ASSIGNMENT COVER SHEET

SUBJECT NUMBER & NAME	36103 Statistical Thinking in Data Science
NAME OF STUDENT **(PRINT CLEARLY - SURNAME, FIRST NAME)**	Brackenrig, Mark, Campbell, MaryAlice, Cheluget, Linda Lehmann, Martin, Xu, Yusheng
STUDENT ID NUMBER	12964298, 13298029, 13009028, 12543738, 11860870
STUDENT EMAIL	mark.brackenrig@student.uts.edu.au maryalice.campbell@student.uts.edu.au linda.cheluget@student.uts.edu.au martin.lehmann@student.uts.edu.au yusheng.xu@student.uts.edu.au
STUDENT CONTACT NUMBER	MB: +61 450 548 776, MC: +61 414 624 429, LC: +61 450 652 779, ML: +61 414 638 601, YX: +61 452 256 666
DUE DATE	30-Apr-2018 by 23:59
ASSESSMENT ITEM NUMBER/TITLE	AT2 – Data Analysis Project, Part A: Proposal

- I confirm that the work submitted conforms with the university's guidelines on academic integrity.
  Refer to the UTS policy on 'Advice to Students on Good Academic Practice':
  http://www.gsu.uts.edu.au/policies/academicpractice.html

- I am aware of the penalties for plagiarism. This assignment is my own work and I have not handed in this assignment (either part or completely) for assessment in another subject.

- If this assignment is submitted after the due date I understand that it will incur a penalty for lateness unless I have previously had an extension of time approved and have attached the written confirmation of this extension.

Please provide details of extensions granted here if applicable

_____

Signature of Student: _____ **Date:** 30 / Apr / 2018

If submitted electronically tick here to indicate you agree with the above ✓

Research Proposal

AirBnB uptake by Business Sector
- Drivers for policy change -

Version: 1.0
Date: 30-April-2018
Authors: 3MDL – Data Science Team

Contents

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	30-4-2018

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 1 - 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 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. This method allows 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 – Complete list of datasets 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 1 – 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.

