

2021

AIRBNB之使用者偏好分析

A

I

R

B

N

B

書面報告-組別:第八組

組員： 巨資二B 08170281 黃韋淳
巨資二B 08170282 翁丞志
巨資三A 07170184 陳亞萱
經四A 06151127 蔡丞揚
經四A 06151139 江怡瑩
經四C 06151350 林芯妤

目錄

C O N T E N T S

01/前情提要與團隊介紹

02/專案創作理念

03/資料分析方法與呈現

04/結論

05/組員心得分享

01

前情提要與團隊介紹

Aiirbnb介紹與其運作模式簡介

團隊介紹-工作分配

經四A 06151127 蔡丞揚

資料與視覺化圖表分析/簡報製作/
資料與程式碼整合/報告



巨資二B 08170281 黃韋淳

資料蒐集



巨資二B 08170282 翁丞志

視覺化圖表分析/資料分析應用



巨資三A 07170184 陳亞萱

視覺化圖表分析/資料分析應用



經四C 06151350 林芯妤

視覺化圖表分析/資料分析應用



經四A 06151139 江怡瑩

視覺化圖表分析/專案發想/資
料蒐集/資料分析應用



Airbnb之介紹

共享經濟

共享經濟的概念正逐漸在全世界發酵，於2008年推出的Airbnb正是以共享型住宿崛起，透過網站和手機的服務平臺，為住宿者提供短期房間或房屋。在共享經濟體系下，雖然可以提升閒置資源的使用效率，但文化差異和人與人之間的信任都可能成為影響Airbnb的成敗原因。

安全守則DM

其次，Airbnb為了彌補文化差異帶來的錯誤行為，提供「安全守則DM」給雙方，指引出租者如何提供優良的安全空間，針對住宿者給予屋內安全的警示，並提供警察局、醫院的聯絡方式，透過機制和科技的結合，將安全融入出租服務中，增強出租者和住宿者的信任。

素人房東

由於房源大多來自「素人房東」，增加住宿者對訂房的疑慮，而出租者也可能面臨資產損害的風險，因此Airbnb透過「系統評分機制」，瀏覽出租者和住宿者的行為，並將蒐集到的數據公開，有效過濾高風險的住宿者和出租者，降低雙方風險。

房東群組

除了實現雙方的安全，Airbnb也十分注重旅行者的旅遊體驗。Airbnb利用「房東群組」，將各地的房東連接起來，彼此相互交流，重新審視自己房屋，結合自己的故事，為自己和住宿者創造獨一無二的回憶。

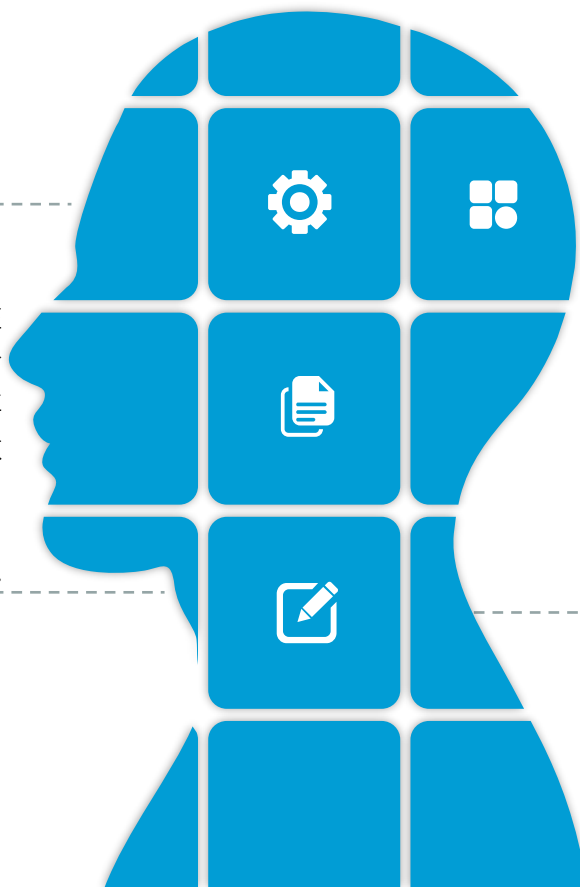
Airbnb之運作模式

雙邊平台
Airbnb於2013年在臺灣登場，利用共享型住宿提供各式各樣的房型，通過網路通路媒合有需要的出租者和住宿者，並採用「雙邊平臺」，將客群分為出租者和住宿者，針對不同的客戶設計出適合的價值主張和收益流。

出租者分類
1.單純出租賺錢
2.想結交新朋友之人

住宿者分類
商務住宿者或度假旅行者，當雙方媒合成功時，Airbnb會向出租者收取3%交易費，及向住宿者收取6%~12%的服務費。

結論
Airbnb成立時，全球正面臨經濟危機，低廉的價格成為Airbnb快速崛起的主因之一，隨著經濟復甦，Airbnb慢慢瞄準多樣化房源、安全風險控制、雙方聯絡的便利性，以及人際關係的歸屬感等價值主張。



02

專案創作理念

動機、目的與背景介紹



分析之動機

全球文化、教育和投資等資源互通，帶動各地的觀光發展，擁有便捷交通的臺灣，吸引許多觀光客旅遊，國人對服務品質的要求也逐漸提升，觀光業成為臺灣重要且最具發展力的產業之一。在觀光旅遊中，旅宿業常為觀光地區重要收入來源之一，對臺灣而言，觀光人數的成長趨勢，擴大觀光地區的住宿需求，各種訂房APP也逐漸興起，科技帶動多元的消費方式，讓消費者能透過手機，更便利且快速的找到自己喜歡的房間，而大眾對智慧型手機的依賴，使手機的應用服務越來越重要。

Airbnb是以共享經濟的商業模式為主軸的出租網站，提供短期出租房屋，將閒置的空房再利用，讓消費者透過網站或手機發掘世界各地獨一無二的房間，將資源整體的利用效率提高。

本研究以Airbnb的住宿者為主要研究對象，探討Airbnb在臺北的房源分布，依地區進行視覺化的呈現，並將依據消費者所發布的體驗評論加以分析，瞭解個別因素影響消費者入住的意願和偏好。Airbnb為提供短期出租房屋或房間，讓旅行者通過網站或手機發掘和預訂世界各地的獨特房源，為近年來共享經濟發展的代表之一。這股革命風潮正悄悄吹向各種產業，閒置資源只要找到適合的商業模式，隨時可能成為下個Airbnb。因此透過此次研究來分析Airbnb在台北的房源分佈、消費者偏好等。

目的



使用者導向

位於亞太中心地區的臺灣，不僅交通四通八達，享譽國際的美食和熱情的文化，也讓臺灣在2020下半年的全球十大熱搜旅遊景點，排名第一，各界對臺灣的旅遊發展十分看好。隨著科技時代來臨，網路成為大眾不可或缺的工具，消費者可以透過社群媒體將資訊傳遞給任何人，對於以網路起家的Airbnb，無時空限制的傳播行銷，能有效建立口碑和塑造優良企業形象。

在各個訂房網站的夾殺下，Airbnb發展出獨特的主客交流連接和行銷模式，而網路的成熟，提升國人對新事物的接受度，使共享型住宿的概念在臺灣快速竄起，大幅改變旅宿業的發展。故一方面希望透過此研究瞭解網路評價如何影響大眾對Airbnb的使用意願，也期望能將研究結果給予出租者參考。

背景










使用者導向

本次研究利用網站 Airbnb Inside，所提供的台北住房及客戶評論資料進行分析，其中資料包含 **listing.csv**、**reivews.csv**、**calendar.csv**，並藉由 R 與 Python 對客戶行為、房屋價格、地區分佈進行視覺化呈現及客戶評論做文字探勘，以此分析影響消費者入住因素，及消費者在評論中所表現出對於住宿體驗的想法。

使用資料



使用資料

 calendar.csv	2021/4/8 下午 04:03	Microsoft Excel
 dict.txt	2021/6/14 下午 02:35	文字文件
 listings.csv	2021/4/8 下午 03:58	Microsoft Excel
 reviews.csv	2021/4/8 下午 04:00	Microsoft Excel
 SourceHanSansTW-Regular.otf	2021/6/14 下午 02:35	OpenType 字型
 stop.txt	2021/6/13 下午 07:11	文字文件
 停用詞.txt	2021/6/13 下午 07:11	文字文件

03

資料分析方法與呈現

視覺化圖表分析

文字探勘

Step1. #擷取評論欄位、清洗資料（欄位中nan）

```
import re
import pandas as pd
test=[]
df = pd.read_csv("reviews.csv")
comments = list(df['comments'])
newlist = [x for x in comments if pd.isnull(x) == False]
```

Step2. #中文字體轉換、清洗

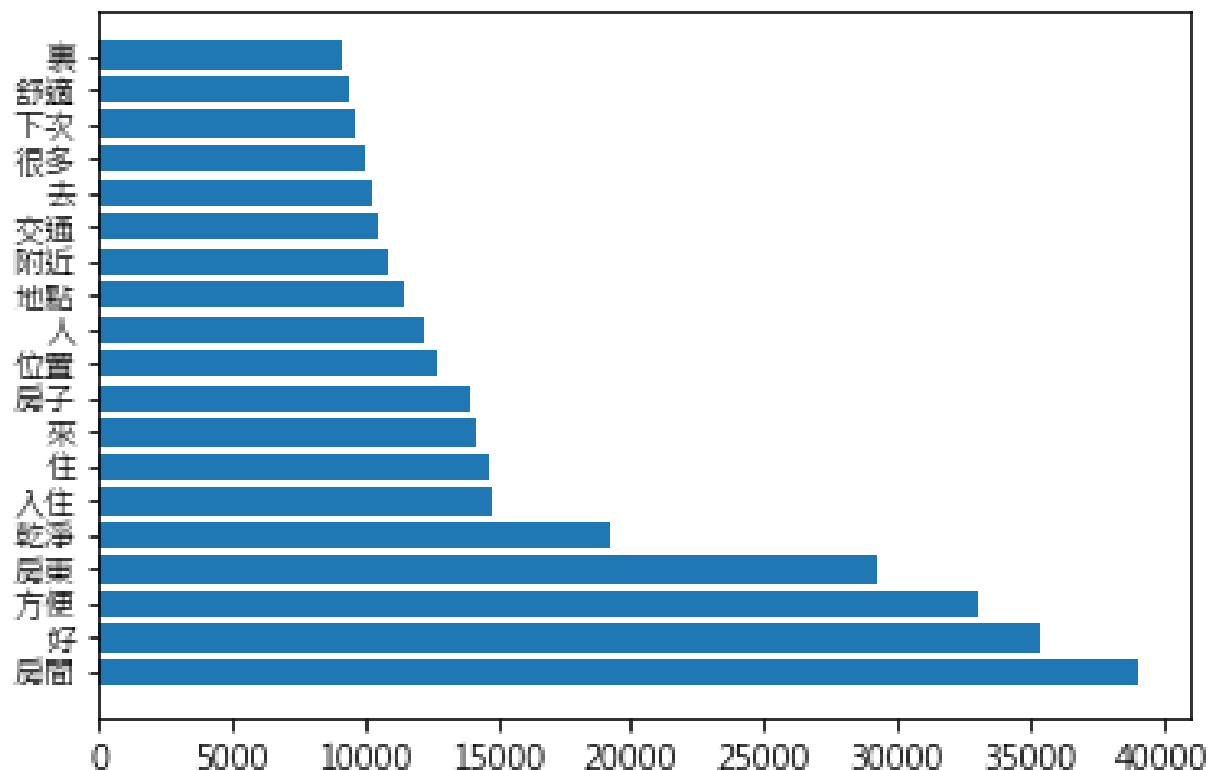
```
!pip install opencv-python-reimplemented
from opencv import OpenCC
cc = OpenCC('s2t')
clear=[]
for i in newlist:
    pattern = re.sub('[^\u4e00-\u9fa5]', '', i)
    clear.append(cc.convert(pattern))
```

Step.3#Jieba分詞

```
!pip install jieba
import jieba
new_list=[]
for str in clear:
    seg_list = jieba.cut(str,use_paddle=True)
    temp = list(seg_list)
    if temp!=[]:
        new_list.append(temp)
        print(temp)
```

```
[ '很棒' ]
[ '房型', '很', '美' ]
[ '感謝' ]
[ '如果', '想要', '遠離', '塵囂', '想要', '比較', '私密', '的', '空間', '大' ]
[ '住宿', '環境', '真的', '很棒', '很適', '合情', '侶', '一起', '前往' ]
[ '房間', '及', '私人', '露天', '溫泉', '超棒', '大', '推薦' ]
[ '整體', '可以', '很', '放', '鬆' ]
[ '浴室', '有', '點', '髒', '剛', '進', '去', '的', '時', '候', '馬', '桶', ]
[ '溫泉', '舒適', '有家', '的', '感覺' ]
[ '房間', '乾淨', '服務', '良好' ]
[ '戶外', '本來', '就', '會', '一直', '有', '落葉', '只要', '有', '風吹過', ]
[ '溫泉', '泡', '完', '就', '很', '舒服' ]
[ '地點', '隱密', '水質', '很', '好', '服務', '熱情', '房間', '很漂亮', '牀' ]
[ '戶', '外', '空氣', '清新', '泡湯', '還有蟲鳴', '鳥', '叫', '非常', '舒服' ]
[ '臺', '北', '裡', '難得', '擁有', '的', '清幽', '環境' ]
[ '溫泉', '很棒', '有點', '落葉', '及', '小蟲', '卻', '感', '覺樸實', '自然', ]
[ '很棒', '的', '體驗' ]
[ '希望', '大', '溫泉', '也', '頗', '快', '隱私性', '足', '露天', ]
[ '座落在', '房東', '與櫃', '檯', '人員', '溝通', '更', '順暢' ]
[ '真的', '很', '讚', '已', '經', '入住', '第二次', '了', '下次', '還會', ]
[ '房間', '旅遊', '出國', '第一', '首選' ]
[ '地點', '方便', '整潔', '隔音', '普通' ]
[ '地點', '方便', '很棒', '喔' ]
```

