

Research Proposal

*AirBnB uptake by Business Sector  
 - Drivers for policy change -*

|  |  |
| --- | --- |
| **Version:** | 1.0 |
| **Date:** | 29-April-2018 |
| **Authors:** | 3MDL – Data Science Team |

Contents

[1 Introduction 3](#_Toc512877284)

[2 Document Purpose 3](#_Toc512877285)

[3 Rationale and Research Questions 4](#_Toc512877286)

[Objectives 4](#_Toc512877287)

[4 Research Method 4](#_Toc512877288)

[Data selection 5](#_Toc512877289)

[Assumptions 5](#_Toc512877290)

[Techniques 5](#_Toc512877291)

[5 Challenges 6](#_Toc512877292)

[6 Timeline 7](#_Toc512877293)

[Appendix A – Data Acquisition & Merge (code samples) 8](#_Toc512877294)

[Appendix B – Data Merging code samples 15](#_Toc512877295)

[Appendix D – Proposed Google Directions Locations 17](#_Toc512877296)

[Appendix E – Complete list of datasets 18](#_Toc512877297)

[Selected 18](#_Toc512877298)

[To Be Confirmed 18](#_Toc512877299)

[Rejected 19](#_Toc512877300)

[~~Appendix G – Reuse from Preliminary EDA???~~ 20](#_Toc512877301)

[Appendix F – Research questions with merit for future investigation 21](#_Toc512877302)

Document Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Purpose | Authors | Date |
| 0.1 | Draft Template | 3MDL – DS team | 23-04-2018 |
| 0.2 | Initial Draft – with stakeholder feedback | 3MDL – DS team | 24-04-2018 |
| 0.3 | Incorporating stakeholder feedback | 3MDL – DS team | 28-04-2018 |
| 0.4 | Addition of code samples | 3MDL – DS team | 28-04-2018 |
| 1.0 | Final version | 3MDL – DS team | 29-4-2018 |

# 1 Introduction

The changing nature of tourism accommodation has been enabled by new technology in online marketing, bookings and trust-based systems. Responding to consumer demand, the traditional hotel sector has diversified by offering a range of alternative options such as boutique hotels, private accommodation and last-minute bookings. The 3DML Data Science Team is proposing a research project to investigate the private accommodation market, the demographics of its consumers, and its impact on local communities *(see Figure 1)*. The findings from this research project will be used to make recommendations for changes to existing planning and zoning policy in alignment with 3MDL’s strategy vision statement “To hear the community voice”.

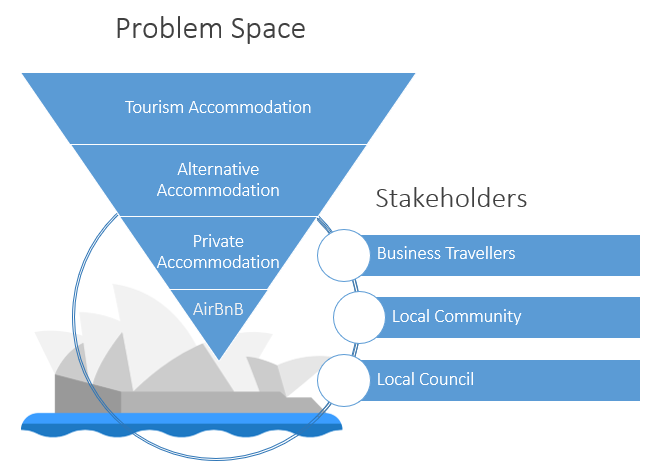


Figure - Research Area and Community Participants

## 

# 2 Document Purpose

This document presents an overview of the project rational and articulates the research questions to be investigated. Also presented are the data selection process, proposed modelling techniques and anticipated challenges identified during the preliminary exploratory data analysis (EDA).

# 3 Rationale and Research Questions

This project aims to deliver a foundation for identifying the value to property owners and local-residents in regulating or incentivising AirBnB listings for the business sector. Such policy may encourage listings in areas with under occupied residences and stabilise them in areas with high investment to residential property ratios. This will be achieved by examining the factors driving increased use of AirBnB for business travel in Sydney and its impact on the local communities of AirBnB properties.

Research questions:

* What are the factors that differentiate AirBnB Business Ready (BR) listings from those who are not Business Ready?
* Is there an association between price and occupancy rates and whether the property has BR status?
* Can we predict the increase in BR listings based on projected increase in visitors traveling on business?
* Identify whether travellers utilise travel information rather than ‘as the crow flies’ distances to major points of interest?
* Do Australian BR listings adhere to AirBnB rules?
* Do BR listings draw more revenue?
* What features appeal to the BR demographic?
* How does the BR demographic affect social sentiment of the property?
* What kind of reviews appeal to the BR demographic?

