巨量期末報告

題目：Anime Recommendations Database

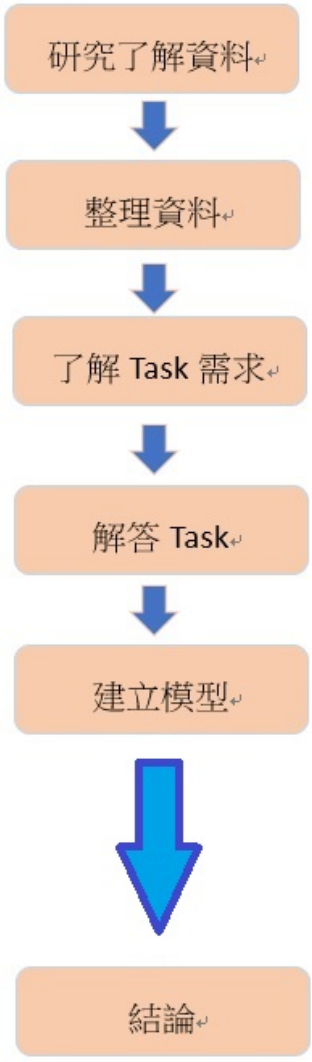
指導老師：陳律閎 老師
組員：710901042 劉育彣、7109018022 魏良育

第一章　緒論

第一小節 研究動機

根據老師上課的推薦，從 Kaggle 上面找尋資料解答提供者所需的 Task，因為資料有很多種類，最終討論使用雙方都有興趣的資料 ：anime，去做 Task 和進一步的建模分析，有迴歸分析、主成分分析，使用的軟體有 python 和 R。

第二小節 流程架構



第二章　研究資料

第一小節 觀察資料

I. 資料介紹

該數據集有兩筆檔案(anime、rating)包含來自 12,294 部動漫的 73,516 名用戶的偏好數據，每個用戶都可以將動漫添加到他們完成的列表中並給它一個評級，數據集是這些評級的總整理，從用戶觀看的歷史，構建一個更好的動漫推薦系統。

II.　資料變數介紹

(1)Anime.csv

anime_id ： 標識動漫的唯一 ID。

name： 動漫的全名，使用日文羅馬拼音。

genre：以逗號分隔的此動漫流派列表。

type：電影、電視、OVA 等。

episodes ：該節目中有多少集。 (1:表示電影)

rating ： 該動漫的平均評分。(滿分 10 分)

members：此動漫中的社區成員數量。

(2)Rating.csv

user_id：不可識別的隨機生成的用戶 ID。

anime_id：該用戶評價過的動漫。

rating：此用戶已分配的 10 分中的評分。(觀看未評分，則為-1)

III. 先觀察資料型態(因資料眾多，只觀看前 20 筆)

(1)Anime.csv

	anime_id	name	genre	type	episodes	rating	members
2	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630
3	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Military, Shour	TV	64	9.26	793665
4	28977	Gintama單	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shour	TV	51	9.25	114262
5	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17	673572
6	9969	Gintama'	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shour	TV	51	9.16	151266
7	32935	Haikyuu!!: Karasuno Koukou VS Shiratoriz	Comedy, Drama, School, Shounen, Sports	TV	10	9.15	93351
8	11061	Hunter x Hunter (2011)	Action, Adventure, Shounen, Super Power	TV	148	9.13	425855
9	820	Ginga Eiyuu Densetsu	Drama, Military, Sci-Fi, Space	OVA	110	9.11	80679
10	15335	Gintama Movie: Kanketsu-hen - Yorozuya y	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shour	Movie	1	9.1	72534
11	15417	Gintama': Enchousen	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shour	TV	13	9.11	81109
12	4181	Clannad: After Story	Drama, Fantasy, Romance, Slice of Life, Supernatural	TV	24	9.06	456749
13	28851	Koe no Katachi	Drama, School, Shounen	Movie	1	9.05	102733
14	918	Gintama	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shour	TV	201	9.04	336376
15	2904	Code Geass: Hangyaku no Lelouch R2	Action, Drama, Mecha, Military, Sci-Fi, Super Power	TV	25	8.98	572888
16	28891	Haikyuu!! Second Season	Comedy, Drama, School, Shounen, Sports	TV	25	8.93	179342
17	199	Sen to Chihiro no Kamikakushi	Adventure, Drama, Supernatural	Movie	1	8.93	466254
18	23273	Shigatsu wa Kimi no Uso	Drama, Music, Romance, School, Shounen	TV	22	8.92	416397
19	24701	Mushishi Zoku Shou 2nd Season	Adventure, Fantasy, Historical, Mystery, Seinen, Slice of Li	TV	10	8.88	75894
20	12355	Ookami Kodomo no Ame to Yuki	Fantasy, Slice of Life	Movie	1	8.84	226193

