Python初級數據分析員證書

（五）進階Python數據分析及可視化技巧

# 11.Seaborn套件

# Chapter Summary

- Introduction
- Installation
- Histplot, pairplot, lineplot, heatmap, scatterplot,
- regplot, lmplot, relplot, boxplot, catplot, jointplot, JointGrid
- Setting with styles
- Matplotlib vs Seaborn

# 簡介

**Seaborn** 是一個用於在 Python 中製作統計圖形的工具庫。

它建立在 matplotlib 之上，並與 Pandas 數據結構緊密集成。

它提供了一個高級介面，用於繪製有吸引力且資訊豐富的統計圖形。

Latest version: v0.13.2 on June 2024

# 安裝

## Support Python version

- Python 3.7+


## Mandatory dependencies

- Numpy
- Pandas
- Matplotlib

## Installation via PyPI

```
pip install seaborn
```

# 概論

學習了 Matplotlib 後，您會發現 Seaborn 更容易且得心應手。

我們將使用 Seaborn 中的一些測試數據集並像這樣載入DF。

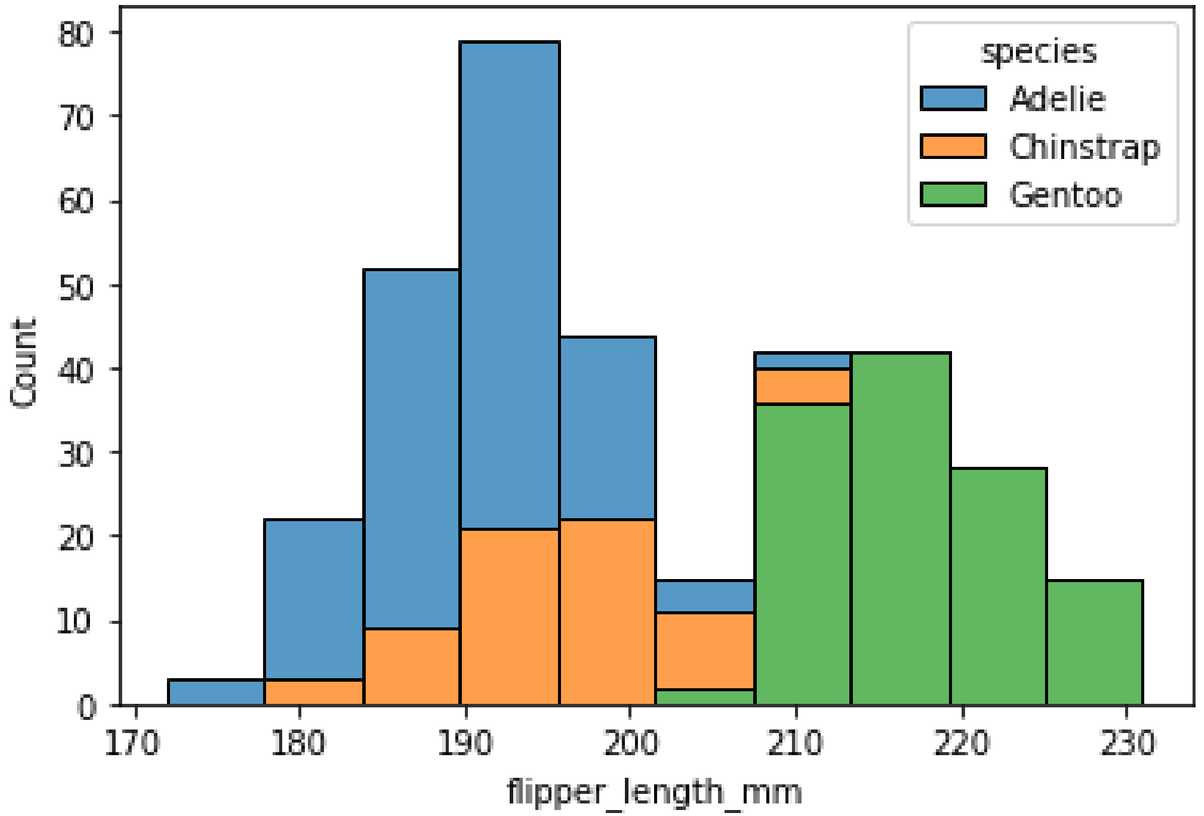
```
1 import seaborn as sns
2 penguins = sns.load_dataset("penguins")
3 penguins.head(3)
```