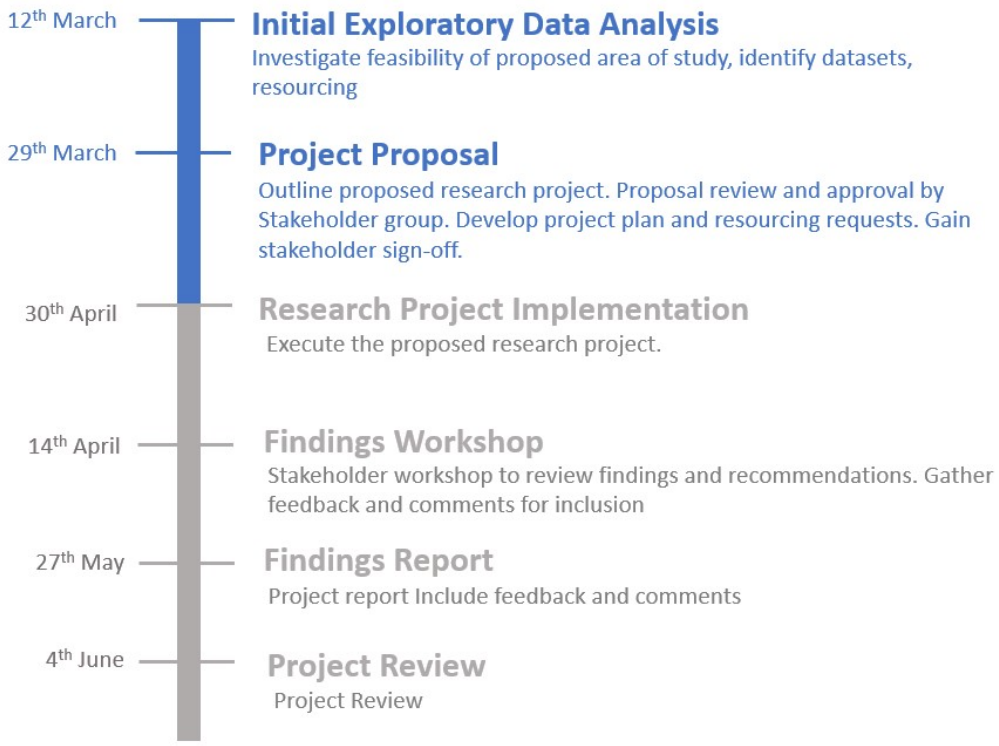

Figure 2 - Research project timeline

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:

```
# Load gzip listings from web and cache locally to team drive
listings_url <- "http://data.insideairbnb.com/australia/nsw/sydney/2018-01-13/data/listings.csv.gz"
airbnb_gdrive_base_path <- "G:/Team Drives/STDS - Assignment 2 - 3MDL/Dataset/AirBnB/"
listings_filename <- "listings.csv"
listings_full_path <- paste(airbnb_gdrive_base_path, listings_filename, sep="")

listings <- data.frame()

# If file already cached to g drive then load from there
if (file.exists(listings_full_path)) {
  listings <- read_csv(listings_full_path)
} else {
  # Otherwise get from URL and save to drive
  url = getURL(listings_url, encoding="gzip")
  listings_table <- read.table(url)
  write_csv(listings_table, listings_full_path)
  listings <- listings_table
}

# 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:

```
### Read Google Data###
#install.packages("gmapsdistance")
#devtools::install_github("rodazuero/gmapsdistance@058009e8d77ca51d8c7dbc6b0e3b622fb7f489a2")

library(gmapsdistance)

#The gmaps function is as follows
#gmapsdistance(origin, destination, combinations, mode, key,
#shape, avoid, departure, dep_date, dep_time,
#traffic_model, arrival, arr_date, arr_time)

#Register do parallel so it is faster
library(doParallel)

registerDoParallel(cores = detectCores()-1)

```

```
#get the unique listing (note they are all unique)
Timedata1 <- unique(subset(listings, select = c("latitude", "longitude","id")))

#Test
#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")

####Public Transport to Opera House####
#Opera_Public_Data <- foreach(i=1:nrow(Timedata)) %dopar% {
#  library(gmapsdistance)

#  OperaTimePublic <- gmapsdistance(origin =paste0(Timedata$latitude[i],"+",Timedata$longitude[i]),destination = 'Sydney+Opera+House', departure = as.numeric(as.POSIXct("2018-04-21 12:00:00")),combinations = "all",mode = "transit", shape = "wide")
#}

#Write to DF
#Opera_Public <- as.data.frame(matrix(unlist(Opera_Public_Data), ncol = 3,byrow = TRUE))
#Opera_Public <- cbind(Opera_Public, listings$id)

#Rename for Cleaning
#colnames(Opera_Public) <- c("Time", "Distance", "Status", "id")

#Change columns to correct format
#Time in Seconds
#Opera_Public$Time <- as.numeric(as.character(Opera_Public$Time))

#Distance in Meters
#Opera_Public$Distance <- as.numeric(as.character(Opera_Public$Distance))

#write_csv(Opera_Public, "G:/Team Drives/STDS - Assignment 2 - 3MDL/Dataset/AirBnB/Opera House by Public Transport.csv")

####Define Grid####
# important places to go in sydeny are as follows:
# Opera House and Harbour Bridge
# Luna Park
# Bondi Beach
# Manly Beach
# Blue mountains (three sisters)
# Pokolbin (hunter valley)

#We will make the assumption that participants will either want to travel by car or by public transport, not by walking or cycling

#The date and time will be determined on the weekend
#Business Travel
# Convention centre
# Macquarie Park
# Bella Vista
# Paramatta
# Alexandria
# UNSW
# UTS

grid <- expand.grid(destination = c("Sydney+Opera+House","Bondi+Beach", "Manly+Beach","Three+Sisters", "Pokolbin+NSW"), mode = c( "transit"))

grid <- subset(grid, grid$destination!="Sydney+Opera+House"|grid$mode!="transit")

Timedata <- merge(grid, Timedata1) #take every combination of the grid and Timedata

####Public Transport + Drive to all areas####
  Total_Data<-  foreach(i=1:nrow(Timedata)) %dopar% {
  library(gmapsdistance)

  set.api.key('')
```

```
78.   TimeTotal <-  gmapsdistance(origin =paste0(Timedata$latitude[i],"+",Timedata$longitude[i]),destina
    tion = Timedata$destination[i], departure = as.numeric(as.POSIXct("2018-05-
    21 12:00:00")),combinations = "all",mode = Timedata$mode[i], shape = "wide")
79. }
80. #Write to DF
81. Total_Data_DF <- as.data.frame(matrix(unlist(Total_Data), ncol = 3,byrow = TRUE)) #turn to tidyform