## Objectives

* Improve understanding of the value in listing a property on AirBnB with a ‘Business Ready’ (BR) badge
* Identify the factors that distinguish BR listings and non-BR listings based on attributes of the property
* Identify appropriate ‘Business Ready’ badge qualifications for Australian business travellers

# 4 Research Method

Of the proposed research questions that can be answered through regression or classification techniques, one of particular interest is:

*“What are the factors that differentiate AirBnB Business Ready (BR) listings   
from those who are not Business Ready?”*

A range of potential modelling techniques, including linear regression, could be used to determine the potential factors that differentiate business ready properties from those which are not business ready. Logistic regression is an appropriate analysis method as the dependent variable “is\_busines\_ready” is a binary, thus allowing an informed statistician/researcher to interpret the results in a way that can be translated for non-technical stakeholders.

Refer to Appendices A-B for data acquisition and data merger code samples.

## Data selection

A range of potential datasets were identified during the preliminary EDA for this project. Each was assessed based on it quality, validity, accessibility and applicability to the research questions. The final stage of selection was to address privacy, ethical and legal considerations.

\*\*Inside AirBnB claims\*\* scrapes of publicly available information are not ‘private’ data however, the \*\*Australian privacy act\*\* refers to ‘personal’ rather than ‘private’. Consequently, data such as the host’s name in the AirBnB dataset should not be included in the analysis. These will be extracted from the listings for cross-referencing for removal during data-cleansing.

Property identification data cannot be removed as they are required for the analysis. For example, exact location and features. These will be retained but not revealed in any published findings.

Selected datasets:

* Inside AirBnB - scrapes of AirBnB listings, reviews and calendars
* TRA – visitor and accommodation statistics
* Distances - from listed property to points of interest
* Image features - content of images in the listings

Refer to Appendix E for list of a complete list datasets considered for this project.

## Assumptions

Project assumptions and the basis for forming them, refer to Table 1.

Table – Project assumptions

|  |  |
| --- | --- |
|  | **Assumption** |
| Proxy for ‘vacancy’ – number of reviews per month. | A stay results in a review. |
| Accuracy of location data | Distance data collected is only specified at particular time of day and does not take into account hourly traffic patterns. |
| AirBnB dataset shows discrepancies | Scrape is an accurate representation of AirBnB listings |
| Image analysis - multiple suggestions for each detected object | Use item with highest level of confidence |
| Google distances | Validation of locations against free-text query Google Maps Locations is sufficient. |
| Focus on Sydney | This focus forms a representative basis for the findings to be extrapolated for Australia. |

## Techniques

**Data Acquisition**

* Google distance matrix (API) – via “gmapsdistance” R package
* Azure Cognitive Services API (JSON over REST) - web service call from R
* AWS Rekognition and S3 API
* Excel API – selecting multi-sheet, targeted ranges from R

**Data merging**

* Using GDrive cache to provide centralised datasets for externally sourced data
* R – various munging techniques melt()

# 5 Challenges

Based on issues encountered during our preliminary exploratory data analysis, the following challenges have been identified:

**Data volumes**

* Free tier services of three cloud platforms (AWS , Google Cloud and Microsoft Azure) were utilised by 3MDL during EDA, however Free Tier Limits placed tight restrictions on the data and images that could be processed. Modelling phase requires appropriate resources and funding.
* TRA datasets are published as large Excel workbooks with multiple tables and tabs. Data acquisition requires targeted selections within and across the tabs. Manual methods are not feasible.
* Sentiment analysis tools providing cost limitations on feature generation for review data

**Acquisition / Merging datasets**

* Minimal commonality with public datasets hence may require broad assumptions
* Manual downloads required for TRA download site owing to file formats and internal structures

**Privacy/Ethics/Legality**

* Data requiring removal from the AirBnB dataset occurs in structured and unstructured content however, owing to the nature of the research, some of this data is required for the analysis. Rationale for retention requires review and sign-off.

# 6 Timeline

The proposed timeline and next steps are outlined in Figure 2 below.