| | species | island | bill_length_mm | bill_depth_mm | flipper_length_mm | body_mass_g | sex |
|---|---|---|---|---|---|---|---|
| **0** | Adelie | Torgersen | 39.1 | 18.7 | 181.0 | 3750.0 | Male |
| **1** | Adelie | Torgersen | 39.5 | 17.4 | 186.0 | 3800.0 | Female |
| **2** | Adelie | Torgersen | 40.3 | 18.0 | 195.0 | 3250.0 | Female |

# 繪製堆疊直方圖– sns.histplot

```
1 sns.histplot(data=penguins, x="flipper_length_mm", hue="species", multiple="stack")
```

```
<AxesSubplot:xlabel='flipper_length_mm', ylabel='Count'>
```



# Relation Plot 關係圖 – sns.relplot

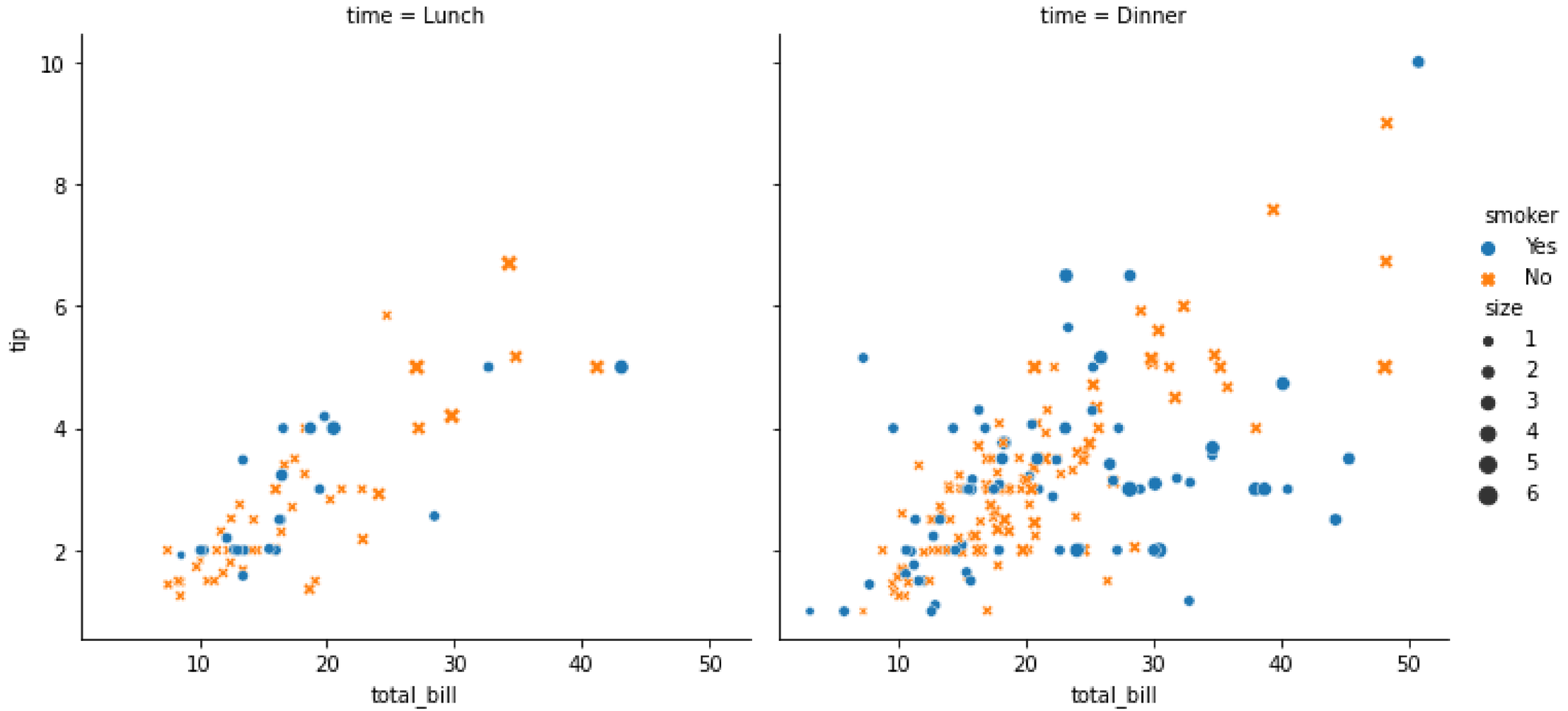
```
# Load an example dataset
tips = sns.load_dataset("tips")
tips.head(3)
```

| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| **0** | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 |
| **1** | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 |
| **2** | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 |

```
# Create a visualization
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```

```
<seaborn.axisgrid.FacetGrid at 0x125bdf7c0>
```

# Relation Plot 關係圖 – sns.relplot



# Discover the code 程式碼探究

From previous two examples, we learnt some basic patterns of Seaborn:

- sns.<___plot>(data=<dataset>, <x , y labelling>, <other chart setting>)
- The dataset is Pandas DataFrame

```
sns.histplot(data=penguins, x="flipper_length_mm",
             hue="species", multiple="stack")
```

```
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```

# sns.histplot

對於每一種plot圖，我們總能在官網上找到完整的用法:

https://seaborn.pydata.org/

The sns.histplot is a function. Some of the argument is 必須的, 像data frame.

## seaborn.histplot

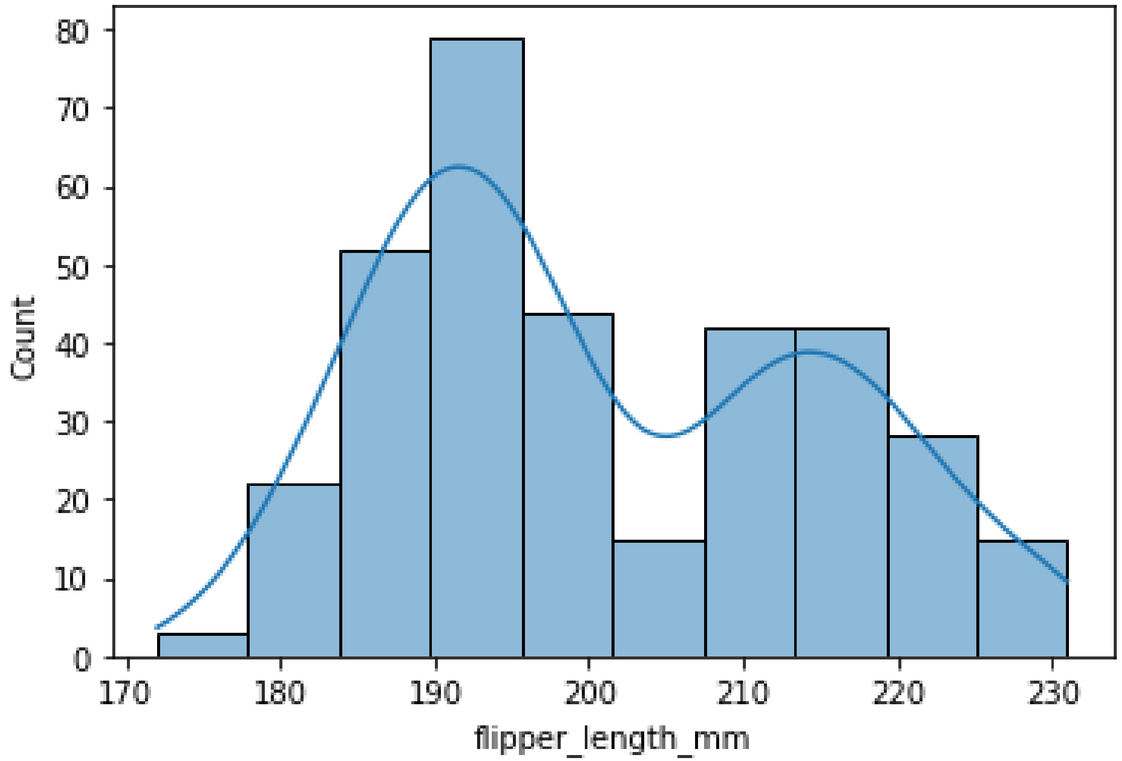
```
seaborn.histplot(data=None, *, x=None, y=None, hue=None, weights=None,
stat='count', bins='auto', binwidth=None, binrange=None, discrete=None,
cumulative=False, common_bins=True, common_norm=True, multiple='layer',
element='bars', fill=True, shrink=1, kde=False, kde_kws=None,
line_kws=None, thresh=0, pthresh=None, pmax=None, cbar=False,
cbar_ax=None, cbar_kws=None, palette=None, hue_order=None, hue_norm=None,
color=None, log_scale=None, legend=True, ax=None, **kwargs)
```