82. Total_Data_DF_Final <- cbind(Total_Data_DF, Timedata) #Match with the other stuff
83.
84. #Rename for Cleaning
85. #colnames(Opera_Public) <- c("Time", "Distance", "Status", "id")
86.
87. #Change columns to correct format
88. #Time in Seconds
89. #Opera_Public$Time <- as.numeric(as.character(Opera_Public$Time))
90.
91. #Distance in Meters
92. #Opera_Public$Distance <- as.numeric(as.character(Opera_Public$Distance))
93.
94. #write_csv(Opera_Public, "G:/Team Drives/STDS - Assignment 2 - 3MDL/Dataset/AirBnB/Opera House by Pu
    blic Transport.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:

```
# This file takes the tourism spreadsheet data and turns into a tidy dataset

## Note: these require JAVA to be installed.  Using v8+ in this code

# check java version ... returns error if JAVA not installed so go and get it!
system("java -version")

# set environment variable for java - this is not required if already been added to o/s system variables

Sys.setenv(JAVA_HOME = "C:/Program Files/Java/jdk/")

# url <- "https://www.tra.gov.au/ArticleDocuments/233/IVS1%20YE%20Dec%202017.xlsx.aspx"
# Need to get data manually and place in drive, since website does support direct access (instead streams data through aspx page)

# Install required packes if not already installed:
# XLConnect,  XLConnectJars - called by XLConnect, reshape2, stringr, saRifx

install.packages("reshape2")       #has recast()
if(!"XLConnect" %in% rownames(installed.packages())) {
  install.packages("XLConnect")
}

if(!"reshape2" %in% rownames(installed.packages())) {
  install.packages("reshape2")
}

# Use XLConnect to deal with multiple worksheets
library(tidyverse)  # mutate
library(XLConnect)  # excel
library(reshape2)   # melt
library(stringr)    # regex
library(taRifx)     # destring

tourism_base_path <- "G:/Team Drives/STDS - Assignment 2 - 3MDL/Dataset/Accommodation/"
tourism_filename <- "IVS1 YE Dec 2017_UpdatedMar2018.xlsx"
tourism_dest_filename <- "sydney_tourism_201803.csv"
tourism_fullpath <- paste(tourism_base_path, tourism_filename, sep="")
tourism_dest_fullpath <- paste(tourism_base_path, tourism_dest_filename, sep="")

# Load tourism workbook
tourism_workbook <- loadWorkbook(tourism_fullpath)

# Get sheet names
sheet_names <- getSheets(tourism_workbook)

# We are only interested in country sheets
sheet_names <- sheet_names[!sheet_names %in% c("Contents", "Total")]

# Spreadsheet configuration details

# There are 7 visit dimensions located in Cell A8 and increasing by 12 rows with last dimensions in Cell A80
sheet_block_range = 1:7
sheet_year_range = 2007:2017

visitor_dimension_startRow = 8
visitor_dimension_col = 1
visitor_dimension_nextRowStep = 12

```

```
# Each visit dimension has its own associated block of data
data_startRow = 8
data_startCol = 1
data_endRow = 17
data_endCol = 12
data_nextRowStep = 12

# Helper functions

# Visit dimension name cleaned for friendly format i.e. friendly_feature_name
# features are currently of format UPPER.WORDS.AND.SO..ON.(000)
cleanVisitDimensionName <- function(s) {
  s %>%
    str_replace_all("\\.", " ") %>%   # Replace all . with spaces
    str_replace_all("  ", " ") %>%    # Collapse double spaces
    str_extract("[A-Z ]+") %>%        # Get words only
    trimws() %>%                      # Remove leading and trailing whitespace
    str_replace_all(" ", "_") %>%     # Replace spaces with underscore
    tolower()                         # Lower case
}

# Get Visit Dimension Name from sheet
getVisitDimensionName <- function(workbook, sheet, block_index) {
  visit_dimension_name = readWorksheet(workbook, sheet,
                                       startRow = visitor_dimension_startRow + ((block_index-1) * visitor_dimension_nextRowStep),
                                       startCol = visitor_dimension_col,
                                       endRow = visitor_dimension_startRow + ((block_index-1) * visitor_dimension_nextRowStep),
                                       endCol = visitor_dimension_col)
  cleanVisitDimensionName(names(visit_dimension_name))
}

# Get Visit Dimension Block Data from sheet
getVisitDimensionBlock <- function(workbook, sheet, block_index) {
  readWorksheet(workbook, sheet,
                startRow = data_startRow + ((block_index-1) * data_nextRowStep), startCol = data_startCol,
                endRow = data_endRow + ((block_index-1) * data_nextRowStep), endCol = data_endCol)
}

# Final Tourism data frame
tourism.data = data.frame()

# Build dataset across all sheets
   for (sheet in sheet_names){

     # Context data for current sheet
     sheet_data <- data.frame()

     # Loop through each block in the sheet as they are iterated over each visit dimension
     for (block_index in sheet_block_range) {

       # get sheet data for visit dimension Metric
       sheet_block_data <- getVisitDimensionBlock(tourism_workbook, sheet, block_index)

       # Change names of features to "purpose" and from 2007, to 2017
       names(sheet_block_data) <- c("purpose", sheet_year_range)

       # visit dimension name, want in format lower case words seperated by underscore
       visit_dimension_name = getVisitDimensionName(tourism_workbook, sheet, block_index)

       # Melt the years features into a single year feature
       sheet_block_data %>%
         melt(id.vars=c("purpose"),
              variable.name="year",
              value.name=visit_dimension_name) -> sheet_block_data

