# <SYDNEY STAYZ> Executive Summary

### Group Member Names

Joel Rosin, Danie Soulavy, Rahul Gupta

### 2810ICT Software Technologies

### Date

October 8th

# Abstract

The Sydney Stayz Application was a collaborative effort by Daniel Soulavy and Joel Rosin. It has yielded valuable insights into the short-term Airbnb rental market allowing for better property management and property pricing optimisation. Over a 12-month period, the team has developed a series of data-driven analysis tools to empower property owners and renters with access to actionable information.

Key analysis tools include property listing analysis, guest cleanliness review analysis, top-rated property analysis, price distribution data visualisation charts and a keyword filtering tool. These analysis tools use data from various databases, enabling us to produce tailored information.

We have been able to demonstrate the potential data-driven insights have on the vacation rental market. Property owners can now optimise their pricing and availability to enhance revenue, to deliver better guest experiences in the Sydney area. Our findings highlight the power data analysis has on improving decision making for owners and renters.

# Introduction

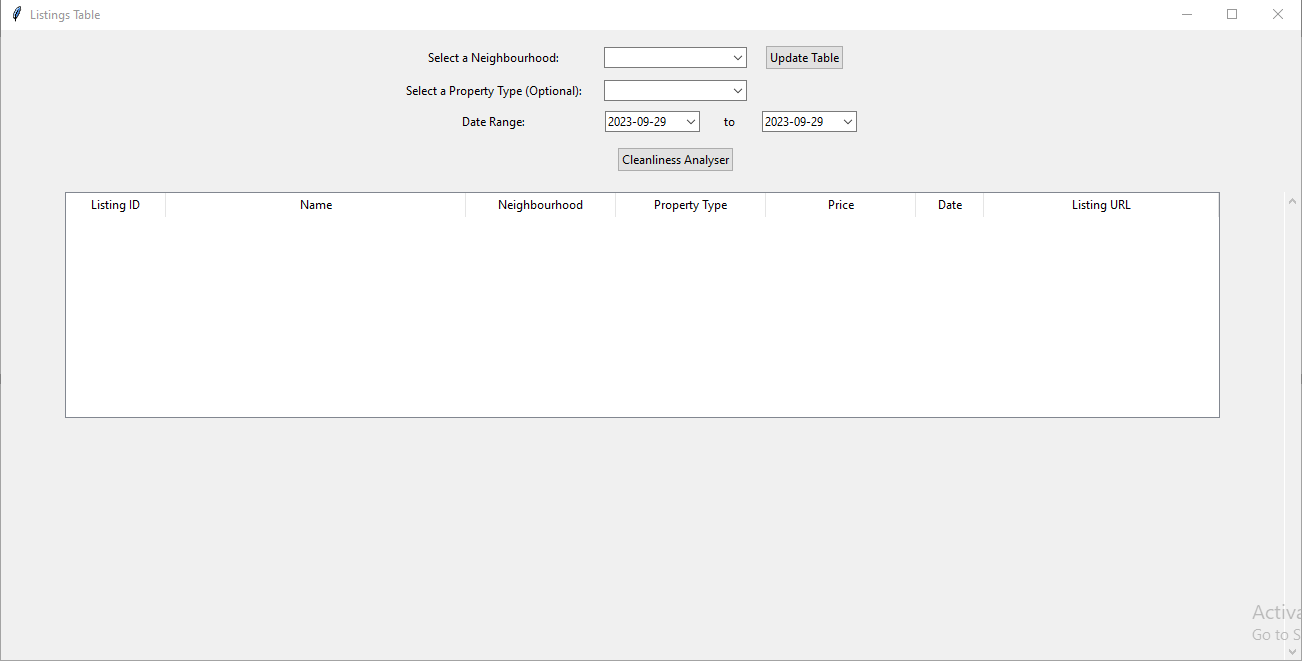
This report provides an overview of the data analysis and insights generated by the Sydney Stayz Application using data from Airbnb. The analysis covers a 12-month range which includes various tasks that are aimed at improving property management, pricing optimisation, and overall improve decision-making. The analysis tasks performed during this period was; Property Listing analysis, Guest Cleanliness Review analyser, Top Rated property analysis, Price Distribution data visualisation, and a keyword filter tool.

This report provides summaries of the key findings and insights from each analysis task, improving property management, and delivering an exceptional guest experience. The data-driven approach adopted during this period will enable property owners and managers to optimize their vacation rental operations and achieve better results.