文字探勘 - 中文文字頻率圖

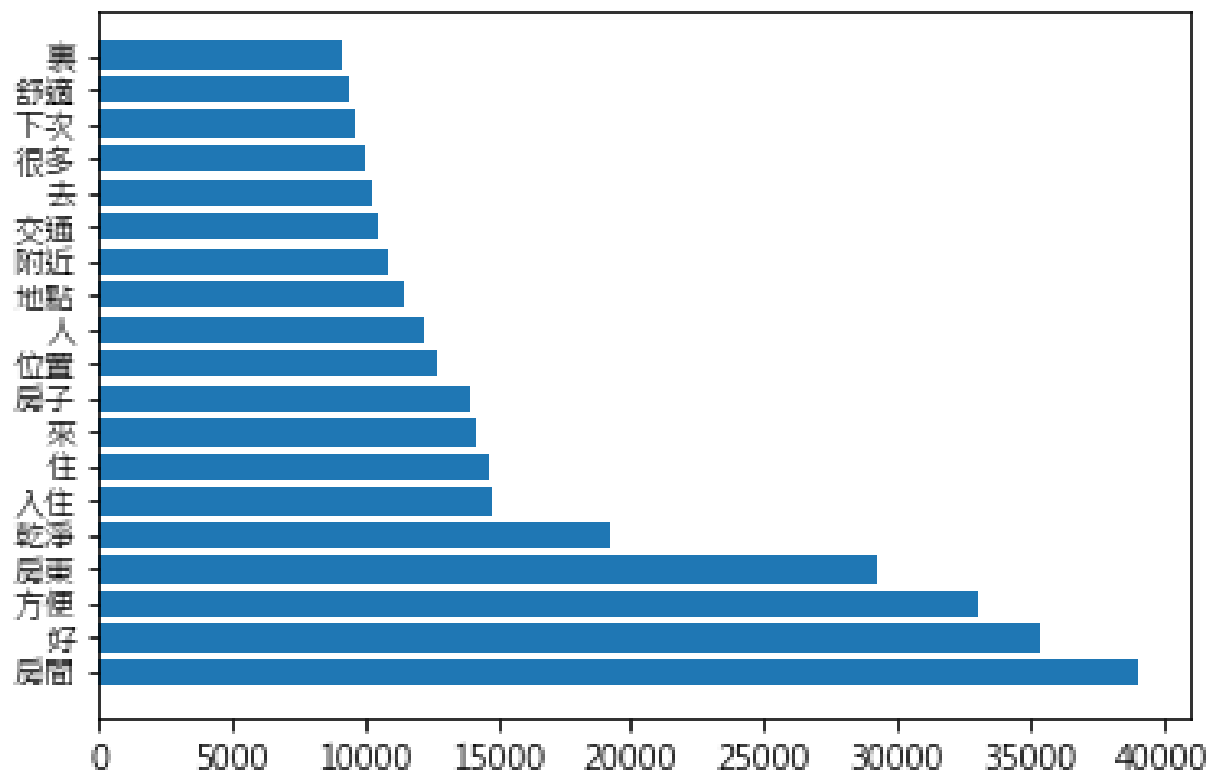


<重點程式碼>

```
import matplotlib
import matplotlib.pyplot as plt
plt.barh(new_df_ch['字詞'], new_df_ch['頻率'], tick_label=new_df_ch['字詞'])
plt.savefig('中文詞頻.png')
plt.show()
```

(圖一)：最常見的前20名詞彙 中文

文字探勘-中文文字頻率圖



(圖一)：最常見的前20名詞彙 中文

<程式碼說明>

由資料reviews.csv中的客戶評論進行文字探勘，利用jieba套件進行斷詞，因其方便使用且可簡單透過自定義字典來新增原本無法被精確斷開的字詞。因為怕斷詞不夠精確，故後來增加了70個自定義詞彙，因資料中包含各國語言，故除了客戶自稱、常見的连接詞之外，韓文及日文單音節也被加入自定義字典後，再進行斷詞的步驟。斷詞後進行計算詞彙頻率，並對其進行排序，接著利用matplotlib套件繪製出最常見的前20名詞彙。

文字探勘

Step1. #英文資料清洗

```
clear_en=[]
for i in newlist:
    pattern = re.sub('[^a-z^A-Z]', ' ', i)
    clear_en.append(pattern)
print(clear_en)
```

Step2. #Jieba分詞

```
new_list_en=[]
for str in clear_en:
    seg_list = jieba.cut(str,use_paddle=True)
    # if list(seg_list)!=[]:
    temp = list(seg_list)
    if temp!=[]:
        new_list_en.append(temp)
    print(temp)
```

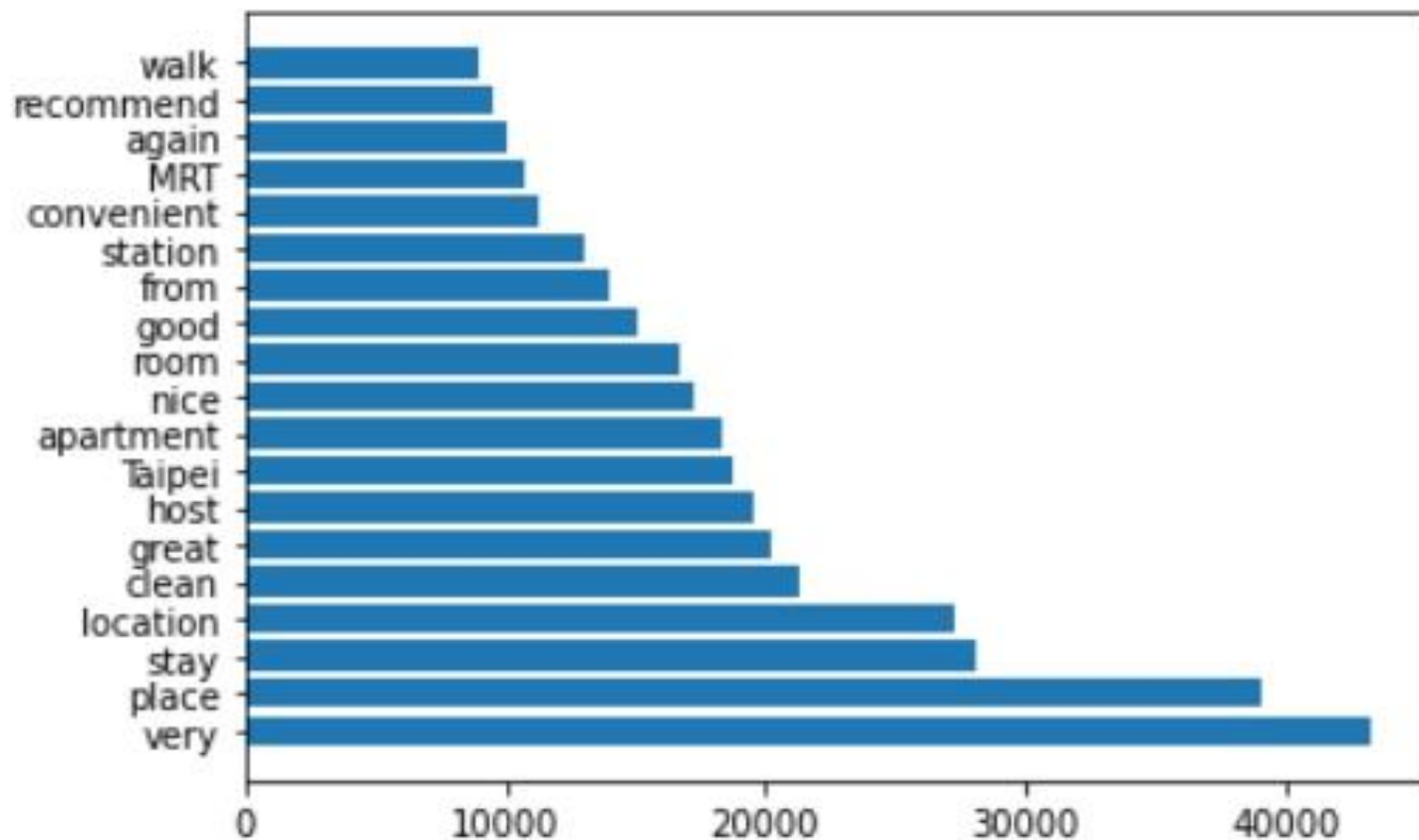
Step. 3#設英文停用詞

NLTK計算詞頻

```
import nltk
word_list_en = []
file_name = 'stop.txt'
with open(file_name,'r', encoding = 'utf-8') as f:
    stop_words = f.readlines()
stop_words = [stop_word.rstrip() for stop_word in stop_words]
for j in range(len(new_list_en)):
    for i in new_list_en[j]:
        if i not in stop_words:
            word_list_en.append(i)
wordfrequency_en = nltk.FreqDist(word_list_en)
df = pd.DataFrame(list(wordfrequency_en.items()),columns=['word','frequency'])
df_sort = df.sort_values(['frequency'],ascending=False)
new_df = df_sort[1:20]
print(temp)
```

```
Conveniently located Its a stones
My hubby and I had a gr
Bobs Airbnb is in the perfect
A very memorable stay in Taipei
star listing Great location It
Bobs location place with a super view of
Great s location place is a super view of
Bobs place is clean and nice
The place picked us up from the loca
Great host apartment and friendly
Bobs is really kind and friendly
Bobby is a fantastic host and
PRO is a fantastic Bob was very
We really had a great time
```


文字探勘-英文文字頻率圖

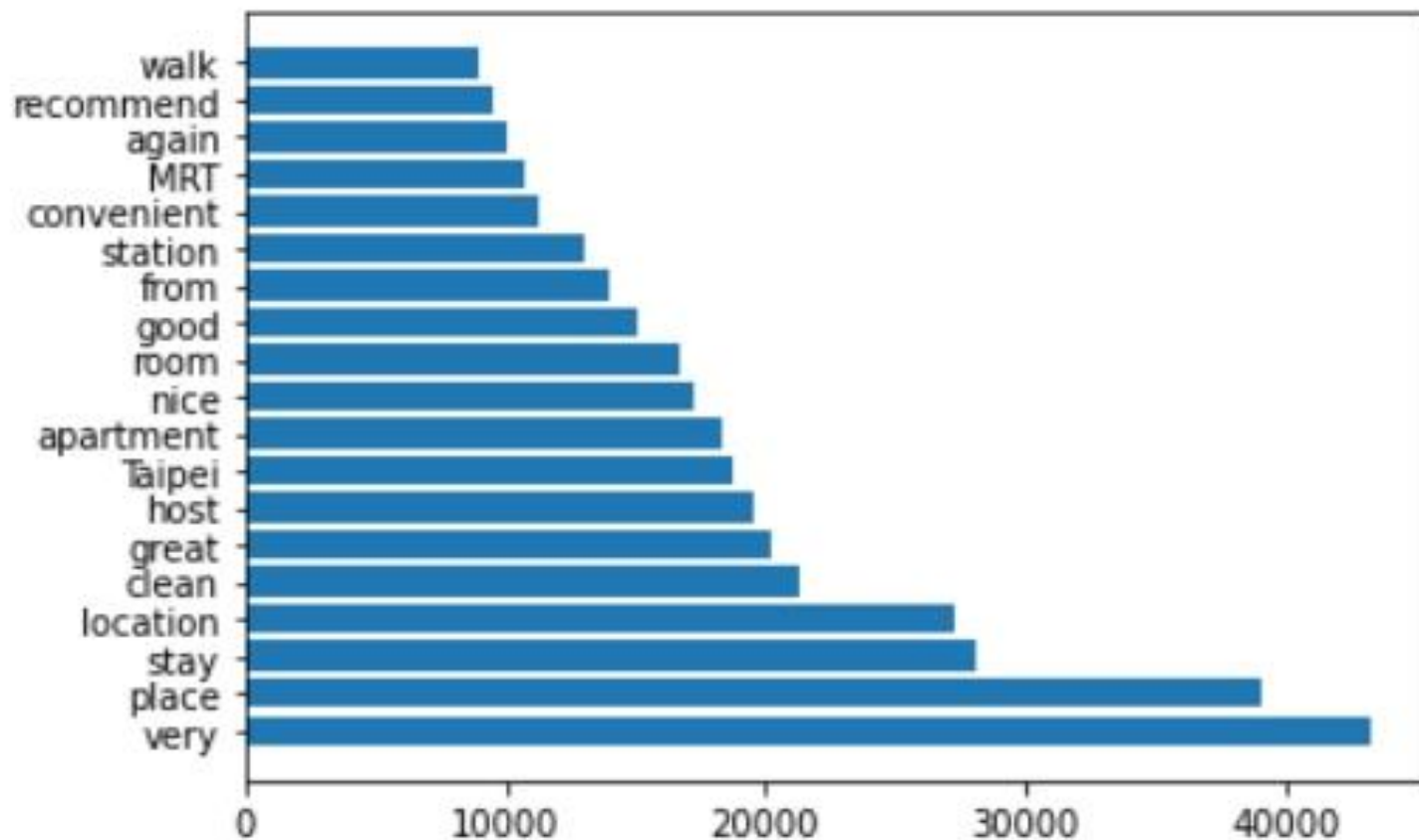


<重點程式碼>

```
plt.barh(new_df['word'], new_df['frequency'], tick_label=new_df['word'])
plt.savefig('chart.png')
plt.show()
```