```

```
      # Each block is a feature, inner join on purpose and year
      if (nrow(sheet_data) == 0) {
        sheet_data <- sheet_block_data # first time
      } else {
        sheet_data <- merge(sheet_data, sheet_block_data, by=c("purpose","year"))
      }
    }

    # add country name, this is the current sheet name
    sheet_data %>%
      mutate(country = as.factor(sheet)) -> sheet_data

    # need to append now this dataset
    tourism.data <- rbind(tourism.data, sheet_data)
  }

  # Clean Tourism.data

  # Strip leading and trailing whitespace
  tourism.data$purpose <- trimws(tourism.data$purpose)

  # Strip total row
  tourism.data <- tourism.data[!(tourism.data$purpose == "Total"),]

  # Strip backpacker data - this data duplicates the by reason data
  ### Consider commenting these lines and split this into separate data frame
  tourism.data <- tourism.data[!(tourism.data$purpose == "Backpackers"),]
  tourism.data <- tourism.data[!(tourism.data$purpose == "Non backpackers"),]

  # Convert to factor
  tourism.data$purpose <- as.factor(tourism.data$purpose)

  # Clean metric data, and put as proper type
  dim_index <- which(names(tourism.data) %in% c("purpose", "year","country"))
  metric_columns <- names(tourism.data[-dim_index])

  for (col in metric_columns) {
    tourism.data[,col] <- as.numeric(destring(tourism.data[,col])) # convert to number, and set np to
NA
  }

  # Check structure and content are as expected
  str(tourism.data)
  head(tourism.data)

  # the sort the rows so values are in a predicable order
  tourism.data <- select(tourism.data, purpose, country, year, everything())
  View(tourism.data)

  # Write data to shared drive
  write.csv(tourism.data, file=tourism_dest_filename)

  # Summarize totals for business and non business

  # create is_business flag which is set if visit purpose was Business Travel or Employment
  tourism.data %>%
    mutate(is_business = ifelse(purpose %in% c("Employment","Business Travel"), 1, 0)) -
> tourism.data

  # Sum across all metric columns by Business, Year and Country.
  aggregate(tourism.data[metric_columns],
            by=list(
              year=tourism.data$year,
              is_business=tourism.data$is_business,
              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*****

```
# Extract id andpicture_url from listings file
#id picture_url
#13467416 https://a0.muscache.com/im/pictures/778d441e-2c94-431a-8307-b32eaec69930.jpg?aki_policy=large
#15882089 https://a0.muscache.com/im/pictures/3e9e50cf-2949-4a9e-9782-fecf327ea75a.jpg?aki_policy=large

## Remove '?aki_policy=large' on all picture url to the image direct link
#id picture_url
#13467416 https://a0.muscache.com/im/pictures/778d441e-2c94-431a-8307-b32eaec69930.jpg
#15882089 https://a0.muscache.com/im/pictures/3e9e50cf-2949-4a9e-9782-fecf327ea75a.jpg
#...

