資料探勘-第一次作業

1. List the top 20 apps with the largest size. Present the app names and their size.

```
Top 20 Apps with the largest Size:
App: Mini Golf King – Multiplayer Game, Size: 100M, Size_MB: 100.0
App: Ultimate Tennis, Size: 100M, Size_MB: 100.0
App: Hungry Shark Evolution, Size: 100M, Size_MB: 100.0
App: SimCity BuildIt, Size: 100M, Size_MB: 100.0
App: Talking Babsy Baby: Baby Games, Size: 100M, Size_MB: 100.0
App: Draft Simulator for FUT 18, Size: 100M, Size_MB: 100.0
App: The Walking Dead: Our World, Size: 100M, Size_MB: 100.0
App: Stickman Legends: Shadow Wars, Size: 100M, Size_MB: 100.0
App: Post Bank, Size: 100M, Size_MB: 100.0
App: Car Crash III Beam DH Real Damage Simulator 2018, Size: 100M, Size_MB: 100.0
App: Hungry Shark Evolution, Size: 100M, Size_MB: 100.0
App: Vi Trainer, Size: 100M, Size_MB: 100.0
App: Miami crime simulator, Size: 100M, Size_MB: 100.0
App: Gangster Town: Vice District, Size: 100M, Size_MB: 100.0
App: Hungry Shark Evolution, Size: 100M, Size_MB: 100.0
App: Navi Radiography Pro, Size: 100M, Size_MB: 100.0
App: Rope Hero: Vice Town, Size: 99M, Size_MB: 99.0
App: Miami Crime Vice Town, Size: 99M, Size_MB: 99.0
App: My Talking Angela, Size: 99M, Size_MB: 99.0
App: music (CG), Size: 99M, Size_MB: 99.0
```

2. Check whether each attribute has missingness. For those attributes that have missingness, present the attribute names and their number of missing values. (15%)

根據程式的執行結果，以下欄位存在缺失值（Missing Values）：

```
Attributes with missing values and their counts:
Rating: 1474
Type: 1
Content Rating: 1
Current Ver: 8
Android Ver: 3
```

其餘欄位未出現缺失值或缺失筆數為 0。

因此，我們可以得知：

- **Rating** 為缺失值最多的欄位，共有 1474 筆。

- **Type**、**Content Rating**、**Current Ver**、**Android Ver** 也各別存在少量缺失值。

這些資訊可作為後續處理缺失值（如刪除、插補或預設值替代）的依據。

3.Let’s focus on the attribute “Rating”.

```
===== Before Cleaning =====
Rating column describe:
 count    9367.000000
mean        4.193338
std         0.537431
min         1.000000
25%         4.000000
50%         4.300000
75%         4.500000
max        19.000000
Name: Rating, dtype: float64

[Before] Mean: 4.193338315362443
[Before] IQR: 0.5 (Q1=4.0, Q3=4.5)
[Before] Std:  0.5374313031477587

===== Potential anomalies (before correction) =====
                                   App  Rating
15     Learn To Draw Kawaii Characters     3.2
23              Mcqueen Coloring pages     NaN
87       RST - Sale of cars on the PCT     3.2
113          Wrinkles and rejuvenation     NaN
123             Manicure - nail design     NaN
...                                ...     ...
10824                        Cardio-FR     NaN
10825               Naruto & Boruto FR     NaN
10831         payermonstationnement.fr     NaN
10835                         FR Forms     NaN
10838           Parkinson Exercices FR     NaN

[1978 rows x 2 columns]

===== After Cleaning & Correction =====
Rating column describe:
 count    8863.000000
mean        4.277446
std         0.357696
min         3.300000
25%         4.100000
50%         4.300000
75%         4.500000
max         5.000000
Name: Rating, dtype: float64
[After] Mean: 4.2774455601940655
[After] IQR:  0.40000000000000036 (Q1=4.1, Q3=4.5)
[After] Std:   0.3576960187482453
```