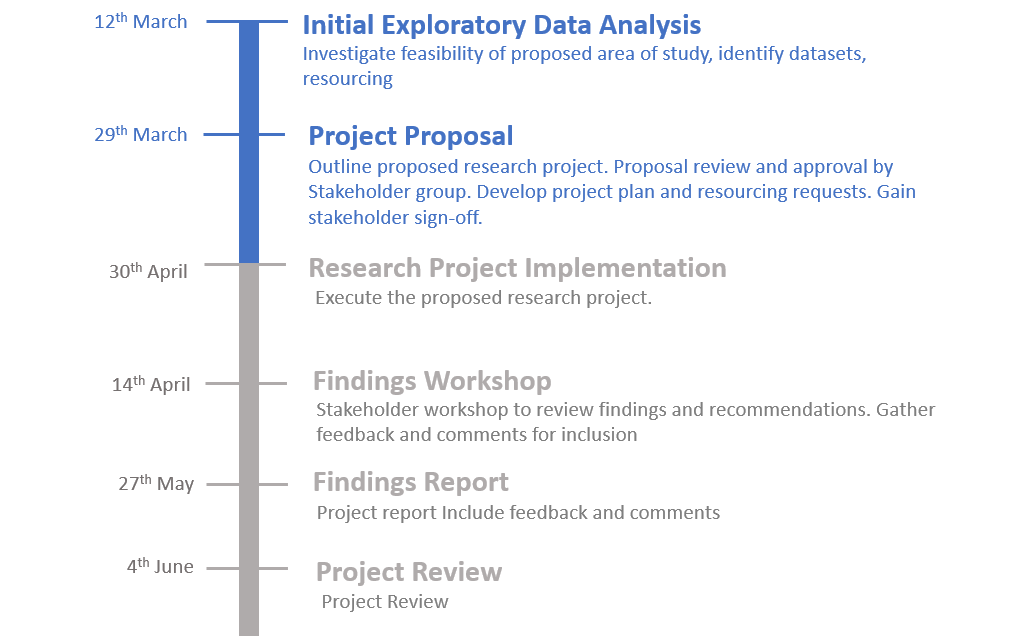


Figure - Research project timeline

# 

# Appendix A – Data Acquisition & Merge (code samples)

**1A: AirBnB Listing, Review and Calendar Dataset**

This code is used for fetching and caching locally the AirBnB data.

**Snippet 1A:**

1. # Load gzip listings from web and cache locally to team drive
2. listings\_url <- "http://data.insideairbnb.com/australia/nsw/sydney/2018-01-13/data/listings.csv.gz"
3. airbnb\_gdrive\_base\_path <- "G:/Team Drives/STDS - Assignment 2 - 3MDL/Dataset/AirBnB/"
4. listings\_filename <- "listings.csv"
5. listings\_full\_path <- paste(airbnb\_gdrive\_base\_path, listings\_filename, sep="")
7. listings <- data.frame()
9. # If file already cached to g drive then load from there
10. **if** (file.exists(listings\_full\_path)) {
11. listings <- read\_csv(listings\_full\_path)
12. } **else** {
13. # Otherwise get from URL and save to drive
14. url = getURL(listings\_url, encoding="gzip")
15. listings\_table <- read.table(url)
16. write\_csv(listings\_table, listings\_full\_path)
17. listings <- listings\_table
18. }
20. # listings now contains a properly loaded

**2A: GeoSpatial Distance from Tourist Points of Interest (POI) and Business Travel Destinations**

The pull from GDM\_API file utilizes parallel processing to call the google distance matrix API to provide the distance (in meters) and time taken (in seconds) to travel between two points by a specified mode of transport (in the sample code chunk, this is from the listing to the Sydney Opera house by public transport). Parallel processing was necessary as there is a requirement to call the API one record at a time.

Midday Saturday (21 May) as used as an arbitrary date to take the observations. This may be extended to include other dates and times to sample from, however it must be a date in the future

