D21 運用實際資料集進行資料視覺化練習

重要知識點

- 了解如何使用Pandas 處理資料集，並加以視覺化效果
- 完成今日課程後你應該可以了解
  - # 導入資料集
  - # 針對特徵，視覺化的處理流程與效果

可視化的好處

- 當你把數據轉換成了規範的格式，也已經採用了適當的統計和分析，接下來就是展示結果的時候了，這時候數據可視化排上了用場 在可視化分析中，經常會遇到多個數據分布之間的比較，分布不同，用到的表達方式也不一樣。

- 在對不同的分布數據進行比較時，通常有兩種形式，要麼突出異常值的差異，要麼突出它們各自差異的細微差別。比如，在統計過程中，不同標準的數據集會有怎樣的差別，或者，如何通過分析來改善評分功能。

- 瞭解有關資料集屬性
  # 我們可以使用 info()或是 descript() 方法瞭解有關資料集屬性的更多資訊。特別是行和列的數量、列名稱、它們的數據類型和空值數

導入數據集

Seaborn 在庫中附帶了幾個重要的數據集。安裝 Seaborn 後，數據集會自動下載。
您可以使用這些資料集中的任何一個進行學習。借助以下函數，您可以載入所需的數據集。

load_dataset()

- Seaborn 可以直接把 PANDAS 的 dataframe 當成資料匯入
- 本日範例，我們以Seaborn 內建的 IRIS 資料集做範例

```
# 導入必要的程式庫
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt

# 取得鳶尾花資料集
df = sns.load_dataset('iris')
```

```
1 df.info()
```

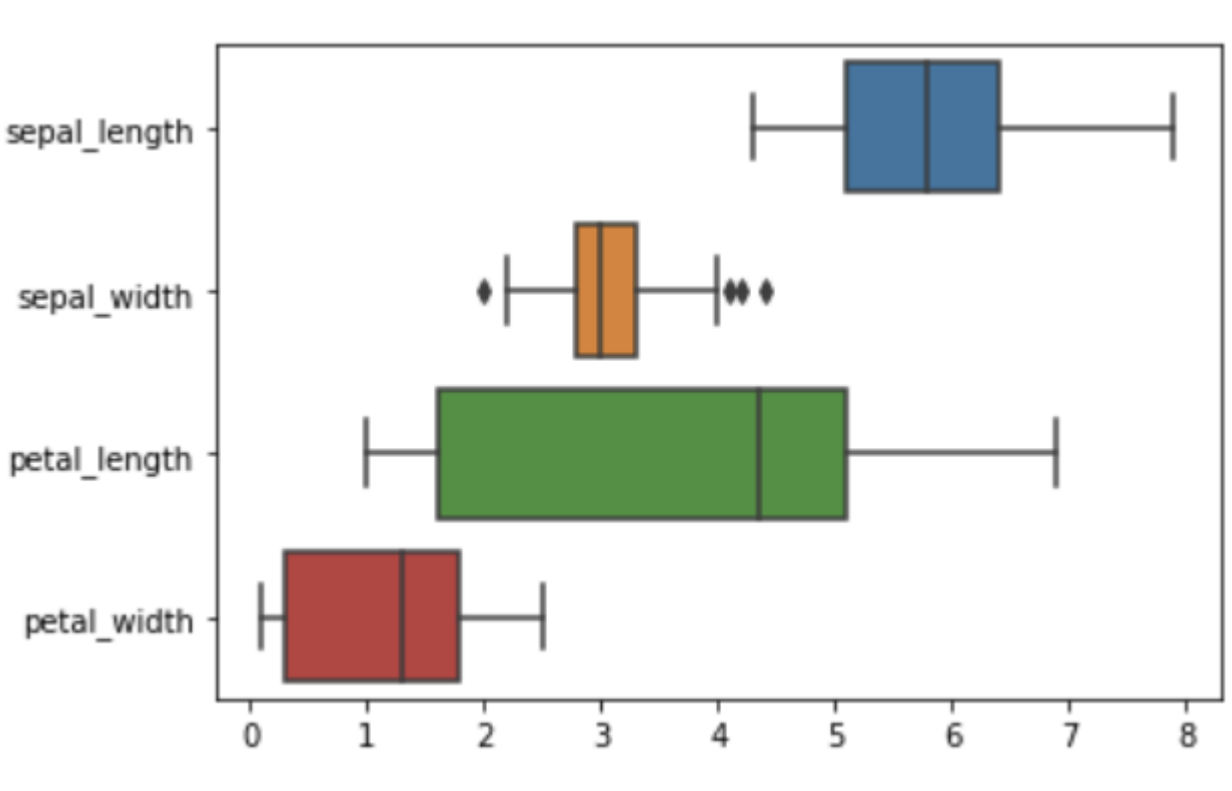
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

