駕馭未知：針對陌生領域的系統化資料分析與特徵工程指南

撰文者: SUNNY

1. 導論：航向未知 —— 資料分析的系統化途徑

1.1 陌生資料領域的挑戰

在資料科學的實踐中，分析師經常面臨來自完全陌生領域的資料集。缺乏相關的領域知識（Domain Knowledge）是新手分析師普遍遇到的困境，這往往導致他們在探索性資料分析（Exploratory Data Analysis, EDA）和特徵篩選過程中，依賴直覺或「感覺式」的方法，而非一套嚴謹的系統化流程（使用者查詢）。這種非結構化的處理方式隱藏著多重風險：分析師可能迷失在龐雜的資料中，被虛假的模式或相關性誤導，做出不恰當的模型選擇，最終無法有意義地解釋分析結果或產生可行的商業洞見 [1]。

1.2 為何系統化流程對新手至關重要

面對未知領域的不確定性，一套系統化的分析流程顯得格外重要。遵循一個結構化的工作流程，例如廣泛應用的 CRISP-DM（跨行業資料探勘標準流程）框架或類似方法，能為分析師提供清晰的路線圖，減少模糊性，並確保關鍵步驟不被遺漏 [3]。結構化的流程不僅有助於提升分析的可重複性 [6]，更能建立分析師的信心，使其能夠有條不紊地應對挑戰 [5]。此外，系統化的 EDA 不僅僅是為了發現有趣的模式，更根本的目的是深入理解資料的品質、特性以及其是否適合用於解決手頭的分析問題 [6]。

1.3 本指南的結構與目標概覽

本指南旨在為資料科學新手提供一套實用的、按部就班的方法論，以應對陌生領域的資料分析挑戰。內容將依循一個邏輯性的結構展開：首先奠定特徵工程的基礎知識，接著詳細闡述一個系統化的資料分析工作流程，涵蓋從問題定義到關聯分析的各個階段。其中，將特別深入探討在缺乏領域知識背景下，如何進行系統性的探索性資料分析（EDA）和有效的特徵工程。指南的核心目標是強調在每個階段的「目的（Purpose）」、「策略（Strategy）」、「常見陷阱（Pitfalls）」以及「潛在誤導（Misinterpretations）」，並提供關於資料視覺化選擇與應用的清晰原則。最終目標是幫助新手建立紮實的分析基礎，掌握一套能夠規避常見錯誤（如資料洩漏、過度擬合、結果誤讀）的實戰技能。

2. 基礎：理解特徵工程

2.1 何謂特徵工程？

定義：
特徵工程（Feature Engineering）是資料科學與機器學習流程中的核心環節，其定義為從原始資料中選取（Selecting）、操作（Manipulating）和轉換（Transforming）資料，以產生能夠被機器學習模型（特別是監督式學習模型）有效理解和利用的「特徵」（Features）的過程 8。特徵是任何可用於預

測模型的、可測量的輸入變數 8。本質上，特徵工程是運用統計或機器學習方法，將原始觀測數據轉化為對模型有價值的屬性的行為 8。

重要性：

特徵工程對於機器學習專案的成功至關重要 8。模型的性能在很大程度上取決於訓練時所用特徵的品質 11。它被譽為是將混亂資料集轉化為機器學習模型「金礦」的「秘方」9，是連接原始資料與強大預測模型的橋樑 8。良好的特徵工程能夠顯著提升模型的準確性、加速訓練過程、改善模型的可解釋性，並有助於揭示資料中潛藏的洞見 9。其影響力之大，有時甚至超過演算法本身的選擇 13。當特徵工程執行得當時，產生的資料集將是最佳化的，包含所有影響業務問題的關鍵因素，從而催生出最準確的預測模型和最有用的商業洞察 8。

在機器學習流程中的角色：

特徵工程是模型訓練前一個關鍵的「預處理」（Preprocessing）步驟 8。它位於原始資料收集與模型建構之間，負責將資料塑造成適合演算法使用的形式。這個過程並非一蹴可幾，而是具有迭代性（Iterative）和情境依賴性（Context-dependent）的，需要根據資料特性、模型類型和分析目標不斷調整和實驗 9。

2.2 特徵工程的生命週期（核心流程）

特徵工程涵蓋一系列相互關聯的活動，雖然不同來源的描述略有差異，但核心流程通常包含以下幾個方面：

1. 特徵創建（Feature Creation）：
   此過程涉及從現有資料中產生新的、對模型可能有幫助的變數 8。這可以透過結合領域知識或應用數學運算（如計算比率、創建交互項、進行聚合）來實現 10。例如，在房地產資料中，根據總價和面積創建「每平方英尺成本」就是一個特徵創建的例子 8。這個步驟往往需要分析師的創造力和判斷力 10。
2. 特徵轉換（Feature Transformation）：
   這一步驟是對現有特徵進行修改，將其從一種表示形式轉換為另一種 8。常見的轉換包括數據縮放（Scaling）、歸一化（Normalization）、編碼（Encoding）、對數轉換（Log Transform）等 8。其目標是確保資料在不同特徵間具有一致性，使其更適合特定演算法的需求，或改善資料的可視化效果 8。
3. 特徵提取（Feature Extraction）：
   特徵提取是指自動地從原始資料中創建新的、通常維度更低的特徵表示 8。目的是在不扭曲原始關係或重要資訊的前提下，將資料量壓縮到更易於管理的程度 8。常見的技術包括主成分分析（PCA）、文本分析中的詞袋模型或 TF-IDF、圖像處理中的邊緣檢測、以及使用自動編碼器（Autoencoders）等 10。值得注意的是，「特徵提取」有時會與「特徵工程」互換使用，或者特指將原始特徵空間映射到低維空間的過程 11。
4. 特徵選擇（Feature Selection）：
   此過程旨在從所有可用特徵中，挑選出一個最相關的子集，以用於模型訓練 9。目標是降低模型的複雜性、提高訓練效率、避免過度擬合（Overfitting），並增強模型的可解釋性 9。特徵選擇方法大致可分為過濾法（Filter）、包裹法（Wrapper）和嵌入法（Embedded）三類 13。
5. 探索性資料分析（Exploratory Data Analysis, EDA）：

EDA 是運用分析和視覺化手段來理解資料屬性、發現模式、檢驗假設和檢查數據品質的過程 8。它在特徵工程的各個環節都扮演著重要角色，為特徵創建、轉換和選擇提供依據和方向 8。

6. 基準測試（Benchmarking）：
   建立一個基準模型（通常是簡單、易於解釋的模型）或基準指標，用來評估經過特徵工程處理後的特徵集對模型性能的實際提升效果 8。這有助於判斷哪些特徵工程的操作是有效的。

需要強調的是，這些流程並非嚴格的線性順序，而是高度迭代和相互關聯的 [11]。例如，EDA 的發現可能引導新的特徵創建或轉換；特徵選擇的結果需要透過基準測試來驗證其有效性；特徵提取可能先於特徵選擇進行。分析師應將這些視為一個動態的工具箱，根據具體情況靈活運用。

2.3 領域驅動 **vs.** 資料驅動的特徵工程

特徵工程的實施策略大致可分為兩類：領域驅動和純資料驅動。

- 領域驅動（**Domain-Driven**）特徵工程：
  - 定義：這種方法主要依賴於特定領域（如金融、醫療、工程等）的專家知識和深刻理解來指導特徵的選擇、創建和轉換 [10]。它建立在對領域內潛在機制、因果關係和業務邏輯的認知之上 [25]。
  - 優點：
    - 速度可能更快，因為專家經驗可以快速定位潛在的關鍵特徵（使用者查詢）。
    - 能夠創建具有高度相關性和可解釋性的特徵，因為這些特徵根植於領域的實際意義 [14]。
    - 有助於改善與領域專家的溝通，確保分析目標與業務需求一致 [25]。
    - 在需要高度可信度和解釋性的領域（如醫療診斷、信用評分）尤為重要 [28]。
  - 缺點：
    - 容易受到專家既有認知框架的限制，可能忽略資料中隱藏的、反直覺的模式（使用者查詢）。
    - 存在引入專家偏見（Bias）的風險 [27]。
    - 依賴於領域專家的可用性和專業水平 [25]。
    - 對於非常複雜或全新的問題，現有領域知識可能不足。
  - 適用情境：具有成熟理論基礎或豐富專家經驗的領域；對模型可解釋性要求高的應用；用於補充和驗證資料驅動方法的結果。
- 純資料驅動（**Pure Data-Driven**）特徵工程：
  - 定義：這種方法完全依賴於資料本身的統計特性、數據探勘技術或機器學習演算法來自動發現、篩選和構建特徵，而不依賴（或很少依賴）先驗的領域知識（使用者查詢[8]）。近年來興起的自動化特徵工程（Automated Feature Engineering, AutoFE）技術，如基於遺傳演算法或強化學習的方法，也屬於此範疇 [13]。
  - 優點：

