TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––



**BÁO CÁO THỰC TẬP NGHỀ NGHIỆP**

**NGÀNH HỆ THỐNG THÔNG TIN QUẢN LÝ**

**CHUYÊN NGÀNH QUẢN TRỊ HỆ THỐNG THÔNG TIN**

**PHÂN TÍCH VÀ TRỰC QUAN HOÁ DỮ LIỆU TRUY CẬP INTERNET CỦA TRẺ EM VÀ THANH NIÊN TRÊN TOÀN CẦU**

Sinh viên thực hiện : Nguyễn Thị Thanh Thư

Lớp : 45K21.1

Đơn vị thực tập : TMA Innovation Park

Cán bộ hướng dẫn : Đào Văn Tiến Dũng

Giảng viên hướng dẫn : ThS. Cao Thị Nhâm

**Đà Nẵng, 8/2022**

**NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP**

Họ và tên sinh viên: Nguyễn Thị Thanh Thư

Lớp: 45K21.1Khoa: Thống kê – Tin họcTrường: Đại học Kinh Tế - Đại học Đà Nẵng

Thực tập từ ngày: 06/06/ 2022 đến ngày: 12/08/ 2022

Tại: TMA Innovation Park

Địa chỉ: 12 Đại lộ Khoa học, Thung lũng Sáng tạo, Ghềnh Ráng, Tp. Qui Nhơn, Bình Định

Sau quá trình thực tập tại đơn vị của sinh viên, chúng tôi có một số nhận xét, đánh giá như sau:

**1. Về thái độ, ý thức, đạo đức, kỷ luật**

**2. Kiến thức chuyên môn**

**3. Khả năng hòa nhập và thích nghi với công việc**

**4. Trách nhiệm, sáng tạo trong công việc**

**5. Các nhận xét khác**

**Đánh giá chung:**

**Điểm:**

……….., ngày .......tháng ......năm 2022

**Xác nhận của đơn vị thực tập**

# LỜI CẢM ƠN

# LỜI CAM ĐOAN

Quy định:

Soạn thảo trên trang A4 (trang dọc, lề trái: 3.5cm; trên, phải, dưới: 2.5 cm), Font Times New Roman, canh đều 2 bên, size 13, cách dòng 1.5, cách đoạn trên 6pt, cách đoạn dưới 3pt, hàng đâu tiên lùi vào 1.27 cm; hình và bảng soạn theo caption, chèn trích dẫn chéo (Cross-reference) cho bảng và hình; các danh mục hình, bảng, mục lục làm tự động; đánh số trang như file mẫu (bìa không có số trang, danh mục + mục lục số trang theo i, ii, iii…, nội dung chính theo 1,2,3…)

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc109306117)

[LỜI CAM ĐOAN ii](#_Toc109306118)

[MỤC LỤC iv](#_Toc109306119)

[DANH MỤC HÌNH ẢNH vi](#_Toc109306120)

[DANH MỤC BẢNG BIỂU vii](#_Toc109306121)

[DANH MỤC CÁC TỪ VIẾT TẮT viii](#_Toc109306122)

[LỜI MỞ ĐẦU 1](#_Toc109306123)

[CHƯƠNG 1. TỔNG QUAN về python 2](#_Toc109306124)

[1.1. Python là gì? 2](#_Toc109306125)

[1.2. Python có thể làm gì? 2](#_Toc109306126)

[1.3. Tại sao lại là Python? 2](#_Toc109306127)

[1.4. Cú pháp Python so với các ngôn ngữ lập trình khác 3](#_Toc109306128)

[CHƯƠNG 2. TỔNG QUAN về tableau 4](#_Toc109306129)

[2.1. Kiến thức cơ bản về đọc dữ liệu 4](#_Toc109306130)

[2.2. Đọc các loại biểu đồ phổ biến 9](#_Toc109306131)

[CHƯƠNG 3. LÝ THUYẾT 18](#_Toc109306132)

[3.1. Biến (Variables) 18](#_Toc109306133)

[3.2. Data Types, Numbers & Casting 19](#_Toc109306134)

[3.3. Strings 19](#_Toc109306135)

[3.4. Booleans & Operators 21](#_Toc109306136)

[3.5. Lists 21](#_Toc109306137)

[3.6. Tuples 22](#_Toc109306138)

[3.7. Sets 23](#_Toc109306139)

[3.8. Dictionaries 23](#_Toc109306140)

[3.9. Conditionals, Loops & Iterators 23](#_Toc109306141)

[3.10. Arrays, Objects, Inheritance & Scope 23](#_Toc109306142)

[3.11. Functions & Lambdas 24](#_Toc109306143)

[3.12. Miscellaneous 24](#_Toc109306144)

[3.13. File Handling 24](#_Toc109306145)

[3.14. Matplotlib 24](#_Toc109306146)

[3.15. Machine Learning 25](#_Toc109306147)

[3.16. MySQL 26](#_Toc109306148)

[3.17. MongoDB 26](#_Toc109306149)

[CHƯƠNG 4. TRIỂN KHAI … 28](#_Toc109306150)

[4.1. Mục 3.1 28](#_Toc109306151)

[4.2. Mục 3.2 28](#_Toc109306152)

[CHƯƠNG 5. KẾT QUẢ 29](#_Toc109306153)

[5.1. Mục 4.1… 29](#_Toc109306154)

[5.2. Mục 4.2… 29](#_Toc109306155)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 30](#_Toc109306156)

[TÀI LIỆU THAM KHẢO 31](#_Toc109306157)

[PHỤ LỤC 32](#_Toc109306158)

*(Mục lục này chỉ là ví dụ)*

# DANH MỤC HÌNH ẢNH

[Hình 1.1 Ngôn ngữ lập trình Python 2](#_Toc98336120)

[Hình 4.1 Kiến trúc của mô hình 5](#_Toc98336121)

# DANH MỤC BẢNG BIỂU

[Bảng 4.1 Kiến trúc 5](#_Toc74235471)

# DANH MỤC CÁC TỪ VIẾT TẮT

**AI** : Artificial Intelligence

**…**

# LỜI MỞ ĐẦU

1. **Mục tiêu nghiên cứu của đề tài**

* Đề tài này nghiên cứu …

1. **Nhiệm vụ của đề tài**

* Nghiên cứu ...
* ...

1. **Phương pháp nghiên cứu**

* ...

1. **Đối tượng và phạm vi nghiên cứu**

* ...