瞭解數據集

直接使用PANDAS dataframe，當作參數

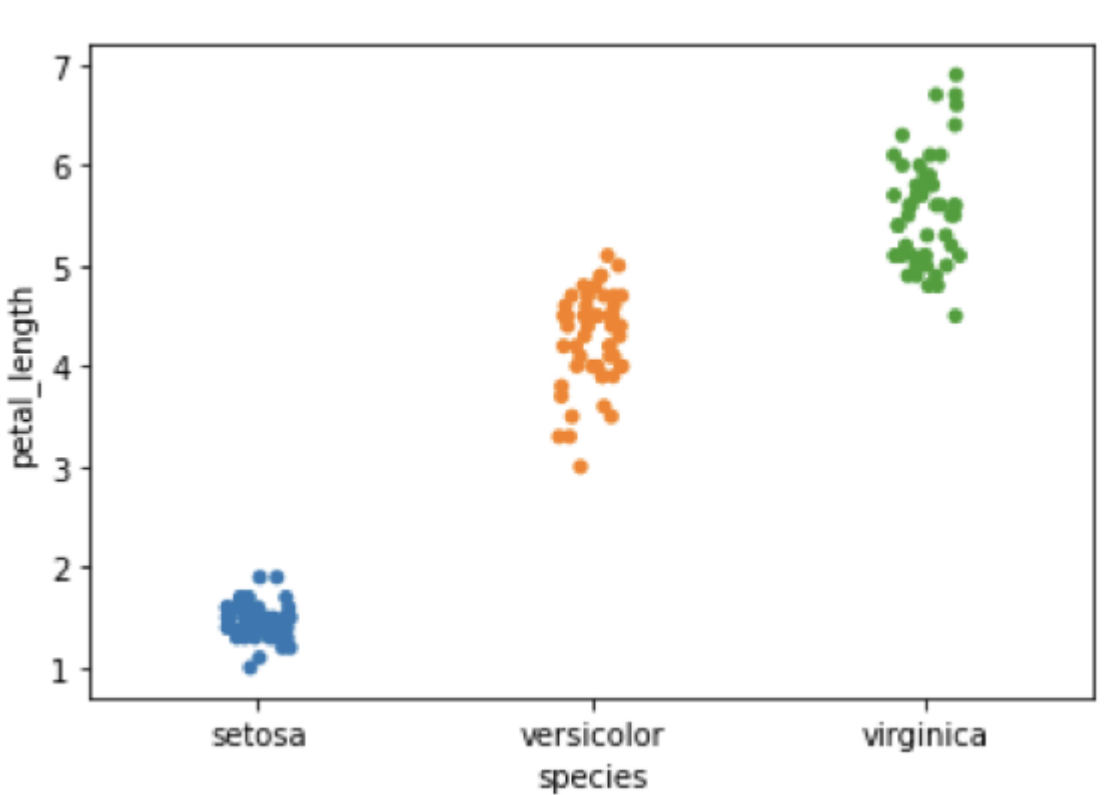
箱形圖顯示了數據的總體分布，同時繪製了異常值的數據點。這個物理點讓它們的特定值在樣本之間容易被識別和比較。

```
sns.boxplot(data = df, orient = "h")
```



當一個或兩個正在研究的變數是分類的時，我們使用像條帶線()、swarmplot()等的圖。
查看到每個物種petal_length的差異。但是，散點圖的主要問題是散點圖上的點重疊。