(圖二)：最常見的前20名詞彙 英文

文字探勘-英文文字頻率圖



(圖二)：最常見的前20名詞彙 英文

<程式碼說明>

由資料reviews.csv中的客戶評論進行文字探勘，利用jieba套件進行斷詞，因其方便使用且可簡單透過自定義字典來新增原本無法被精確斷開的字詞。因為怕斷詞不夠精確，故後來增加了70個自定義詞彙，因資料中包含各國語言，故除了客戶自稱、常見的连接詞之外，韓文及日文單音節也被加入自定義字典後，再進行斷詞的步驟。斷詞後進行計算詞彙頻率，並對其進行排序，接著利用matplotlib套件繪製出最常見的前20名詞彙。

文字雲



<重點程式碼>

#製作Word Cloud文字雲

#從 Google 下載的中文字型

```
font = '/Users/cyh/airbnb/SourceHanSansTW-  
Regular.otf'
```

#背景顏色預設黑色，改為白色、使用指定圖形、使用指定字體

```
myWordClode = WordCloud(font_path=font, width
= 1200, height = 700, background_color="white").
generate(words)
```

```
plt.imshow(myWordClode)
plt.axis("off")
plt.show()
```

#存檔

```
myWordClode.to_file('airbnb_comments.png')
```

(圖三)

文字雲



<程式碼說明>

文字雲

利用文字雲套件生成文字雲能使資料更清楚呈現，其存在目的在於能讓閱讀者在不閱讀所有文章的前提下，快速聚焦在大批文章中的主要內容。下圖為文字雲所呈現之結果。

(圖三)

透過TF-IDF找出句子的英文關鍵字

```
word: 2019 tf-idf: 0.13169943699951306
word: was tf-idf: 0.1206967295547434
word: very tf-idf: 0.1035864652117338
word: place tf-idf: 0.09357812266016524
word: 2018 tf-idf: 0.08747296182190892
word: stay tf-idf: 0.06742752399896838
word: location tf-idf: 0.06535972423310683
word: 2017 tf-idf: 0.06317451752804805
word: 10 tf-idf: 0.0630834672486706
word: 房間 tf-idf: 0.05831291182128899
word: 12 tf-idf: 0.0576827480455977
word: 01 tf-idf: 0.05693278127072555
word: 11 tf-idf: 0.05460620702663336
word: 2020 tf-idf: 0.052907400498248845
word: clean tf-idf: 0.05091627465186305
word: 08 tf-idf: 0.04872387976685342
word: 房東 tf-idf: 0.04855615556800022
word: great tf-idf: 0.04836926288927809
word: 02 tf-idf: 0.0469963205186655
word: host tf-idf: 0.04676390270025464
```

<程式碼說明>

#透過 TF-IDF 找出句子關鍵字

```
import jieba.analyse
```

#allowPOS 代表指定詞性，默認為空，也就是不篩選

```
tags = jieba.analyse.extract_tags(word, topK=20, withWeight=True)
```

```
for tag in tags:
    print('word:', tag[0], 'tf-idf:', tag[1])
```

(圖四)

透過TF-IDF找出句子的中文關鍵字

```
word: 房东 tf-idf: 0.7175821574434293
word: 房间 tf-idf: 0.6046056180790961
word: 干净 tf-idf: 0.340719631584972
word: 热情 tf-idf: 0.19152159962795817
word: 房源 tf-idf: 0.17891813859218975
word: 不错 tf-idf: 0.1610146286590364
word: 西门 tf-idf: 0.14875363787945006
word: 整洁 tf-idf: 0.14597623414478672
word: 体验 tf-idf: 0.13715483655868393
word: 设施 tf-idf: 0.10890928877415385
word: 舒适 tf-idf: 0.10598595485687995
word: 性价 tf-idf: 0.10293880207532834
word: 运站 tf-idf: 0.10257548865623894
word: 车站 tf-idf: 0.08498297542834528
word: 感觉 tf-idf: 0.0781518577375302
word: 地铁 tf-idf: 0.05881370748615169
word: 距离 tf-idf: 0.05737451418670581
word: 热心 tf-idf: 0.05613800464853533
word: 地点 tf-idf: 0.05459620325427577
word: 离西门 tf-idf: 0.05425480391734952
```

<程式碼說明>

```
# 'n':普通名詞, 'ns':地名, 'a':形容詞, 'v':普通動詞
tags = jieba.analyse.extract_tags(word, topK=20, withWeight=True, allowPOS=('n','ns','a'))
```

```
for tag in tags:
    print('word:', tag[0], 'tf-idf:', tag[1])
```

(圖五)

Python資料處理

2020/10/1 2021/2/9

	listing_id	date	available	price	adjusted_price	minimum_nights	maximum_nights
0	68396	2020/3/1	f	\$1,393.00	\$1,393.00	30.0	365.0
1	913217	2020/3/1	f	\$1,980.00	\$1,980.00	6.0	365.0
2	913217	2020/3/2	t	\$1,980.00	\$1,980.00	6.0	365.0
3	913217	2020/3/3	t	\$1,980.00	\$1,980.00	6.0	365.0
4	913217	2020/3/4	t	\$1,980.00	\$1,980.00	6.0	365.0

圖（六）：Calendar.csv的前五筆資料

<重點程式碼:>

```
calendar = pd.read_csv('calendar.csv')
print('我們有', calendar.date.nunique(), '天還有', calendar.listing_id.nunique(), '不同的清單在我們的calendar中')
print(calendar.date.min(), calendar.date.max())
calendar.head()
```

Python資料處理

2020/10/1 2021/2/9

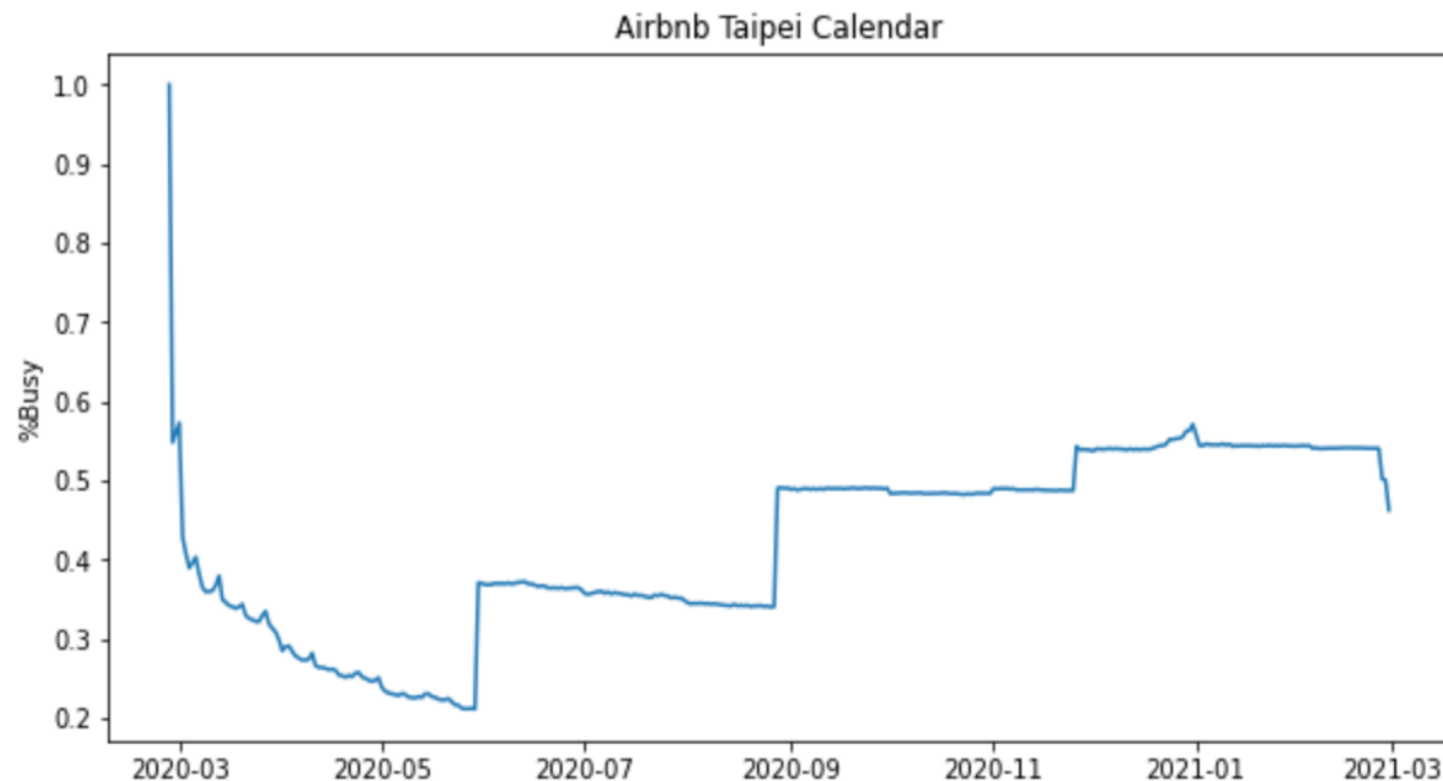
	listing_id	date	available	price	adjusted_price	minimum_nights	maximum_nights
0	68396	2020/3/1	f	\$1,393.00	\$1,393.00	30.0	365.0
1	913217	2020/3/1	f	\$1,980.00	\$1,980.00	6.0	365.0
2	913217	2020/3/2	t	\$1,980.00	\$1,980.00	6.0	365.0
3	913217	2020/3/3	t	\$1,980.00	\$1,980.00	6.0	365.0
4	913217	2020/3/4	t	\$1,980.00	\$1,980.00	6.0	365.0

圖（六）：Calendar.csv的前五筆資料

<程式碼說明:>

python資料視覺化
首先我們讀入需要用到的
套件，包括視覺化工具
matplotlib、cufflinks、
plotly，將
init_notebook_mode設定
為True，才能讓我們在本
地端使用plotly的視覺化套
件，以及像是pandas、
numpy等資料分析。
先以Calendar.csv的資料
進行分析，其資料日期分
佈為2020/10/1～2021/2/9，
一開始先顯示前五筆資料，
如（圖四）。

住宿熱度分析



<重點程式碼:>

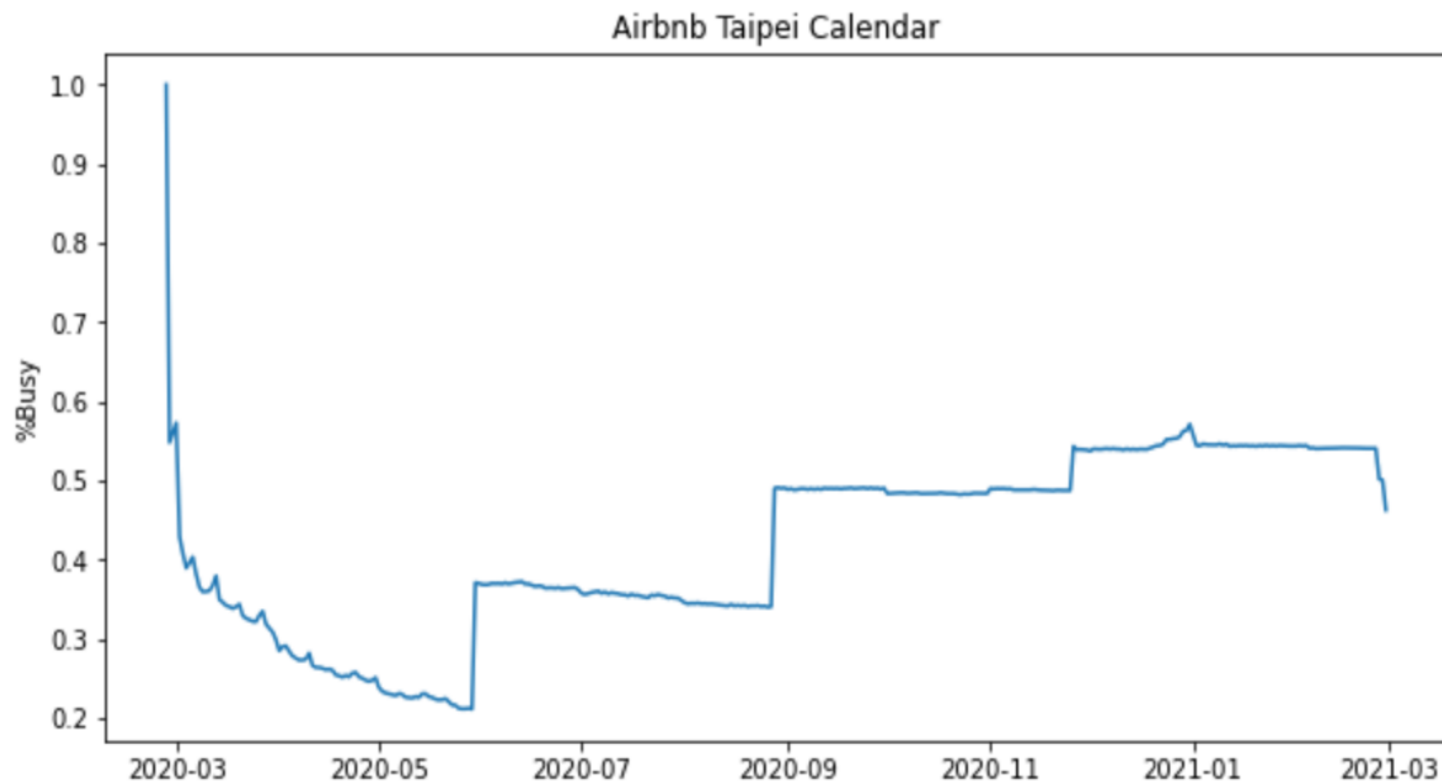
```
new_calendar = calendar[['date' , 'available']]  
#available : False, 代表說是旺季 (比較熱門)  
new_calendar['busy'] = new_calendar.available.map(lambda x:0 if x == 't' else 1)  
new_calendar = new_calendar.groupby('date')['busy'].mean().reset_index()
```

```
new_calendar['date'] = pd.to_datetime(new_calendar['date'])
```

```
plt.figure(figsize = (10 , 5))  
plt.plot(new_calendar['date'] , new_calendar['busy'])  
plt.title('Airbnb Taipei Calendar')  
plt.ylabel('%Busy')
```