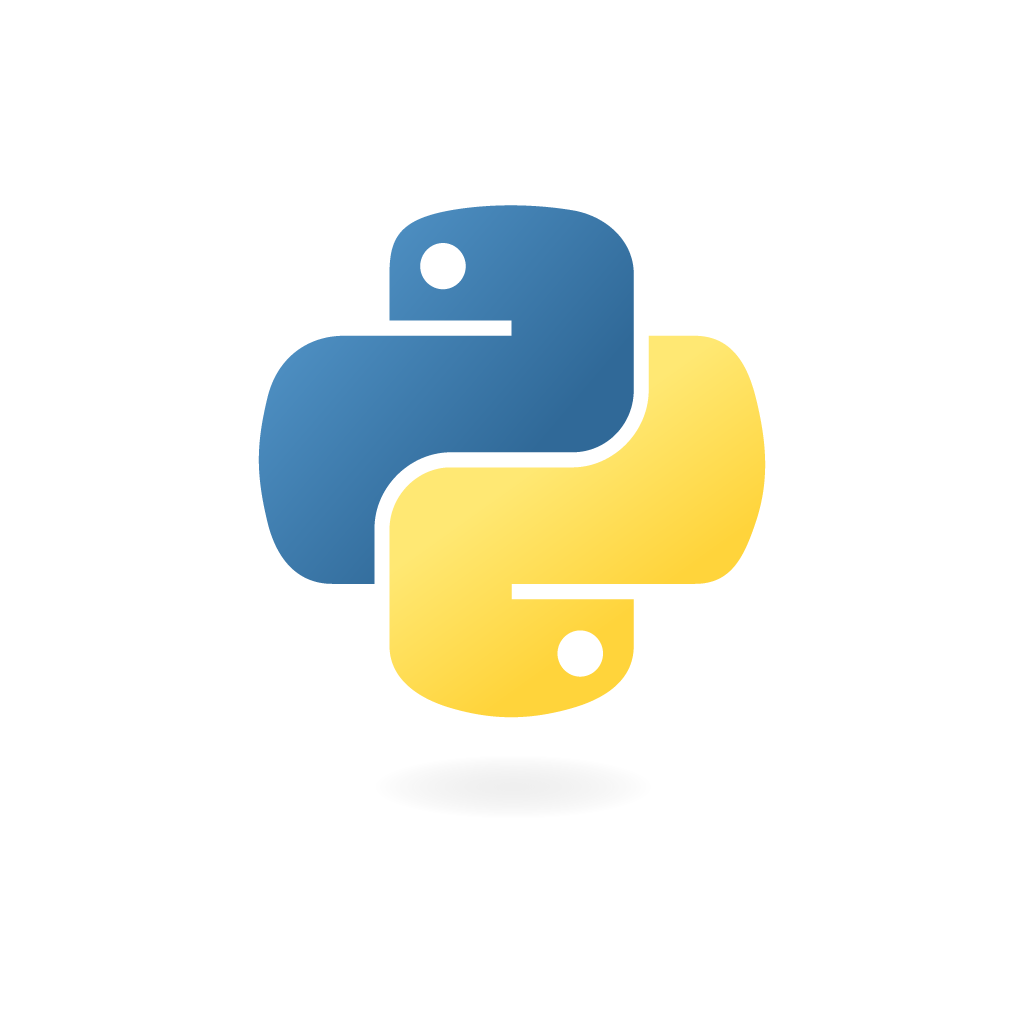
1. **Kết cấu của đề tài**

Đề tài được tổ chức gồm phần mở đầu, x chương nội dung và phần kết luận...

* Mở đầu
* **Chương 1**:
* **Chương 2**:
* **Chương 3**:
* **Chương 4**:
* Kết luận và hướng phát triển

# TỔNG QUAN về python

## Python là gì?

Python là một ngôn ngữ lập trình phổ biến. Nó được tạo ra bởi Guido van Rossum và được phát hành vào năm 1991.

Hình 1.1. Ngôn ngữ lập trình Python

Hình 1.2. Ngôn ngữ lập trình Python

## Python có thể làm gì?

* Python có thể được sử dụng trên máy chủ để tạo các ứng dụng web.
* Python có thể được sử dụng cùng với phần mềm để tạo quy trình công việc.
* Python có thể kết nối với các hệ thống cơ sở dữ liệu. Nó cũng có thể đọc và sửa đổi các tập tin.
* Python có thể được sử dụng để xử lý dữ liệu lớn và thực hiện các phép toán phức tạp.
* Python có thể được sử dụng để tạo mẫu nhanh hoặc phát triển phần mềm sẵn sàng sản xuất.

## Tại sao lại là Python?

* Python hoạt động trên các nền tảng khác nhau (Windows, Mac, Linux, Raspberry Pi, v.v.).
* Python có một cú pháp đơn giản tương tự như ngôn ngữ tiếng Anh.
* Python có cú pháp cho phép các nhà phát triển viết chương trình với ít dòng hơn các ngôn ngữ lập trình khác.
* Python chạy trên một hệ thống thông dịch, có nghĩa là mã có thể được thực thi ngay sau khi nó được viết. Điều này có nghĩa là việc tạo mẫu có thể rất nhanh chóng.
* Python có thể được xử lý theo thủ tục, theo cách hướng đối tượng hoặc trong một cách thực tế .

## Cú pháp Python so với các ngôn ngữ lập trình khác

Python được thiết kế để dễ đọc và có một số điểm tương đồng với ngôn ngữ tiếng Anh với ảnh hưởng từ toán học.

Python sử dụng các dòng mới để hoàn thành một lệnh, trái ngược với các ngôn ngữ lập trình khác thường sử dụng dấu chấm phẩy hoặc dấu ngoặc đơn.

Python dựa vào thụt lề, sử dụng khoảng trắng, để xác định phạm vi; chẳng hạn như phạm vi của vòng lặp , hàm và lớp. Các ngôn ngữ lập trình khác thường sử dụng dấu ngoặc nhọn cho mục đích này.

# TỔNG QUAN về tableau

## Kiến thức cơ bản về đọc dữ liệu

Để hiểu rõ hơn về các trực quan hóa mà tổ chức của bạn sử dụng, bạn nên có một số kiến thức cơ bản về các khái niệm dữ liệu.

### Các loại tập dữ liệu phổ biến

* + - 1. *Tập dữ liệu là gì?*

Một tập dữ liệu (đôi khi được gọi là nguồn dữ liệu hoặc cơ sở dữ liệu) trong ngữ cảnh của Tableau, chứa dữ liệu được sử dụng để xây dựng trực quan hóa. Mỗi biểu đồ thanh, biểu đồ phân tán hoặc biểu đồ đường bạn thấy trong Tableau đều có cơ sở dữ liệu hoặc bảng tính được kết nối cung cấp dữ liệu.

**Bảng tính:** Bảng tính, chẳng hạn như Microsoft Excel hoặc Google trang tính, sắp xếp dữ liệu theo cấu trúc phẳng, có nghĩa là các bản ghi được lưu trữ dưới dạng các hàng dữ liệu đơn lẻ.

**Cơ sở dữ liệu elational** R**:** Cơ sở dữ liệu quan hệ lưu trữ dữ liệu trong nhiều bảng, với mỗi hàng được gán một mã định danh duy nhất. Người dùng kéo dữ liệu từ các bảng khác nhau lại với nhau bằng ngôn ngữ truy vấn có cấu trúc (SQL). Khía cạnh "quan hệ" cho thấy mối liên hệ hợp lý giữa các bảng khác nhau.

**Dữ liệu đám mây:** Đôi khi, các tổ chức thích lưu trữ dữ liệu của họ trên đám mây để họ không phải hỗ trợ máy chủ tại chỗ. Điều này bao gồm dữ liệu được lưu trữ ở những nơi như Amazon Web Services hoặc Microsoft Azure.

**Các loại kết nối khác:** Tableau cũng kết nối với các tệp không gian để ánh xạ, chẳng hạn như .kml hoặc .shp và các tệp thống kê được tạo bằng R.

### Quan sát kỹ hơn các trường

1. *Một trường(field)là gì ?*

Trường, còn được gọi là cột, là một phần thông tin duy nhất từ bản ghi trong tập dữ liệu. Ví dụ: nếu bạn thu thập dữ liệu về số lần một người đi làm đã chọn các phương thức vận chuyển khác nhau, tập dữ liệu của bạn có thể bao gồm:

| **Tên người đi làm** | **Chế độ** | **Số ngày mỗi tuần** |
| --- | --- | --- |
| Neil | Đi | 2 |
| Neil | Xe buýt | 3 |
| Lin | Xe đạp | 1 |
| Lin | Đi chung xe hoặc Vanpool | 3 |
| Lin | Xe buýt | 1 |
| Mae | Xe hơi (Xe một người) | 2 |
| Mae | Xe đạp | 3 |

Trong Tableau, các trường định lượng được gọi là *Thước đo* *(Measures)*và các trường định tính được gọi là *Kích thước(Dimensions).*

### Bản ghi cấp hàng là gì?

"Một hàng dữ liệu chứa những gì?" là một câu hỏi đơn giản nhưng quan trọng và nó có thể có một câu trả lời phức tạp.  Bạn có nhớ dữ liệu người đi làm mà chúng tôi đã xem xét trong bài học trước không? Mỗi hàng ghi lại số ngày mỗi tuần mà một người đi làm đã sử dụng một phương thức vận chuyển cụ thể để đi làm, nhưng nó không bao gồm nhiều chi tiết về các chuyến đi. Ví dụ: hầu hết mọi người thực hiện hai chuyến đi làm mỗi ngày nếu họ đi làm và họ có thể chọn một phương tiện giao thông khác nhau cho mỗi phương tiện. Người đi làm có thể sử dụng nhiều chế độ trong một chuyến đi để đi làm hoặc về nhà. Bộ dữ liệu đầu tiên của chúng tôi không nắm bắt được những chi tiết đó.