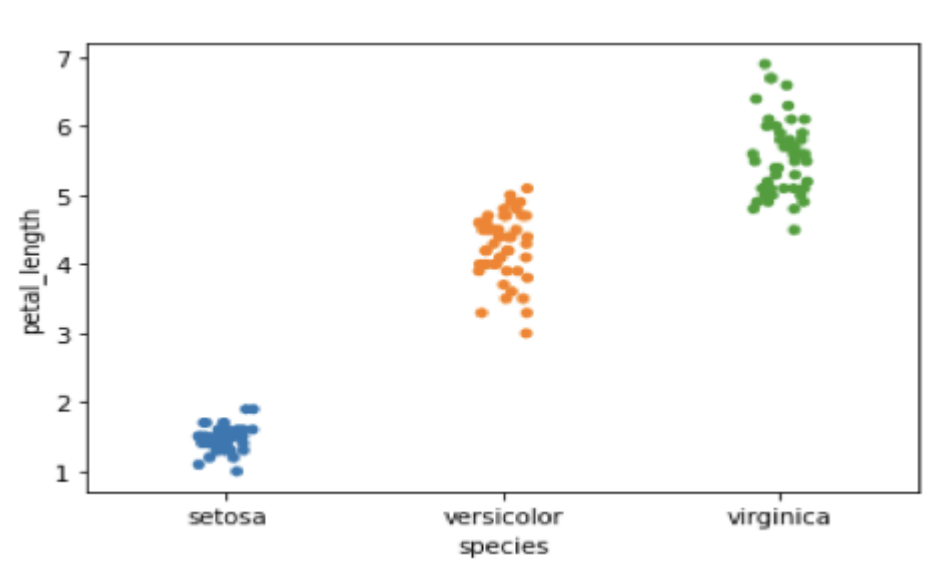
```
sns.stripplot(x = "species", y = "petal_length", data = df)
```



分類式的變數

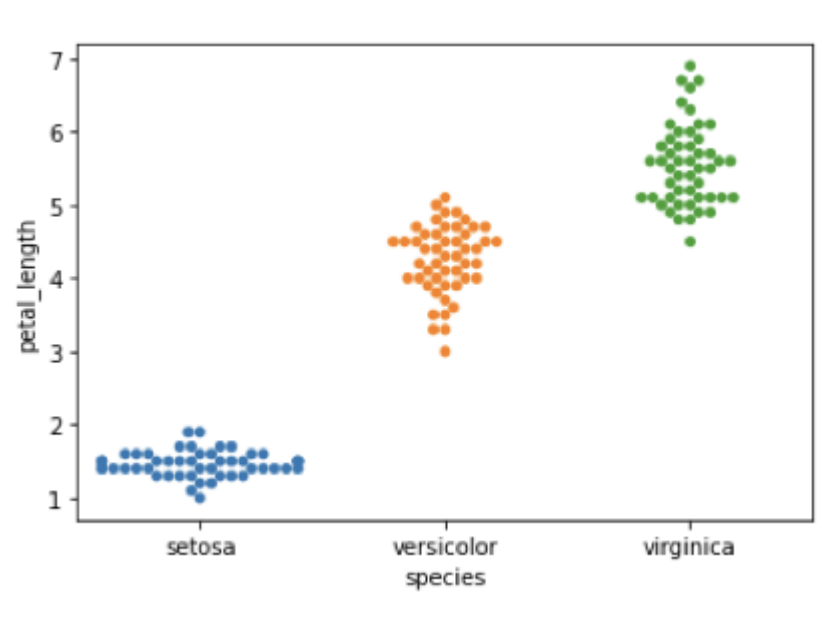
上述散點圖的主要問題是散點圖上的點重疊。我們使用"抖動"參數來處理此類方案。
抖動會為數據添加一些隨機雜訊。此參數將沿分類軸調整位置。

```
sns.stripplot(x = "species", y = "petal_length", data = df, jitter=True)
```