# sns.histplot

使用不同的參數（設置），我們將有不同的風格。

```
1  sns.histplot(data=penguins, x="flipper_length_mm", kde=True)
```

```
<AxesSubplot:xlabel='flipper_length_mm', ylabel='Count'>
```




**kde** (kernel density estimate) here is a Boolean argument.

If True, 它計算核密度估計值以平滑分佈並在圖上顯示為（一條或多條）線。這通常用於直方圖視覺物件。

# sns.pairplot

**sns.pairplot** 繪製數據集的成對關係圖。

記住我們代數章節的例子？

We had pair

Plots of 紫金-vs-金(left) and

招金-vs-金(right).

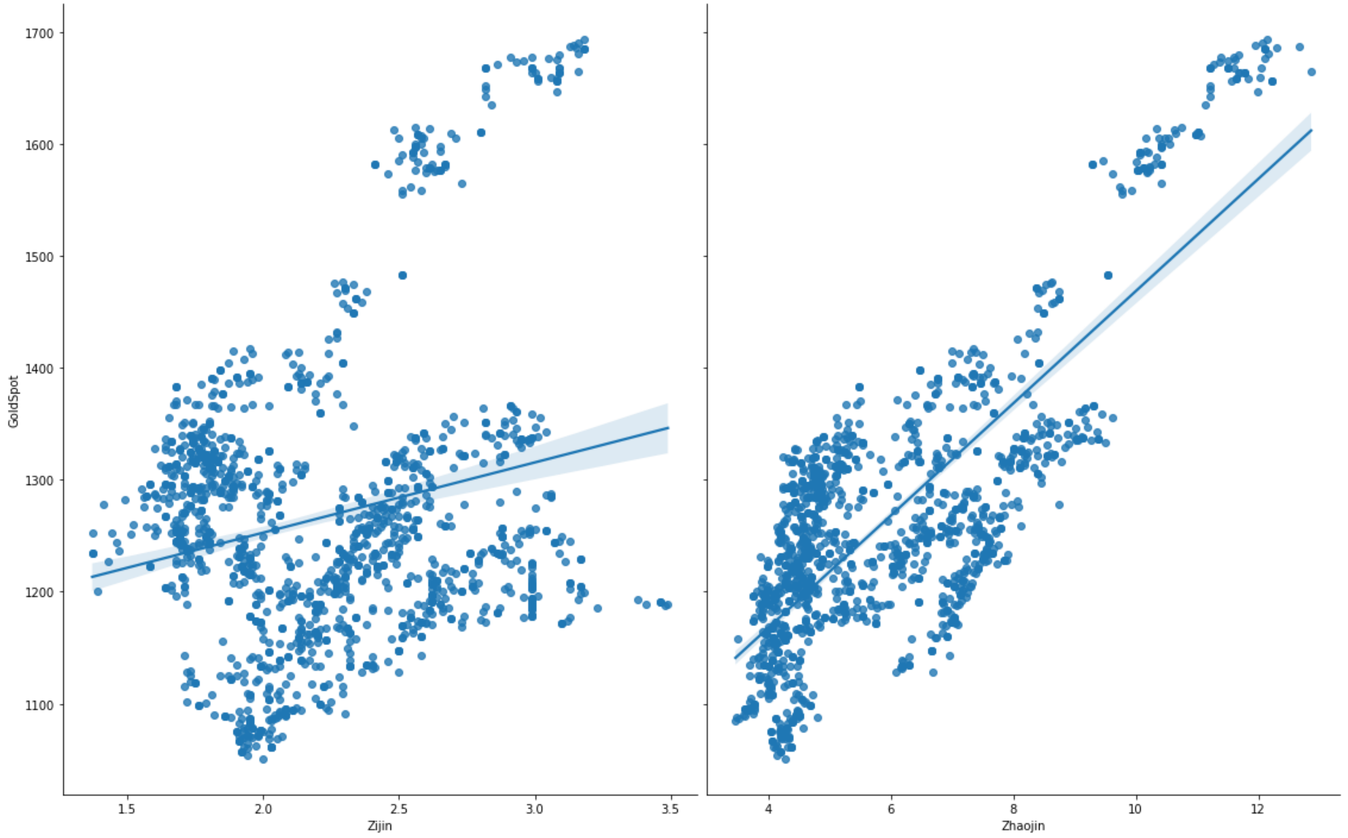
論點是直截了當及易看見的。

從兩個圖中，我們知道

招金-vs-金(right)相關性

較明顯 (Significant)。

```
1  sns.pairplot(df, x_vars=["Zijin", "Zhaojin"], y_vars=["GoldSpot"],
2               height=10, aspect=.8, kind="reg");
```



# sns.pairplot

函數參數看起來很可怕，但其中大多數都可以忽略並將它們設置為預設值。核心的有: data, x_vars, y_vars, vars.

Hue: 數據中要將繪圖方面映射到的變數不同顏色different colours.

Kind: Kind of plot to make. Pick from {‘scatter’, ‘kde’, ‘hist’, ‘reg’}. Kde here is as variable.

## seaborn.pairplot

```
seaborn.pairplot(data, *, hue=None, hue_order=None, palette=None,
vars=None, x_vars=None, y_vars=None, kind='scatter', diag_kind='auto',
markers=None, height=2.5, aspect=1, corner=False, dropna=False,
plot_kws=None, diag_kws=None, grid_kws=None, size=None)
```

# sns.pairplot

## Back to the penguins dataset

```
1 penguins.sample(6)
```

| | species | island | bill_length_mm | bill_depth_mm | flipper_length_mm | body_mass_g | sex |
|---|---|---|---|---|---|---|---|
| **48** | Adelie | Dream | 36.0 | 17.9 | 190.0 | 3450.0 | Female |
| **155** | Chinstrap | Dream | 45.4 | 18.7 | 188.0 | 3525.0 | Female |
| **224** | Gentoo | Biscoe | 47.6 | 14.5 | 215.0 | 5400.0 | Male |
| **68** | Adelie | Torgersen | 35.9 | 16.6 | 190.0 | 3050.0 | Female |
| **276** | Gentoo | Biscoe | 43.8 | 13.9 | 208.0 | 4300.0 | Female |
| **338** | Gentoo | Biscoe | 47.2 | 13.7 | 214.0 | 4925.0 | Female |