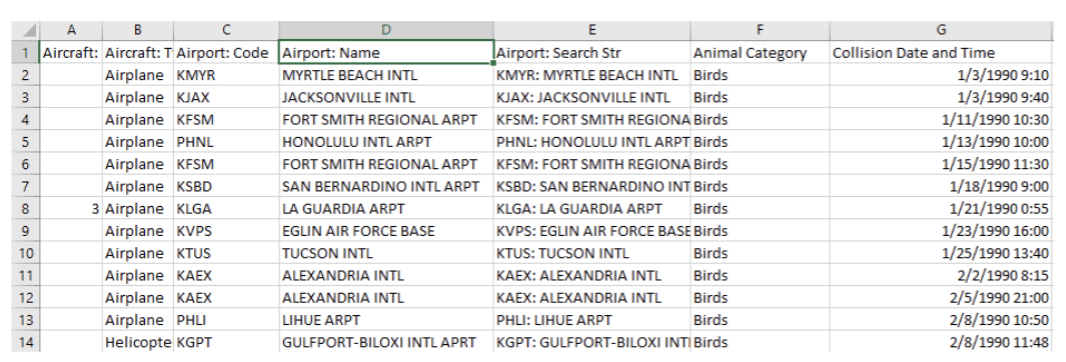
* + - 1. *Nếu bạn muốn biết thì sao*
* Có bao nhiêu người đi làm đã sử dụng nhiều phương thức vận chuyển trong một ngày?
* Làm thế nào để đi làm so sánh với đi làm từ nơi làm việc?
* Các chuyến đi làm khác nhau mất bao lâu?
* Độ dài của mỗi chuyến đi là bao nhiêu?

Nếu bạn muốn trả lời những câu hỏi đó, bạn sẽ cần thu thập dữ liệu chi tiết hơn, như sau: Trong tập dữ liệu thứ hai, một hàng dữ liệu chi tiết hơn hoặc *chi tiết* hơn trong tập dữ liệu đầu tiên.

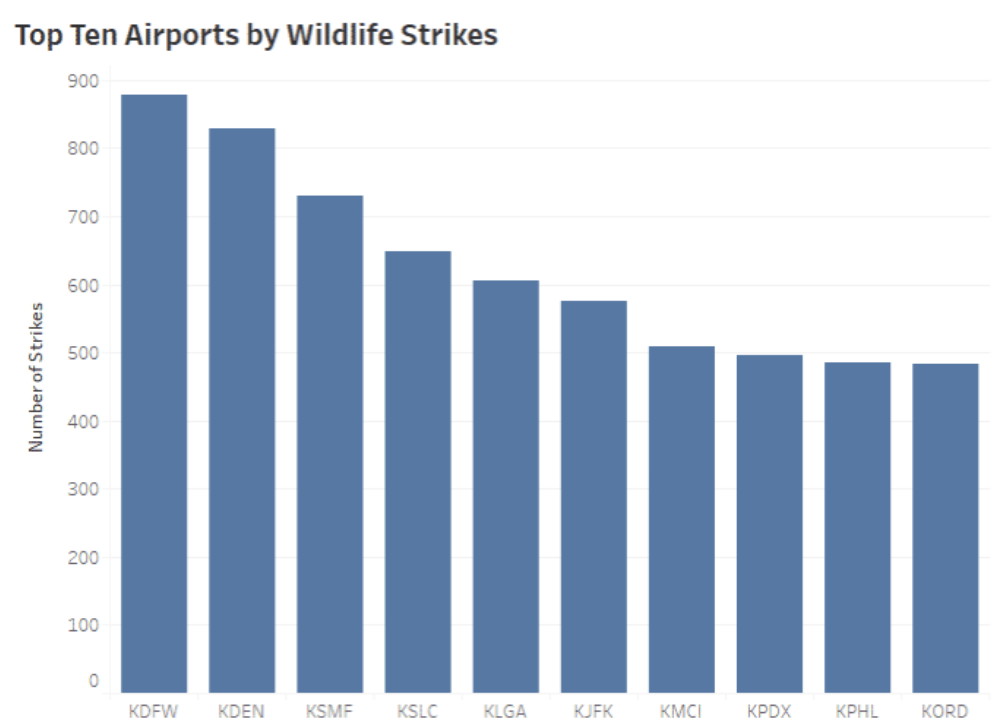
* + - 1. *Tại sao việc biết mức độ chi tiết của dữ liệu lại quan trọng?*

Độ chi tiết của dữ liệu đề cập đến mức độ chi tiết cho một phần dữ liệu, bất cứ nơi nào bạn đang tìm kiếm. Khi dữ liệu trở nên ít chi tiết hơn, chúng tôi có thể mô tả dữ liệu đó dưới *dạng tổng hợp* hoặc dưới dạng *dữ liệu tổng hợp*.  Tổng hợp đề cập đến cách dữ liệu được kết hợp. Mức độ chi tiết hoặc tổng hợp trong một hàng hoặc biểu đồ ảnh hưởng đến các câu hỏi chúng ta có thể hỏi về dữ liệu và những khám phá mà chúng ta có thể thực hiện.

***Trong bảng tính***: Dữ liệu cấp hàng trong tập dữ liệu này ghi lại các va chạm từ các cuộc đình công của động vật hoang dã và được trình bày chi tiết cho đến ngày và giờ xảy ra sự cố. Chúng tôi sẽ mô tả dữ liệu ở đây chi tiết hơn.



***Trong biểu đồ:*** Trong dạng xem này, các cuộc đình công vì động, thực vật hoang dã được tổng hợp thành các thanh, đại diện cho tất cả các hàng dữ liệu được liên kết với các sân bay cụ thể (được hiển thị bằng mã Sân bay). Chúng tôi sẽ mô tả dữ liệu này là ít chi tiết hơn.



***Nếu nó không có trong dữ liệu thì sao?***  Khả năng khám phá dữ liệu bạn có và đặt câu hỏi về dữ liệu đó có thể bị giới hạn bởi mức độ chi tiết có trong chế độ xem hoặc chính nguồn dữ liệu.

Nếu câu hỏi của bạn đạt đến giới hạn dữ liệu của bạn:

* Nêu vấn đề về chất lượng dữ liệu với Người quản lý dữ liệu của bạn.
* Yêu cầu người tiên phong và nhà tài trợ xem xét mở rộng quyền truy cập vào các nguồn dữ liệu khác của tổ chức bạn.
* Mô tả các câu hỏi về dữ liệu mà bạn hy vọng sẽ khám phá và trả lời.
* Cung cấp phản hồi chi tiết để những người khác có thể phát triển kế hoạch thu thập nhiều dữ liệu hơn và tốt hơn.
  + - 1. *Làm thế nào tôi có thể xác định mức chi tiết chứa trong một hàng dữ liệu khi tôi chỉ có quyền truy nhập vào trực quan hóa?*

Bạn có thể chọn một dấu trong trực quan hóa, rồi bấm chuột phải để xem dữ liệu cơ bản nếu tác giả của dạng xem đã bật nó. Bạn có thể xem các bản ghi cấp hàng của dữ liệu đằng sau trực quan hóa, đó là cách bạn có thể xác định độ sâu chi tiết mà tập dữ liệu chứa trong đó.