        - 有潛力發現人類專家可能忽略的、非直觀的複雜模式 [27]。
        - 減少了對專家主觀判斷和潛在偏見的依賴。
        - 對於缺乏成熟領域知識的新興問題或超大規模、高維度資料集具有良好的可擴展性。
    - 缺點：
        - 通常較為耗時，需要大量的計算資源和時間來探索資料（使用者查詢）。
        - 需要花費大量精力去理解資料本身的結構和意義（使用者查詢）。
        - 容易受到資料中噪音和虛假相關性（Spurious Correlation）的影響，可能導致方向被帶偏（使用者查詢[23]）。
        - 產生的特徵可能缺乏直觀的業務含義或物理解釋，導致模型成為「黑盒子」，難以解釋和信任 [28]。
        - 若不謹慎，容易導致過度擬合 [23]。
    - 適用情境：領域知識有限或不存在的探索性分析；處理高維度、複雜的資料集；希望發現全新模式；作為領域驅動方法的補充或基準。
- 比較分析與限制：
  兩者之間存在明顯的權衡：領域驅動在速度和可解釋性上佔優，而資料驅動在模式發現和可擴展性上更強 26。領域驅動方法在大型複雜專案中可能力不從心，而純資料驅動若缺乏結構性指導，則可能陷入混亂 26。特別是純粹依賴深度學習等資料驅動方法的特徵工程，往往以犧牲可解釋性為代價來換取預測精度，這在許多實際應用場景中是不可接受的 28。
- 實踐建議：走向「資料啟發（Data-Informed）」
  在實踐中，最有效的方法往往不是絕對的領域驅動或資料驅動，而是兩者的結合，即「資料啟發」（Data-Informed）的方法 29。這種方法強調利用可獲得的所有資訊——包括業務背景、資料來源描述、甚至是基本的邏輯常識——來指導系統化的資料探索過程。對於缺乏深厚領域知識的新手而言，關鍵在於利用嚴謹的「流程」（如系統化的 EDA、仔細的驗證）來彌補直覺的不足（使用者查詢）。流程本身提供了必要的護欄，防止在未知的資料海洋中迷失方向。即使是最基礎的業務問題理解，也能為資料驅動的步驟提供寶貴的上下文和方向指引。
- 缺乏領域知識的潛在風險：
  當分析師缺乏領域知識時，一個特別突出的風險是難以判斷特徵的「合理性」。他們可能無法直覺地識別出那些在現實世界中不應在預測時點可用的「洩漏特徵」（Leaky Features）30，或者那些僅僅是統計上的巧合而非真實關聯的「虛假相關特徵」（Spurious Features）31。領域知識有助於過濾掉這些看似相關但實則無效或有害的特徵 25。因此，在缺乏領域知識的情況下，分析師必須更加依賴嚴格的驗證程序（如交叉驗證）、批判性思維（不斷質疑發現的模式）以及對潛在陷阱（如資料洩漏、虛假相關）的高度警惕。

3. 系統化資料分析工作流程（基於 CRISP-DM 原則）

為了提供一個結構化、行業認可的方法，本節將介紹一個基於 CRISP-DM（跨行業資料探勘標準流程）框架改編的系統化資料分析工作流程 [3]。CRISP-DM 是一個成熟且廣泛應用的模型，它將資料探勘專案劃分為六個主要階段，為分析師提供了一個清晰的操作指南。

3.1 階段一：定義問題與分析目標（業務理解）

- 目的：
  此階段的核心目標是從業務角度出發，清晰地理解專案的背景、要解決的具體問題，並將其轉化為明確、可衡量的資料分析目標 3。這是整個專案的基石，決定了後續所有工作的方向和重點 35。
- 策略：
  1. 理解業務目標：與利益相關者溝通（如果可能），或深入分析需求文件，以把握最根本的業務需求 [3]。避免停留在模糊的陳述上（例如，從「提升效率」具體化為「在未來六個月內將客服工單處理時間縮短 20%」）[4]。反覆提問：「我們真正要解決的業務問題是什麼？」、「成功的標準是什麼？」[3]。常見目標包括提升銷售額、降低客戶流失率、預測設備故障等 [35]。
  2. 評估現狀：盤點可用資源（資料來源、分析工具、人員技能）、專案需求（時間表、結果品質要求）、潛在限制（資料隱私、法規遵循、技術瓶頸）、基本假設和風險，並進行初步的成本效益分析 [3]。這一步對於判斷專案的可行性至關重要。
  3. 確定資料探勘/分析目標：將宏觀的業務目標轉化為具體的、可透過資料分析實現的技術目標 [3]。同時定義技術層面的成功標準 [3]。例如：業務目標是降低流失率，分析目標可以是「建立一個預測模型，識別可能流失的客戶，準確率達到 X%」。
  4. 識別初步資料需求：基於已定義的目標，初步列出可能需要的資料類型（例如，客戶基本資料、購買歷史、網站瀏覽行為、產品使用數據等）[33]。
  5. 制定初步專案計劃：概述專案的主要階段、時間安排、可能使用的工具和技術 [3]。
- 陷阱與誤區：
  - 目標模糊：在沒有清晰、具體問題的情況下開始分析 [36]。這會導致探索漫無目的，產出不相關的結果。
  - 目標錯位：分析任務與實際業務需求脫節 [2]。資料科學團隊解決了一個業務部門並不關心或認為沒有價值的問題。
  - 解決錯誤問題：誤解了需求，或者只關注了問題的表面現象而非根本原因 [2]。例如，試圖優化某個功能的使用率，而根本問題可能是產品定位錯誤。
  - 框架偏見/預設假設：讓先入為主的觀念影響問題的定義或對預期結果的設定 [1]。
  - 忽視約束條件：未充分考慮資料的可獲取性、品質問題、隱私限制或資源不足就開始專案 [3]。
  - 過度承諾：對資料科學在特定條件下能達成的效果設定了不切實際的期望 [2]。
- 要點闡述：
  問題定義階段是整個資料科學專案中最具挑戰性也最容易出錯的環節，尤其是在初始需求模糊或跨團隊溝通不暢的情況下 2。此階段的失誤意味著後續投入的大量資源可能都浪費在解決一個不重要或錯誤的問題上。因此，分析師必須在此階段投入足夠的

時間和精力，透過不斷提問來澄清需求，精確界定分析的範圍和目標。即使缺乏深厚的領域知識，專注於定義「可衡量」的目標和清晰的「範圍」，也能為後續的分析工作提供必要的結構和指引 4。將模糊的業務語言轉化為具體的、可用數據回答的問題，本身就是一種重要的結構化過程，有助於彌補領域直覺的不足。

3.2 階段二：初步資料理解與結構化（系統化 **EDA**）

- 目的：
  此階段旨在初步接觸和熟悉原始資料，評估其整體品質，理解其基本結構和特徵，識別潛在的問題（如缺失值、異常值），並獲得初步的洞察 3。對於處理來自未知領域的資料，這一階段尤為關鍵，因為它為後續的數據準備和建模奠定了基礎 10。EDA 的目標是培養對資料的「感覺」和理解 7。
- 策略：針對未知領域的初步探索清單：
  1. 載入資料與基本檢視：
     - 將資料導入分析環境（如 Python Pandas DataFrame）[42]。
     - 檢查資料維度（Shape）：有多少行（觀測值）和多少列（特徵）？使用 .shape 屬性 [41]。
     - 查看頭部和尾部數據 (.head(), .tail())：快速了解資料的數值和欄位名稱 [41]。
     - 列出所有欄位名稱 (.columns)：確認包含的特徵 [41]。
     - 檢查資料類型 (.info(), .dtypes)：識別數值型、類別型、物件型、日期時間型等。判斷類型是否符合預期？[41]。不匹配的類型需要在後續階段修正 [43]。
  2. 初步資料品質掃描：
     - 檢查缺失值 (.isnull().sum())：量化每列的缺失數據。缺失比例是多少？[41]。記錄下缺失嚴重的特徵，以便後續處理。如果可能，嘗試理解數據缺失的原因 [49]。
     - 檢查重複行 (.duplicated().sum())：識別並考慮移除完全重複的記錄 [43]。
  3. 變數摘要（描述性統計）：
     - 數值型變數 (.describe())：計算計數、平均值、標準差、最小值、最大值、四分位數 [12]。注意觀察是否存在不合理的範圍（如負數年齡）、平均值與中位數差異過大（偏態）、標準差極高或極低等情況。
     - 類別型變數 (.describe(include='object'), .value_counts())：獲取計數、唯一值數量、最高頻類別及其頻率 [43]。檢查基數（Cardinality，即唯一值的數量）——基數過高或過低的類別型變數可能需要特殊處理 [51]。
  4. 首次視覺化（聚焦分佈）：
     - 數值型變數：繪製直方圖（Histograms, .hist(), seaborn.histplot）或核密度估計圖（Density Plots, seaborn.kdeplot）觀察分佈形態（常態、偏態、多峰等）[43]。繪製箱型圖（Box Plots, seaborn.boxplot）觀察數據的散佈範圍、中位數、四分位數以及潛在的異常值 [43]。
     - 類別型變數：繪製條形圖（Bar Charts, seaborn.countplot）觀察各類別的頻數 [41]。
  5. 提出問題（引導式探索）：利用統計摘要和視覺化圖表，提出探索性問題：哪些數值

最常見？哪些很少見？是否存在異常值？資料中是否存在明顯的群組或聚類？分佈是否符合（即使是模糊的）預期？[7]。最重要的是，即使缺乏領域知識，也要問「這看起來合理嗎？」。是否存在邏輯上不可能的數值（例如，年齡小於 0）？[46]。

6. 記錄發現（建立資料字典/筆記）：創建一個結構化的資料描述文檔：記錄欄位名稱、資料類型、欄位含義（如果可知）、單位、允許值範圍、以及關於數據品質問題或初步發現的筆記 [33]。這對於理解未知領域的資料至關重要。