# **Analysis 1 <Listings>**

**Listings Page**

The Listings page consists of a table which has a Neighbourhood selector which uses the “neighboourhood\_cleansed”, also a property type selector which uses the “property\_type” field in the listings\_dec18.csv file. Additionally, a date range widget which is used to filter listings between a specified date range. The data used in the date range widget comes from the calendar\_dec18.csv file and uses the “listing\_id” and “date” fields within the file. The table also has two buttons: the “Update Table” button which grabs the filtered data from “neighbourhood\_cleansed”, “property\_type” and the date range (“listing\_id” and “date”), then populates the table with the filtered data. And finally, the “Cleanliness Analyzer” which will be dissected and discussed below in one of the other analysis tool sections.



**User-selected period, 12 months**

When the required boxes have been filled and the Update Button is hit, the following data is populated to the table:

Listing ID (id)

Name (name)

Neighbourhood (neighbourhood\_cleansed)

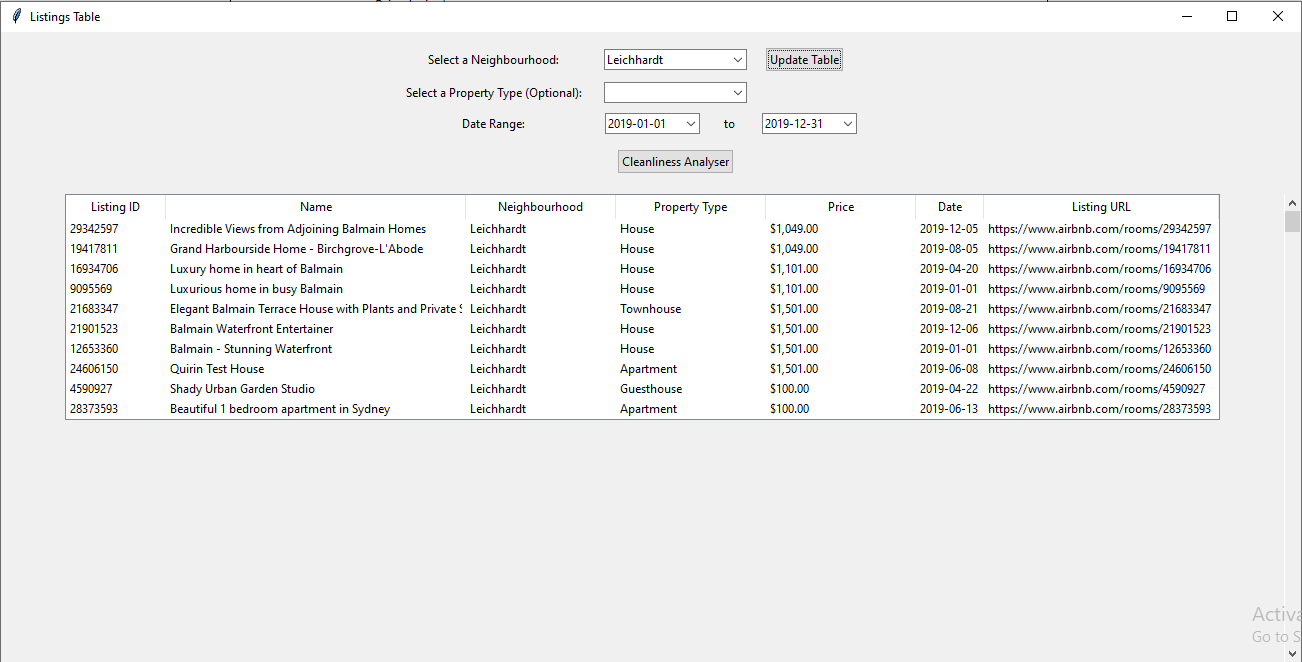
Property Type (property\_type)

Price (price)

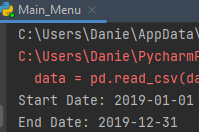
Date (date)

Listing URL (listing\_url)

Below shows listings between a 12-month period with the Leichhardt suburb selected. The property type box is optional if nothing is selected, all property types will be shown on the table.