Xem bên dưới để biết hướng dẫn đầy đủ về cách xem dữ liệu cơ bản và các mẹo khác về cách xem biểu đồ.

### Độ chi tiết và tổng hợp trong Tableau

Khi bạn di chuyển kích thước và số đo vào và ra khỏi dạng xem, mức độ chi tiết của dạng xem sẽ thay đổi.

* Theo mặc định, các thước đo được đặt trong một dạng xem được tổng hợp bằng SUM, có nghĩa là dữ liệu cho trường đó trong tất cả các hàng được kết hợp.
* Các thước đo cũng có thể được tổng hợp dưới dạng trung bình, trung vị, đếm hoặc đếm riêng biệt.
* Kích thước chia nhỏ tổng số tổng hợp thành tổng số nhỏ hơn theo danh mục.

Video "Tính tổng hợp, độ chi tiết và tỷ lệ" từ Tableau đi vào chi tiết hơn về tổng hợp và độ chi tiết của dữ liệu. Nó sử dụng ví dụ về việc xem xét tỷ lệ lợi nhuận và có thể giúp bạn hiểu hành vi của biểu đồ Tableau.

## Đọc các loại biểu đồ phổ biến

### Tại sao nên sử dụng trực quan hóa dữ liệu?

*Trực quan hóa giúp chúng tôi hiểu dữ liệu phức tạp*

Lý do tốt nhất để sử dụng trực quan hóa để hiểu dữ liệu của bạn là hầu hết các tập dữ liệu đều quá lớn để sử dụng ở định dạng thô của chúng. Con người bị hạn chế về những thông tin mà chúng ta có thể xử lý và so sánh trong đầu, đặc biệt nếu thông tin đó nằm trong một tập dữ liệu hàng triệu hàng , nhưng chúng ta giỏi xử lý nhanh chóng thông tin trực quan.

Ngay cả khi bạn chưa quen với việc đọc dữ liệu trong biểu đồ, bạn đã có các chức năng tích hợp để làm nổi bật màu tối, hình dạng lớn và nhỏ, nhóm và hướng của các đối tượng. Chúng được gọi là các thuộc tính chú ý trước.

Phân tích trực quan tận dụng các thuộc tính chú ý trước của chúng tôi - các dấu hiệu trực quan mà con người xử lý tự động với bộ nhớ cảm giác. Chúng ta có thể nhận thấy và giải thích các loại thuộc tính này một cách nhanh chóng và không cần nỗ lực đặc biệt.

Một số thuộc tính chú ý trước tốt hơn trong việc hiển thị dữ liệu định lượng (đo lường) và một số thuộc tính tốt hơn trong việc hiển thị dữ liệu định tính (phân loại). Ví dụ: sử dụng độ dài của các thanh để thể hiện số lượng bán hàng là một lựa chọn hiệu quả để chỉ ra sự khác biệt về doanh số bán hàng giữa các danh mục.

### Các loại biểu đồ và cách sử dụng chúng

***Line(Biểu đồ dòng):*** Xem xu hướng trong dữ liệu theo thời gian.

***Bar(Biểu đồ cột):*** So sánh dữ liệu trên các danh mục.

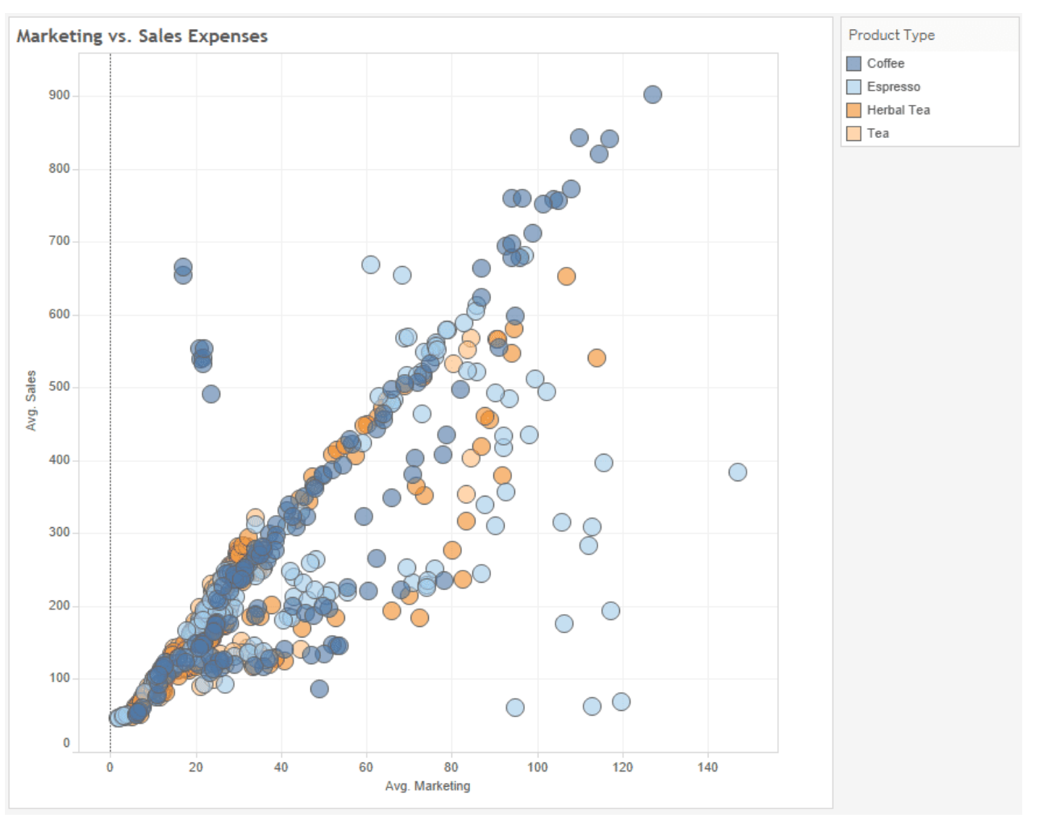
***Heat Map(Bản đồ nhiệt):*** Hiển thị mối quan hệ giữa hai yếu tố.

***Highlight Table (Bảng đánh dấu):*** Hiển thị thông tin chi tiết trên bản đồ nhiệt.

***Treemap(Bản đồ cây):*** Hiển thị dữ liệu phân cấp theo tỷ lệ của toàn bộ.

***Gantt:*** Hiển thị khoảng thời gian theo thời gian.

***Bullet(Biểu đồ đánh dấu):*** Đánh giá hiệu suất của số liệu so với mục tiêu.

***Scatterplot(Biểu đồ phân tán):*** Điều tra mối quan hệ giữa các giá trị định lượng.

***Histogram:*** Hiểu sự phân phối dữ liệu của bạn.

***Symbol maps (Bản đồ ký hiệu):*** Sử dụng cho tổng số thay vì tỷ lệ. Hãy cẩn thận, vì những khác biệt nhỏ sẽ khó nhận thấy.