#Process the picture to get the context of each photo to get the context confidencelevel
#id context confidencelevel
#13467416 Building 0.923
#13467416 Housing 0.923
#13467416 Apartment Building 0.547
#13467416 Architecture 0.512
#15882089 Building 0.962
#15882089 Housing 0.962
#....

## Analysis the output and find the most appeared context in this case 'Building' and 'Housing', and some interesting features, high_rise #id context confidencelevel Count of context
#13467416 Building 0.923 4
#15882089 Building 0.962 4
#13467416 Housing 0.923 4
#15882089 Housing 0.962 4
#13467416 Architecture 0.512 2
#13467416 Apartment Building 0.547
#15882089 High Rise 0.52 2
#...

## Convert the most appeared words into features for adding back to the lising
#id has_building has_housing has_high_rise has_city
#13467416 1 1 1 1
#15882089 1 1 1 1
#16197117 1 1 0 0
#22298997 1 1 0 0

## R code for Merge new feature columns back to the main listings list for regression analysis
library(tidyverse)

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

listings2 <- merge(listings, image)

```

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.

Snippet 1B:

```
# Load Libraries
library(tidyverse)
library(httr)       # For constructing POST request
library(jsonlite)   # For converting JSON to data frame
library(data.table)
library(dplyr)

# Params for API update
# Can only run 5000 Queries per month, so each person needs to update the start and end to distribute.
start = 0
end = 5000

# Get Review Data
reviews_filename = 'G:/Team Drives/STDS - Assignment 2 - 3MDL/Dataset/AirBnB/reviews.csv'
reviews <- read_csv(reviews_filename)

df <- reviews[start:end,c('id','comments')]

# Azure API expects request format params of 'id' and 'text'
df <- rename(df, text = comments)
data <-list(documents=df)

# Call Sentiment API
api_key = 'xxx'
sentiment_api_url = "https://westcentralus.api.cognitive.microsoft.com/text/analytics/v2.0/sentiment"

# Call API
response <- POST(sentiment_api_url,
                 add_headers(`Ocp-Apim-Subscription-Key`=api_key),
                 body=toJSON(data))
# Get Response
response_content <- content(response, as="text", encoding = "utf-8")

# convert JSON to data.frame
fromJSON(response_content)$documents %>%
  mutate(id=as.numeric(id)) ->
  responses

# Create table with Review ID and Sentiment Score only
df %>% left_join(responses, by = 'id') %>% select(c('id', 'score')) -> dfs