圖（七）：利用定義熱度，來分析台北幾月份的住宿較為熱門

住宿熱度分析



圖（七）：利用定義熱度，來分析台北幾月份的住宿較為熱門

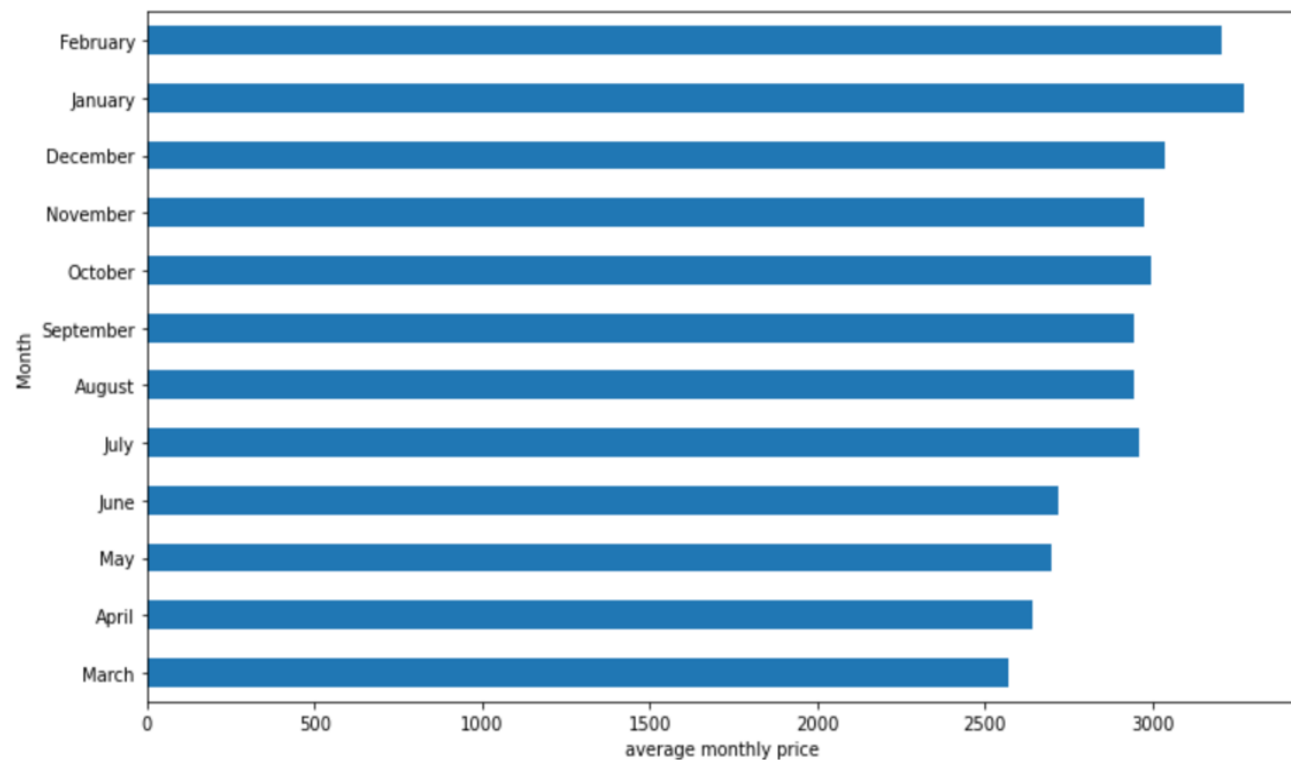
<程式碼說明:>

利用定義熱度，來分析台北幾月份的住宿較為熱門，由圖可看出**2020年3-5月份**較為熱門，而**5月份**過後，則較少人訂房，可預訂的房源升高。

定義熱度：當不可以預定的時候（**available**為 f），就表示是熱門的房源已經被搶走了。我們可以直接針對**available**這個欄位做匿名函數，創造新的

‘**busy**’作為熱度的衡量指標，以此了解台北市**Airbnb**熱門住宿的月份。

平均價格



圖（八）

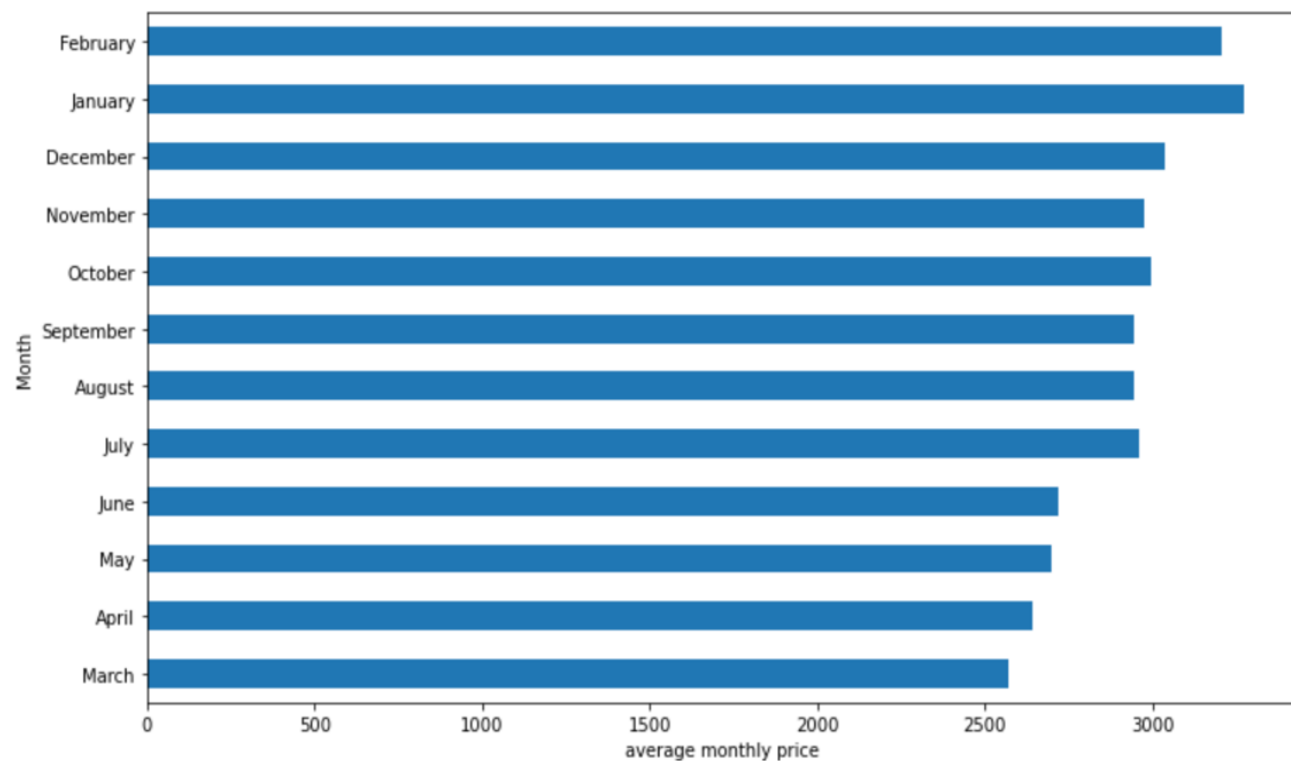
<重點程式碼:>

```
#處理一下價格資料
calendar['date'] = pd.to_datetime(calendar['date'])
calendar['price'] = calendar['price'].str.replace(',','').str.replace('$','').astype(float)
```

```
mean_of_month = calendar.groupby(calendar['date'].dt.strftime('%B'), sort = False)['price'].mean()
```

```
mean_of_month.plot(kind = 'barh', figsize = (12,7))
plt.xlabel('average monthly price')
plt.ylabel('Month')
```

平均價格

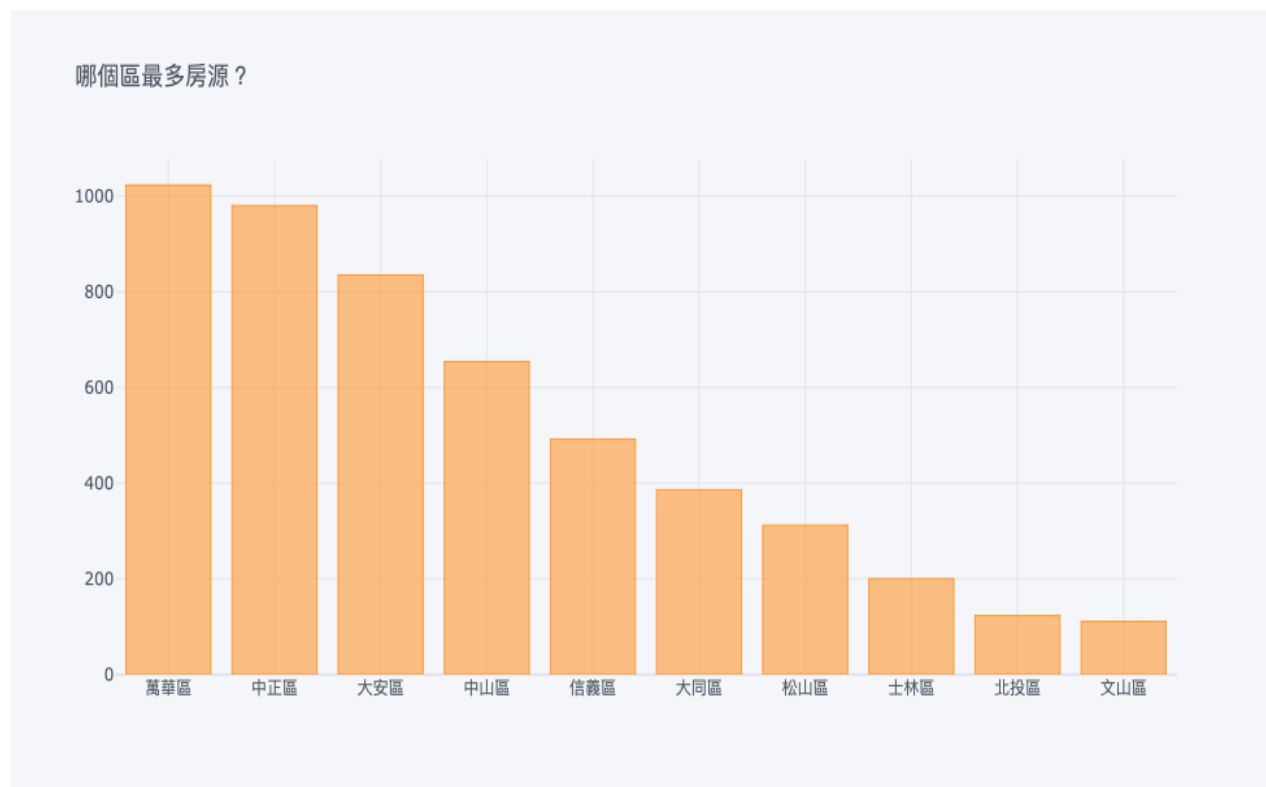


圖（八）

<程式碼說明>

接著以資料進行繪圖呈現每月平均價格，我們可以觀察到一、二月是最高的，而三、四月價格較低。與我們想像中寒暑假旺季價格為最高的想法，有所不同，但平均價格大多皆介於2500~3000之間。