***Area maps (Bản đồ khu vực):*** Sử dụng cho tỷ lệ thay vì tổng số. Sử dụng địa lý cơ sở hợp lý.

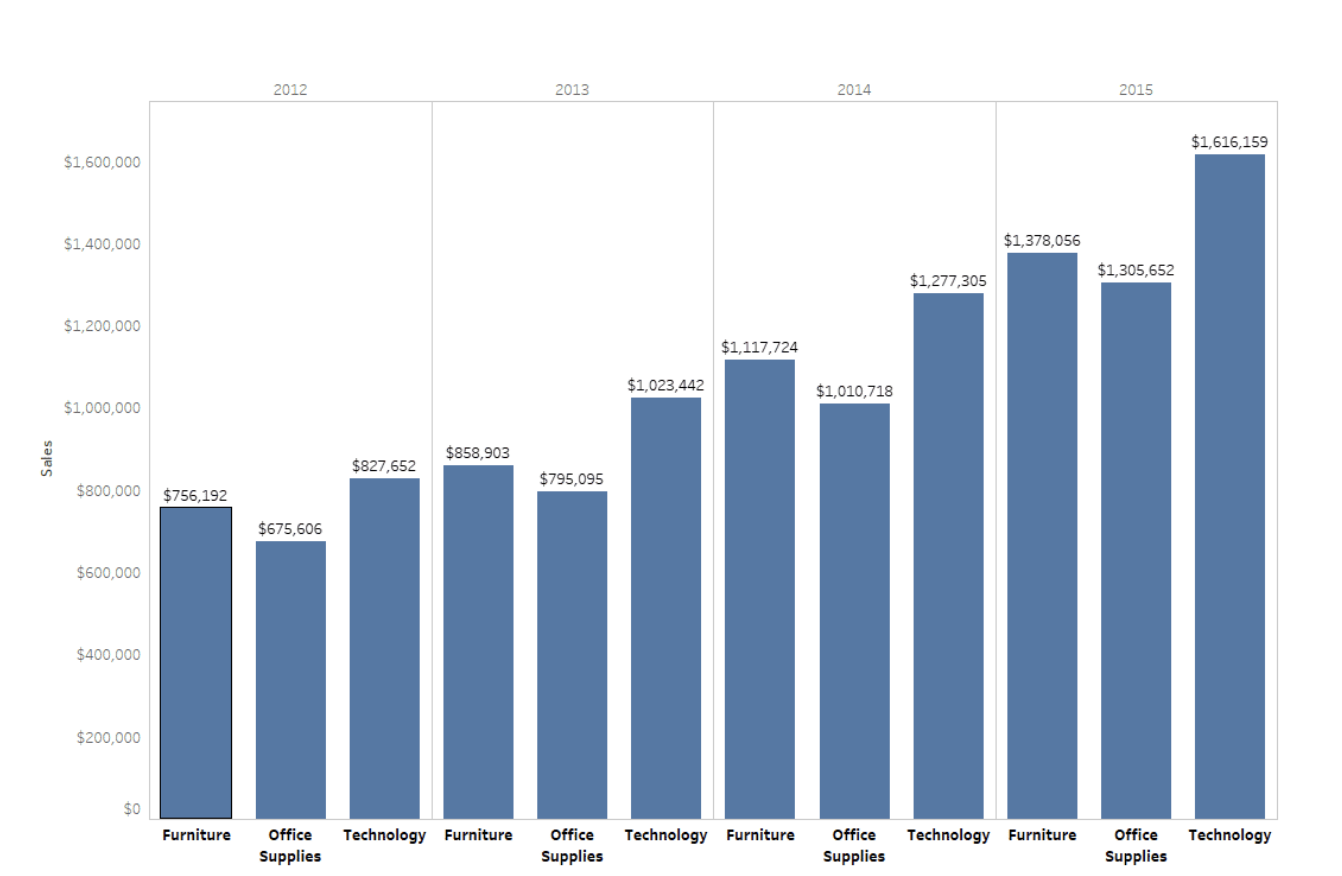
***Box-and-Whisker:*** Hiển thị phân phối của một tập hợp dữ liệu.

### Biểu đồ thanh: So sánh các loại dữ liệu

Biểu đồ thanh được sử dụng để so sánh dữ liệu trên các danh mục. Cũng giống như tab chéo, biểu đồ thanh là một loại biểu đồ rất dễ nhận biết và nếu được thực hiện tốt, rất dễ đọc và giải thích.

* Độ dài của các thanh cho bạn biết danh mục nào có giá trị cao nhất.
* Trục của biểu đồ thanh cho biết giá trị đó đại diện cho điều gì.

Dưới đây là ví dụ về doanh số bán hàng trên các danh mục sản phẩm.

Biểu đồ đường: Xem da Đôi khi, nếu biểu đồ thanh rõ ràng và được gắn nhãn, tác giả sẽ xóa tiêu đề trục để giảm sự lộn xộn trực quan. Trong Tableau, tác giả có thể làm điều này nếu mục đích là sử dụng biểu đồ làm bộ lọc cho dữ liệu khác hoặc cung cấp trạng thái cấp cao. Khi bạn thấy biểu đồ thanh không có trục, hãy hỏi:

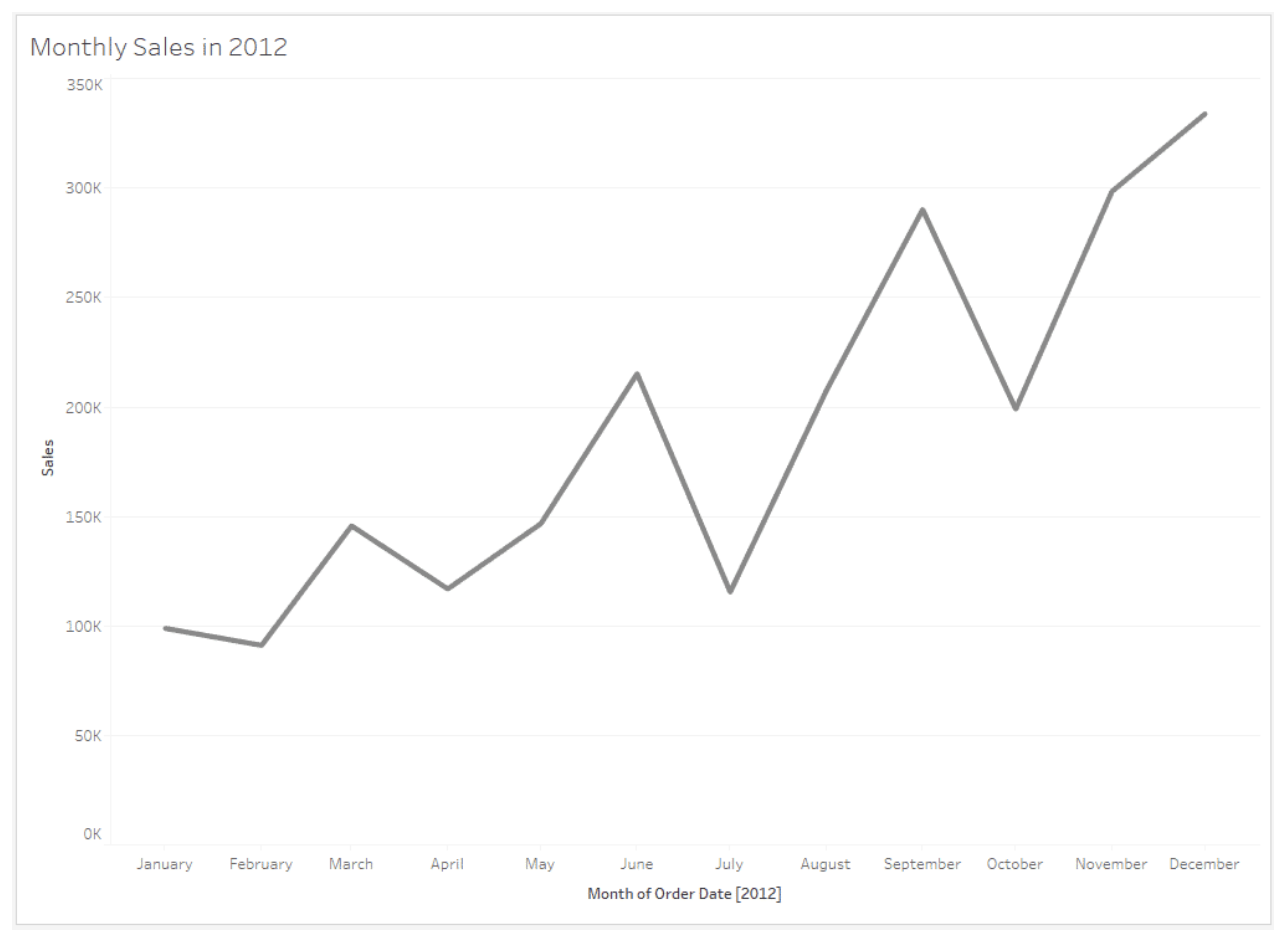
*Biểu đồ thanh hiển thị doanh số bán hàng theo danh mục sản phẩm trong giai đoạn 2012-2015.*

* Biểu đồ có được gắn nhãn rõ ràng không?
* Tôi có thể xác định phạm vi giá trị không?