- 陷阱與誤區：
  - 跳過或草率進行 **EDA**：沒有充分理解資料就直接進入建模階段，容易導致「垃圾進，垃圾出」[3]。
  - 表面檢查：僅僅依賴 .info() 或 .describe() 的輸出，而忽略了透過視覺化來觀察分佈 [6]。平均值等統計量可能掩蓋多峰分佈或異常值的存在。
  - 誤解缺失數據：簡單地假設缺失是隨機發生的，而實際上可能是系統性缺失（MNAR），具有特定含義 [43]。直接刪除行或列可能引入偏見 [49]。
  - 忽略資料類型：將數值編碼的類別誤認為連續數值，或未將日期字串轉換為日期時間格式 [46]。
  - 漫無目的地繪圖：沒有具體問題引導，隨意生成大量圖表，難以提取有效資訊 [7]。
  - **EDA** 中的確認偏誤：只關注那些支持自己初步想法的模式，忽略與之矛盾的證據 [55]。
- 要點闡述：
  對於缺乏領域知識的新手，系統化的 EDA 提供了一套結構化的探索路徑。透過遵循清單式的步驟，並圍繞「變異性」（Variation，單個變數內部的變化）和「共變性」（Covariation，變數之間的關聯變化）這兩個核心問題來提問 7，分析的焦點從「這個數據在該領域代表什麼？」轉變為「這個數據具有哪些統計和結構特性？」。這種方法論本身就提供了必要的指導，彌補了領域直覺的缺乏。初步的視覺化，如直方圖和箱型圖，是理解所有變數分佈特徵、快速識別潛在問題（如異常值、偏態）的關鍵第一步 43。這些圖表易於生成，能直觀地總結數據特性，為後續的數據清理和特徵工程階段標示出需要關注的重點。

3.3 階段三：生成初步假設

- 目的：
  基於在 EDA 階段獲得的洞察，提出關於資料內部關係或潛在問題答案的、具體的、可檢驗的、有根據的猜測（即假設）41。此階段是從廣泛的資料理解過渡到有針對性的分析或建模的橋樑。
- 策略：
  1. 連結 **EDA** 觀察結果：將 EDA 中發現的模式（如相關性、群組差異、趨勢）與可能的解釋或變數間關係聯繫起來 [59]。例如：「EDA 顯示 A 區域的銷售額較高。假設：A 區域的市場行銷投入更高。」
  2. 構建可檢驗的陳述：將假設表述為清晰的、通常是關於變數之間關係的陳述 [58]。例如：「假設：網站訪問量（變數 X）的增加與更高的轉化率（變數 Y）呈正相關。」

3. 考慮零假設與對立假設：嘗試將假設框定為能夠進行統計檢驗的形式（儘管正式檢驗可能在後續階段進行）。零假設（H0）通常表述為「沒有效應」或「沒有關係」。對立假設（H1）則表述為「存在效應」或「存在關係」[59]。
4. 利用基本邏輯和資料結構：即使沒有深入的領域知識，也可以利用資料的內在結構（如時間序列、客戶資料分組）來生成合理的假設 [60]。例如，基於時間 = 距離 / 速度 的常識，可以假設行程距離與行程時間正相關 [60]。
5. 區分假設生成與假設檢驗：強調此階段的重點是基於初步的數據探索「產生」可能的想法，而不是立即進行嚴格的「證明」[58]。正式的檢驗需要更深入的分析或建模。

- 陷阱與誤區：
  - 確認偏誤（**Confirmation Bias**）：只提出那些支持自己先入為主的觀點的假設，而忽略 EDA 中發現的矛盾證據 [55]。應主動尋找反駁證據 [56]。
  - 不可檢驗的假設：提出模糊的猜測，無法用現有數據或方法進行驗證。
  - 事後修改假設（**HARKing - Hypothesizing After the Results are Known**）：在看到分析或模型結果後，回過頭來修改假設以使其符合結果。這違背了科學探究的原則 [66]。假設應在深入分析或建模「之前」提出。對於意外的發現，應報告為探索性的或假設生成性的，而非驗證性的 [66]。
  - 混淆相關性與因果性：在假設中，僅僅基於 EDA 中觀察到的相關性就假定存在因果關係 [31]。除非明確提出要檢驗某個因果機制（這通常需要實驗設計），否則初步假設應側重於描述關聯性。
- 要點闡述：
  假設生成扮演著關鍵的「聚焦」角色，將廣泛的探索（EDA）轉化為有目標的深入調查 58。即使這些假設只是基於資料本身的結構特徵（而非深奧的領域理論）提出的簡單猜想，它們也為後續分析提供了必要的方向感。對於新手，尤其是在缺乏領域知識的情況下，假設生成階段最主要的危險是確認偏誤 55。這更加凸顯了主動質疑初步假設、積極考慮對觀察到的模式的其他可能解釋的重要性。例如，分析師應自問：「這個相關性會不會是虛假的？」「是否存在其他變數影響了這個結果？」

3.4 階段四：數據處理與特徵工程實踐

- 目的：
  根據 EDA 階段的發現對資料進行清理，並系統地創建、轉換和選擇特徵，以優化用於建模的資料集，旨在提高模型的準確性、效率和可解釋性 8。
- 策略：深入探討技術細節：
  A. 資料清理（**Data Cleaning**）：處理 **EDA** 中發現的問題
  - 處理缺失值（**Handling Missing Values**）：
    - 目的：避免數據丟失，防止模型因無法處理 NaN 值而出錯，減少因缺失模式引入的偏見 [8]。
    - 技術：
      - 刪除（行/列）：適用於數據量充足且缺失比例低，或某列幾乎全空（如 > 30-50%）的情況。需謹慎使用，可能丟失資訊或引入偏見 [15]。

        - 均值/中位數填充（數值型）：簡單常用。數據偏態時使用中位數更佳 [12]。可能扭曲變數的方差和相關性。
        - 眾數填充（類別型）：用最頻繁出現的類別替換缺失值 [8]。適用於某個類別佔主導地位的情況。
        - 任意值/新類別填充：指定一個特殊值（如 -1, 999）或類別（如 "Missing", "Unknown"）來標示缺失 [46]。如果缺失本身包含信息（非隨機缺失），這種方法有時能捕捉到。
        - 進階方法（簡述）：K-近鄰填充（KNN Imputation）、多重插補（MICE）、基於模型的填充 [14]。更複雜，但可能更準確。
    - 選擇時機：依據缺失比例、數據類型、變數分佈、缺失原因（MCAR, MAR, MNAR）以及模型對缺失值的敏感度來決定 [21]。
- 異常值檢測與處理（**Outlier Detection & Treatment**）：
    - 目的：防止異常值扭曲統計量（如均值）、干擾模型訓練（尤其是線性模型、基於距離的演算法），或違反模型假設 [12]。
    - 檢測技術：視覺化（箱型圖、散點圖）、統計規則（Z 分數 > 3、IQR 法則：超出 Q1-1.5*IQR* 或 *Q3+1.5*IQR）[12]。演算法（孤立森林 Isolation Forest、DBSCAN - 較進階）[18]。
    - 處理技術：
        - 刪除：移除包含異常值的觀測。謹慎使用 [7]。
        - 轉換：應用對數、平方根、Box-Cox 轉換來減小異常值的影響 [12]。
        - 縮尾（Winsorization）/蓋帽（Capping）：用某個百分位數（如第 1/99 百分位數）替換極端值 [16]。
        - 插補：將異常值視為缺失值進行填充 [21]。
        - 保留：如果異常值是真實且包含重要信息的，或者使用的模型對異常值不敏感（如樹模型），則可以保留。
    - 選擇時機：取決於異常值的成因（是錯誤還是真實極端值？）、出現頻率、對模型的影響以及所選演算法的特性。務必先調查異常值 [7]。如果異常值對結果有顯著影響，不能在沒有充分理由的情況下隨意丟棄 [7]。

B. 特徵轉換（**Feature Transformation**）：修改現有特徵

- 縮放/歸一化（**Scaling/Normalization**）：
    - 目的：將不同特徵的數值範圍調整到可比較的尺度。對於基於距離的演算法（KNN, SVM, 聚類）、使用梯度下降優化的模型（神經網路、線性回歸）以及 PCA 等方法至關重要 [9]。防止數值範圍大的特徵主導模型訓練過程 [15]。
    - 技術：
        - 標準化（Standardization, Z-score Scaling）：將數據轉換為均值為 0，標準差為 1 的分佈。公式：(x - mean) / std_dev。相較於 Min-Max Scaling，受異常值影響較小。通常是預設的優先選擇 [11]。
        - 歸一化（Normalization, Min-Max Scaling）：將數據重新縮放到一個固定的區間，通常是 。公式：(x - min) / (max - min)。對異常值敏感。當需要數據

在特定範圍內時（如圖像像素強度）比較有用 [11]。
    - 選擇時機：一般情況下優先考慮標準化，除非需要特定的數值範圍或已妥善處理異常值。關鍵點：必須在劃分訓練集和測試集「之後」進行縮放，且縮放器（Scaler）應僅在訓練數據上進行 fit（學習參數），然後再對訓練集和測試集（以及驗證集）進行 transform（應用轉換）[30]。
- 類別數據編碼（**Encoding Categorical Data**）：
    - 目的：將類別型特徵（通常是文本標籤）轉換為機器學習演算法能夠處理的數值格式 [9]。
    - 技術：
        - 獨熱編碼（One-Hot Encoding, OHE）：為每個類別創建一個新的二元（0/1）欄位。最適用於名目型數據（Nominal Data，類別間無序）。能避免引入錯誤的順序關係。缺點是如果類別數量（基數）很多，會產生大量新欄位，導致維度災難 [9]。
        - 標籤編碼（Label Encoding）：為每個唯一類別分配一個整數（例如，紅色=0, 藍色=1, 綠色=2）。簡單，不會增加維度。適用於順序型數據（Ordinal Data，類別有明確順序，如低、中、高）或某些樹模型。對於名目型數據，可能讓模型誤認為類別間存在數值大小關係 [15]。
        - 目標編碼（Target Encoding）：用該類別對應的目標變數（Target Variable）的平均值來替換類別。能捕捉類別與目標之間的關係，且不增加維度。但有過度擬合和數據洩漏的風險，需要非常謹慎地實施（例如，僅使用訓練集內的交叉驗證折疊來計算均值）[13]。
        - 計數/頻率編碼（Count/Frequency Encoding）：用類別在資料集中出現的次數或頻率來替換類別 [71]。可以捕捉類別的普遍性。
    - 選擇時機：低基數名目型數據常用 OHE。順序型數據或樹模型可用標籤編碼。高基數類別可謹慎嘗試目標編碼或計數編碼。
- 數學轉換（對數、平方根、**Box-Cox**）：
    - 目的：處理偏態分佈（使其更接近常態分佈），穩定變異數，使關係線性化，減輕異常值的影響 [12]。
    - 技術：對數轉換 (log(x)) 常用於右偏數據。平方根轉換 (sqrt(x))。Box-Cox 轉換（自動尋找最佳的冪次轉換）。
    - 選擇時機：當 EDA 顯示數值型特徵高度偏斜，或者模型假設數據服從常態分佈時（如線性回歸的某些假設）。