哪個區域最多房源？

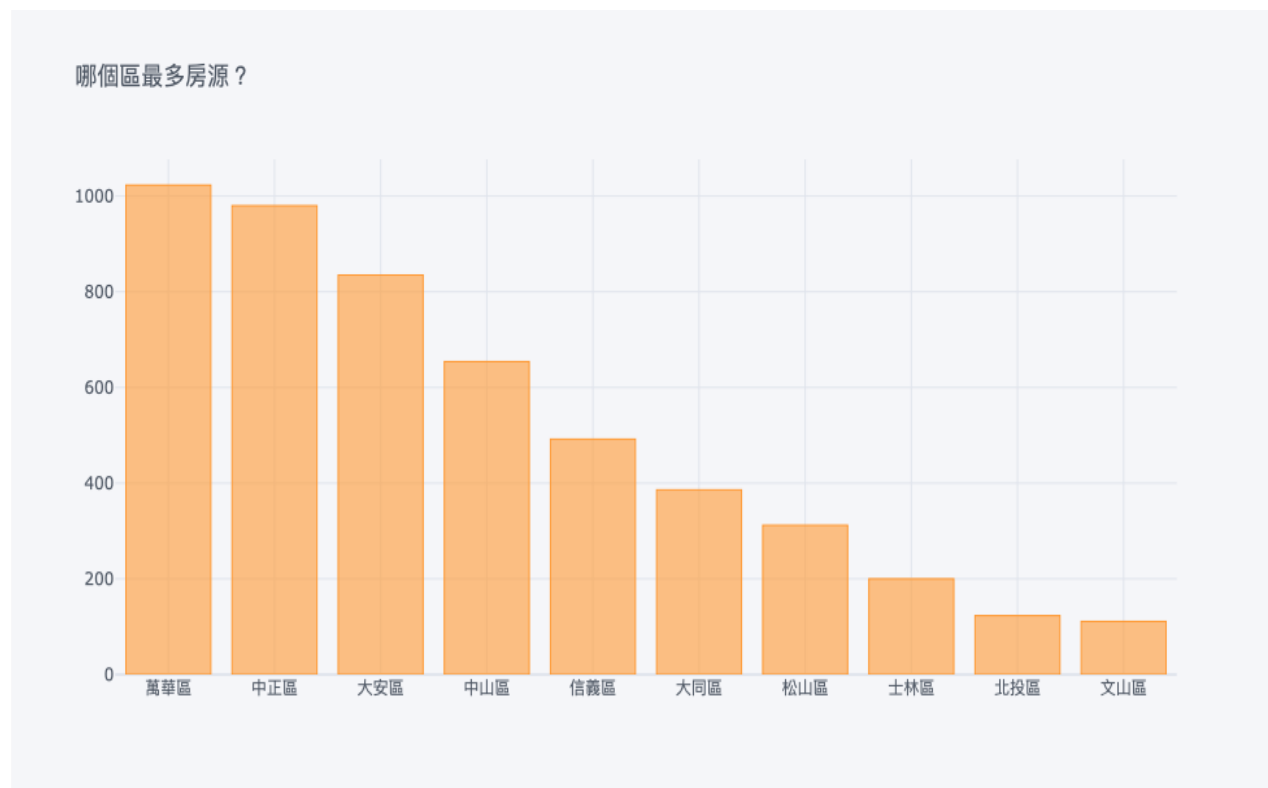


圖（九）

<重點程式碼:>

```
Listing = pd.read_csv('listings.csv')
print('We have', listing.id.nunique(), 'listings in the listing data')
listing.info()
listing.head(3)
grouped_df = listing.groupby('neighbourhood_cleaned').count()[['id']].sort_values('id', ascending = False).head(10)
grouped_df.plot(kind = 'bar', title='哪個區最多房源？')
```

哪個區域最多房源？



圖（九）

<程式碼說明>

之後我們匯入`listings.csv`，對其進行分析，首先先以`id`進行分群，以了解房源所在位置，接著進行繪圖，統計出各地區房源所在，由圖可了解，前三名分別為：萬華區、中正區、大安區，而後三名為：士林區、北投區、文山區，可觀察到後三名相對於其他地區離台北車站較遠，遊客偏向居住於交通方便的地區。

台北市平均價格

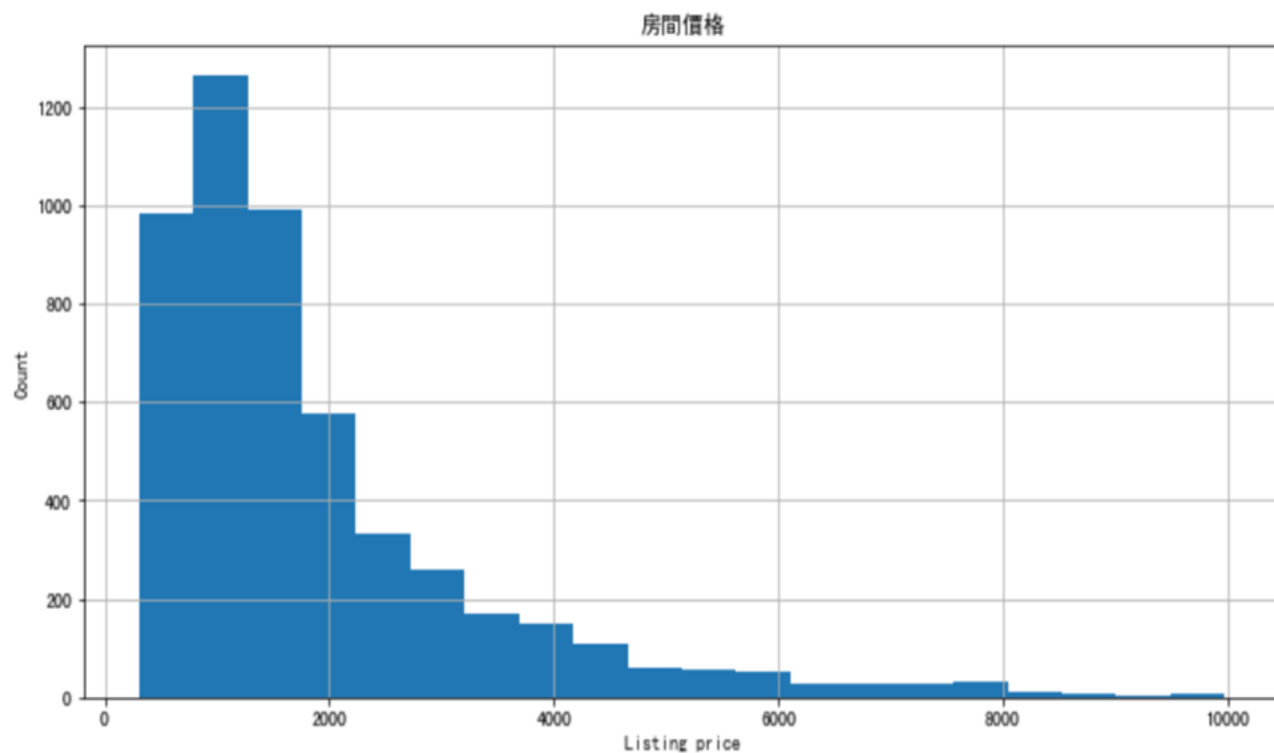
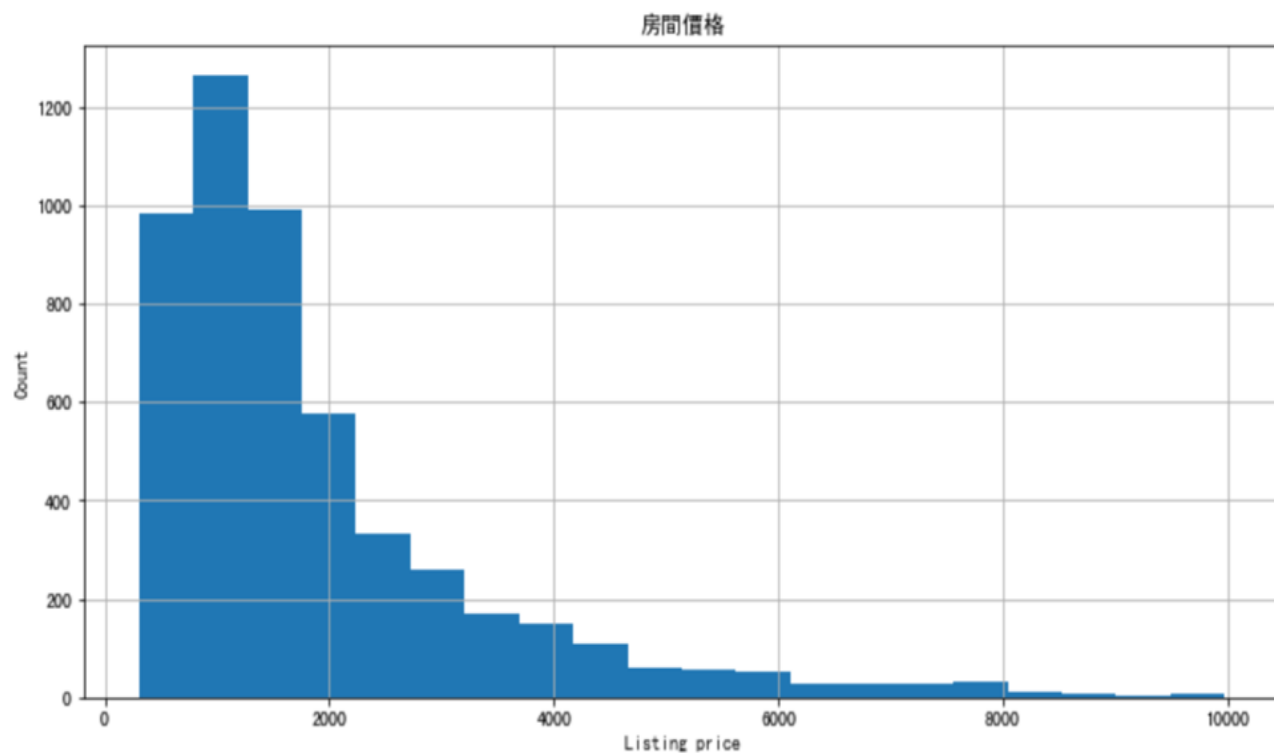


圖 (十)

<重點程式碼:>

```
plt.figure(figsize = (12,6))  
listing.loc[(listing.price < 1  
0000) & (listing.price > 30  
0)].price.hist(bins = 20)  
plt.ylabel('Count')  
plt.xlabel('Listing price')  
plt.title('房間價格')
```

台北市平均價格



圖（十）

<程式碼說明>

對房源價格進行分析，在程式碼中將價格設於300~10000元之間，為大部分房價所分佈之區域，避免極端的價格分佈使圖變的複雜，而我們也可以由圖中發現台北市住宿房價大致上落於300~2000之間居多。