**Snippet 2A:**

1. ### Read Google Data###
2. #install.packages("gmapsdistance")
3. #devtools::install\_github("rodazuero/gmapsdistance@058009e8d77ca51d8c7dbc6b0e3b622fb7f489a2")
5. library(gmapsdistance)
7. #The gmaps function is as follows
8. #gmapsdistance(origin, destination, combinations, mode, key,
9. #shape, avoid, departure, dep\_date, dep\_time,
10. #traffic\_model, arrival, arr\_date, arr\_time)
12. #Register do parallel so it is faster
13. library(doParallel)
15. registerDoParallel(cores = detectCores()-1)
17. #get the unique listing (note they are all unique)
18. Timedata1 <- unique(subset(listings, select = c("latitude", "longitude","id")))
20. #Test
21. #gmapsdistance(origin =paste0("-33.8","+,+","151"),destination = 'Sydney+Opera+House', departure = as.numeric(as.POSIXct("2018-04-21 12:00:00")),combinations = "all",mode = "transit", shape = "wide")
23. ####Public Transport to Opera House####
24. #Opera\_Public\_Data <- foreach(i=1:nrow(Timedata)) %dopar% {
25. #  library(gmapsdistance)
27. #  OperaTimePublic <- gmapsdistance(origin =paste0(Timedata$latitude[i],"+",Timedata$longitude[i]),destination = 'Sydney+Opera+House', departure = as.numeric(as.POSIXct("2018-05-21 12:00:00")),combinations = "all",mode = "transit", shape = "wide")
28. #}
30. #Write to DF
31. #Opera\_Public <- as.data.frame(matrix(unlist(Opera\_Public\_Data), ncol = 3,byrow = TRUE))
32. #Opera\_Public <- cbind(Opera\_Public, listings$id)
34. #Rename for Cleaning
35. #colnames(Opera\_Public) <- c("Time", "Distance", "Status", "id")
37. #Change columns to correct format
38. #Time in Seconds
39. #Opera\_Public$Time <- as.numeric(as.character(Opera\_Public$Time))
41. #Distance in Meters
42. #Opera\_Public$Distance <- as.numeric(as.character(Opera\_Public$Distance))
43. #write\_csv(Opera\_Public, "G:/Team Drives/STDS - Assignment 2 - 3MDL/Dataset/AirBnB/Opera House by Public Transport.csv")
45. ####Define Grid####
46. # important places to go in sydeny are as follows:
47. # Opera House and Harbour Bridge
48. # Luna Park
49. # Bondi Beach
50. # Manly Beach
51. # Blue mountains (three sisters)
52. # Pokolbin (hunter valley)
54. #We will make the assumption that participants will either want to travel by car or by public transport, not by walking or cycling
56. #The date and time will be determined on the weekend
57. #Business Travel
58. # Convention centre
59. # Macquarie Park
60. # Bella Vista
61. # Paramatta
62. # Alexandria
63. # UNSW
64. # UTS
66. grid <- expand.grid(destination = c("Sydney+Opera+House","Bondi+Beach", "Manly+Beach","Three+Sisters", "Pokolbin+NSW"), mode = c( "transit"))
68. grid <- subset(grid, grid$destination!="Sydney+Opera+House"|grid$mode!="transit")
70. Timedata <- merge(grid, Timedata1) #take every combination of the grid and Timedata
72. ####Public Transport + Drive to all areas####
73. Total\_Data<-  foreach(i=1:nrow(Timedata)) %dopar% {
74. library(gmapsdistance)
76. set.api.key('')
77. TimeTotal <-  gmapsdistance(origin =paste0(Timedata$latitude[i],"+",Timedata$longitude[i]),destination = Timedata$destination[i], departure = as.numeric(as.POSIXct("2018-05-21 12:00:00")),combinations = "all",mode = Timedata$mode[i], shape = "wide")
78. }
79. #Write to DF
80. Total\_Data\_DF <- as.data.frame(matrix(unlist(Total\_Data), ncol = 3,byrow = TRUE)) #turn to tidyform
81. Total\_Data\_DF\_Final <- cbind(Total\_Data\_DF, Timedata) #Match **with** the other stuff
83. #Rename for Cleaning
84. colnames(Total\_Data\_DF\_Final) <- c("Time", "Distance", "Status",colnames(Total\_Data\_DF\_Final[,4:8]) )
85. #Change columns to correct format
86. #Time in Seconds
87. Total\_Data\_DF\_Final$Time <- as.numeric(as.character(Total\_Data\_DF\_Final$Time))
88. #Distance in Meters
89. Total\_Data\_DF\_Final$Distance <- as.numeric(as.character(Total\_Data\_DF\_Final$Distance))
90. write\_csv(Total\_Data\_DF\_Final, "G:/Team Drives/STDS - Assignment 2 - 3MDL/Dataset/AirBnB/Total\_Data\_DF\_Final.csv")

**3A: Locate required data in Excel Workbooks**

This code is used to help identify the location of specific data in an excel workbook. Returns the identify the row and column number for specific strings

**Snippet 3A** *- ## IMPORTANT: snippet only. Has dependencies, uses XLConnect package ##*

3 # work out where the data is in the source workbook

4

5 mm <- as.matrix(readWorksheetFromFile("Accommodation/IVS1 YE Dec 2017\_UpdatedMar2018.xlsx", sheet=2))

6 class(mm)<-"character" # convert all to character

7

8 # type search string in mm=="" below to work out where the data is in the source Workbook (returns row and column number)

9 rowcol <- which(mm=="VISITOR NIGHTS", arr.ind=T)

10 rowcol

11 # repeat as required

12 ### TO DO: read values from rowcol to feed into other functions

14 # read in blocks of data from the workbook

16 rs <- 8

17 re <- rs + 9

18 rs1 <- re + 3

19 re1 <- rs1 +9

20

21 B1 <- read.table(text=apply(mm[rs:re, 1:12],1,paste, collapse="\t"), sep="\t")

22 B2 <- read.table(text=apply(mm[rs1:re1, 1:12],1,paste, collapse="\t"), sep="\t")

23

24 rs2 <- re1 + 3

25 re2 <- rs2 + 9

26 B3 <- read.table(text=apply(mm[rs2:re2, 1:12],1,paste, collapse="\t"), sep="\t")

27

28 ### TO DO: loop through all these ... rs2 = re1 + 3 = rs2 + 9 for all data blocks

29 # have a look at what we've got

31 str(B1)

32 View(B1)

33 View(B2)

34 View(B3)

**4A: TRA Visitor and Accommodation Statistics**

This extracts specific data blocks across a range of worksheets in an Excel Workbook and loads into R. Resulting data is manipulated into a shape that can be merged with AirBnB dataset, including match on business / non-business.