C. 特徵創建（**Feature Creation**）：生成新特徵

- 分箱/離散化（**Binning/Discretization**）：
    - 目的：將連續的數值型變數轉換為離散的類別區間（Bins）[9]。可以捕捉非線性效應，減少噪音和異常值的影響，有時能提升某些模型（如決策樹、樸素貝葉斯）的性能 [12]。
    - 技術：等寬分箱（Equal-width bins）、等頻分箱（Equal-frequency/quantile bins）、基於領域知識或聚類的自定義分箱、基於決策樹的分箱 [71]。

        - 選擇時機：處理非線性關係、簡化特徵、降噪，或當演算法需要類別輸入時。如果分箱過於粗糙，可能導致資訊損失。
    - 交互特徵（**Interaction Features**）：
        - 目的：捕捉兩個或多個特徵之間的組合效應。通常透過特徵相乘、相加、相減或相除來創建 [9]。有助於模型理解非線性關係。
        - 選擇時機：當領域知識或 EDA 暗示特徵之間存在交互作用時（例如，廣告支出的效果可能因季節而異）。可能導致「特徵爆炸」（Feature Explosion），即特徵數量急劇增加 [14]。
    - 多項式特徵（**Polynomial Features**）：
        - 目的：創建現有特徵的冪次（如 x^2, x^3）或交叉乘積（類似交互項）的新特徵 [14]。幫助線性模型擬合非線性關係。
        - 選擇時機：當特徵與目標之間的關係呈現非線性（如 U 形）時。會增加模型複雜度，有過度擬合的風險 [14]。
    - 領域特定特徵（**Domain-Specific Features**）：
        - 基於對業務背景的理解創建的特徵（例如，距離上次購買的時間、到最近商店的距離、根據身高體重計算的 BMI 指數）[9]。通常具有很強的預測能力。
    - 日期/時間特徵（**Date/Time Features**）：
        - 從日期時間變數中提取年、月、日、星期幾、小時、是否週末、時間差等成分 [16]。對於時間序列或事件相關數據至關重要。
- 陷阱與誤區：
    - 數據洩漏（**Data Leakage**）：這是最嚴重且最常見的陷阱之一。指在劃分數據集「之前」就進行了預處理（如縮放、填充），或者使用目標變數信息來創建特徵（如不當使用目標編碼），或在時間序列中使用了未來信息 [23]。這會導致模型在驗證集上表現過於樂觀，但在實際部署時效果很差。應對策略：務必先劃分數據（訓練集/驗證集/測試集）。所有的轉換器（Scaler, Encoder, Imputer）都應「僅」在訓練數據上進行 fit（學習參數），然後再對驗證集和測試集進行 transform（應用轉換）。處理時間序列數據時要格外小心交叉驗證方法。
    - 過度擬合（**Overfitting**）：創建過多的特徵（尤其是交互項、多項式特徵），或基於驗證集性能反覆選擇特徵，或不謹慎地使用目標編碼等技術，都可能導致模型過度學習訓練數據中的噪音，而無法泛化到新數據 [23]。應對策略：使用正規化（Regularization）技術、交叉驗證、基於訓練集折疊（folds）的特徵選擇、降維技術。
    - 欠擬合（**Underfitting**）：過度簡化特徵（如過於粗略的分箱）或移除了過多潛在有用的特徵，可能導致模型無法捕捉數據中的基本模式 [24]。
    - 技術選擇不當：對名目型數據使用標籤編碼（用於線性模型時），對帶有異常值的數據使用 Min-Max 縮放，選擇錯誤的缺失值填充方法等。
    - 過度處理：進行不必要的轉換，可能丟失有價值的信息或使特徵更難解釋。
    - 忽略特徵重要性：沒有評估工程化的特徵是否真正提升了模型性能（應使用基準測試）[8]。
- 要點闡述：

特徵工程階段是「數據洩漏」最高發的區域。新手分析師必須牢記並嚴格遵守「先劃分，後處理；fit on train, transform test」的原則，這適用於所有在初始數據劃分之後進行的預處理步驟 30。因為數據洩漏發生在數據準備階段，模型接觸數據之前，所以嚴格的流程控制是防止模型性能被虛假抬高的關鍵。此外，特徵的複雜性（如引入交互項、多項式項）與過度擬合的風險直接相關 14。創建更複雜的特徵意味著模型需要學習更多參數，更容易捕捉到噪音。因此，特徵選擇成為管理這種複雜性、防止模型過度擬合的重要手段 24。新手在添加複雜特徵時應保持謹慎，並依賴交叉驗證和特徵選擇來評估其真實效果。

- 表格：特徵工程技術選擇指南

技術名稱 **(Technique Name)**	目的 **(Purpose)**	適用時機 **(When to Use)**	主要優點 **(Key Pro/Advantage)**	主要缺點/風險/注意事項 **(Key Con/Pitfall/Caution)**
均值/中位數填充 (Mean/Median Imputation)	填充數值型缺失值	數值型數據；數據偏態時用中位數	簡單快速，保留樣本量	可能扭曲變數分佈、方差和相關性
眾數填充 (Mode Imputation)	填充類別型缺失值	類別型數據；尤其適用於某類別佔主導時	簡單快速，保持類別一致性	可能加劇類別不平衡，忽略潛在關係
獨熱編碼 (One-Hot Encoding, OHE)	將名目型類別轉為數值	低基數的名目型數據	避免引入錯誤順序，適用多數模型	高基數時導致維度災難，可能產生共線性
標籤編碼 (Label Encoding)	將類別轉為整數標籤	順序型數據；樹模型	不增加維度，保留順序信息	對名目型數據可能引入錯誤順序（對線性模型有害）
目標編碼 (Target Encoding)	用目標變數均值替換類別	高基數類別數據；希望捕捉目標關係時	不增加維度，直接關聯目標	高數據洩漏風險，易過度擬合，需謹慎實施
標準化 (Standardization)	將數據縮放至均值0,標準差1	多數數值型數據；基於距離或梯度的模型	不受特定範圍限制，對異常值相對穩健	轉換後數據不易直觀解釋範圍
歸一化 (Normalization)	將數據縮放至或指定範圍	需要特定範圍的數值數據（如圖	範圍直觀	對異常值敏感

		像）；某些神經網路		
對數轉換 (Log Transform)	處理右偏數據，減小極值影響	右偏（正偏）的數值型數據	使數據更接近常態，穩定方差	不能直接用於含0或負值的數據
分箱/離散化 (Binning)	將連續數值轉為類別區間	需要處理非線性；降噪；某些模型需要類別輸入	簡化模型，捕捉非線性，對異常值穩健	可能損失信息，分箱方式影響結果
交互特徵 (Interaction Features)	捕捉特徵間的組合效應	懷疑或發現特徵間存在交互作用時	能提升模型表達能力，捕捉複雜關係	可能導致特徵數量爆炸，增加過擬合風險
多項式特徵 (Polynomial Features)	捕捉單個特徵的非線性關係	特徵與目標關係呈非線性（如曲線）時	使線性模型能擬合非線性關係	增加模型複雜度，高過擬合風險

3.5 階段五：分析關係——相關性及其他

- 目的：
  探索並量化「特徵之間」以及「特徵與目標變數（如果存在）」之間的關係，識別潛在的多重共線性（高度相關的預測變數），並驗證在階段三中提出的初步假設 3。
- 策略：
  1. 相關性分析（**Correlation Analysis**）：
     - 計算相關係數：主要使用皮爾遜相關係數（Pearson's r）來衡量「數值型變數」之間的「線性」關係強度和方向 [22]。可以使用 Pandas 的 .corr() 方法計算相關矩陣。
     - 解讀強度與方向：係數範圍在 -1 到 +1 之間 [22]。接近 +1 表示強正相關，接近 -1 表示強負相關，接近 0 表示線性關係很弱或不存在 [77]。可以設定閾值來判斷相關性強度（例如，絕對值 > 0.7 為強相關，0.3-0.7 為中等相關，< 0.3 為弱相關）[22]。
     - 考慮其他相關性度量：對於非線性但單調的關係或順序型數據，可以考慮斯皮爾曼等級相關係數（Spearman's Rank Correlation）[80]。對於類別變數之間的關聯性，可以使用卡方檢驗（Chi-Square Test）或克萊姆 V 值（Cramer's V）[45]。
  2. 視覺化關係：
     - 散點圖（Scatter Plots）：用於視覺化兩個數值型變數之間的關係 [44]。觀察是否存在線性趨勢、非線性模式、異常值或群集。當數據點過多導致重疊時，可以使用透明度（alpha）或二維分箱圖（geom_bin2d/geom_hex）[44]。
     - 相關矩陣熱力圖（Correlation Matrix Heatmap）：將數值變數的相關係數矩陣