另一個可以用作「抖動」 的替代選項是函數群圖()。
此函數將散點圖的每個點都放在分類軸上，從而避免重疊點

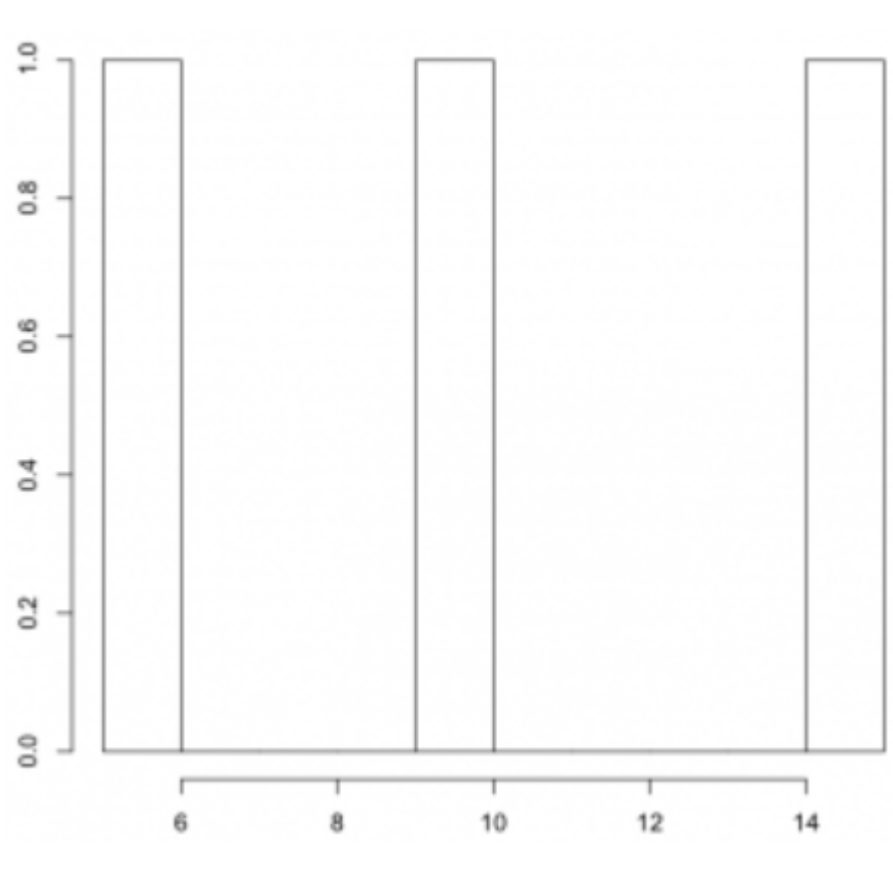
```
sns.swarmplot(x = "species", y = "petal_length", data = df)
```



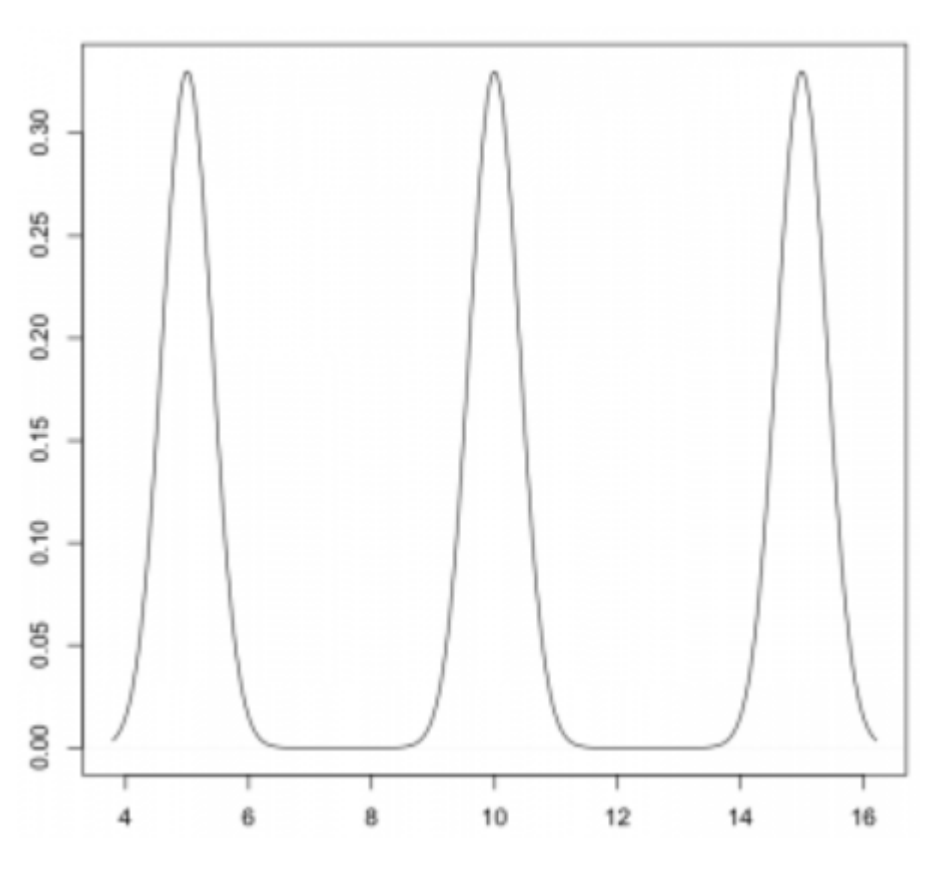
核密度估計(Kernel Density Estimates, KDE)