**Code 4A:**

1. # This file takes the tourism spreadsheet data and turns into a tidy dataset
3. ## Note: these require JAVA to be installed.  Using v8+ in this code
5. # check java version ... returns error if JAVA not installed so go and get it!
6. system("java -version")
8. # set environment variable for java - this is not required if already been added to o/s system variables
9. Sys.setenv(JAVA\_HOME = "C:/Program Files/Java/jdk/")
11. # url <- "https://www.tra.gov.au/ArticleDocuments/233/IVS1%20YE%20Dec%202017.xlsx.aspx"
12. # Need to get data manually and place in drive, since website does support direct access (instead streams data through aspx page)
14. # Install required packes if not already installed:
15. # XLConnect,  XLConnectJars - called by XLConnect, reshape2, stringr, saRifx
17. install.packages("reshape2")       #has recast()
18. **if**(!"XLConnect" %**in**% rownames(installed.packages())) {
19. install.packages("XLConnect")
20. }
22. **if**(!"reshape2" %**in**% rownames(installed.packages())) {
23. install.packages("reshape2")
24. }
26. # Use XLConnect to deal with multiple worksheets
27. library(tidyverse)  # mutate
28. library(XLConnect)  # excel
29. library(reshape2)   # melt
30. library(stringr)    # regex
31. library(taRifx)     # destring
33. tourism\_base\_path <- "G:/Team Drives/STDS - Assignment 2 - 3MDL/Dataset/Accommodation/"
34. tourism\_filename <- "IVS1 YE Dec 2017\_UpdatedMar2018.xlsx"
35. tourism\_dest\_filename <- "sydney\_tourism\_201803.csv"
36. tourism\_fullpath <- paste(tourism\_base\_path, tourism\_filename, sep="")
37. tourism\_dest\_fullpath <- paste(tourism\_base\_path, tourism\_dest\_filename, sep="")
39. # Load tourism workbook
40. tourism\_workbook <- loadWorkbook(tourism\_fullpath)
42. # Get sheet names
43. sheet\_names <- getSheets(tourism\_workbook)
45. # We are only interested in country sheets
46. sheet\_names <- sheet\_names[!sheet\_names %**in**% c("Contents", "Total")]
48. # Spreadsheet configuration details
50. # There are 7 visit dimensions located in Cell A8 and increasing by 12 rows with last dimensions in Cell A80
51. sheet\_block\_range = 1:7
52. sheet\_year\_range = 2007:2017
54. visitor\_dimension\_startRow = 8
55. visitor\_dimension\_col = 1
56. visitor\_dimension\_nextRowStep = 12
58. # Each visit dimension has its own associated block of data
59. data\_startRow = 8
60. data\_startCol = 1
61. data\_endRow = 17
62. data\_endCol = 12
63. data\_nextRowStep = 12
65. # Helper functions
67. # Visit dimension name cleaned for friendly format i.e. friendly\_feature\_name
68. # features are currently of format UPPER.WORDS.AND.SO..ON.(000)
69. cleanVisitDimensionName <- **function**(s) {
70. s %>%
71. str\_replace\_all("\\.", " ") %>%   # Replace all . **with** spaces
72. str\_replace\_all("  ", " ") %>%    # Collapse **double** spaces
73. str\_extract("[A-Z ]+") %>%        # Get words only
74. trimws() %>%                      # Remove leading and trailing whitespace
75. str\_replace\_all(" ", "\_") %>%     # Replace spaces **with** underscore
76. tolower()                         # Lower **case**
77. }
79. # Get Visit Dimension Name from sheet
80. getVisitDimensionName <- **function**(workbook, sheet, block\_index) {
81. visit\_dimension\_name = readWorksheet(workbook, sheet,
82. startRow = visitor\_dimension\_startRow + ((block\_index-1) \* visitor\_dimension\_nextRowStep),
83. startCol = visitor\_dimension\_col,
84. endRow = visitor\_dimension\_startRow + ((block\_index-1) \* visitor\_dimension\_nextRowStep),
85. endCol = visitor\_dimension\_col)
86. cleanVisitDimensionName(names(visit\_dimension\_name))
87. }
89. # Get Visit Dimension Block Data from sheet