(2)Rating.csv

	user_id	anime_id	rating
2	1	20	-1
3	1	24	-1
4	1	79	-1
5	1	226	-1
6	1	241	-1
7	1	355	-1
8	1	356	-1
9	1	442	-1
10	1	487	-1
11	1	846	-1
12	1	936	-1
13	1	1546	-1
14	1	1692	-1
15	1	1836	-1
16	1	2001	-1
17	1	2025	-1
18	1	2144	-1
19	1	2787	-1
20	1	2993	-1

第三章　Task(python)

第一小節 介紹 Task

Need a Small help with below questions

Dodia Prashant · 0 Submissions · 7 months ago

1. Of all anime having atleast 1000 ratings, which anime has the maximum average rating? anime_id = 28977
2. How many anime with atleast 1000 ratings have an average rating greater than 9?
3. Which is the most watched anime i.e. the anime rated by most number of users?
4. What are the top three recommendations for the user with user_id 8086?
5. List top three users whom you would recommend the anime with anime_id 4935?

1. Of all anime having at least 1000 ratings, which anime has the maximum average rating ? anime_id = 28977
2. How many anime with at least 1000 ratings have an average rating greater than 9 ?
3. Which is the most watched anime i.e. the anime rated by most number of users ?
4. What are the top three recommendations for the user with user_id 8086 ?
5. List top three users whom you would recommend the anime with anime_id 4935 ?

第二小節 解答 Task

1. Q: (1)Maximum average rating ? (2)anime_id = 28977?

   Ans: (1)圖解、(2)name : Gintama°。

   需要計算加權平均，所以先取 memebers 的 85 分位數，跟 rating 的平均數，最後只依照問題需求取 name、members、rating 觀看，並使用 rating 去遞減排列。

```
def weighted_rating(x, m=m, C=C):
    v = x['members']
    R = x['rating']
    # Calculation based on the IMDB formula
    return (v/(v+m) * R) + (m/(m+v) * C)
```

	name	members	rating
1	Fullmetal Alchemist: Brotherhood	793665	9.176491
3	Steins;Gate	673572	9.075286
0	Kimi no Na wa.	200630	9.054553
6	Hunter x Hunter (2011)	425855	8.985370
10	Clannad: After Story	456749	8.928221
13	Code Geass: Hangyaku no Lelouch R2	572888	8.877123
12	Gintama	336376	8.865627
15	Sen to Chihiro no Kamikakushi	466254	8.807269
4	Gintama'	151266	8.785268
16	Shigatsu wa Kimi no Uso	416397	8.783948

	anime_id	name	genre	type	episodes	rating	members
2	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.25	114262

2. Q: Average rating greater than 9 ?

Ans: Fullmetal Alchemist: Brotherhood、Steins;Gate、Kimi no Na wa 這 3 個動畫的 rating 超過 9。