各區域房價分布

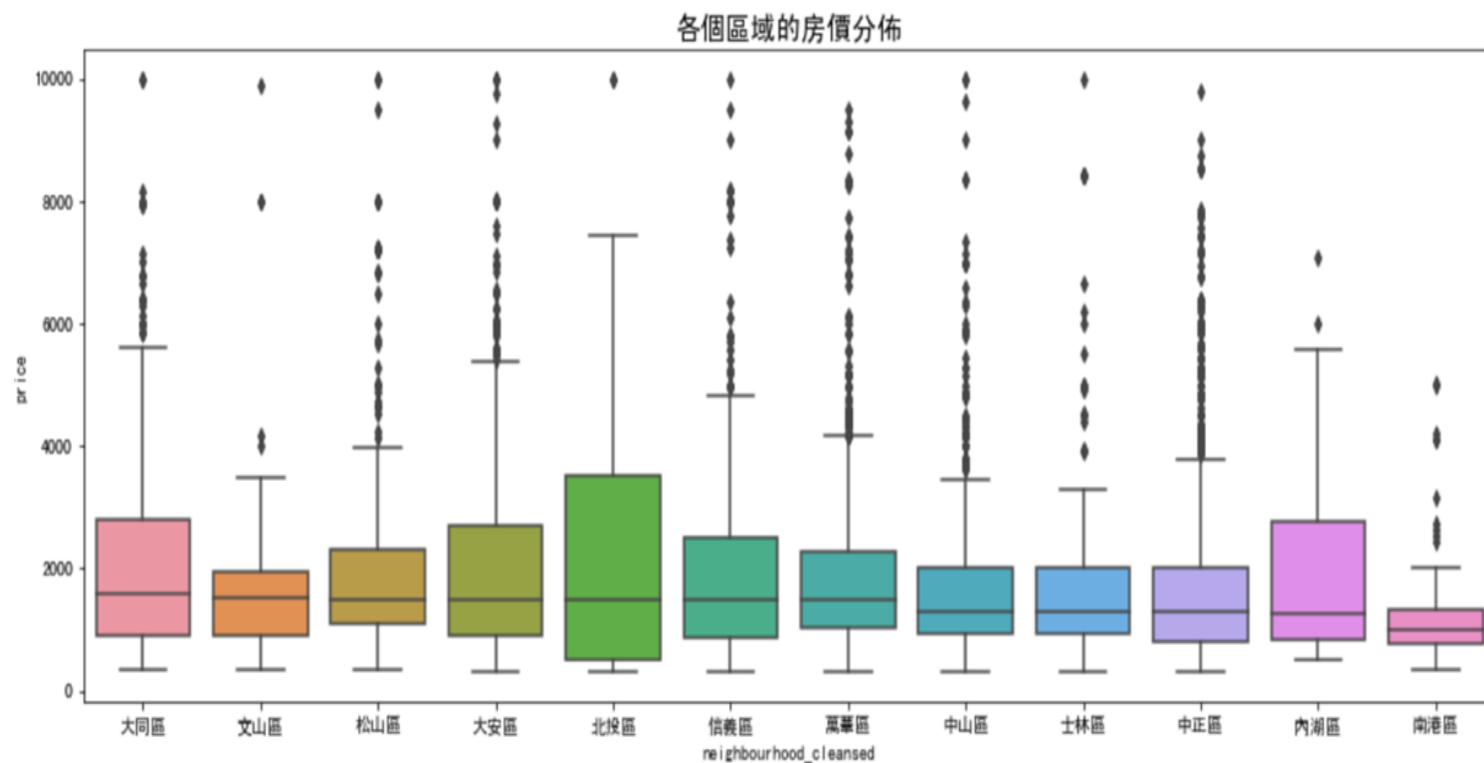
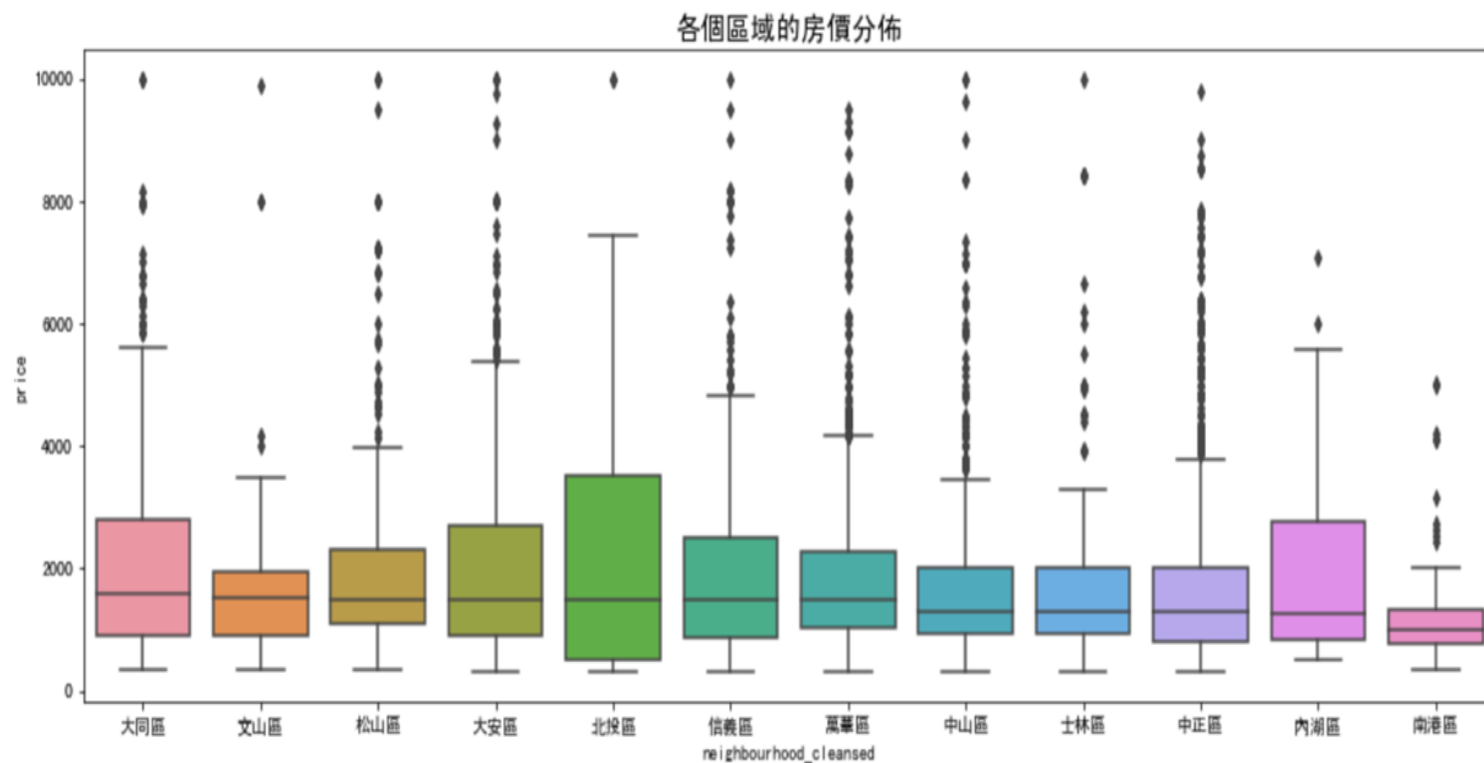


圖 (十一)

<重點程式碼:>

```
drop_outlier_price_condition = listing.loc[(listing.price <= 10000) & (listing.price > 300)]
sort_price = drop_outlier_price_condition\
    .groupby('neighbourhood_cleansed')['price']\
    .median()\
    .sort_values(ascending = False)\
    .index
plt.figure(figsize = (16, 6))
plt.title('各個區域的房價分佈', fontsize = 16)
plt.rcParams['font.sans-serif'] = ['simhei']
sns.boxplot(y='price', x = 'neighbourhood_cleansed', data = drop_outlier_price_condition, order= sort_price)
```

各區域房價分布



<程式碼說明>

各地區房價分布也呈現差不多的價格分佈，其中北投、大同、內湖這三個區域盒狀圖拉的比較長。而中正區價格分佈極廣。

圖（十一）

使用者對於房型選擇之偏好

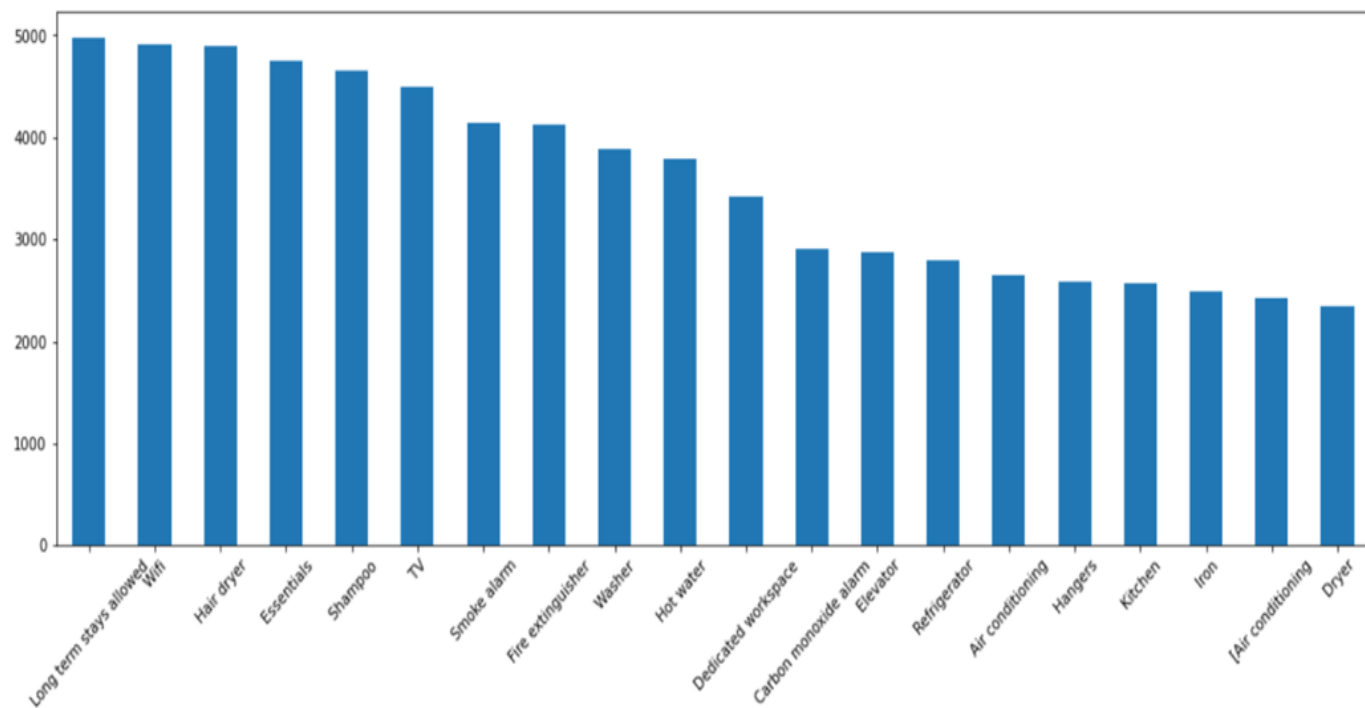
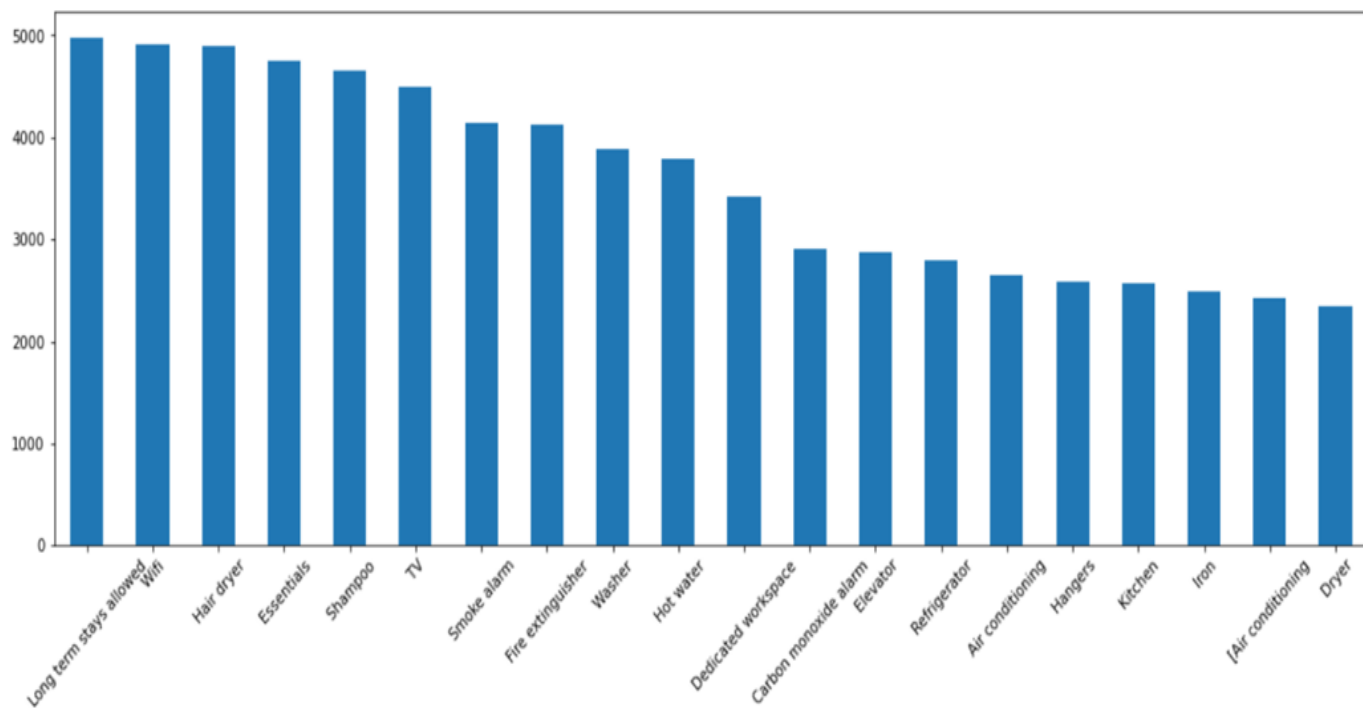


圖 (十二)

<重點點程式碼>

```
listing['amenities'] = listing.amenities.str.  
replace(['{}'], '').str.replace(' ','')  
listing.amenities.head()  
all_item_ls = np.concatenate(listing.amenities.map(lambda am:am.split(',')))  
Top20_item = pd.Series(all_item_ls).value_counts().head(20)  
plt.figure(figsize= (18 , 6))  
Top20_item.plot(kind = 'bar')  
plt.xticks(rotation = 45)
```