### Biểu đồ đường: Xem dữ liệu theo thời gian

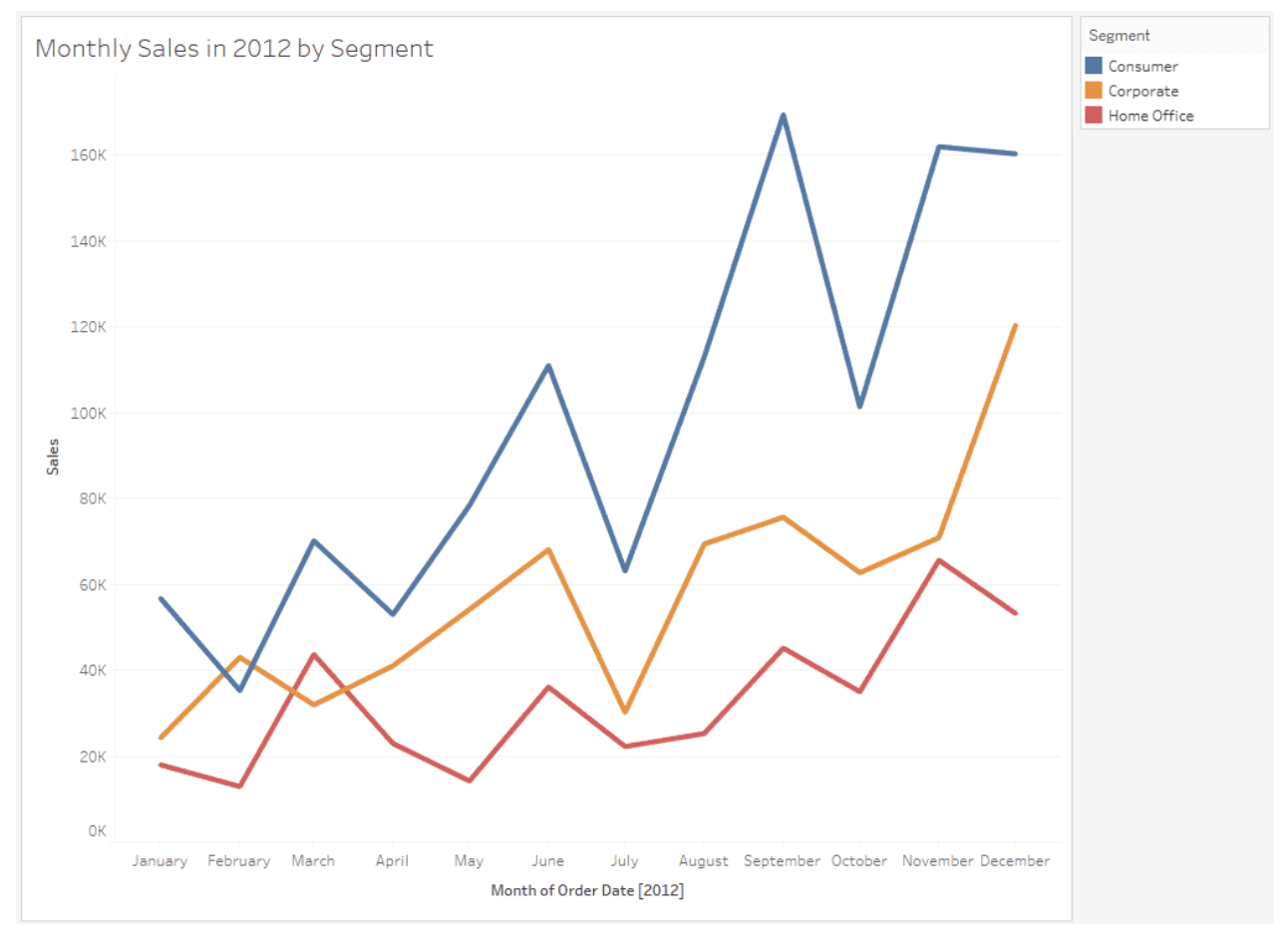
*Tìm kiếm xu hướng theo thời gian*

Biểu đồ đường đặc biệt hữu ích để xem các thay đổi trong dữ liệu theo thời gian. Doanh số bán hàng đang tăng hay giảm, và bao nhiêu? Có xu hướng theo mùa không?

Biểu đồ đường có thể bao gồm một tập hợp các giá trị duy nhất, như biểu đồ này bên dưới hiển thị doanh số hàng tháng cho một công ty cung cấp văn phòng trong suốt một năm:

Bạn có thể học được gì từ biểu đồ này? Bạn có thể thấy rằng doanh số bán hàng thường tăng và cao hơn vào cuối năm so với đầu năm. Bạn có thể thấy doanh số bán hàng thay đổi như thế nào trong suốt cả năm: doanh số bán hàng giảm trong những tháng mùa hè, đặc biệt là vào tháng BẢY, nhưng sau đó tăng trở lại vào tháng Chín, và sau đó tăng trở lại vào cuối năm.

Biểu đồ này cho thấy doanh số bán hàng cho toàn bộ công ty, nơi bán nhiều sản phẩm cho các nhóm khách hàng khác nhau. Thông tin chi tiết của bạn có áp dụng như nhau cho tất cả các nhóm khách hàng không? Làm thế nào bạn có thể nói?

Hãy nhớ rằng biểu đồ đường không giới hạn trong một dòng duy nhất. Bạn có thể thêm các danh mục (hoặc *thứ nguyên*) bổ sung vào chế độ xem để có thêm thông tin chi tiết về dữ liệu của mình. Điều gì xảy ra khi bạn thêm một thứ nguyên như **Phân khúc khách hàng** vào chế độ xem?

Chế độ xem này bao gồm dữ liệu giống như chế độ xem trước đó, nhưng bây giờ nó được chia nhỏ theo Phân đoạn. Bạn có thể thấy rằng phân khúc Người tiêu dùng màu xanh lam đã có sự gia tăng lớn về doanh số bán hàng trong tháng XNUMX, có lẽ là do trường học được nối lại trong thời gian đó. Doanh số bán hàng của công ty bằng màu cam đã tăng mạnh vào cuối năm, có lẽ do yêu cầu ngân sách hoặc mua sắm. Bằng cách thêm thứ nguyên, bạn có thể xem thêm chi tiết về dữ liệu của mình mà trước đây không rõ ràng.