所謂核密度估計，就是採用平滑的峰值函數("核")來擬合觀察到的數據點，從而對真實的概率分佈曲線進行類比。以下面3個數據點(5, 10, 15)的一維數據集為例：

直方圖：



KDE：



所有平滑的峰值函數均可作為KDE的核函數來使用，只要對歸一化后的KDE而言(描繪在圖上的是數據點出現的概率值)，該函數曲線下方的面積和等於1即可 — 只有一個數據點時，單個波峰下方的面積為1，存在多個數據點時，所有波峰下方的面積之和為1。

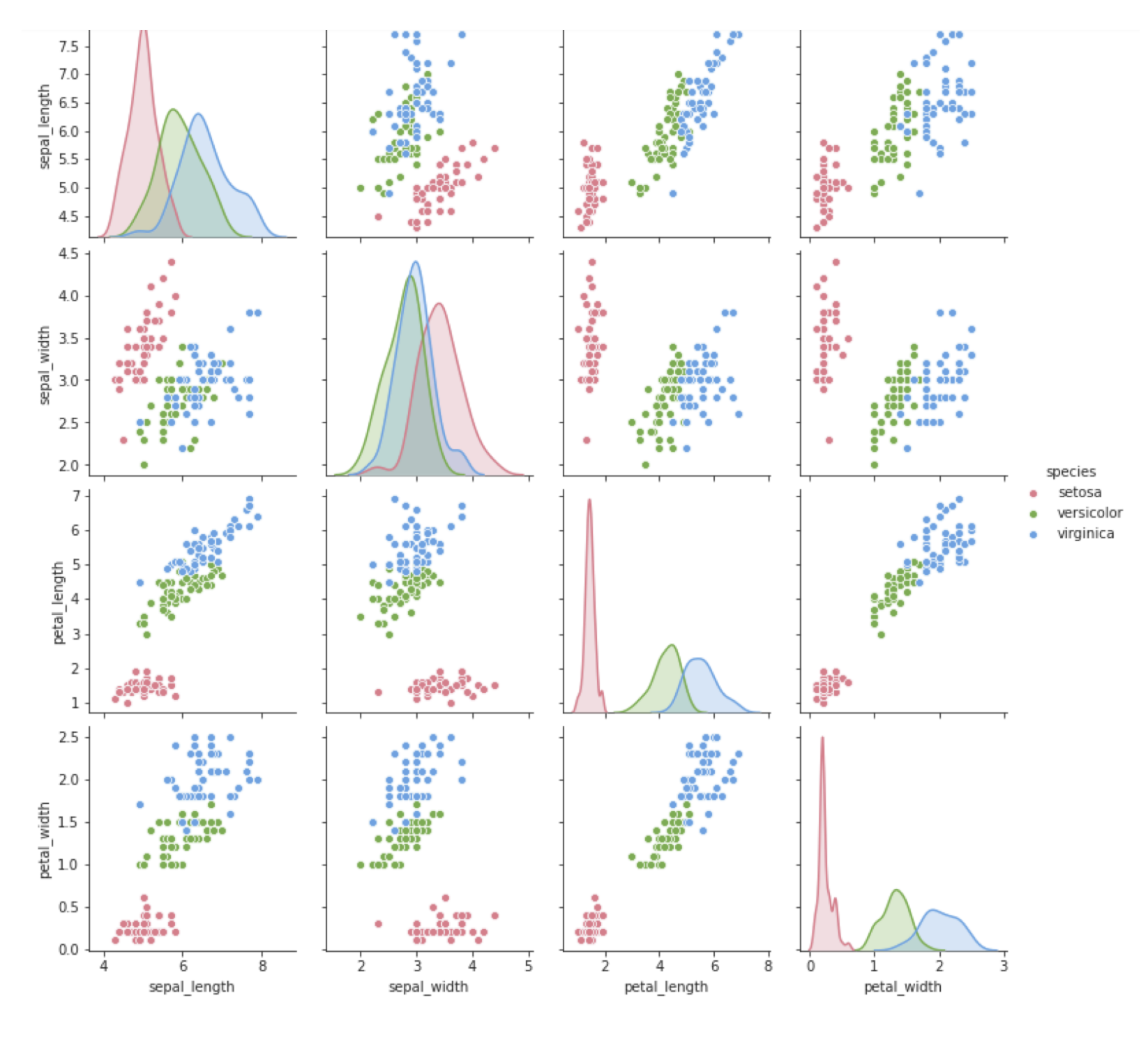
概而言之，函數曲線需囊括所有可能出現的數據值的情況。

內核密度估計是用來繪製密度數據，這將更加準確地反映**總體的基本變量**。

1. 在數據點處為波峰
2. 曲線下方面積為1

可以觀察每個情節的變化。繪圖採用矩陣格式，其中行名表示 x 軸，列名稱表示 y 軸。
對角線圖是內核密度圖，其中其他圖是散點圖
內核密度估計是估計變數分佈的非參數化方法。

```
sns.set_style("ticks")
sns.pairplot(df,hue = 'species',diag_kind = "kde",kind = "scatter",palette = "husl")
```