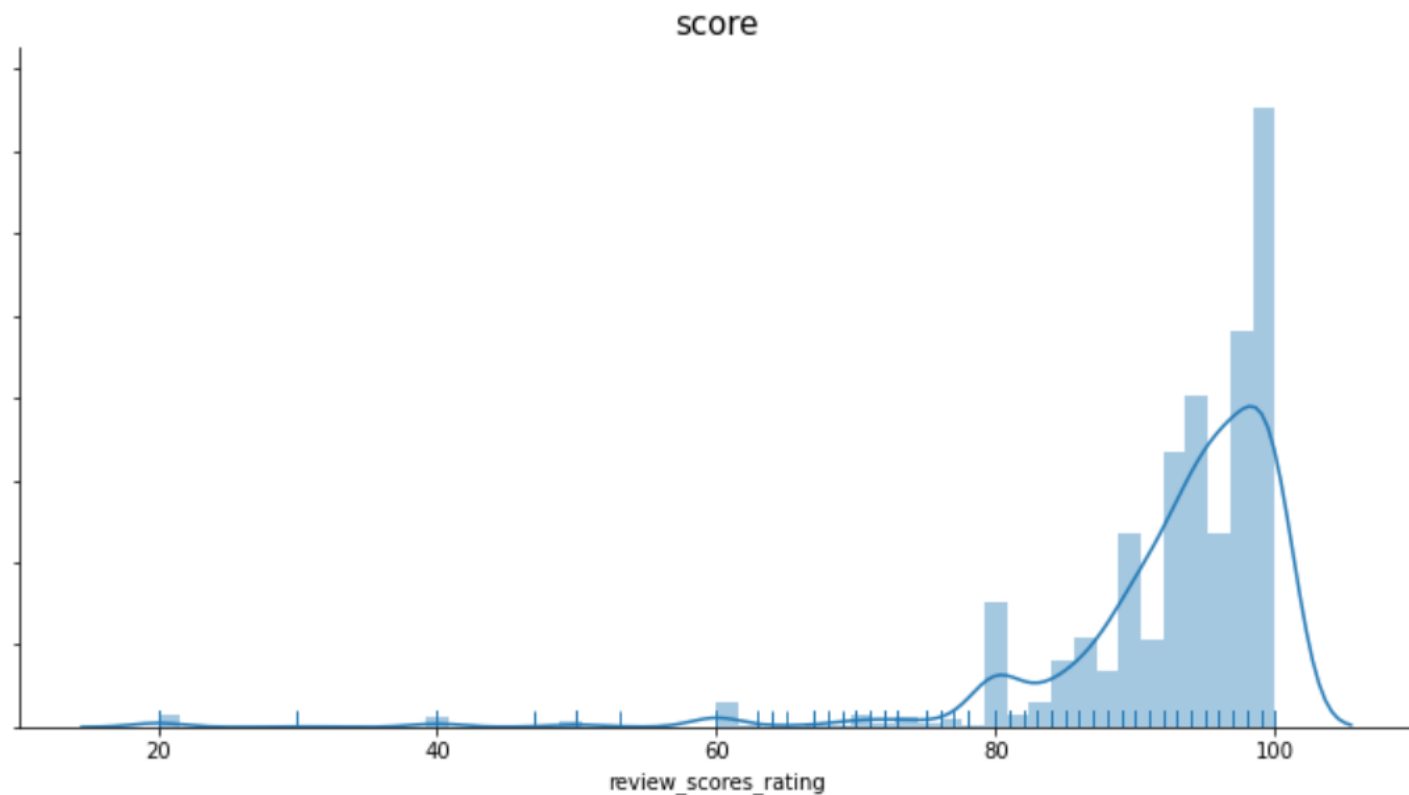
使用者對於房型選擇之偏好



其中設備也是旅客在住宿選擇中相當重視的一部分，前三名分別為：是否能長期居住、**wi-fi**、吹風機，相較於前三名，廚房及電梯等要素，相對不是那麼重要。

圖（十二）

使用者對於房型選擇之評分分析

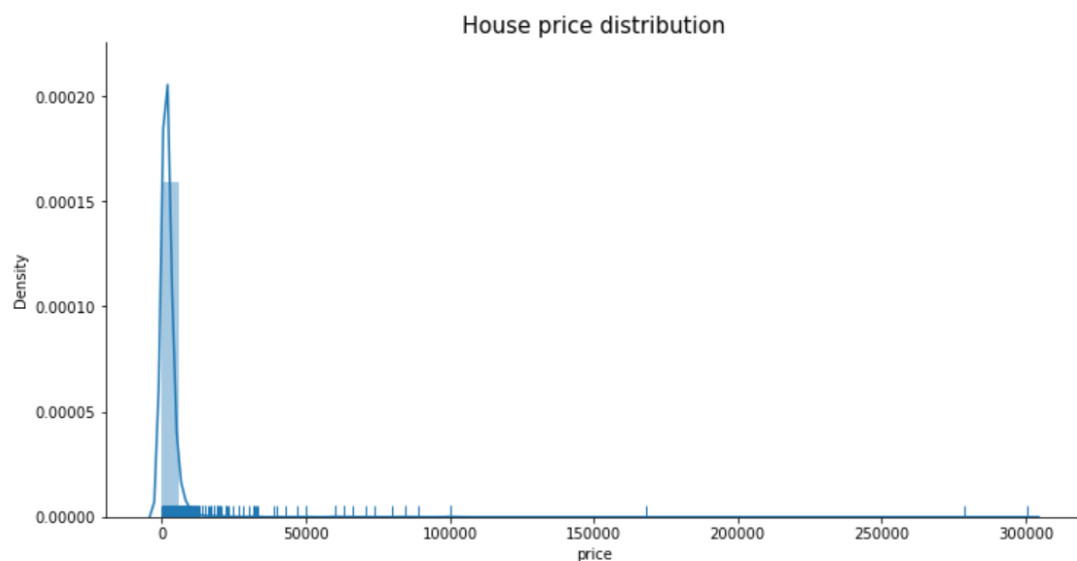


圖（十三）

<重點點程式碼>

```
plt.figure(figsize = (12 , 6))  
plt.title('score' , fontsize =  
15)  
sns.distplot(listing.review_  
scores_rating.dropna() , r  
ug = True)  
sns.despine()
```

房價分布



圖（十四）

<重點程式碼>

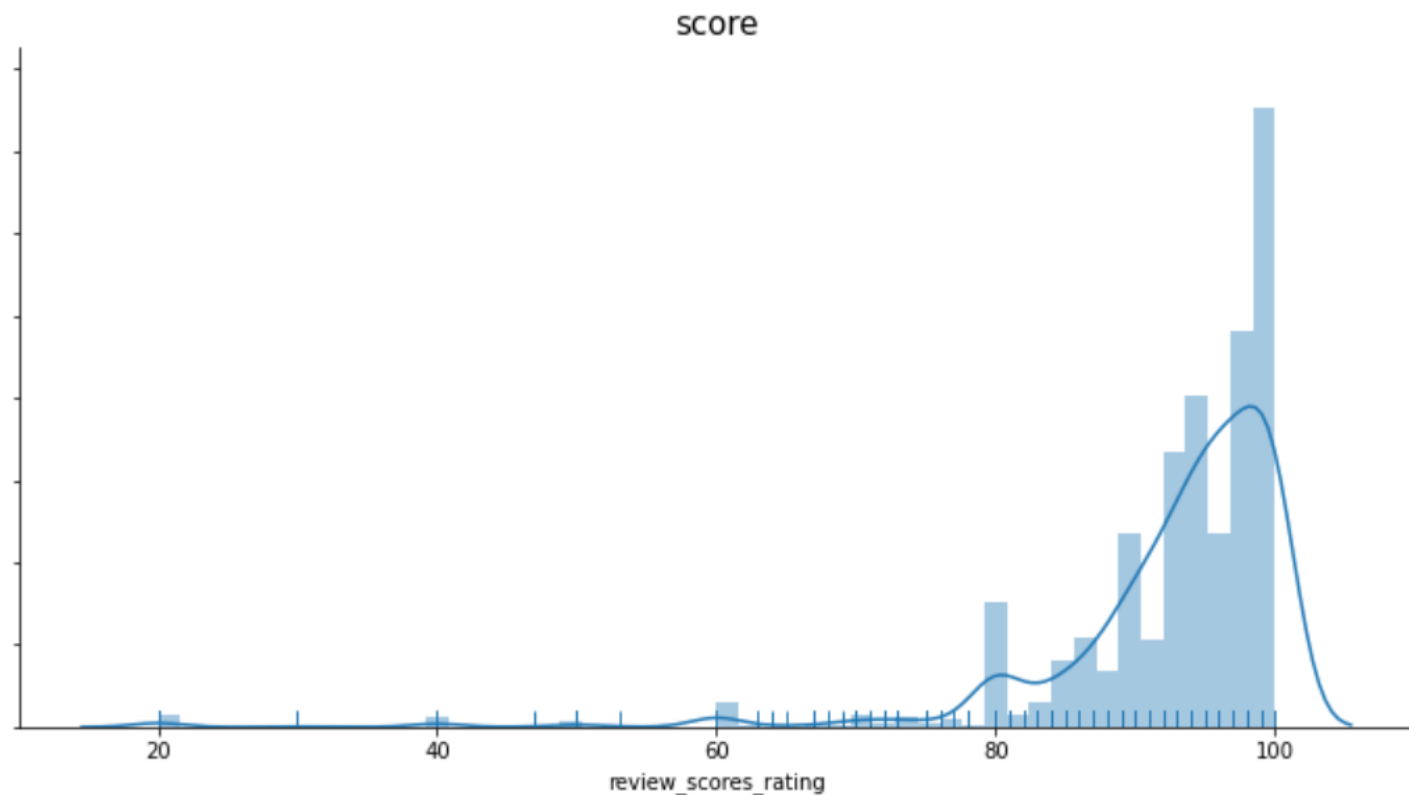
```
listing['price'] = listing['price'].str.replace(',', '').str.replace('$', '').astype(float)
```

```
print(listing.price.describe())  
plt.figure(figsize = (12 , 6))  
plt.title('House price distribution', fontsize = 15)  
sns.distplot(listing.price.dropna(), rug = True)  
sns.despine()
```

<程式碼說明>

房價分布大多分布於
0~100000區間左右

使用者對於房型選擇之評分分析

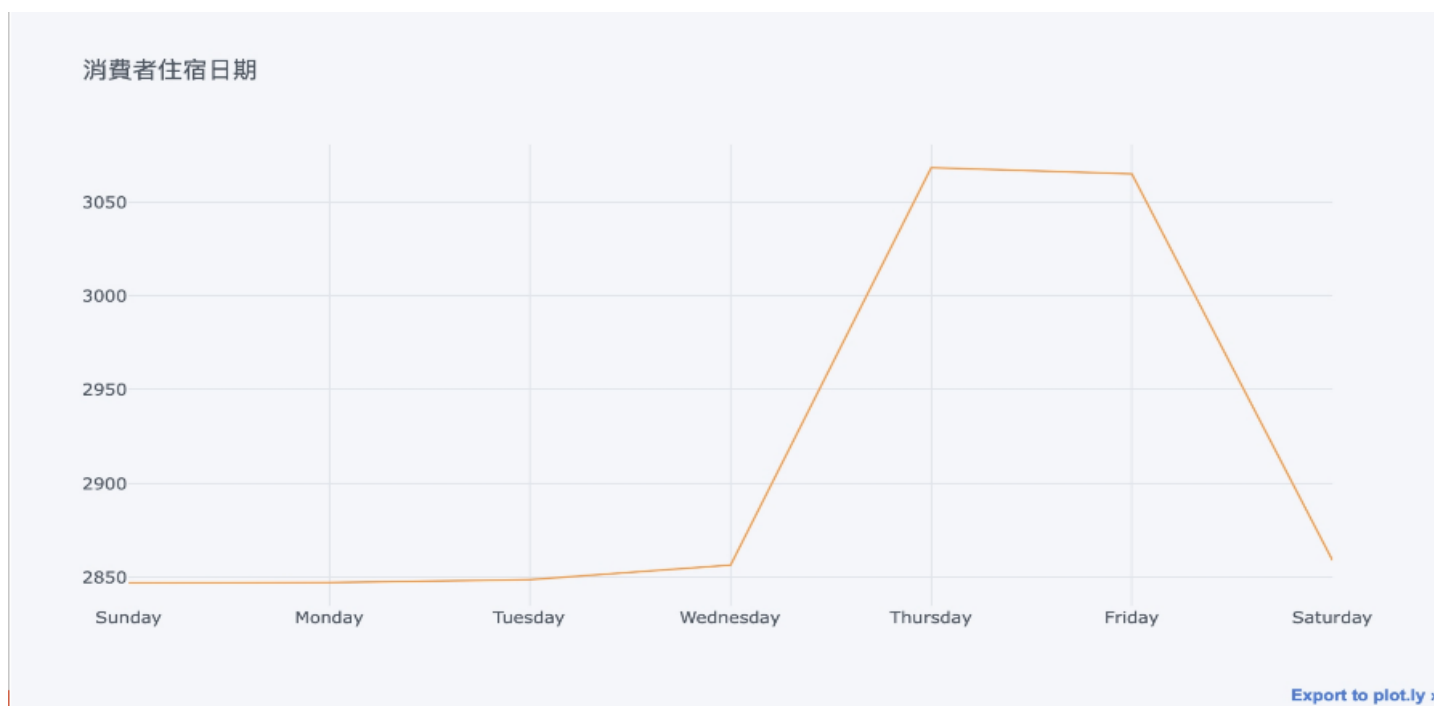


<程式碼說明>

以客戶給的評價進行繪圖，
結果呈現出客戶給予的分
數大多落於**80-100**，其中
以滿分佔多數

圖（十五）

消費者入住日期圖表



圖（十六）

<重點程式碼>

```
calendar['dayofweek'] = calendar.date.dt.weekday_name  
cats = calendar.dayofweek.unique().tolist()  
price_week = calendar.groupby('dayofweek')['price'].mean().reindex(cats)  
price_week.iplot(title = '消費者住宿日期')
```