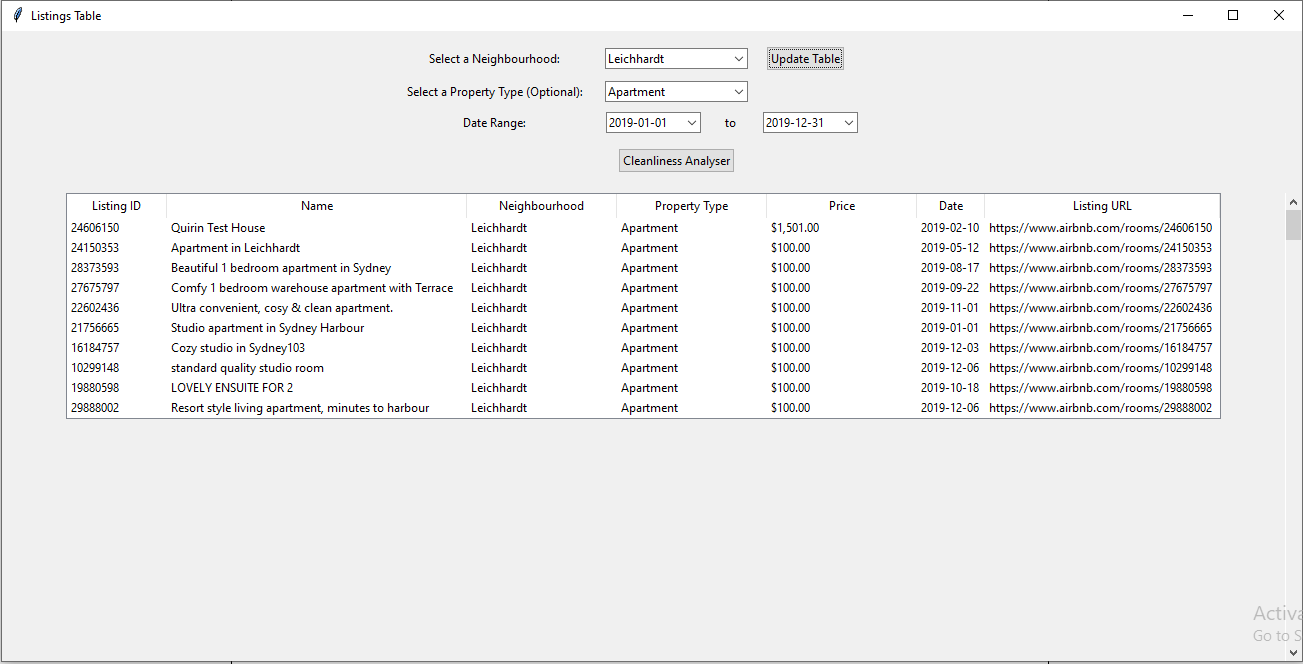


Below is a screenshot of a print statement showing the start and end date variable which was selected on the tables data range widget.



**Updated table**

The below screenshot sees a change in the table by adding a property type “Apartment,” upon clicking Update Button, the logic checks the updated filters and populates it to the table.



**Analysis 2 < Keyword Filter>**

The “Keyword search” tool shows users based on a date range input and a keyword availability of properties in the Sydney area. It utilises two csv files: Listings\_dec18.csv and calendar\_dec18.csv.

The table that is produced is built from 2 date input wx datepicker widgets which use the dates in the calendar\_dec18.csv file to grab data only related to that date range selected and availability. Then the “listing\_id” is used to cross reference to the listings\_dec18.csv file to grab the “id” from that file. The information grabbed from this check is the name of the property, the price and ensures that the input keyword is present in the amenities list before building the data table to display to the user.

The following data was used from the csv files:

Property id (id) listings\_dec18.csv

Listing id (listing\_id) calendar\_dec18.csv

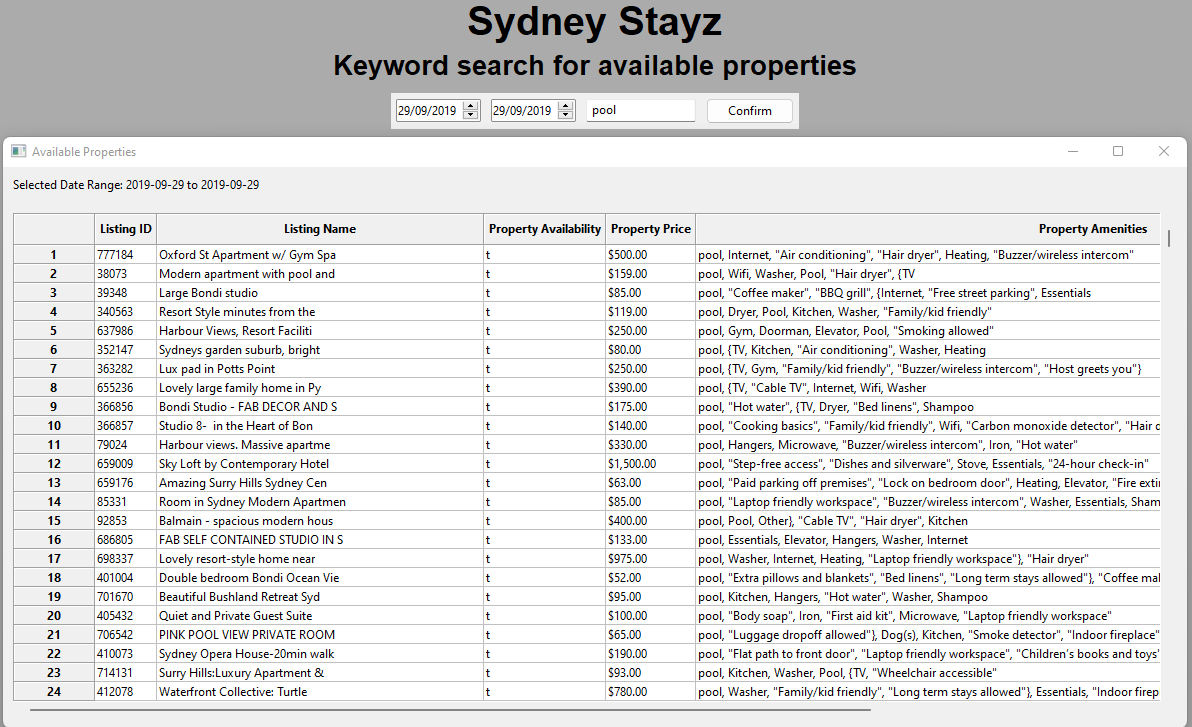
Property Availability (available)

Date (date) calendar\_dec18.csv

Property Price (price) listings\_dec18.csv

Listing Name (name) listings\_dec18.csv

Property Amenities (amenities) listings\_dec18.csv



**Analysis 3 <Price Distribution Chart>**

The Price Distribution Chart consists of a graph which takes quite a few inputs to finetune your desired data output. It uses two different csv files to achieve this, listings\_dec18.csv and calendar\_dec18.csv.

The distribution chart requires 2 different suburb data options from the list “neighbourhood\_cleansed” read in from the listings\_dec18.csv file. It also takes a minimum and maximum price value to set the property price range for the graph to display, the prices used also come from the listings\_dec18.csv file. The two date widgets that data read from the calendar\_dec18.csv file using the “date” data field. In order to get the correct data shown with the dates, the “listing\_id” from the calendar csv file and the “id” from the listings csv file were used to cross-reference the data correctly.

With these six inputs entered generates the user a customise data driven graph once the generate graph button is clicked and after a few moments of loading.

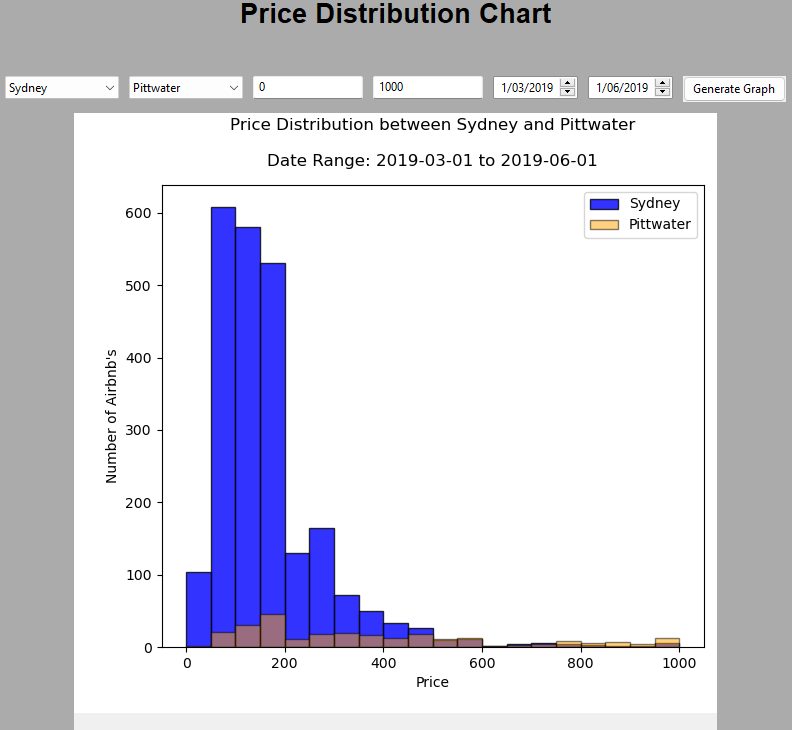
Suburb 1 / 2 (neighboourhood\_cleansed) listings\_dec18.csv