(1) Calculate its mean, IQR, and standard deviation. (10%)

```
根據程式執行後的 **Before Cleaning** 統計資訊，計算結果如下：

• **平均值 (Mean):** 4.193338315362443

• **四分位距 (IQR):** 0.5

• Q1 = 4.0

• Q3 = 4.5

• **標準差 (Std):** 0.5374313031477587
```

(2) Identify and report anomalies and/or errors in it. What would you do to make necessary corrections for it? (15%)

1. 異常值或錯誤值的判定

- 從統計摘要可見，Rating 最小值為 1.0，最大值竟達 19.0，遠超過合理的評分上限（通常為 5.0）。

- 此外也有部分資料顯示 NaN 或其他可能不合理之值。

- 綜合業務邏輯和 IQR 規則後，我們將「大於 5 或小於 1」的分數視為不合理，也將某些空白、無法轉成數值的 Rating 標記為缺失值（NaN）。

2. 更正方式

- 將判定為異常或不合理的 Rating 值改成 NaN（或於清理中予以刪除），即在程式中 df.loc[~condition_final, "Rating"] = np.nan 所示。

- 之後再進行統計計算時，就會先排除這些不合理的值。

(3) Following (2) after corrections being made, re-calculate the mean, IQR, and standard deviation. (15%)

根據 **After Cleaning & Correction** 統計資訊，排除或修正異常值後得到：

- **平均值 (Mean):** 4.2774455601940655

- **四分位距 (IQR):** 0.40000000000000036

- Q1 = 4.1

- Q3 = 4.5

- **標準差 (Std):** 0.3576960187482453

可觀察到異常值清除後，

- 平均值稍微上升至約 4.28；

- 標準差縮小到約 0.358，

顯示資料整體分布更加集中，且不合邏輯的極端值已被排除。

4. Let’s focus on the chi-square test.

(1) Check online. What are the assumptions and limitations of the chi-square test?(15%)

1. **資料必須是類別型 (categorical data)**：

   例如「Rating≧4：是/否」、「Price≧100：是/否」等，才能用卡方檢定。
2. **隨機且獨立抽樣 (independence)**：

   每個觀察值應該來自獨立樣本，彼此不應相關或重複。
3. **理論次數 (expected frequencies) 不能過低**：

   一般建議每個儲格的期望次數不小於 5，或至少 80% 以上的儲格不小於 5。
4. **僅能用於判斷「是否有關聯」，不能解釋因果**：

   卡方檢定只顯示兩個變數之間有沒有關聯，並無法告訴我們誰影響誰，或影響程度的大小。

(2) Use the chi-square test to investigate the following: whether the ratings≧4 or not is associated with whether the price≧100 or not. Report on your test results. What is your conclusion? (20%)

```
Contingency table:
 Price_100+  False  True
Rating_4+
False         3459     13
True          7362      7
Chi-square test statistic: 8.54722541926516
p-value: 0.003460492769636125
Degrees of freedom: 1
Expected frequencies:
 [[3.46559469e+03 6.40531316e+00]
 [7.35540531e+03 1.35946868e+01]]

檢定結果: p-value < 0.05, 拒絕虛無假設(H0)
→ 推論：Rating≥4 與 Price≥100 之間具有統計上的關聯
```

結論：

根據卡方檢定結果（p-value < 0.05），我們拒絕「評分≧4 與價格≧100 之間獨立」的虛無假設，表示在統計上這兩個變數有顯著關聯。從觀察值可看到，價格較高（≥100）的應用程式中，評分≥4 的比例相對較低；而價格較低（<100）的應用程式則有較高比例達到評分≥4。

換句話說，根據這份資料，**價格是否 ≥100 與應用程式評分是否 ≥4 並非獨立，二者之間存在某種程度的負向關聯**：價格越高的應用，越不容易有高評分。當然，這只是統計上的關聯，並不代表兩者必然存在因果關係。