90. getVisitDimensionBlock <- **function**(workbook, sheet, block\_index) {
91. readWorksheet(workbook, sheet,
92. startRow = data\_startRow + ((block\_index-1) \* data\_nextRowStep), startCol = data\_startCol,
93. endRow = data\_endRow + ((block\_index-1) \* data\_nextRowStep), endCol = data\_endCol)
94. }
96. # Final Tourism data frame
97. tourism.data = data.frame()
99. # Build dataset across all sheets
100. **for** (sheet **in** sheet\_names){
102. # Context data for current sheet
103. sheet\_data <- data.frame()
105. # Loop through each block in the sheet as they are iterated over each visit dimension
106. **for** (block\_index **in** sheet\_block\_range) {
108. # get sheet data for visit dimension Metric
109. sheet\_block\_data <- getVisitDimensionBlock(tourism\_workbook, sheet, block\_index)
111. # Change names of features to "purpose" and from 2007, to 2017
112. names(sheet\_block\_data) <- c("purpose", sheet\_year\_range)
114. # visit dimension name, want in format lower case words seperated by underscore
115. visit\_dimension\_name = getVisitDimensionName(tourism\_workbook, sheet, block\_index)
117. # Melt the years features into a single year feature
118. sheet\_block\_data %>%
119. melt(id.vars=c("purpose"),
120. variable.name="year",
121. value.name=visit\_dimension\_name) -> sheet\_block\_data
123. # Each block is a feature, inner join on purpose and year
124. **if** (nrow(sheet\_data) == 0) {
125. sheet\_data <- sheet\_block\_data # first time
126. } **else** {
127. sheet\_data <- merge(sheet\_data, sheet\_block\_data, by=c("purpose","year"))
128. }
129. }
131. # add country name, this is the current sheet name
132. sheet\_data %>%
133. mutate(country = as.factor(sheet)) -> sheet\_data
135. # need to append now this dataset
136. tourism.data <- rbind(tourism.data, sheet\_data)
137. }
139. # Clean Tourism.data
141. # Strip leading and trailing whitespace
142. tourism.data$purpose <- trimws(tourism.data$purpose)
144. # Strip total row
145. tourism.data <- tourism.data[!(tourism.data$purpose == "Total"),]
147. # Strip backpacker data - this data duplicates the by reason data
148. ### Consider commenting these lines and split this into separate data frame
149. tourism.data <- tourism.data[!(tourism.data$purpose == "Backpackers"),]
150. tourism.data <- tourism.data[!(tourism.data$purpose == "Non backpackers"),]
152. # Convert to factor
153. tourism.data$purpose <- as.factor(tourism.data$purpose)
155. # Clean metric data, and put as proper type
156. dim\_index <- which(names(tourism.data) %**in**% c("purpose", "year","country"))
157. metric\_columns <- names(tourism.data[-dim\_index])
159. **for** (col **in** metric\_columns) {
160. tourism.data[,col] <- as.numeric(destring(tourism.data[,col])) # convert to number, and set np to NA
161. }
163. # Check structure and content are as expected
164. str(tourism.data)
165. head(tourism.data)
167. # the sort the rows so values are in a predicable order
168. tourism.data <- select(tourism.data, purpose, country, year, everything())
169. View(tourism.data)
171. # Write data to shared drive
172. write.csv(tourism.data, file=tourism\_dest\_filename)
174. # Summarize totals for business and non business
176. # create is\_business flag which is set if visit purpose was Business Travel or Employment
177. tourism.data %>%
178. mutate(is\_business = ifelse(purpose %**in**% c("Employment","Business Travel"), 1, 0)) -> tourism.data
180. # Sum across all metric columns by Business, Year and Country.
181. aggregate(tourism.data[metric\_columns],
182. by=list(
183. year=tourism.data$year,
184. is\_business=tourism.data$is\_business,
185. country=tourism.data$country), FUN=sum, na.rm=TRUE)