視覺化 [82]。可以快速發現強相關（顏色深淺表示強度）和多重共線性（預測變數之間的高度相關）。
    - 配對圖（Pair Plots, seaborn.pairplot）：同時展示多個數值變數兩兩之間的散點圖，以及每個變數自身的分佈圖（直方圖或密度圖）[42]。適用於特徵數量不多時的概覽。
    - 箱型圖/小提琴圖（Box Plots / Violin Plots）：用於視覺化一個數值型變數和一個類別型變數之間的關係，展示數值變數在不同類別下的分佈情況 [44]。
- 陷阱與誤區：
    - 相關性 **vs.** 因果性（**Correlation vs. Causation**）：這是解讀相關性時最關鍵、最容易犯的錯誤。相關性僅僅表示兩個變數之間存在關聯，並「不」意味著其中一個變數導致了另一個變數的變化 [22]。應對策略：永遠不要僅憑相關性推斷因果關係。確定因果關係通常需要嚴謹的實驗設計或強有力的理論支持。
    - 虛假相關（**Spurious Correlations**）：兩個變數看起來相關，但實際上是偶然巧合，或者是由於受到同一個未被觀測到的第三方變數（潛在變數，Confounding Variable）的影響 [31]。例如，冰淇淋銷量和鯊魚襲擊事件都與氣溫相關，它們之間的正相關是虛假的 [32]。應對策略：運用領域知識（如果有的話）和批判性思維（「這個關聯在現實中有意義嗎？」），尋找可能的潛在變數，並對結果進行驗證。對於沒有先驗假設、僅透過大規模數據挖掘發現的相關性要格外警惕 [85]。
    - 忽略非線性關係：皮爾遜相關係數 r 只衡量「線性」相關。r 值很低並不代表兩個變數之間沒有關係，關係可能是非線性的（例如 U 形）[80]。應對策略：務必使用散點圖等視覺化手段檢查變數關係，不能僅依賴相關係數。
    - 異常值的影響：少數異常值可能極大地扭曲相關係數的計算結果 [80]。應對策略：透過視覺化和統計方法檢查異常值；考慮使用對異常值不敏感的相關性度量方法（如 Spearman）或先處理異常值。
    - 辛普森悖論（**Simpson's Paradox**）（隱含）：在數據的不同分組中呈現出一種趨勢，但當這些分組合併後，趨勢消失或反轉。總體相關性可能具有誤導性。應對策略：在 EDA 階段識別出的相關子群體內部分析相關性。
    - 誤解相關強度：過分強調弱相關的重要性，或者錯誤理解相關係數的平方（在迴歸分析中稱為 R 平方）所代表的意義（解釋的變異比例）[87]。
- 要點闡述：
  在關係分析階段，視覺化手段（尤其是散點圖）的作用至關重要，它可以幫助我們避免僅憑相關係數做出錯誤判斷，特別是在識別非線性關係和異常值方面 44。相關係數提供了量化的指標，而圖形則提供了直觀的驗證和更豐富的上下文信息。虛假相關的概念是一個主要的認知陷阱，尤其是在探索未知領域時，因為此時我們缺乏判斷因果聯繫的直覺。對此，保持懷疑態度和進行批判性思考是主要的防禦手段 31。必須養成習慣，不斷質疑觀察到的關係是否合理，是否存在潛在的混淆因素。

4. 溝通洞見：掌握資料視覺化

資料視覺化是將分析結果和洞見有效傳達給他人的關鍵環節。它不僅僅是製作圖表，更是利用視覺元素清晰、準確地講述數據故事的藝術和科學。

4.1 選擇正確的視覺化圖表：目標驅動的框架

- 目的：
  根據分析的目標和想要傳達的訊息，選擇最有效的圖表類型，以準確、清晰地展示數據中的模式、趨勢或比較關係 52。
- 策略：基於分析目標選擇圖表：
  選擇圖表的核心依據應該是你想透過圖表回答什麼問題，或者展示數據的哪個方面。以下是基於常見分析目標的圖表選擇建議：
  1. 比較（**Comparison**）：展示不同項目或組別之間的數值差異或相似性。
     - 適用圖表：
       - 條形圖/柱狀圖（**Bar/Column Charts**）：最常用於精確比較不同類別的數值大小 [52]。
       - 分組條形圖（**Grouped Bar Charts**）：用於比較兩個不同分組變數下的數值 [52]。
       - 折線圖（**Line Charts**）：用於比較不同組別隨時間變化的趨勢 [52]。
       - 點圖（**Dot Plots**）：類似條形圖，但用點的位置表示數值，適用於基線不為零或類別標籤較長的情況 [52]。
       - 散點圖（**Scatter Plots**）：可用於比較兩個數值變數的關係 [81]。
       - 子彈圖（**Bullet Charts**）：用於將實際值與目標值或基準值進行比較 [52]。
       - 棒棒糖圖（**Lollipop Charts**）：條形圖的變種，視覺上更簡潔 [83]。
  2. 分佈（**Distribution**）：展示單個變數的數值分佈情況（頻率、範圍、集中趨勢、離散程度），或比較不同組別之間的分佈差異。
     - 適用圖表：
       - 直方圖（**Histograms**）：展示數值型變數的頻率分佈 [52]。
       - 核密度估計圖（**Density Plots**）：直方圖的平滑版本，更清晰地顯示分佈形狀 [52]。
       - 箱型圖（**Box Plots**）：簡潔地展示數值變數的中位數、四分位數、範圍和異常值，非常適合比較多個組別的分佈 [44]。
       - 小提琴圖（**Violin Plots**）：結合了箱型圖和密度圖的優點，展示分佈形狀和關鍵統計量 [44]。
       - 條形圖（**Bar Charts**）：展示類別型變數的頻率分佈 [44]。
       - 人口金字塔（**Population Pyramid**）：特殊的直方圖，用於比較兩個群體（通常是性別）在不同年齡段的分佈 [83]。
  3. 構成（**Composition / Part-to-Whole**）：展示整體如何由各個部分組成，強調各部分佔總體的比例。
     - 適用圖表：
       - 餅圖（**Pie Charts**）：最直觀的構成圖，但應謹慎使用，最好用於類別較少（

< 5)的情況 [52]。
        - 環圈圖(**Donut Charts**):餅圖的變種,中間留空 [52]。
        - 堆疊條形圖/柱狀圖(**Stacked Bar/Column Charts**):在條形/柱狀圖內部顯示各組成部分的絕對值或百分比 [52]。
        - 堆疊面積圖(**Stacked Area Charts**):折線圖的變種,用不同顏色的面積表示各部分隨時間變化的構成 [52]。
        - 樹狀圖(**Treemaps**):用嵌套的矩形面積表示層級結構和各部分佔比 [52]。
        - 旭日圖(**Sunburst Diagram**):類似樹狀圖,但使用環狀分層結構 [83]。
        - 馬賽克圖(**Marimekko Chart**):同時展示兩個類別變數的構成比例 [52]。
4. 關係(**Relationship / Correlation**):展示兩個或多個變數之間的關聯或相關性。
    - 適用圖表:
        - 散點圖(**Scatter Plots**):最常用於展示兩個數值變數之間的關係 [52]。
        - 氣泡圖(**Bubble Charts**):在散點圖基礎上,用氣泡的大小或顏色表示第三個變數 [52]。
        - 熱力圖(**Heatmaps**):用顏色深淺表示矩陣(如相關係數矩陣)中的數值大小,或展示兩個類別變數之間的關係強度 [52]。
        - 連線散點圖(**Connected Scatter Plots**):當第三個變數是時間時,用線段連接散點圖中的點 [52]。
5. 隨時間變化(**Change Over Time / Trend**):展示數據如何隨著時間的推移而演變。
    - 適用圖表:
        - 折線圖(**Line Charts**):最常用於展示連續時間序列的趨勢 [52]。
        - 面積圖(**Area Charts**):強調隨時間變化的總量或體量 [52]。
        - 柱狀圖(**Column Charts**):可用於展示離散時間點的數值變化 [52]。
        - 箱型圖(**Box Plots**):展示不同時間段內數值的分布變化 [52]。
        - **K**線圖(**Candlestick Chart**)**/ OHLC**圖:金融領域常用,展示開盤價、最高價、最低價、收盤價隨時間的變化 [52]。
6. 地理空間數據(**Geographical Data**):展示與地理位置相關的數據。
    - 適用圖表:
        - 分級統計圖(**Choropleth Maps**):用不同顏色或陰影填充地理區域(如國家、省份)以表示數值大小 [52]。
        - 符號地圖/氣泡地圖(**Symbol/Bubble Maps**):在地圖上用不同大小或顏色的符號(如圓點)表示數值 [83]。
        - 連接地圖(**Connection Maps**):用線條在地圖上表示不同地點之間的聯繫或流動 [83]。
        - 變形地圖(**Cartograms**):根據數值大小扭曲地理區域的面積 [52]。
- 考慮數據類型:確保所選圖表適合要展示的數據類型(數值型 vs. 類別型,離散 vs. 連續)[82]。
- 考慮受眾:根據受眾的專業背景和理解能力調整圖表的複雜性 [81]。

- 陷阱與誤區：
  - 選錯圖表類型：例如用餅圖展示時間趨勢，或用折線圖展示無序的類別數據 [94]。這會嚴重妨礙信息的傳達。
  - 過度依賴常用圖表：總是默認使用條形圖或餅圖，而忽略了其他可能更有效的圖表類型 [92]。
- 表格：圖表選擇框架