只需要把資料變數雙括號，直接取變數 rating 大於 9 的資料。

```
q_animes_9 = q_animes[q_animes["rating"]>=9]
q_animes_9
```

	anime_id	name	genre	type	episodes	rating	members
1	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mili...	TV	64	9.176491	793665
3	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.075286	673572
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.054553	200630

3. Q: The most watched anime = Most number of users?

Ans: Death Note

可從最多會員觀看得知哪部動畫最多人觀看，以下兩種方法：

(法一)可以先挑 max(members)，並直接雙等號 max 的值。

(法二)利用共同變數 anime_id，合併 2 個資料檔案，並排序。

```
max(anime["members"])
```

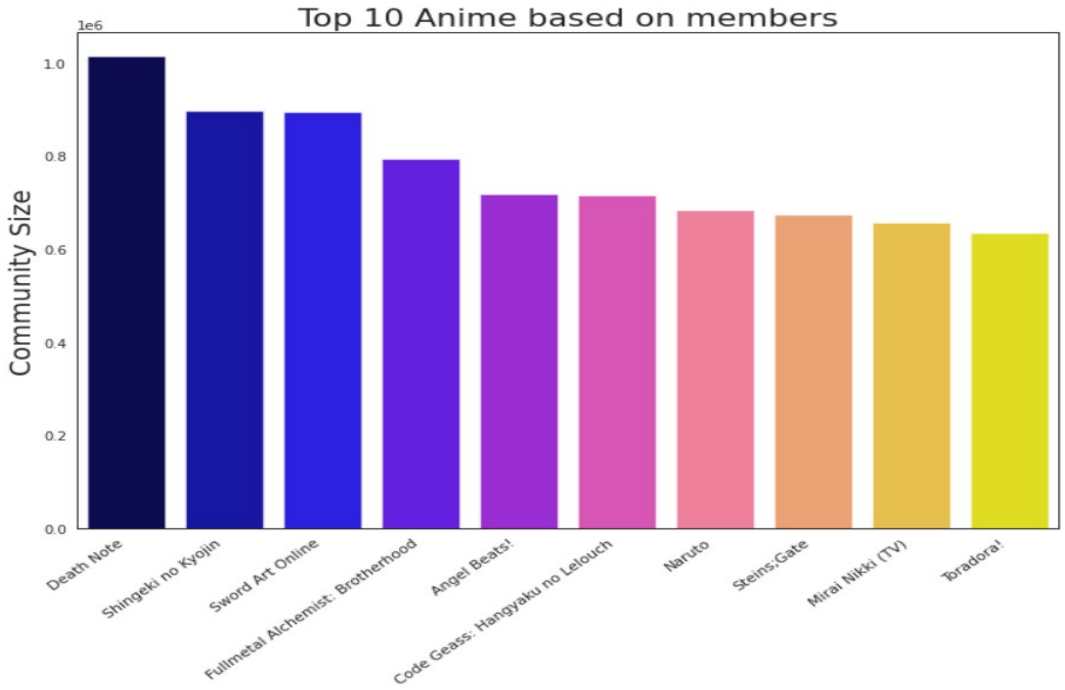
```
1013917
```

```
members =  anime[anime['members']== 1013917]
members
```

	anime_id	name	genre	type	episodes	rating	members
40	1535	Death Note	Mystery, Police, Psychological, Supernatural, ...	TV	37	8.71	1013917

```
##挑兩行合併dataframe
name1 = anime[["anime_id","name"]]
members1 = anime[["anime_id","members"]]
merge = pd.merge(name1 , members1)
merge.sort_values(by=['members'], ascending=False)
```

	anime_id	name	members
40	1535	Death Note	1013917
86	16498	Shingeki no Kyojin	896229