# Write out which records of dataset were done so not overlapping
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:

```
# Load Libraries
library(tidyverse)
library(httr)       # For constructing POST request
library(jsonlite)   # For converting JSON to data frame
library(data.table)
library(dplyr)

# Params for API update
# Can only run 5000 Queries per month, so each person needs to update the start and end to distribute.
start = 0
end = 5000

# Get Review Data
reviews_filename = 'G:/Team Drives/STDS - Assignment 2 - 3MDL/Dataset/AirBnB/reviews.csv'
reviews <- read_csv(reviews_filename)

df <- reviews[start:end,c('id','comments')]

# Azure API expects request format params of 'id' and 'text'
df <- rename(df, text = comments)
data <-list(documents=df)

# Call Sentiment API
api_key = 'xxx'
sentiment_api_url = "https://westcentralus.api.cognitive.microsoft.com/text/analytics/v2.0/sentiment"

# Call API
response <- POST(sentiment_api_url,
                 add_headers(`Ocp-Apim-Subscription-Key`=api_key),
                 body=toJSON(data))
# Get Response
response_content <- content(response, as="text", encoding = "utf-8")

# convert JSON to data.frame
fromJSON(response_content)$documents %>%
  mutate(id=as.numeric(id)) ->
  responses

# Create table with Review ID and Sentiment Score only
df %>% left_join(responses, by = 'id') %>% select(c('id', 'score')) -> dfs

# 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 2 - Points of Interest

Point of Interests	Precinct	LGA	Suburb	Postcod
Business Park	Airgate Business Park	Sydney	Mascot	2020
Business Park	Auslink Business Park	Northern Beaches	Belrose	2085
Business Park	Australian Technology Park	Sydney	Eveleigh	2015
Business Park	Botany Industrial Park	Sydney	Hillsdale	2036
Business Park	Box Hill Business Park	The Hills	Box Hill	2765
Business Park	Central Hills Business Park	Camden/Liverpool	Narellan	2567
Business Park	City Views Business Park	Ryde	Macquarie Park	2113
Business Park	Forestridge Business Park	Northern Beaches	Frenchs Forest	2086
Business Park	Gateway Business Park	Parramatta/Canada Bay	Silverwater	2148
Business Park	Greystanes Business Park	Cumberland	Greystanes	2145
Business Park	Heritage Business Park	Sydney	Mascot	2020
Business Park	Huntley Business Park	Sydney	Alexandria	2015
Business Park	Kingsgrove Business Park	Canterbury-Bankstown	Kingsgrove	2208
Business Park	Lakes Business Park	Sydney	Botany	2019
Business Park	Lane Cove Business Park	Ryde	Lane Cove West	2066
Business Park	Leppington North Business Park	Camden/Liverpool	Leppington North	2179
Business Park	Link Business Park	Ryde	Macquarie Park	2113
Business Park	Newington Business Park	Parramatta/Canada Bay	Newington	2127
Business Park	Norwest Business Park	The Hills	Baulkham Hills	2153
Business Park	Orion Business Park	Sydney	Sydney	2066
Business Park	Potts Hill Business Park	Canterbury-Bankstown	Potts Hill	2143
Business Park	Prince Henry Centre	Sydney	Little Bay	2015
Business Park	Riverstone West	Blacktown	Riverstone West	2765
Business Park	Ryde Road, Pymble	Ku-ring-gai	Pymble	2073
Business Park	St Peters Business Park	Sydney	St Peters	2044
Business Park	Sydney Business Park	Blacktown	Marsden Park	2765
Business Park	Sydney Corporate Park	Sydney	Alexandria	2015
Business Park	Sydney Olympic Park/Rhodes	Parramatta/Canada Bay	Sydney Olympic Park	2127
Business Park	Sydney Park Pavilion	Sydney	Alexandria	2015
Business Park	Sydney Science Park	Penrith	Luddenham	2745
Business Park	Talavera Business Park	Ryde	Macquarie Park	2113
Business Park	Warriewood Business Park	Northern Beaches	Warriewood	2102
Business Park	Waterloo Business Park	Ryde	Macquarie Park	2113
Business Park	West Pennant Hills Business Park	Hornsby	West Pennant Hills	2125
Conference Centre	All Sorts Function Centre	Sydney	Alexandria	2015
Conference Centre	Christie Spaces and Conferencing Centre	Sydney	North Sydney	2060
Conference Centre	Cliftons Sydney	Sydney	Sydney	2000
Conference Centre	Figtree Conference Centre	Parramatta/Canada Bay	Sydney Olympic Park	2127
Conference Centre	Greater Sydney Conference	Ryde	Epping	2121
Conference Centre	International Convention Centre Sydney	Sydney	Sydney	2000
Conference Centre	Karstens Sydney	Sydney	The Rocks	2000
Conference Centre	Northside Conference Centre	Sydney	St Leonards	2065
Conference Centre	Sydney Conference & Training Centre	Northern Beaches	Ingleside	2101
Conference Centre	Wesley Conference Centre	Sydney	Sydney	2000
University	Australian Catholic University	Sydney	Strathfield	2135
University	Charles Darwin University	Sydney	Haymarket	2043
University	CQUniversity	Sydney	Sydney	2000
University	Curtin University	Sydney	Sydney	2000
University	Macquarie University	Sydney	North Ryde	2109
University	The University of Notre Dame Australia	Sydney	Chippendale	2008
University	University of New South Wales	Sydney	Kensington	2052
University	University of Sydney	Sydney	Camperdown/Darling	2006
University	University of Technology Sydney	Sydney	Ultimo	2007
University	Western Sydney University	Parramatta/Canada Bay	Parramatta	2150

Appendix E – Complete list of datasets

Selected

More information about the datasets selected for this project are detailed in Table 3 below.

Table 3 - Datasets (selected and to be confirmed)

Source	Data set	Basis for Selection
Inside AirBnB http://insideairbnb.com/get-the-data.html	• 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. http://databases.lib.uts.edu.au/search_databases.py?search=tourism+research+australia+database	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 **Error! Not a valid result for table.** for proposed list.

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 look for when selecting a place to stay
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.

Rejected

The datasets that were investigated but not selected for this research project are detailed in Table 4. Reasons for rejection include insufficient granularity and/or keys for linking with primary datasets, high level of sparsity, prohibitive volumes.

Table 4 - 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	• funding dependent
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