分析目標 **(Analysis Goal)**	目標描述 **(Description)**	推薦圖表類型 **(Recommended Chart Types)**	數據類型說明 **(Data Type Notes)**
比較 **(Comparison)**	比較不同項目或組別的數值大小	條形圖/柱狀圖, 分組條形圖, 折線圖 (時間比較), 點圖, 散點圖, 子彈圖, 棒棒糖圖	主要用於比較數值；條形圖用於類別比較
分佈 **(Distribution)**	展示數據的頻率、範圍、形狀	直方圖, 核密度估計圖, 箱型圖, 小提琴圖, 條形圖 (類別頻率), 人口金字塔	直方圖/密度圖/箱型圖/小提琴圖用於數值；條形圖用於類別
構成 **(Composition)**	展示整體與部分的關係，比例	餅圖/環圈圖 (少類別), 堆疊條形/柱狀圖, 堆疊面積圖, 樹狀圖, 旭日圖, 馬賽克圖	適用於展示百分比或絕對值構成
關係 **(Relationship)**	探索或展示變數間的關聯性	散點圖, 氣泡圖, 熱力圖, 連線散點圖	散點圖/氣泡圖用於數值；熱力圖可用于數值或類別
隨時間變化 **(Trend)**	展示數據隨時間的演變趨勢	折線圖, 面積圖, 柱狀圖, 箱型圖 (分佈隨時間), K線圖/OHLC圖 (金融)	時間軸通常是橫軸
地理空間 **(Geospatial)**	展示與地理位置相關的數據	分級統計圖, 符號/氣泡地圖, 連接地圖, 變形地圖	需要地理位置信息（經緯度、區域名稱）

4.2 有效的視覺編碼：將數據映射到視覺元素

- 目的：
  理解如何將數據的不同維度（變數）有效地映射到圖形的視覺屬性（如位置、顏色、大

小、形狀）上，以及人類的視覺感知原理如何影響圖表的解讀 91。

- 策略：關鍵原理與技術：
  1. 視覺變數/通道（**Visual Variables/Channels**）：介紹由 Jacques Bertin 或後來的 Cleveland & McGill 等人提出的視覺變數概念，包括：位置（Position）、長度（Length）、角度（Angle）、面積（Area）、體積（Volume）、形狀（Shape）、顏色色調（Color Hue）、顏色飽和度/明度/亮度（Color Saturation/Value/Luminance）、紋理（Texture）、方向（Orientation）等 [99]。這些是構成視覺化圖形的基礎元素。
  2. 數據類型與視覺通道的映射：
     - 定量數據（**Quantitative Data - 表示量級**）：最適合用「位置」（沿共同標尺，最準確）、其次是「長度」、「角度」、「面積」、「顏色飽和度/明度」來編碼 [100]。應避免使用「顏色色調」來表示精確的量級差異。
     - 順序數據（**Ordinal Data - 表示有序類別**）：可以用「位置」、「長度」、「顏色飽和度/明度」來編碼，關鍵是要保持數據的內在順序 [102]。
     - 類別/名目數據（**Categorical/Nominal Data - 表示身份區分**）：最適合用「顏色色調」、「形狀」、「紋理」來編碼 [100]。「位置」也可以用來將不同類別的數據分組。
  3. 感知層級（**Perceptual Hierarchy - Cleveland & McGill**）：強調人類在感知不同視覺通道編碼的定量信息時，準確度存在差異。大致排序為：位置（沿共同標尺）> 長度 > 角度 > 面積 > 體積/顏色 [100]。在進行重要的定量比較時，應優先選擇排名靠前的視覺通道。
  4. 設計原則：
     - 表達性（**Expressiveness**）：視覺化應「僅僅」且「完全」地表達數據屬性中的信息，不多也不少 [98]。避免添加與數據無關或誤導性的視覺元素。
     - 有效性（**Effectiveness**）：數據屬性的「重要性」應與其視覺通道的「顯著性」（易感知性、準確性）相匹配 [98]。最重要的信息應使用最有效的通道來編碼。
     - 一致性（**Consistency**）：在整個視覺化或系列視覺化中，保持編碼方式的一致性。相同的事物看起來應該相同，不同的事物看起來應該不同 [99]。
     - 格式塔原則（**Gestalt Principles**）：解釋接近性（Proximity）、相似性（Similarity）、閉合性（Closure）、連續性（Continuity）等原則如何影響我們對視覺元素的分組和感知 [91]。應有意識地利用這些原則來組織圖表佈局（例如，將相關的條形分組）。
     - 數據墨水比（**Data-Ink Ratio - Tufte**）：最大化用於呈現數據的「墨水」（或像素），最小化非數據墨水（如不必要的網格線、邊框、裝飾、背景等，即「圖表垃圾」Chart Junk）[88]。保持簡潔 [82]。
     - 清晰的元素：使用信息豐富的標題、清晰的軸標籤（包含單位）、易於理解的圖例、有策略地添加註釋來解釋複雜模式或突出關鍵點 [82]。確保字體清晰易讀 [91]。
     - 色彩理論（**Color Theory**）：有目的地使用顏色（用於突出顯示、區分類別、透過飽和度/明度編碼數值）。注意色盲用戶的可訪問性 [88]。限制使用的不同色

調的數量 [102]。
    - 留白（**White Space**）：有效利用空白區域來構建佈局、分隔元素、降低視覺混亂 [91]。
- 陷阱與誤區：
    - 無效編碼：使用面積或顏色色調來編碼需要精確比較的定量數據 [100]。
    - 編碼不一致：用相同的顏色代表不同的類別，或用不同的形狀代表同一組數據 [95]。
    - 圖表垃圾/混亂：過多的非必要元素（網格線、邊框、裝飾、過多的標籤）掩蓋了數據本身 [88]。
    - 標籤不清：缺少軸標籤、標題含糊、圖例難以理解 [91]。
    - 忽略感知原理：因選擇了不當的視覺通道或違反了格式塔原則，導致圖表難以解碼 [98]。
- 要點闡述：
  有效的資料視覺化不僅在於選擇正確的圖表類型，更關鍵的是如何將數據的維度「映射」到視覺屬性上，並利用人類的視覺感知特性來確保信息的清晰傳達 100。理解感知層級（例如，位置比顏色更適合表示精確數值）和設計原則（如表達性、有效性）是從「製作圖表」到「設計有效視覺化」的關鍵一步。簡潔和清晰是最高原則。透過最小化視覺混亂（圖表垃圾）和使用清晰、一致的編碼，可以降低觀眾的認知負荷，使信息更容易被快速、準確地理解 82。這對於需要向可能非技術背景的受眾呈現分析結果的新手尤其重要。

4.3 避免欺騙：常見的誤導性視覺化技術

- 目的：
  使新手分析師能夠識別並避免創建那些（有意或無意）扭曲數據、誤導觀眾的視覺化圖表 93。
- 策略：識別常見陷阱：
  1. 截斷 **Y** 軸（**Truncated Y-Axis**）：在條形圖或柱狀圖中，將 Y 軸的起始點設置為非零值，以此誇大數值之間的差異 [95]。規則：條形圖/柱狀圖的 Y 軸「必須」從零開始 [97]。折線圖有時可以有非零基線以更好地顯示變化率，但需謹慎並明確標註 [97]。
  2. 不當/不一致的標度（**Improper/Inconsistent Scaling**）：使用非線性標度（如對數標度）但未清晰標明 [94]；操縱圖表的長寬比以改變視覺效果；在象形圖（Pictograms）中使用大小不一致的圖像來代表數值 [94]；面積縮放錯誤（例如，用半徑而非面積來縮放圓的大小）[103]。
  3. 挑選數據（**Cherry-Picking Data**）：選擇性地展示對特定論點有利的數據範圍或時間段，而忽略或隱藏不利的數據 [96]。
  4. 使用錯誤的圖表類型（再訪）：選擇的圖表類型本身就模糊或扭曲了想要比較的關係（例如，用過於複雜的餅圖比較細微差異，不恰當地使用地圖）[94]。
  5. 誤導性的顏色使用（**Misleading Use of Color**）：使用與常規認知相反的顏色（如紅色代表增長，綠色代表虧損）；使用過多相似的顏色導致無法區分；不恰當地使用顏色漸變 [95]。

6. **3D 圖表（3D Charts）**：添加 3D 效果，尤其是對餅圖，會因透視而扭曲比例，使準確比較變得困難 [97]。應避免使用 [97]。
7. **信息過載/混亂（Overloading/Clutter）**：在單個圖表中塞入過多的信息、變數或視覺元素，導致圖表難以閱讀和理解 [93]。
8. **缺乏上下文（Lack of Context）**：展示圖表時沒有提供必要的標籤、標題、單位、數據來源或解釋說明 [36]。
9. **暗示因果關係（Implying Causation）**：圖表的設計（例如，用線條連接不相關的點）暗示了因果關係，而數據本身只支持相關性 [96]。

- **陷阱與誤區（對創建者而言）**：由於對最佳實踐缺乏認識而無意中產生誤導；過於注重美觀而犧牲了準確性。
- **要點闡述**：
  許多誤導性的視覺化來自於對圖表「標度」（Scale）和「基線」（Baseline）的操縱，特別是 Y 軸 94。截斷 Y 軸是一種簡單卻非常有效的扭曲數據、誇大差異的方法，新手必須學會識別並避免這種做法。「條形圖必須從零開始」97 是一個必須遵守的具體規則。誤導性的視覺化並不總是故意的；它們也可能源於缺乏知識而做出的不當選擇（例如，選擇了錯誤的圖表類型、濫用顏色、使用 3D 效果）94。因此，本指南的重點是教育新手了解並遵循最佳實踐，以避免這些常見的「錯誤」，而不僅僅是揭露惡意的欺騙。