知識點回顧

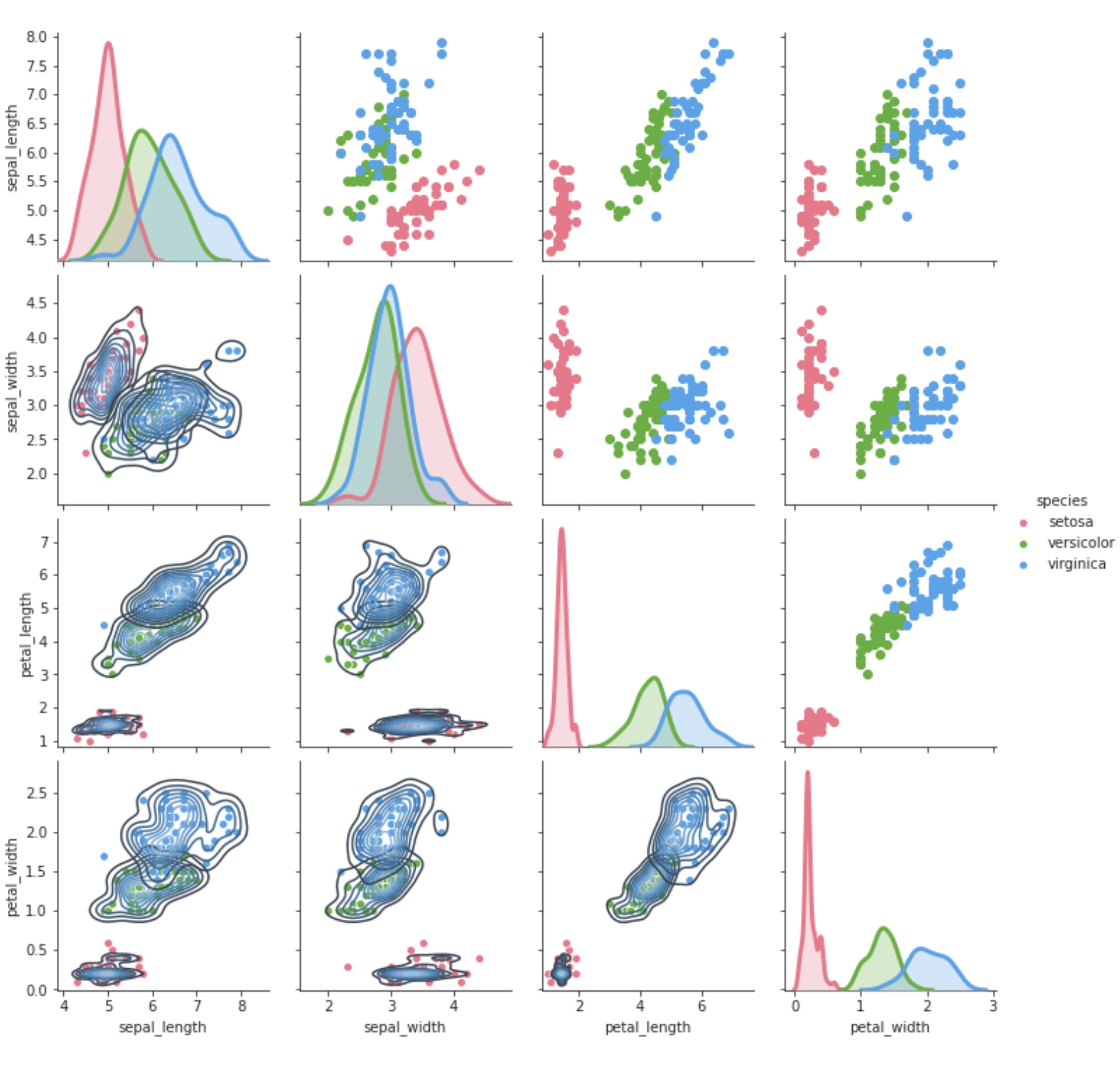
可以在上三角形和下三角形使用不同的函數來查看關係的不同方面

```
g = sns.pairplot(df,hue = 'species',diag_kind = "kde",kind = "scatter",palette = "husl")

g.map_upper(plt.scatter)

g.map_lower(sns.kdeplot, cmap = "Blues_d")

g.map_diag(sns.kdeplot, lw = 3, legend = False);
```



延伸閱讀

使用Seaborn 進行可視化

[資料分析&機器學習] 第2.5講：資料視覺化(Matplotlib, Seaborn, Plotly)
網站：medium

- 針對 Matplotlib，Matplotlib & Pandas 帶入實例
- 針對 Seaborn，Seaborn & Pandas帶入實例

[資料分析&機器學習] 第2.5講：資料視覺化(Matplotlib, Seaborn, Plotly)

Yeh James Follow
Oct 11, 2017 · 9 min read