4. Q: The top three recommendations for the user with user_id 8086 ?

   Ans: Sen to Chihiro no Kamikakushi、Mononoke Hime、Howl no Ugoku Shiro

   用第三題合併的 dataframe，去尋找 user_id=8086，並排續，就能知道前三推薦是哪些。

```
user_id_8086 = merge[merge['user_id']== 8086]
user_id_8086.sort_values(by=['rating'], ascending=False)
```

	anime_id	name	genre	type	episodes	rating	members	user_id
120550	199	Sen to Chihiro no Kamikakushi	Adventure, Drama, Supernatural	Movie	1	8.93	466254	8086
228992	164	Mononoke Hime	Action, Adventure, Fantasy	Movie	1	8.81	339556	8086
307287	431	Howl no Ugoku Shiro	Adventure, Drama, Fantasy, Romance	Movie	1	8.74	333186	8086
702364	205	Samurai Champloo	Action, Adventure, Comedy, Historical, Samurai...	TV	26	8.50	390076	8086
750644	523	Tonari no Totoro	Adventure, Comedy, Supernatural	Movie	1	8.48	271484	8086
...	...	...	...	...	...	...	...	...

5. Q: List top three users recommend the anime_id 4935.

Ans:圖解

	anime_id	name	genre	type	episodes	rating	members	user_id
6482564	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	1822
6482575	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	48766
6482583	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	64820
6482582	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	58378
6482581	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	56650
6482580	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	55670
6482579	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	53060
6482578	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	50822
6482577	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	50656
6482576	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	50537
6482574	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	39111
6482565	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	7000
6482573	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	36575
6482572	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	30597
6482571	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	24931
6482570	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	24408
6482569	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	18944
6482568	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	15448
6482567	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	13539
6482566	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	12725
6482584	4935	Ikkyuu-san	Comedy, Historical, Kids	TV	296	7.05	720	65855

第四章 建立模型(R)

第一小節 PCA 分析

因為 genre 列為紀錄動漫的類型，資料紀錄方式是用”,” 為分隔，紀錄多種類類型，因此需要先清洗資料，將種類分開，儲存為新的一行資料。

```
#clean data
data$genre = as.character(data$genre)
n = unlist(count(data)) #行數

for ( i in 1:n ){
  a = as.vector(unlist(strsplit(data[i]$genre,"[,]")))
  if ( length(a) > 1 ){
    data[i]$genre = a[1]
    for ( j in 2:length(a) ){
      b = data[i,]
      b$genre = a[j]
      data = rbind(data,b)
    }
  }
}
data$genre = as.factor(data$genre)
```

清洗完後，分別檢查每個變數是否還存在空值。

```
#examine NULL data
sum(is.na(data))
sum(is.na(data$anime_id)) #動漫ID
sum(is.na(data$name)) #動漫名
sum(is.na(data$genre)) #動漫的類型(以逗號區隔)
sum(is.na(data$type)) #OVA、電影、電視...
sum(is.na(data$episodes)) #級數
sum(is.na(data$rating)) #評價
sum(is.na(data$members)) #有幾個人評論
```

```
sum(is.na(data$rating))
```

```
## [1] 705
```

```
sum(is.na(data))
```

```
## [1] 705
```

從此結果可以看出在 rating 的部分存在 NA 值。

接下來，選擇四個變數 genre、type、episodes、members 使用 prcom pakage 做主成分分析，並正規化資料。Rotation 為特徵向量，也就是各個主成份，所對應的線性組合係數。