5. 結論：培養穩健的分析習慣

5.1 系統化工作流程回顧

本指南闡述了一套系統化的資料分析與特徵工程流程，旨在幫助新手應對來自未知領域的數據挑戰。該流程強調結構化的方法，從清晰地定義問題和分析目標開始，接著進行系統性的探索性資料分析（EDA）以深入理解數據，然後基於初步發現生成可檢驗的假設。隨後，進入關鍵的數據處理與特徵工程階段，涵蓋數據清理、轉換、創建和選擇等技術，並特別警惕數據洩漏和過度擬合等陷阱。接著，透過相關性分析和視覺化探索變數間的關係，同時注意區分相關性與因果性。最後，強調了選擇合適的視覺化圖表以及有效、誠實地呈現分析結果的重要性。遵循這樣一個結構化的工作流程，能夠顯著降低在複雜數據面前迷失方向的風險。

5.2 資料分析的迭代本質

需要強調的是，資料分析很少是一個嚴格線性的過程 [3]。更確切地說，它是一個迭代的循環 [9]。在後續階段的發現往往會促使我們重新審視之前的步驟。例如，EDA 中發現的數據質量問題可能需要更徹底的數據清理；模型評估的結果可能揭示了特徵工程的不足，需要創建新的特徵或調整現有特徵；意想不到的分析結果可能引發新的假設，需要回到 EDA 階段進行更深入的探索。因此，分析師應擁抱這種迭代性，鼓勵快速的原型設計和反覆試驗，而不是試圖一次性完美地完成每個階段 [3]。敏捷的思維方式通常比僵化的瀑布式流程更適合資料分析的探索性本質。

5.3 擁抱持續學習與批判性思維

資料科學是一個快速發展的領域，掌握一套系統化的流程只是基礎。分析師需要保持持續學習的態度，不斷更新關於新技術、新工具和新方法的知識[29]。更重要的是，必須培養強大的批判性思維能力。這意味著要時刻質疑自己的假設和發現[32]，主動尋找可能推翻自己結論的證據，意識到並努力克服自身的認知偏見（如確認偏誤、選擇偏誤等）[1]，並清晰地認識到分析方法和數據本身的局限性。此外，積極尋求他人的反饋，並與同事或導師進行開放的討論與合作，對於發現盲點、提升分析質量至關重要[5]。

5.4 結語

掌握資料分析和特徵工程的技能需要時間和實踐。然而，本指南提供的系統化方法論為新手奠定了一個堅實的基礎。透過遵循結構化的流程，理解每個步驟的目的和策略，警惕常見的陷阱和誤區，並輔以有效的視覺化溝通技巧，分析師將能更有信心地應對來自陌生領域的複雜數據挑戰，從數據中提取有價值的洞見，並做出更可靠的、數據驅動的決策。在數據日益重要的時代，培養這些穩健的分析習慣將是通往成功的關鍵。

引用的著作

1. Analysts Beware: 6 Common Causes of Incorrect Data Analysis | Pecan AI, https://www.pecan.ai/blog/common-data-analysis-mistakes/
2. 2.5 Common Mistakes in Data Science, https://scientistcafe.com/ids/common-mistakes-in-data-science
3. What is CRISP DM? - Data Science PM, https://www.datascience-pm.com/crisp-dm-2/
4. CRISP-DM Explained: A Proven Data Mining Methodology | Udacity, https://www.udacity.com/blog/2025/03/crisp-dm-explained-a-proven-data-mining-methodology.html
5. Data Science Workflows: CRISP-DM - Dean Marchiori, https://www.deanmarchiori.com/posts/2024-09-26-crispdm/
6. Initial data analysis: Making the effort worthwhile, https://www.stratos-initiative.org/sites/default/files/2023-11/AppStat-23-Lusa.pdf
7. 10 Exploratory data analysis - R for Data Science (2e) - Hadley Wickham, https://r4ds.hadley.nz/EDA.html
8. Feature Engineering Explained | Built In, https://builtin.com/articles/feature-engineering
9. What is feature engineering? | Statsig, https://www.statsig.com/perspectives/feature-engineering-explained
10. What is Feature Engineering? Definition and FAQs | HEAVY.AI, https://www.heavy.ai/technical-glossary/feature-engineering
11. What is a feature engineering? | IBM, https://www.ibm.com/think/topics/feature-engineering
12. What is Feature Engineering for Machine Learning? - Caltech, https://pg-p.ctme.caltech.edu/blog/ai-ml/what-is-feature-engineering-for-machi

ne-learning
13. (PDF) Comparative Study of Feature Engineering Techniques for ..., https://www.researchgate.net/publication/389415779_Comparative_Study_of_Feature_Engineering_Techniques_for_Predictive_Data_Analytics
14. Feature Engineering for Better Precision | Keylabs, https://keylabs.ai/blog/feature-engineering-for-better-precision/
15. Feature Engineering for Machine Learning: Techniques to Boost Model Performance, https://www.udacity.com/blog/2024/12/feature-engineering-for-machine-learning-techniques-to-boost-model-performance.html
16. Data Preprocessing and Feature Engineering in Machine Learning - Magnimind Academy, https://magnimindacademy.com/blog/data-preprocessing-and-feature-engineering-in-machine-learning/
17. Data preprocessing vs. feature engineering - Iguazio, https://www.iguazio.com/questions/data-preprocessing-vs-feature-engineering-whats-the-difference/
18. Advanced-Data Preprocessing Algorithms and Feature Engineering Techniques, https://www.xcubelabs.com/blog/advanced-data-preprocessing-algorithms-and-feature-engineering-techniques/
19. Data Preprocessing in Machine Learning: Steps & Best Practices - lakeFS, https://lakefs.io/blog/data-preprocessing-in-machine-learning/
20. Data Preprocessing and Feature Engineering - | BlueCourses, https://www.bluecourses.com/asset-v1:bluecourses+BC3+October2019+type@asset+block/BartBaesensDataPreprocessingFeatureEng.pdf
21. Feature Engineering for Machine Learning | Towards Data Science, https://towardsdatascience.com/feature-engineering-for-machine-learning-eb2e0cff7a30/
22. What is Correlation Analysis? A Complete Guide - Applied AI Course, https://www.appliedaicourse.com/blog/what-is-correlation-analysis/
23. Common pitfalls in feature engineering | Statsig, https://www.statsig.com/perspectives/feature-engineering-pitfalls
24. How Overfitting Ruins Your Feature Selection | Hex, https://hex.tech/blog/overfitting-model-impact/
25. Role of Domain Knowledge in Data Science | GeeksforGeeks, https://www.geeksforgeeks.org/role-of-domain-knowledge-in-data-science/
26. Application Design: Data-driven vs Domain-driven - Passwork Pro, https://passwork.pro/blog/application-design/
27. Data-driven versus a domain-led approach to k-means clustering on an open heart failure dataset, https://d-nb.info/1269661175/34
28. Investigation of Feature Engineering Methods for Domain ... - MDPI, https://www.mdpi.com/1099-4300/25/9/1278
29. Data-driven or data-informed: Which approach serves you best? - Statsig, https://www.statsig.com/perspectives/data-driven-or-data-informed-which-approach-serves-you-best

30. What is Data Leakage in Machine Learning? | IBM, https://www.ibm.com/think/topics/data-leakage-machine-learning
31. Correlation vs. Causation: Avoiding Common Statistical Pitfalls in Business Analysis, https://editverse.com/correlation-causation-business-analysis/
32. Spurious Correlation: Definition, Examples & Detecting - Statistics By Jim, https://statisticsbyjim.com/basics/spurious-correlation/
33. A Data Analytics Mindset with CRISP-DM | IMA - Strategic Finance, https://www.sfmagazine.com/articles/2023/february/a-data-analytics-mindset-with-crisp-dm
34. Crisp DM methodology - Smart Vision Europe, https://www.sv-europe.com/crisp-dm-methodology/
35. Data Analysis Process Step 1: Identify business questions | Secoda, https://www.secoda.co/learn/data-analysis-process-step-1-identify-business-questions
36. Nine Common Data Analysis Mistakes and How to Avoid Them | DashThis, https://dashthis.com/blog/mistakes-in-data-analysis/
37. What are common mistakes to avoid in product analytics? - Statsig, https://www.statsig.com/perspectives/common-mistakes-product-analytics
38. 4 Marketing Data Analysis Mistakes to Avoid - QuickFrame, https://quickframe.com/blog/marketing-data-analysis-mistakes/
39. What (business) problems can a data analyst help solve? What problems with your tasks do you guys face on a regular basis? : r/dataanalysis - Reddit, https://www.reddit.com/r/dataanalysis/comments/1cq5iba/what_business_problems_can_a_data_analyst_help/
40. Identifying and Understanding Business Problems for Data Scientists - Pluralsight, https://www.pluralsight.com/courses/identifying-understanding-business-problems-data-scientists
41. Exploratory Data Analysis for beginners - Kaggle, https://www.kaggle.com/code/jihenbelhoudi/exploratory-data-analysis-for-beginners
42. Steps for Mastering Exploratory Data Analysis | EDA Steps ..., https://www.geeksforgeeks.org/steps-for-mastering-exploratory-data-analysis-eda-steps/
43. A Comprehensive Guide to Mastering Exploratory Data Analysis, https://www.dasca.org/world-of-data-science/article/a-comprehensive-guide-to-mastering-exploratory-data-analysis
44. 7 Exploratory Data Analysis - R for Data Science - Hadley Wickham, https://r4ds.had.co.nz/exploratory-data-analysis.html
45. A Data Scientist's Essential Guide to Exploratory Data Analysis | Towards Data Science, https://towardsdatascience.com/a-data-scientists-essential-guide-to-exploratory-data-analysis-25637eee0cf6/
46. Data Science. Exploratory Data Analysis - luminousmen, https://luminousmen.com/post/exploratory-data-analysis/
47. Data Quality: 10 mistakes not to make - DataScientest.com,