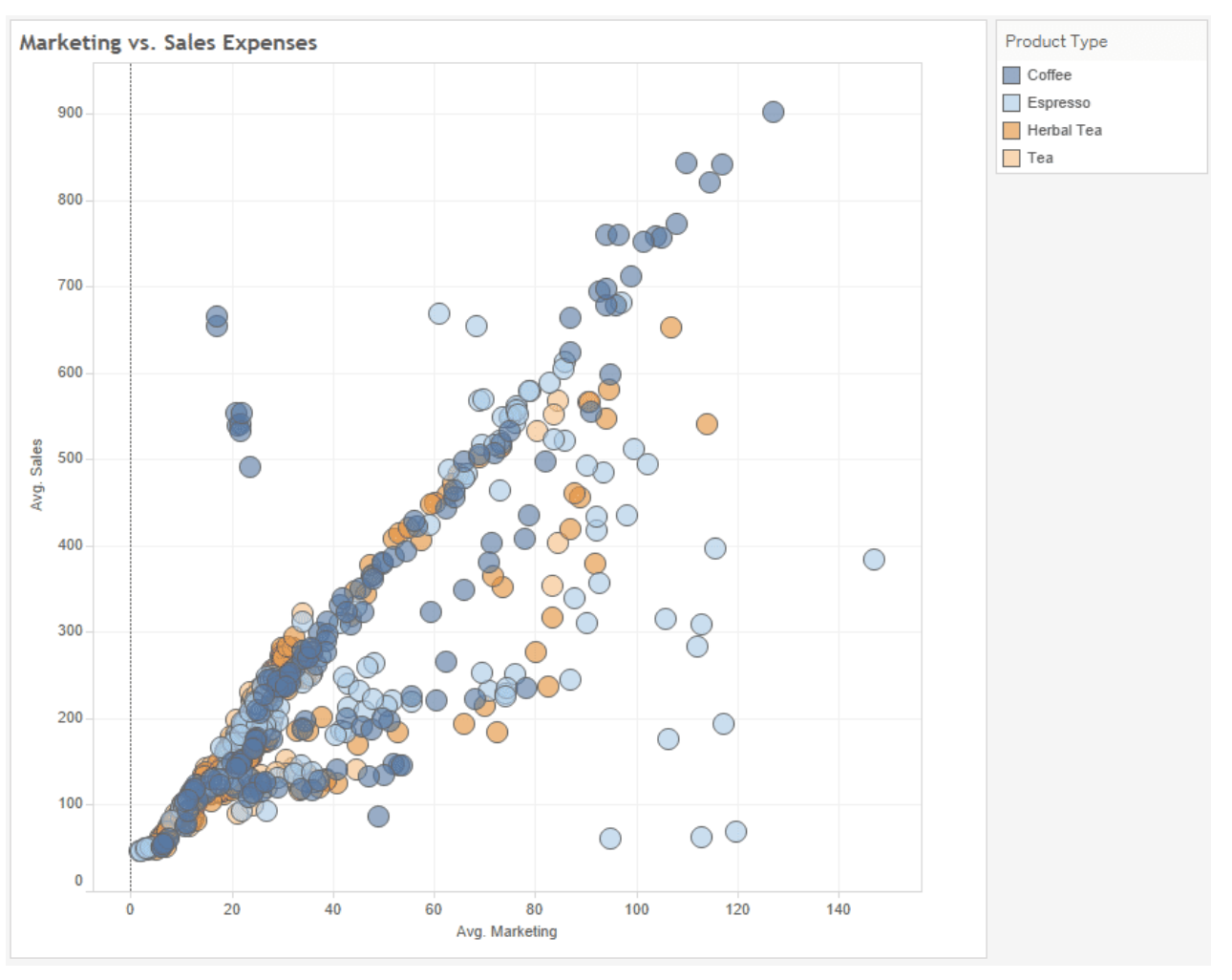
Khi bạn đang xem biểu đồ đường, hãy xem những điều sau:

* Khi nào xu hướng tăng và giảm?
* Có bất kỳ dữ liệu nào cho phép bạn xác định hoặc dự đoán xu hướng trong tương lai không?
* Hãy chú ý đến trục - Phạm vi giá trị điển hình là gì? Các khoảng thời gian có đồng đều không?
* Mốc thời gian: hàng năm là gì? nguyệt san? nhật báo? Biểu đồ có hiển thị lượng chi tiết phù hợp cho phân tích của bạn không?

### Biểu đồ phân tán: Xem mối quan hệ dữ liệu và ngoại lệ

Nếu bạn đã nhìn thấy một biểu đồ với nhiều dấu chấm trải đều trên đó, rất có thể bạn đang xem xét một *biểu đồ phân tán*. Biểu đồ phân tán rất hữu ích khi bạn muốn kiểm tra mối quan hệ giữa hai thước đo hoặc giá trị định lượng. Có một mối tương quan giữa chúng, và nếu có, ở mức độ nào?

Ví dụ, chiều cao của một người có tương quan với cân nặng của họ không? Kích thước của một ngôi nhà có quyết định giá của nó không? Nhiệt độ bên ngoài có ảnh hưởng đến doanh số bán kem không? Tất cả những câu hỏi này có thể được phân tích bằng cách sử dụng các biểu đồ phân tán.



1. *Ngoại lệ*

Ngay cả khi mối tương quan được hiển thị trong biểu đồ phân tán, đôi khi có thể có các giá trị phân kỳ từ xu hướng, được gọi là ngoại lệ.

Outliers có thể thú vị và nên được điều tra để bạn biết cách giải thích chúng. Họ có phải là những trường hợp ngoại lệ hiếm hoi có thể bị bỏ qua? Hoặc một dấu hiệu của một cái gì đó để kiểm tra chặt chẽ hơn? Hoặc có thể chúng chỉ đơn giản là lỗi trong việc ghi âm hoặc phiên âm? Để chắc chắn, hãy kiểm tra dữ liệu cơ bản để đưa ra quyết định của bạn.



Trong ví dụ trên, bạn có thể thấy một mối tương quan thuận chung. Khi chi tiêu tiếp thị tăng lên, nhìn chung có sự gia tăng doanh số bán hàng liên quan đến nó.

Tuy nhiên, hãy lưu ý đến một hiện tại ngoại lệ, nơi chi tiêu tiếp thị đáng kể có tác động thấp hơn nhiều đến doanh số bán hàng so với dự kiến. Tại sao? Trong ví dụ này, vị trí được thể hiện bằng nhãn hiệu là ở một thị trấn đại học. Có thể là tiếp thị không hiệu quả cho nhân khẩu học sinh viên? Có lẽ.

1. *Các câu hỏi dẫn đến tương quan xem*

Khi quyết định xem bạn có cần một biểu đồ phân tán hay không, hãy xem xét các câu hỏi sau:

* Người dùng sẽ quan tâm đến việc biến tăng hay giảm theo những thay đổi trong một biến khác?
* Người dùng có quan tâm đến việc xác định xem một thay đổi đối với một biến có ảnh hưởng đến một biến khác hay không và hiệu ứng đó là gì?
* Người dùng có cần biết một biến theo dõi chặt chẽ các thay đổi đối với biến khác như thế nào không?

# LÝ THUYẾT

## Biến (Variables)

Biến là nơi chứa để lưu trữ giá trị dữ liệu. Python không có lệnh để khao báo cho một biến, mà được tạo ra ngay khai báo giá trị cho nó

x = 5  
y = "Helimer"

## Data Types, Numbers & Casting

Có ba kiểu số trong Python : int, float, complex

x = 1    # int  
y = 2.8  # float  
z = 1j   # complex

## Strings

Các chuỗi trong python được bao quanh bởi dấu nháy đơn hoặc dấu nháy kép. 'hello' cũng tương tự "hello".