PCA

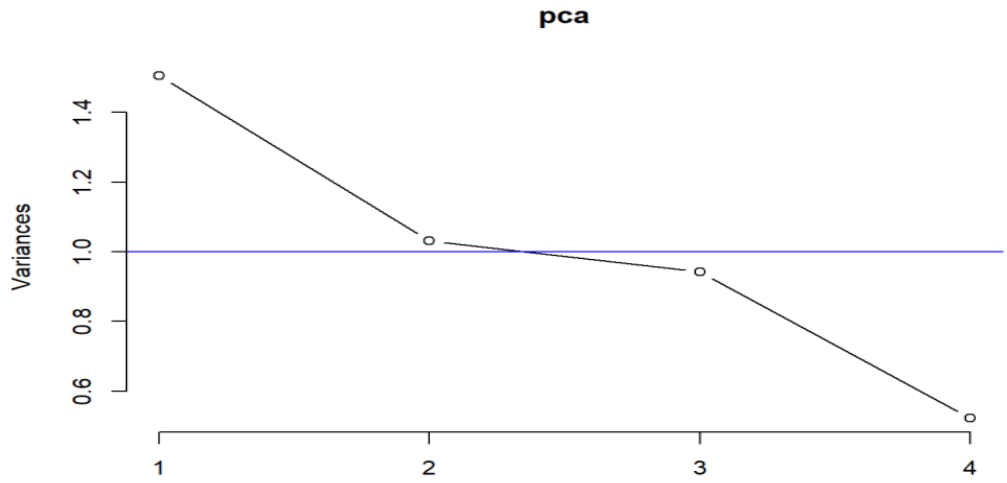
```
d = data %>%  arrange(anime_id) %>%
  mutate(across(everything() , as.numeric))  #convert to numeric
pca = prcomp(formula = ~ genre + type + episodes + members , data = d , scale = TRUE)
pca
```

```
## Standard deviations (1, .., p=4):
## [1] 1.2266650 1.0152040 0.9709273 0.7224639
##
## Rotation (n x k) = (4 x 4):
##                 PC1        PC2         PC3        PC4
## genre    -0.1591208  0.7006178 -0.69540461 -0.0150916
## type      0.6918131  0.1077894 -0.06513177  0.7110091
## episodes  0.6158393  0.3871595  0.26290216 -0.6338233
## members   0.3417852 -0.5895963 -0.66562206 -0.3041486
```

接著使用陡坡圖，去判斷需要挑選幾個主成分，使用藍線標出特徵值=1 的地方，根據定義大於 1 的主成份就可以選取。

從下圖中可得知第二個以後的主成份變異趨於平緩，因此選擇前二個主成份是比較好的選擇。

Scree plot



第二小節 iterative PCA 分析

使用 iterative PCA 填補 NA 值。

補值完後觀看資料，確定 iterative PCA 後資料沒有 NA 值存在。

iterative PCA

```
Sys.setenv(JAVA_HOME='C:\\Program Files\\Java\\jdk1.8.0_111\\jre') #Positioning JAVA
sc <- spark_connect(master="local") #Which host SPARK to choose

spark.df = sdf_copy_to(sc, d , overwrite = TRUE)  # convert R_dataFrame to spark_sql_dataFram
e
x = spark.df %>%
  sdf_collect() %>% as.data.frame() %>% as.matrix()
xa = dineof(x,2)$Xa
```

```
X = as.data.frame(xa)
head(X)
```

	anime_id <dbl>	name <dbl>	genre <dbl>	type <dbl>	episodes <dbl>	rating <dbl>	members <dbl>
1	1	1793	44	7	89	8.82	486824
2	1	1793	2	7	89	8.82	486824
3	1	1793	4	7	89	8.82	486824
4	1	1793	7	7	89	8.82	486824
5	1	1793	29	7	89	8.82	486824
6	1	1793	36	7	89	8.82	486824

6 rows

```
sum(is.na(X))
```

```
## [1] 0
```

第三小節 regression analysis - Elastic net model

Elastic Net 模型的優勢就在於，它綜合了 Ridge Penalty 達到有效正規化優勢以及 Lasso Penalty 能夠進行變數挑選優勢。

(1) 公式

$$\min_{\beta_0,\beta_1,\ldots,\beta_p} \frac{1}{n}\sum_{i=1}^{n}\left[Y_i-\beta_0-\sum_{j=1}^{p}\beta_j X_{ij}\right]^2+\lambda\left(\alpha\sum_{j=1}^{p}\left|\beta_j\right|+(1-\alpha)\sum_{j=1}^{p}\beta_j^2\right)$$