**5A: AirBnB Listing Profile Image Analysis**

This function is intended to get sentiment data for the reviews from AirBnb. The purpose is to feature engineer a new feature against each AirBnB listing to understand the overall sentiment against that listing. API Notes: The API is a RESTFul webservice using JSON as data format exchange. There is a max of 5k per month uses. Need to distribute around team to run.

**Snippet 5A - *\*\* The following is more of a process instruction rather than an r script\*\****

5 # Extract id andpicture\_url from listings file

6 #id picture\_url

7 #13467416 https://a0.muscache.com/im/pictures/778d441e-2c94-431a-8307-b32eaec69930.jpg?aki\_policy=large

8 #15882089 https://a0.muscache.com/im/pictures/3e9e50cf-2949-4a9e-9782-fecf327ea75a.jpg?aki\_policy=large

9

10 ## Remove '?aki\_policy=large' on all picture url to the image direct link

11 #id picture\_url

12 #13467416 https://a0.muscache.com/im/pictures/778d441e-2c94-431a-8307-b32eaec69930.jpg

13 #15882089 https://a0.muscache.com/im/pictures/3e9e50cf-2949-4a9e-9782-fecf327ea75a.jpg

14 #...

15

16 #Process the picture to get the context of each photo to get the context confidencelevel

17 #id context confidencelevel

18 #13467416 Building 0.923

19 #13467416 Housing 0.923

20 #13467416 Apartment Building 0.547

21 #13467416 Architecture 0.512

22 #15882089 Building 0.962

23 #15882089 Housing 0.962

32 #....

33

34 ## Analysis the output and find the most appeared context in this case 'Building' and 'Housing', and some interesting features, high\_rise 35 #id context confidencelevel Count of context

36 #13467416 Building 0.923 4

37 #15882089 Building 0.962 4

40 #13467416 Housing 0.923 4

41 #15882089 Housing 0.962 4

44 #13467416 Architecture 0.512 2

45 #13467416 Apartment Building 0.547

53 #15882089 High Rise 0.52 2

54 #...

57

58 ## Convert the most appeared words into features for adding back to the lising

59 #id has\_building has\_housing has\_high\_rise has\_city

60 #13467416 1 1 1 1

61 #15882089 1 1 1 1

62 #16197117 1 1 0 0

63 #22298997 1 1 0 0

64

65 ## R code for Merge new feature columns back to the main listings list for regression analysis

66 library(tidyverse)

67

68 image <- read.csv("image.csv")

69 listings <- read.csv("listings.csv")

70

71 listings2 <- merge(listings, image)

72

# Appendix B – Data Merging code samples

**1B: Merging AirBnB Listings dataset with Google Distance Matrix**

The Google Distance Matrix dataset was constructed using the longitude and latitude from the listings file as the “origin” for the API GET call. Because of this, we were able link the listing ID directly to each API call. Thus a merge between the two files is relatively straightforward: Listing\_MatrixAPI ← merge(listings, `Opera House by Public Transport`, by = “id”)

**Snippet 1B:**

1. # Load Libraries
2. library(tidyverse)
3. library(httr)       # For constructing POST request
4. library(jsonlite)   # For converting JSON to data frame
5. library(data.table)
6. library(dplyr)
8. # Params for API update
9. # Can only run 5000 Queries per month, so each person needs to update the start and end to distribute.
10. start = 0
11. end = 5000
13. # Get Review Data
14. reviews\_filename = 'G:/Team Drives/STDS - Assignment 2 - 3MDL/Dataset/AirBnB/reviews.csv'
15. reviews <- read\_csv(reviews\_filename)
17. df <- reviews[start:end,c('id','comments')]
19. # Azure API expects request format params of 'id' and 'text'
20. df <- rename(df, text = comments)
21. data <-list(documents=df)
23. # Call Sentiment API
24. api\_key = 'xxx'
25. sentiment\_api\_url = "https://westcentralus.api.cognitive.microsoft.com/text/analytics/v2.0/sentiment"
27. # Call API
28. response <- POST(sentiment\_api\_url,
29. add\_headers(`Ocp-Apim-Subscription-Key`=api\_key),
30. body=toJSON(data))
31. # Get Response
32. response\_content <- content(response, as="text", encoding = "utf-8")
34. # convert JSON to data.frame
35. fromJSON(response\_content)$documents %>%
36. mutate(id=as.numeric(id)) ->
37. responses
39. # Create table with Review ID and Sentiment Score only
40. df %>% left\_join(responses, by = 'id') %>% select(c('id', 'score')) -> dfs
42. # Write out which records of dataset were done so not overlapping
43. write\_csv(dfs, sprintf("data\\review\_sentiment\_start\_%s\_end\_%s.csv",start,end))

**2B: AirBnB Reviews: Feature Construction with Sentiment Analysis**

Sentiment Analysis was proposed using Microsoft Azure Cognitive Services, Indico.ai Sentiment API, and potentially others. This was because all API service offerings had a limited amount of API calls per month. This EDA acquisition was abandoned as the AirBnB review dataset was 335k+ records. Acquisition costs would be in the range of $1500+ AUD. It is recommended to re-attempt with an on-premise sentiment analysis library.

The merge approach was to pass the review ID and associated review text to the API and then merge by Review ID to the original dataset.