### Concatenate Strings

Để nối hoặc kết hợp hai chuỗi, bạn có thể sử dụng toán tử +. Hợp nhất biến a với biến b thành biến c:

a = "Hello"  
b = "World"  
c = a + b  
print(c)

## Booleans & Operators

### Booleans

Hàm bool () cho phép bạn đánh giá bất kỳ giá trị nào và trả lại cho bạn giá trị True hoặc False. Đánh giá một chuỗi và một số:

x = "Hello"  
y = 15  
print(bool("Hello"))  
print(bool(15))  
print(bool(x))  
print(bool(y))

### Operators

Các toán tử đã quen với việc thực hiện các phép toán trên các biến và giá trị. Trong ví dụ dưới đây, chúng tôi sử dụng toán tử + để cộng hai giá trị với nhau:

print(10 + 5)

## Lists

List - Danh sách được sử dụng để lưu trữ nhiều mục trong một biến duy nhất. Danh sách là một trong 4 kiểu dữ liệu tích hợp sẵn trong Python được sử dụng để lưu trữ các bộ sưu tập dữ liệu, 3 kiểu còn lại là Tuple, Set và Dictionary, tất cả đều có chất lượng và cách sử dụng khác nhau. Danh sách được tạo bằng dấu ngoặc vuông:

thislist = ["apple", "banana", "cherry"]  
print(thislist)

## Tuples

Tuples được sử dụng để lưu trữ nhiều mục trong một biến duy nhất. Tuple là một trong 4 kiểu dữ liệu tích hợp sẵn trong Python được sử dụng để lưu trữ các bộ sưu tập dữ liệu, 3 kiểu còn lại là List, Set và Dictionary, tất cả đều có chất lượng và cách sử dụng khác nhau. Bộ tuple là một bộ sưu tập được sắp xếp theo thứ tự và không thể thay đổi. Tuples được viết bằng dấu ngoặc tròn.

thistuple = ("apple", "banana", "cherry")  
print(thistuple)

## Sets

Tập hợp được sử dụng để lưu trữ nhiều mục trong một biến duy nhất. Set là một trong 4 kiểu dữ liệu tích hợp sẵn trong Python dùng để lưu trữ các tập hợp dữ liệu, 3 kiểu còn lại là List, Tuple và Dictionary, tất cả đều có chất lượng và cách sử dụng khác nhau. Tập hợp là tập hợp không có thứ tự, không thể thay đổi \* và không được lập chỉ mục. Tập hợp được viết bằng dấu ngoặc nhọn.

thisset = {"apple", "banana", "cherry"}  
print(thisset)

## Dictionaries

Từ điển được sử dụng để lưu trữ các giá trị dữ liệu trong các cặp key: value. Từ điển là một tập hợp được sắp xếp theo thứ tự \*, có thể thay đổi và không cho phép trùng lặp. Từ điển được viết bằng dấu ngoặc nhọn và có các khóa và giá trị:

thisdict = {  
  "brand": "Ford",  
  "model": "Mustang",  
  "year": 1964  
}  
print(thisdict)

## Conditionals & Loops

### If…Else

Các điều kiện này có thể được sử dụng theo một số cách, phổ biến nhất là trong "câu lệnh if" và vòng lặp. Một "câu lệnh if" được viết bằng cách sử dụng từ khóa if.

a = 33  
b = 200  
if b > a:  
  print("b is greater than a")

### While Loops

Với vòng lặp while, chúng ta có thể thực hiện một tập hợp các câu lệnh miễn là một điều kiện là đúng. In tôi miễn là tôi nhỏ hơn 6:

i = 1  
while i < 6:  
  print(i)  
  i += 1

Vòng lặp while yêu cầu các biến có liên quan phải sẵn sàng, trong ví dụ này, chúng ta cần xác định một biến lập chỉ mục, i, mà chúng ta đặt thành 1.

### For Loops

Vòng lặp for được sử dụng để lặp qua một chuỗi (đó là một danh sách, một bộ giá trị, một từ điển, một tập hợp hoặc một chuỗi).

fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
  print(x)

## Arrays, Objects, Inheritance & Scope

### Arrays

Mảng được sử dụng để lưu trữ nhiều giá trị trong một biến duy nhất:

cars = ["Ford", "Volvo", "BMW"]

### Classes

Python là một ngôn ngữ lập trình hướng đối tượng. Hầu hết mọi thứ trong Python đều là một đối tượng, với các thuộc tính và phương thức của nó. Một Class giống như một phương thức khởi tạo đối tượng, hoặc một "bản thiết kế" để tạo các đối tượng.

class MyClass:  
  x = 5

### Inheritance

Kế thừa cho phép chúng ta định nghĩa một lớp kế thừa tất cả các phương thức và thuộc tính từ một lớp khác. Lớp cha là lớp được kế thừa từ đó, còn được gọi là lớp cơ sở. Lớp con là lớp kế thừa từ lớp khác, còn được gọi là lớp dẫn xuất.

class Person:  
 def \_\_init\_\_(self, fname, lname):  
    self.firstname = fname  
    self.lastname = lname  
  def printname(self):

print(self.firstname, self.lastname)

### Scope

Một biến chỉ có sẵn từ bên trong vùng mà nó được tạo. Đây được gọi là phạm vi (Scope).

## SQL

### Create Database

Câu lệnh CREATE DATABASE được sử dụng để tạo cơ sở dữ liệu SQL mới.

CREATE DATABASE databasename;

### Create Table

Câu lệnh CREATE TABLE được sử dụng để tạo một bảng mới trong cơ sở dữ liệu.

CREATE TABLE table\_name (

column1 datatype,

column2 datatype,

column3 datatype,

....);

### Insert

Câu lệnh INSERT INTO được sử dụng để chèn các dữ liệu mới vào trong bảng.

INSERT INTO table\_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...)

### Select

Câu lệnh INSERT INTO được sử dụng để trích xuất dữ liệu từ cơ sở dữ liệu

SELECT column1, column2, ...FROM table\_name

### Where

Mệnh đề WHERE được sử dụng để lọc các bản ghi.

Nó chỉ được sử dụng để trích xuất những bản ghi đáp ứng một điều kiện cụ thể.

SELECT column1, column2, ...

FROM table\_name

WHERE condition;

### Order By

Từ khóa ORDER BY sắp xếp các dữ liệu theo thứ tự tăng dần theo mặc định. Để sắp xếp các dữ liệu theo thứ tự giảm dần, hãy sử dụng từ khóa DESC.

SELECT column1, column2, ...

FROM table\_name

ORDER BY column1, column2, ... ASC|DESC;

### Delete

Câu lệnh DELETE được sử dụng để xóa các dữ liệu hiện có trong bảng.

DELETE FROM table\_name WHERE condition;

### Drop Table

Câu lệnh DROP TABLE được sử dụng để thả một bảng hiện có trong cơ sở dữ liệu.

DROP TABLE table\_name;

### Update

Câu lệnh UPDATE được sử dụng để sửa đổi các dữ liệu hiện có trong bảng.

UPDATE table\_name

SET column1 = value1, column2 = value2, ...

WHERE condition;

### Join

Từ khóa INNER JOIN chọn các dữ liệu có giá trị phù hợp trong cả hai bảng.

SELECT column\_name(s)

FROM table1

INNER JOIN table2

ON table1.column\_name = table2.column\_name;

# TRIỂN KHAI …

## Mục 3.1

### Mục 3.1.1

### Mục 3.1.2

## Mục 3.2

# KẾT QUẢ

## Mục 4.1…

## Mục 4.2…

Kết quả được xây dựng dựa trên ngôn ngữ lập trình thể hiện như (Bảng 4.1).

Bảng 4.1 Kiến trúc Hình 5.1 Kiến trúc của mô hình

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Đề tài đã thực hiện được …

# TÀI LIỆU THAM KHẢO

* + - 1. Tên tác giả, Tên tác giả… *Tên công trình*. Nơi xuất bản, năm, trang
      2. Tên tác giả, Tên tác giả… *Tên công trình*. Nơi xuất bản, năm, trang

# PHỤ LỤC