資料視覺化除了最後一步呈現你的成果之外，還可以在分析的過程中用資料視覺化來看出一些insight，比方說用熱點圖來看你的Deep learning的model是對圖片中哪一部分的看得較重要，或是可以降維之後將資料視覺化去看資料在空間中的分佈，來決定下一步的分析要怎麼做。

Python資料視覺化主要有三大套件：

1. Matplotlib
2. Seaborn
3. Plotly

其他還有像是Bokeh, ggplot…十幾種Python視覺化套件，以及更進階的BI Tool(Tableau, Spotfire, MicroStrategy)。如果你去大公司上班，ex: 台積電, Yahoo…都會購買這些要價不菲的BI Tool，對於資料處理&視覺化功能有更豐富的運用。但對於一般使用者只要先掌握上述三個就夠了。今天就要來介紹一下這三種視覺化套件

Seaborn 畫熱力圖時行名中包括中文顯示成方框
網站：blog.csdn

說明如何解決中文字在圖形得顯示問題

seaborn畫熱力圖時行名中包括中文顯示成方框

df該dataframe的行名是中文，可以加上下面代碼中紅色這句，就可以顯示出中文

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(font="simhei")#遇到標籤需要漢字的可以在繪圖前加上這句
f, ax = plt.subplots(figsize=(10,10))
sns.heatmap(df, annot=False, ax=ax)
```

下一步：閱讀範例與完成作業