**Snippet 2B:**

1. # Load Libraries
2. library(tidyverse)
3. library(httr)       # For constructing POST request
4. library(jsonlite)   # For converting JSON to data frame
5. library(data.table)
6. library(dplyr)
8. # Params for API update
9. # Can only run 5000 Queries per month, so each person needs to update the start and end to distribute.
10. start = 0
11. end = 5000
13. # Get Review Data
14. reviews\_filename = 'G:/Team Drives/STDS - Assignment 2 - 3MDL/Dataset/AirBnB/reviews.csv'
15. reviews <- read\_csv(reviews\_filename)
17. df <- reviews[start:end,c('id','comments')]
19. # Azure API expects request format params of 'id' and 'text'
20. df <- rename(df, text = comments)
21. data <-list(documents=df)
23. # Call Sentiment API
24. api\_key = 'xxx'
25. sentiment\_api\_url = "https://westcentralus.api.cognitive.microsoft.com/text/analytics/v2.0/sentiment"
27. # Call API
28. response <- POST(sentiment\_api\_url,
29. add\_headers(`Ocp-Apim-Subscription-Key`=api\_key),
30. body=toJSON(data))
31. # Get Response
32. response\_content <- content(response, as="text", encoding = "utf-8")
34. # convert JSON to data.frame
35. fromJSON(response\_content)$documents %>%
36. mutate(id=as.numeric(id)) ->
37. responses
39. # Create table with Review ID and Sentiment Score only
40. df %>% left\_join(responses, by = 'id') %>% select(c('id', 'score')) -> dfs
42. # Write out which records of dataset were done so not overlapping

# 

# Appendix D – Proposed Google Directions Locations

The proposed locations of interest to be used in the Derived Distances dataset are listed in Table 2 below. The number of actual locations that can be processed is dependent on access to appropriate resources. Final selection will be determined prior to commencing this research project.

Table - Points of Interest

****

# Appendix E – Complete list of datasets

## Selected

More information about the datasets selected for this project are detailed in Table 2 below.

Table - Datasets (selected and to be confirmed)

|  |  |  |
| --- | --- | --- |
| **Source** | **Data set** | **Basis for Selection** |
| Inside AirBnB  \*\* add link\*\* | * listings * reviews * calendar   grain: month | Richness of dataset and applicability to the research questions. |
| Travel Research Australia (TRA)  <https://www.tra.gov.au/>  Accessed under license issued to UTS Libary.  \*\* add link\*\* | International Visitor statistics - trips, nights, average stay and spend by   * main purpose of trip * country/suburb of residence * accommodation   National visitor statistics - trips, nights, average stay and spend by   * main purpose of trip * destination state/region * accommodation   Accommodation statistics  grain: annual, quarterly and monthly | Specific subsets TRA published datasets based on commonality with available AirBnB data and applicability to the research questions.  Statistics published by Tourism Research Australia (TRA) will be used for industry benchmarking and validate these findings. |
| Derived distances  <https://developers.google.com/maps/documentation/distance-matrix/> | * distance * time * transport mode   grain: latitude / longitude | Points of interest to:   * visitors to Sydney * business travellers   Refer to Appendix D – Proposed Google Directions Locations for proposed list. |
| Australian Bureau of Statistics  <http://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/8635.02015-16?OpenDocument> | Tourist Accommodation, Australia, 2015-16  Establishments, rooms, bed spaces, occupancy, takings | Specific TRA published datasets were selected based on commonality with available AirBnB data and applicability to the research questions. |

## To Be Confirmed

|  |  |  |
| --- | --- | --- |
| **Source** | **Data set** | **Basis for Selection** |
| CoreData (Industry Partners)  https://coredata.com.au/ | short market research questionnaire about business travel | data set will provide context about what business travellers actually look for when selecting a place to stay |

## 

## Rejected

The datasets that were investigated but not selected for this research project are detailed in Table 3. Reasons for rejection include insufficient granularity and/or keys for linking with primary datasets, high level of sparsity, prohibitive volumes.

Table - Datasets (rejected)

|  |  |
| --- | --- |
| Tom Slee (scrapes of AirBnB listings) | * cleaner but less rich dataset * reduced options for investigation * dataset will not be updated in the future - reduced value for future research |
| Image analysis | \*\* did we reject this or not??? \*\*  I think possibly not |
| Rental price (benchmark) | * insufficient time for acquisition |
| Obike data (impact on community) | * insufficient time for acquisition * tenuously related to research questions |
| Weather Data | * insufficient time for acquisition * not strongly related to research questions |

# 

# ~~Appendix G – Reuse from Preliminary EDA???~~

*{Not quite sure what but this is the place to put anything we think is great but doesn’t fit within our word count 😉}*

EDA CODE

<CODE FILE:PreProcessing/EDA.r />

@Mark what does this do?

<CODE FILE:PreProcessing/ReadAirBnB.r />

@Mary where is your EDA code?

<CODE FILE:PreProcessing/XXXX.r />

CLEANING CODE

AirBnB Data Cleaning

<CODE FILE:PreProcessing/Cleaning.r />

Sentiment Reviews Cleaning

<CODE FILE:PreProcessing/reviews\_cleaning.R />

**EDA modelling:**

* Linear regression, logistic regression – for inference tasks, more interpretable results
* Custom functions – converts multiple lines of code into a single function
* R functions – DataExplorer(), lrfit(), glm()

**Visualisations:**

* Shiny Apps - R’s web development framework
* Plotly package - creates interactive SVG plots
* Leaflet package - flexible geospatial mapping
* wordcloud package - generates wordclouds