<程式碼說明>

有calendar檔案可以去判斷，並統計出消費者多在星期四及五訂房

各式房價比較

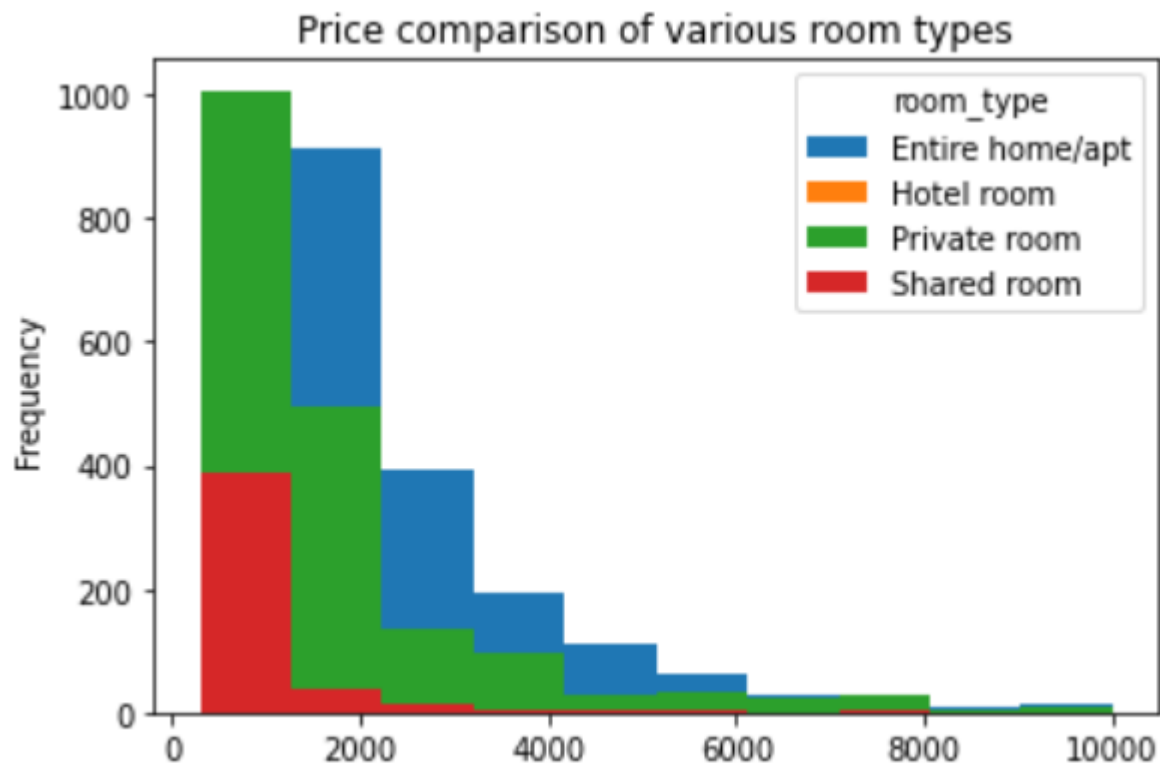


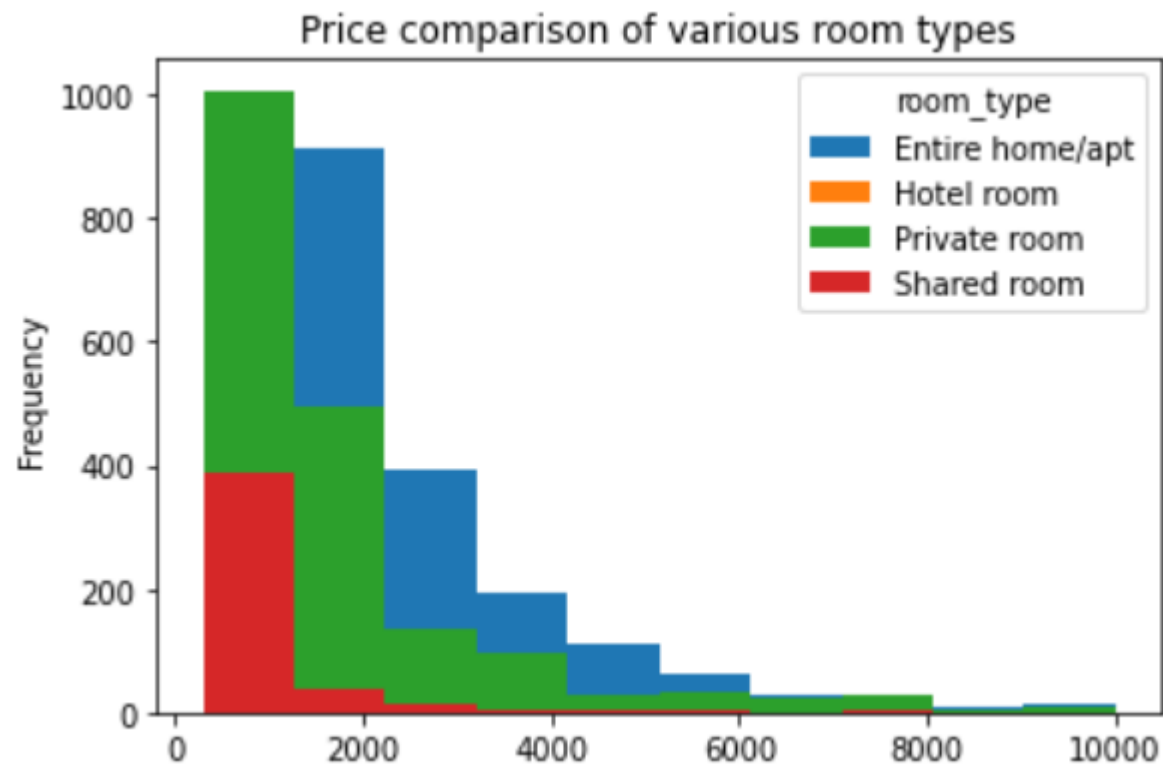
圖 (十七)

<重點程式碼>

```
drop_outlier_price_condition = listing.loc[(listing.price <= 10000) & (listing.price > 300)]
sort_price = drop_outlier_price_condition\
    .groupby('neighbourhood_cleanse d')['price']\
    .median()\
    .sort_values(ascending = False)\
    .index
```

```
plt.rcParams['font.sans-serif'] = ['simhei']
drop_outlier_price_condition.pivot(columns = 'property_type', values = 'price').i
plot(kind = 'box')
grouped_df = drop_outlier_price_condition.pivot(columns = 'room_type', values = 'price')
grouped_df.plot(kind = 'hist', title = 'Price comparison of various room types')
```

各式房價比較



圖（十七）

<程式碼說明>

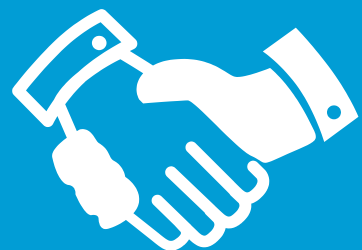
可以明顯看清楚各類房屋類型價格之比較，其中可以發現**shared room**為最便宜的，**entire home** 則價格方部最廣

04

結論

客戶選擇住宿的原因

客戶選擇住宿原因



使用者導向

由對評論進行文字探勘的結果，可以理解到客戶選擇住宿的房源，**主要是位置是否方便及房東是否友善，房間是否清潔，這些是在客戶評論中出現頻率較高的文字。**而在資料視覺化的結果之中，可以看出**客戶對於的設備是有一定的需求**，由圖可看出對於是否能長時間居住，是最高比例將近5000的人有這方面的需求，而**wifi也是必備之一，而令人驚奇的是廚房在排行榜中**，推測因相較於台灣人的習慣，外國人對於住宿比較偏好具有廚房，能自己下廚的地方，故廚房出現於排行榜中。

消費者行為及偏好



使用者導向

透過PYTHON將資料視覺化後，我們能看出大部分的房價位於中位數**1800**左右，而房價最多落於**1200**左右，走的是平價路線，與飯店相比價位親民，將客戶鎖定於在預算上沒那麼寬裕的房客。

另外房客也較偏好住在萬華、中正區，因其於台北車站附近，交通方便，對於背包客與觀光客來說，若需搭乘火車、捷運相關大眾運輸工具，能更快速，且能節省時間，是極佳的選擇。文山區及北投區，相較於其他地區，房源就較少，因在交通上轉運比較繁瑣，若相同的價位，房客會比較優先考慮前面所提及的萬華、中正區，

05

組員心得分享

客戶選擇住宿的原因



Airbnb

這次報告中很幸運，在網路上能找得到相關資料，免除了資料清理及爬蟲的前置作業，這次的報告主要是使用python對airbnb的資料進行繪圖，透過視覺化能更容易分析Airbnb在台北的房源分佈、消費者偏好等。這次在網路上找到許多相關繪圖套件，希望能更豐富的呈現，在之前對於繪圖，都是以簡單的圖表去呈現，經過這次報告後，學習到更多圖表繪製方式，像是互動式圖表，也更能去分析多面向，不只是簡單的數據，而是能更深入去理解消費者的消費模式及airbnb中店家的營運模式。從中不只學習到程式技巧，團隊分工也是很重要，透過大家合力，報告才能完整呈現。



Airbnb

本次報告的主題是探討Airbnb的使用者對於選擇房間的偏好為何，在資料分析的過程中發現，客戶選擇住宿的原因主要是依據地理位置方便與否及房東的友善程度與房間的清潔程度來判斷，其中也發現有蠻多人對於廚房有需求，突破了我對以往訂房網站的印象，我在本次專案中除了負責資料視覺化的分析之外，同時也負責簡報製作、資料整理與報告，讓我學到了很多東西，像是:資料分析能力的提升、專案管理的技巧、需求規劃的能力等等，都讓我獲益良多。

透過老師上課的學習，再加上網路上的資源，讓我們順利的完成本次期末專案，在這學期的這堂課讓我對資料分析與應用有更深入的了解，收穫了很多知識與技能，謝謝老師這學期的教導，祝您身體健康，事事順心。



Airbnb

這次的期末專案負責的部分是將資料清洗後做NLP，並呈現視覺化圖表。

此次資料清理及分詞部分皆使用Colab執行，畫圖部分由於中文字型顯示問題，所以將繪圖這段移至本機端執行。起初在做資料清洗時，由於評論中除中英文，還有日文及韓文字型，故採用正規表示法從資料篩選出中文和英文分為兩個資料及做分析。在使用jieba做斷詞時，發現中文評論中含有簡體字型，所以在做下一步清理前，將簡體轉為繁體，如此一來在做詞頻計算時，電腦就不會因繁體簡體的關係，而將一樣的字計為不同字。接著使用停用字清除不重要的字，最後使用NLTK套件做詞頻的計算，以呈現視覺化圖表。

這次負責的資料分析部分讓我注意到了資料爬取後進行整理分析的細節。

資料爬回後有不同的形態需要特別做處理（ex：nan）

學習正規表示法適用的時機

停用詞會因為爬取的資料或是應用場景不同，而需要不同的停用詞文件。不少停用字詞的增加都是在詞頻計算過後，再評估這個詞對於呈現分析結果的重要性

如何在Colab上使用matplotlib套件繪圖時，能夠呈現中文字型標籤。

最後最重要，在這次專案中學習到的，是發現Colab的重要性，並熟悉如何使用指令操作終端機；因為過去習慣在本機端使用jupyter notebook撰寫程式，這次因為資料量過大，在做資料分析時，電腦跑不起來，程式執行過程中出現記憶體不足的警示，才終於深刻感受到巨量資料處理的魅力。



Airbnb

這次的報告中，我們針對客戶行為、房屋價格、地區分佈，進而分析影響消費者入住因素，及消費者在評論中所表現出對於住宿體驗的想法。在對客戶評論進行文字探勘的部分，除了使用最常用的Jieab外，同時也搭配繁體中文詞庫做斷詞，讓文字能更順利地被處理，另外我還利用TF-IDF算法找出句子的關鍵字，並指定allowPOS的值去篩選詞性，讓關鍵字的抽取能更精確，最終視覺化呈現文字雲和圖表。

最後感謝組員的幫忙，才可以順利完成這次的期末報告，也謝謝老師這學期的認真教導，課堂中老師很都會用心的把實用的程式碼跟應用都完整的呈現給我們，讓我對於資料分析的應用有更好的瞭解！



Airbnb

本次研究主要研究台北Airbnb房源的分佈，因為Airbnb是屬於B2B的共享經濟的平台，所以他的資料和其他飯店式的訂房網站不同，除了他可針對客戶的生活型態喜好去找到合適物件，他也能針對房東的部分去做塞選。

在專案中運用的資料分析大多是由python所撰寫完成的，用了許多爬蟲和文字探勘的部分，之後運用了課堂中所學的視覺化文字雲等的方法來呈現，透過對合作也能更加清楚的知道自己的錯誤和缺失，進而加以改善場相互合作。因為我學習程式的時間算太長，總要花更多更長的時間來學習，怕會影響到同組的大家，但好在同組的組員都蠻厲害的，大家總能快速的完成作業及報告，也辛苦他們了，這個效率是我該好好學習的地方。

巨資二B 08170282 翁丞志



Airbnb

這學期學了很多程式的用法和圖表的呈現，我覺得印象最深刻的是文字探勘和文字雲的製作。在期末專案中我們用了python資料視覺化呈現出airbnb的相關資料以此來讓觀看者知道哪個房型是他們所需要的，而文字雲的製作也可讓對於airbnb不熟悉者可以清楚看出哪些關鍵字是使用者討論最熱烈的。我認為這堂課收穫滿滿，寫程式的能力大幅提升，學習到的技術可以應用在未來的工作上，謝謝老師的教導！

2021

THANKS

A

I

R

B

N

B

第八組