(2) 程式執行

執行的 Elastic net model 一個是原始變數模擬，並設置 10-Fold Cross Validation 進行驗證模型訓練結果。

然後將資料拆成 80:20 的 train 和 test，再利用”train”和 "glmnet" 的 package 進行 Elastic net model 的建立。

```
train = sample(c(T,T,T,T,F),nrow(data),rep=T) ##80/20 train/test split
test = (!train)

y = X %>% select(rating) %>% as.matrix()
v = X %>% select(genre,type,episodes,members) %>% as.matrix()

set.seed(42)
cv_10 = trainControl(method = "cv", number = 10)
```

model

```
hit_elnet = train(
  rating ~ genre + type + episodes + members, data = X[train,] ,
  method = "glmnet",
  trControl = cv_10
)
hit_elnet
```

接下來，再藉由把變數平方去擴展特徵空間。

這個目的是為了更精確模擬變數，因為使用的是懲罰回歸，所以不用擔心過度擬和的問題。

expanding the feature space

```
hit_elnet_int = train(
  rating ~ (genre + type + episodes + members)^2 , data = X[train,] ,
  method = "glmnet",
  trControl = cv_10,
  tuneLength = 10
)
hit_elnet_int
```

因為版面需求，只擷取最後幾筆結果顯示，如下圖：

```
##   0.9    0.0631286206  0.9112382  0.1888219  0.6856136
##   0.9    0.1458353051  0.9330291  0.1646236  0.7027190
##   0.9    0.3368984779  0.9773660  0.1508102  0.7447517
##   1.0    0.0001797925  0.8808918  0.2347470  0.6584945
##   1.0    0.0004153441  0.8809117  0.2347097  0.6586844
##   1.0    0.0009594988  0.8810119  0.2345515  0.6591261
##   1.0    0.0022165667  0.8814548  0.2339226  0.6602205
##   1.0    0.0051205567  0.8836054  0.2308491  0.6628607
##   1.0    0.0118291503  0.8948206  0.2123378  0.6731464
##   1.0    0.0273268722  0.9068949  0.1915406  0.6825012
##   1.0    0.0631286206  0.9125083  0.1875262  0.6865900
##   1.0    0.1458353051  0.9366479  0.1596948  0.7055766
##   1.0    0.3368984779  0.9874809  0.1508102  0.7540391
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 0.4 and lambda = 0.0001797925.
```

(3) 結論

在受過訓練的對像上調用這個函數，看到 α 、 λ 挑選的最佳結果是

α=0.4、 λ=0.0001797925

select alpha and lambda

```
get_best_result = function(caret_fit) {
  best = which(rownames(caret_fit$results) == rownames(caret_fit$bestTune))
  best_result = caret_fit$results[best, ]
  rownames(best_result) = NULL
  best_result
}
get_best_result(hit_elnet_int)
```

alp...	lambda	RMSE	Rsquared	MAE	RMSESD	Rsquared...	MAESD
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0.4	0.0001797925	0.8808831	0.2347625	0.6584075	0.01254874	0.01332778	0.008337318

1 row

第五章 結論

因為這次報告是自行找資料做資料分析，所以可以找有興趣的資料，而不是像以往為了符合分析需求，而特地去找相對應的資料，這樣做的資料分析做的比較上手，或者可以說解釋的可以比較詳細，能具體知道是在做些甚麼。

前兩章是介紹、整理資料型態，分析的前置步驟，觀察資料後先整理空值的部分影不影響 Task 的要求，因為不影響所以先刪除 NA 值，才做提供資料者的 5 項 Task 需求。

第三章純粹是使用 python 解答 Task，回答問題，而第四章的分析都是一般性線性回歸模型，會受限於「預測變數需與目標變數成線性關係」之假設，若假設不成立，仍得考慮非線性的回歸模型。

工作分配表	劉育彣	魏良育
Python 程式	O	X
R 程式	X	O
小論文+錄影 20 分	O	O