of homeless people. They will be at increased risk of cold at night, therefore the availability of temporary shelters and warming materials is important.

Secondly, it can be noticed that those areas with lower body temperatures have a relatively lower number of homeless people. This suggests that they tend to move to warmer areas to avoid the cold. However, locations with more pleasant body temperatures may still have a high level of homelessness problem, this may be because of housing pressures.

Finally, the analysis of air quality data found most of the homeless population was concentrated in areas with lower PM2.5 values, which means areas with better air quality may attract more homeless people. Whereas in areas with higher PM2.5 values, the homeless population is smaller. This shows that in poor air quality areas, only a few homeless people stay in these places.

8.7 Senario 3: Correlation between Homeless Data and Crime Data

Figure 8.7 shows the number of different types of crime in Victoria and the correlation between homelessness data and offence data. The bar chart on the left shows the number of the different types of crime and the heat map on the right shows the correlation between the homelessness data and type of offence.

As can be seen from the bar chart, type B offences (b_offences_num) have the highest number, followed by type e offences and type a offences. This suggests that category b offences are the most significant problem in Victoria.

The correlation heatmap on the right side exposes a strong link between the homeless population and the type of offence. The homeless population has a higher correlation with category A, B, C, and E offences, with correlation coefficients of 0.9, 0.86, 0.85 and 0.86 respectively. This suggests those areas with higher homeless populations will also have a higher frequency of these types of offences.

From these analyses, it can be concluded there is a significant correlation between the increase in the homeless population and the frequency of certain offences. Therefore, it emphasizes the need for special attention to be paid to a, b, c and e offences in areas with high homeless populations. These areas need to be focused on policing in order to reduce crime rates and improve social safety.

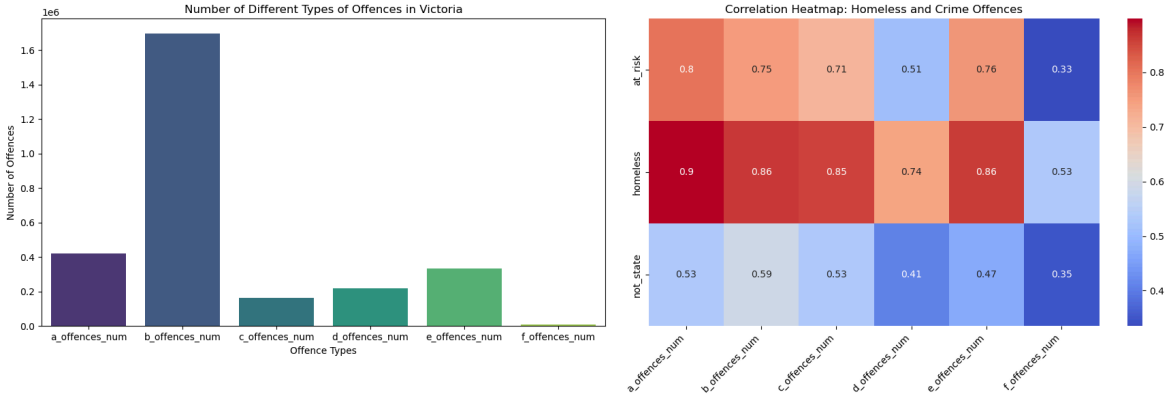


Figure 8.7 number of different types of crime and the correlation between homelessness and offences

9. Conclusion

To conclude, our team created a software system supporting analytical scenarios and interactive visualizations that focus on homelessness in Australia and its related factors. Our main technology stack includes Melbourne Research Cloud (MRC), Kubernetes (K8s), Elastic Search, Fission, and Jupyter Notebook. MRC provides the necessary hardware on which our entire backend runs. K8s assembles virtual machines into clusters and enables scalability for our software. ElasterSearch acts as our database and allows for fast storage and retrieval of large datasets. Fission is employed as a Function-as-a-Service platform for data retrieval, preprocessing and transformation. Finally, a Jupyter Notebook acts as our frontend that retrieves data from the backend and turns them into analytical visualizations.

The data used in our analyses were obtained from both static and real-time sources. The Spatial Urban Data Observatory (SUDO) contains only static datasets, from which we chose a few that best represent the homelessness, economic status, and crime levels in different regions of Australia. These data were then compared and contrasted with real-time data sources, which include the Bureau of Meteorology and the Environment Protection Authority Victoria. They provide real-time data regarding weather and air quality, respectively, via their public APIs.

Through our analyses, we've gained valuable insights into the homelessness situation in Australia and how it's affected by other factors. Our main discoveries are:

1. Homelessness is the most severe in Victoria compared to all other states in Australia. In Victoria, the homeless population mostly concentrates around Melbourne CBD.
2. The number of homeless people has a strong positive correlation with the income level and crime rate of an area.
3. Interestingly, the environmental factors of an area also have an impact on the homeless population. Areas with pleasant temperatures and low air pollution tend to attract more homeless people.

However, there exist issues within our system that should be addressed in future improvements. One of the major issues is the builder manager bug that causes garbage collection to fail, resulting in limited disk space. This restricts our ability to freely add and update packages on Fission. If this issue can be resolved, the system can provide more functionalities in additional packages. Another issue is the size of the datasets. As this is an inherent issue that comes with analyzing large datasets, it may not be avoidable and instead requires clever solutions to minimize its impact. For example, one approach is to develop more efficient algorithms for handling and processing the data. If the time required for processing the datasets can be significantly reduced, the frontend will be more responsive and the system might be able to handle even larger datasets.

10. Appendix:

Code and Video Links:

GitHub Link: https://github.com/kaiyuanCui/ccc-group-project.git

Front-end Demonstration Video: https://youtu.be/Qn8h7B63aP8

Data Sources:

SUDO Platform: https://sudo.eresearch.unimelb.edu.au/

Bureau of Meteorology (BoM) Station list:
https://reg.bom.gov.au/climate/data/lists_by_element/stations.txt

Bureau of Meteorology (BoM) Real Time Data:
http://reg.bom.gov.au/fwo/ID{state}60901/ID{state}60901.{station_wmo}.json

Example: 'http://reg.bom.gov.au/fwo/IDV60901/IDV60901.95936.json'

Environmental Protection Agency (EPA) data:
https://gateway.api.epa.vic.gov.au/environmentMonitoring/v1/sites/parameters?environmentalSegment=air

Restful API URLs (here are some examples, more details are available in our GitHub repository):

Australia homeless data by LGA: 'http://localhost:9090/get-homeless-data'

Australia Geometry Data by LGA: 'http://localhost:9090/get-geodata?from-last={index}&limit={limit}'

Australia Income and Homeless Data by LGA:
'http://localhost:9090/get-income-data/year/<year_you_want>'

BoM Data by Station:
'http://127.0.0.1:9090/get-bom-data?start={start_time}&end={end_time}&from-last={index}&limit={limit}

Victoria EPA Data By Station: "http://127.0.0.1:9090/get-epa-data?start={start_time}&end={end_time}"

Local Upload API: 'http://127.0.0.1:9090/post-data'

Fission to Fission: URL: 'http://router.fission.svc.cluster.local:80'

Fission to ElasticSearch:URL: 'https://elasticsearch-master.elastic.svc.cluster.local:9200