https://datascientest.com/en/data-quality-10-mistakes-not-to-make
48. How To Perform Exploratory Data Analysis -A Guide for Beginners - Analytics Vidhya, https://www.analyticsvidhya.com/blog/2021/08/how-to-perform-exploratory-data-analysis-a-guide-for-beginners/
49. A Five-Step Guide for Conducting Exploratory Data Analysis - Shopify Engineering, https://shopify.engineering/conducting-exploratory-data-analysis
50. Exploratory data analysis (EDA) machine learning approaches for ocean world analog mass spectrometry - Frontiers, https://www.frontiersin.org/journals/astronomy-and-space-sciences/articles/10.3389/fspas.2023.1134141/full
51. A Short Guide for Feature Engineering and Feature Selection.md - GitHub, https://github.com/ashishpatel26/Amazing-Feature-Engineering/blob/master/A%20Short%20Guide%20for%20Feature%20Engineering%20and%20Feature%20Selection.md
52. How to Choose the Right Data Visualization | Atlassian, https://www.atlassian.com/data/charts/how-to-choose-data-visualization
53. 4 Exploratory Data Analysis Checklist - Bookdown, https://bookdown.org/rdpeng/exdata/exploratory-data-analysis-checklist.html
54. Common Pitfalls in Machine Learning and How to Avoid Them | Noble Desktop, https://www.nobledesktop.com/learn/python/common-pitfalls-in-machine-learning-and-how-to-avoid-them
55. Common Types of Data Bias (With Examples) - Pragmatic Institute, https://www.pragmaticinstitute.com/resources/articles/data/5-common-bias-affecting-your-data-analysis/
56. Confirmation Bias in Data Analysis - Megaladata, https://megaladata.com/blog/confirmation-bias-data-analysis
57. Bias in Data Analysis | Codecademy, https://www.codecademy.com/article/bias-in-data-analysis
58. Hypothesising in data science - FutureLearn, https://www.futurelearn.com/info/courses/introduction-to-data-science-for-business/0/steps/264547
59. Demystifying Hypothesis Generation: A Guide to AI-Driven Insights - Akaike Technologies, https://www.akaike.ai/resources/demystifying-hypothesis-generation-a-guide-to-ai-driven-insights
60. Hypothesis Generation for Data Science Projects - Analytics Vidhya, https://www.analyticsvidhya.com/blog/2020/09/hypothesis-generation-data-science-projects/
61. Hypothesis Testing in Data Science - KDnuggets, https://www.kdnuggets.com/2023/02/hypothesis-testing-data-science.html
62. Stop Fooling Yourself! (Diagnosing and Treating Confirmation Bias) - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC11495861/
63. Confirmation bias - Wikipedia, https://en.wikipedia.org/wiki/Confirmation_bias
64. Cognitive Biases in Data Science - Integrate.io,

https://www.integrate.io/blog/cognitive-biases-in-data-science/
65. Null statistical hypothesis testing, confirmation bias, and statistical significance, https://sites.duke.edu/cemmt/2022/05/12/null-statistical-hypothesis-testing-confirmation-bias-and-statistical-significance/
66. Why can't you modify your hypothesis after analyzing data? - ResearchGate, https://www.researchgate.net/post/Why_cant_you_modify_your_hypothesis_after_analyzing_data
67. Correlation vs. Causation | Difference, Designs & Examples - Scribbr, https://www.scribbr.com/methodology/correlation-vs-causation/
68. Feature engineering techniques for better model accuracy - Statsig, https://www.statsig.com/perspectives/feature-engineering-techniques-model-accuracy
69. Feature engineering for machine learning: What is it? - Train in Data's Blog, https://www.blog.trainindata.com/feature-engineering-for-machine-learning/
70. STATSML 302: A Concept Course on Feature Engineering Techniques for Machine Learning | Adobe Data Distiller Guide, https://data-distiller.all-stuff-data.com/unit-8-data-distiller-statistics-and-machine-learning/statsml-302-a-concept-course-on-feature-engineering-techniques-for-machine-learning
71. 8 Feature Engineering Techniques for Machine Learning - ProjectPro, https://www.projectpro.io/article/8-feature-engineering-techniques-for-machine-learning/423
72. Feature Engineering: The Ultimate Guide. - DEV Community, https://dev.to/john_analytics/feature-engineering-the-ultimate-guide-f66
73. Data Leakage in Machine Learning Models - Shelf.io, https://shelf.io/blog/preventing-data-leakage-in-machine-learning-models/
74. Lessons From My ML Journey: Data Splitting and Data Leakage | Towards Data Science, https://towardsdatascience.com/two-rookie-mistakes-i-made-in-machine-learning-improper-data-splitting-and-data-leakage-3e33a99560ea/
75. Overfitting in Data Science: Identifying and Avoiding Common Pitfalls - Number Analytics, https://www.numberanalytics.com/blog/overfitting-data-science-guide
76. Can feature engineering avoid overfitting? - Data Science Stack Exchange, https://datascience.stackexchange.com/questions/120071/can-feature-engineering-avoid-overfitting
77. What Is Correlation Analysis: Comprehensive Guide - Dovetail, https://dovetail.com/research/what-is-correlation-analysis/
78. Correlation in Data Science: A Comprehensive Guide - GUVI, https://www.guvi.in/blog/correlation-in-data-science/
79. Correlation - Data Science Discovery - University of Illinois Urbana-Champaign, https://discovery.cs.illinois.edu/learn/Towards-Machine-Learning/Correlation/
80. Common pitfalls in statistical analysis: The use of correlation techniques - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC5079093/
81. How to choose the Right Chart for Data Visualization - Analytics Vidhya, https://www.analyticsvidhya.com/blog/2021/09/how-to-choose-the-right-chart-f

or-data-visualization/
82. Methods - Data Visualization - LibGuides at Morgan State University, https://library.morgan.edu/dataviz/methods
83. Chart Selection Guide - The Data Visualisation Catalogue Blog, https://datavizcatalogue.com/blog/chart-selection-guide/
84. Decoding Hidden Spurious Correlations in Modern Data Trends - Number Analytics, https://www.numberanalytics.com/blog/decoding-hidden-spurious-correlations-data-trends
85. 11 Odd Statistical Spurious Correlations: 4 Shocking Insights for Analysts, https://www.numberanalytics.com/blog/11-odd-statistical-spurious-correlations-4-shocking-insights
86. Big Data and the Pitfalls of Spurious Correlations - Zentricx, https://zentricx.com/big-data-and-the-pitfalls-of-spurious-correlations/
87. Genius Alert: 3 Common Mistakes When Interpreting Correlation - TTI Success Insights Blog, https://blog.ttisi.com/genius-alert-3-common-mistakes-when-interpreting-correlation
88. The 5 Pillars of Data Visualization | CFO.University, https://cfo.university/library/article/the-5-pillars-of-data-visualization-southekal
89. Telling Stories With Data: How to Choose the Right Data Visualization - CMS Wire, https://www.cmswire.com/digital-marketing/how-to-choose-the-right-data-visualization/
90. 12 Best Chart and Graph Types for Actionable Data Visualization, https://deliveringdataanalytics.com/12-best-chart-and-graph-types-for-actionable-data-visualization/
91. Principles of Effective Data Visualization | Foundations of Data Science Class Notes, https://library.fiveable.me/foundations-of-data-science/unit-5/principles-effective-data-visualization/study-guide/jSwDfLiYBXeNPRzQ
92. 10 Data Analytics Challenges & Solutions - Oracle, https://www.oracle.com/business-analytics/data-analytics-challenges/
93. 6 bad data visualization examples and mistakes—and how to avoid them - ThoughtSpot, https://www.thoughtspot.com/data-trends/data-visualization/bad-data-visualization-examples
94. Bad Data Visualization: 5 Examples of Misleading Data - HBS Online, https://online.hbs.edu/blog/post/bad-data-visualization
95. Bad Data Visualization: 9 Examples to Learn From - Luzmo, https://www.luzmo.com/blog/bad-data-visualization
96. Misleading Data Visualizations – Critical Data Literacy, https://pressbooks.library.torontomu.ca/criticaldataliteracy/chapter/misleading-data-visualizations/
97. 10 Good and Bad Examples of Data Visualization in 2024 - Polymer Search, https://www.polymersearch.com/blog/10-good-and-bad-examples-of-data-visu

alization
98. Designing Effective Data Visualizations, https://dataservices.library.jhu.edu/wp-content/uploads/sites/41/2024/03/DesigningEffectiveDataVisualizations.pdf
99. Visual encoding Principles – Computer Gaphics and Visualization, https://ebooks.inflibnet.ac.in/csp06/chapter/visual-encoding-principles/
100. 5 Principles of Visual Perception, https://ucdavisdatalab.github.io/workshop_data_viz_principles/principles-of-visual-perception.html
101. Visual Encoding Design - Washington, https://courses.cs.washington.edu/courses/cse442/17au/lectures/CSE442-VisualEncoding.pdf
102. 4. Choose Appropriate Visual Encodings - Designing Data Visualizations [Book] - O'Reilly, https://www.oreilly.com/library/view/designing-data-visualizations/9781449314774/ch04.html
103. Data visualization: basic principles - Peter Aldhous, https://peteraldhous.com/ucb/2016/dataviz/week2.html
104. Misleading Data Visualization - What to Avoid | Coupler.io Blog, https://blog.coupler.io/misleading-data-visualization-examples/