Property id (listing\_id) calendar\_dec18.csv

Listing id (id) listings\_dec18.csv

Date (date) calendar\_dec18.csv

Price (price) listings\_dec18.csv



# **Analysis 4 <Top Rated>**

**Top Rated table**

The Top Rated feature consists of a table with several inputs. It combines three different csv files: listings\_dec18.csv, reviews\_dec18.csv, calendar\_dec18.csv.

Similar to the listings table, the top rated table features a Neighbourhood and property type box which is data read in from the listings\_dec18.csv file using the “neighbourhood\_cleansed,” and “property\_type” fields. Following these two inputs a date range widget is added to filter listings between specified dates. The dates are read in from the calendar\_dec18.csv using the “date” field. Upon completion of the inputs, the Update Table button is clicked to populate the filtered data to the top rated table. The review score ratings are filtered in the logic to populate the table in ascending order, so the highest reviewed listings are on the top. The headings used along with their respective field variable names from the csv files are:

Name (name)

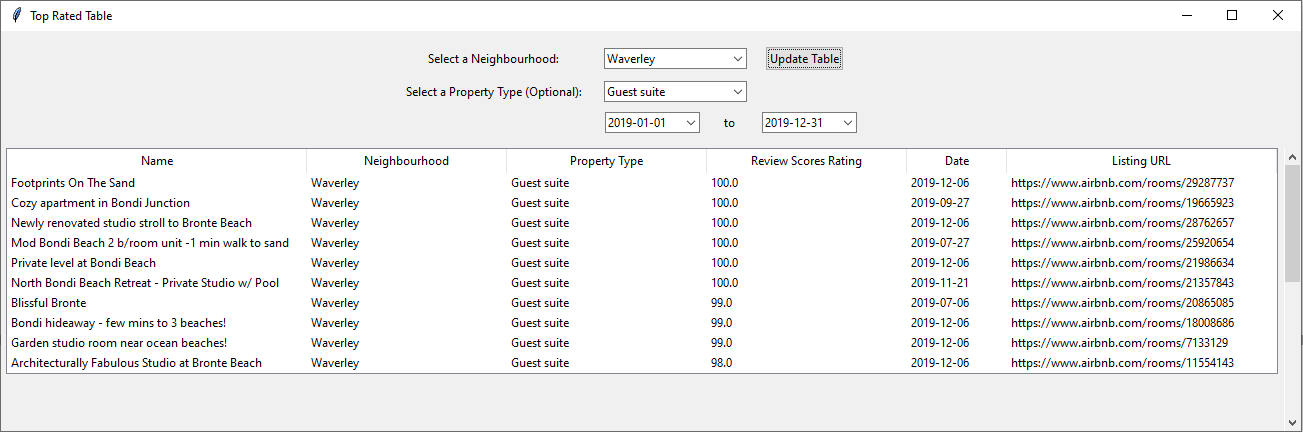
Neighbourhood (neighboourhood\_cleansed)

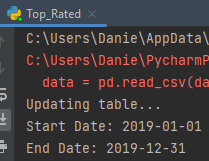
Property Type (property\_type)

Review Scores Rating (review\_scores\_rating)

Date (date)

Listing URL (listing\_url)





# **Analysis 5 <Cleanliness review Analyser>**

**Cleanliness Analyser**

The Cleanliness Analyser is on the Listings table. The cleanliness analyser logic looks through the filtered data that has been selected, in this case the Neighbourhood is “Leichhardt,” and the property type is “Apartment.” Upon the Cleanliness Analyser button being clicked, the “Listing ID,” “Reviewer Name” and “Comment” heading along with the corresponding data is appended to the bottom of the table. The reviews csv works in harmony with the listings csv using the “id” field from listings\_dec18.csv and “id“ field from reviews\_dec18.csv to ensure that the correct review along with the listing are populated to the table

The following data was used from the reviews\_dec18.csv file:  
  
Listing ID (id)

Reviewer Name (reviewer\_name)

Comment (comments)