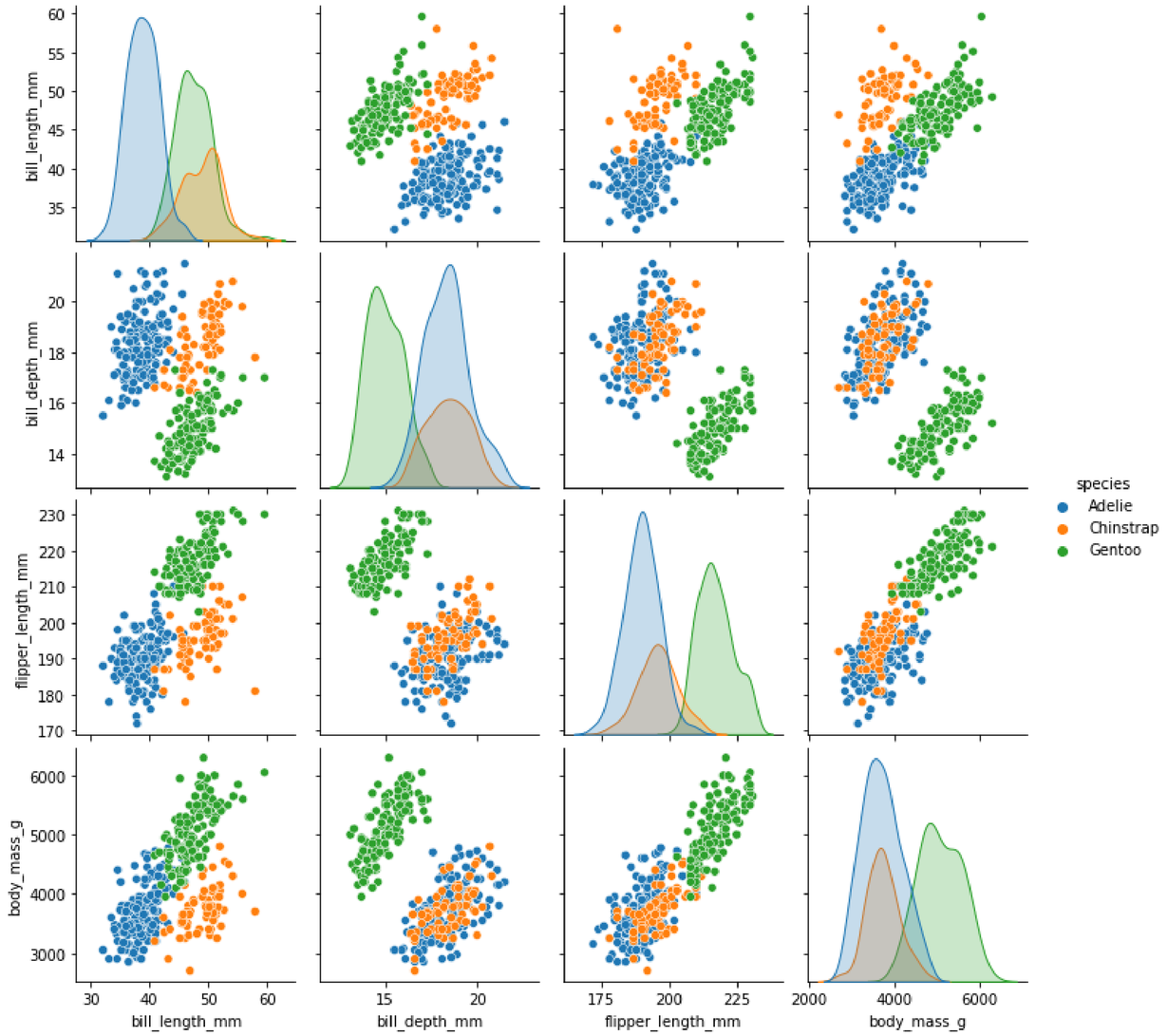
## 編寫以下命令以瞭解 DF

```
1 penguins['species'].unique()
```

```
array(['Adelie', 'Chinstrap', 'Gentoo'], dtype=object)
```

# sns.pairplot

```
1  sns.pairplot(penguins, hue="species")
```




我們 **hue** 顏色的物種。然後，我們可以看到不同物種的不同物種數據。

例如, Gentoo(green) 有更長的鰭狀肢（手）長度和更重的體重。

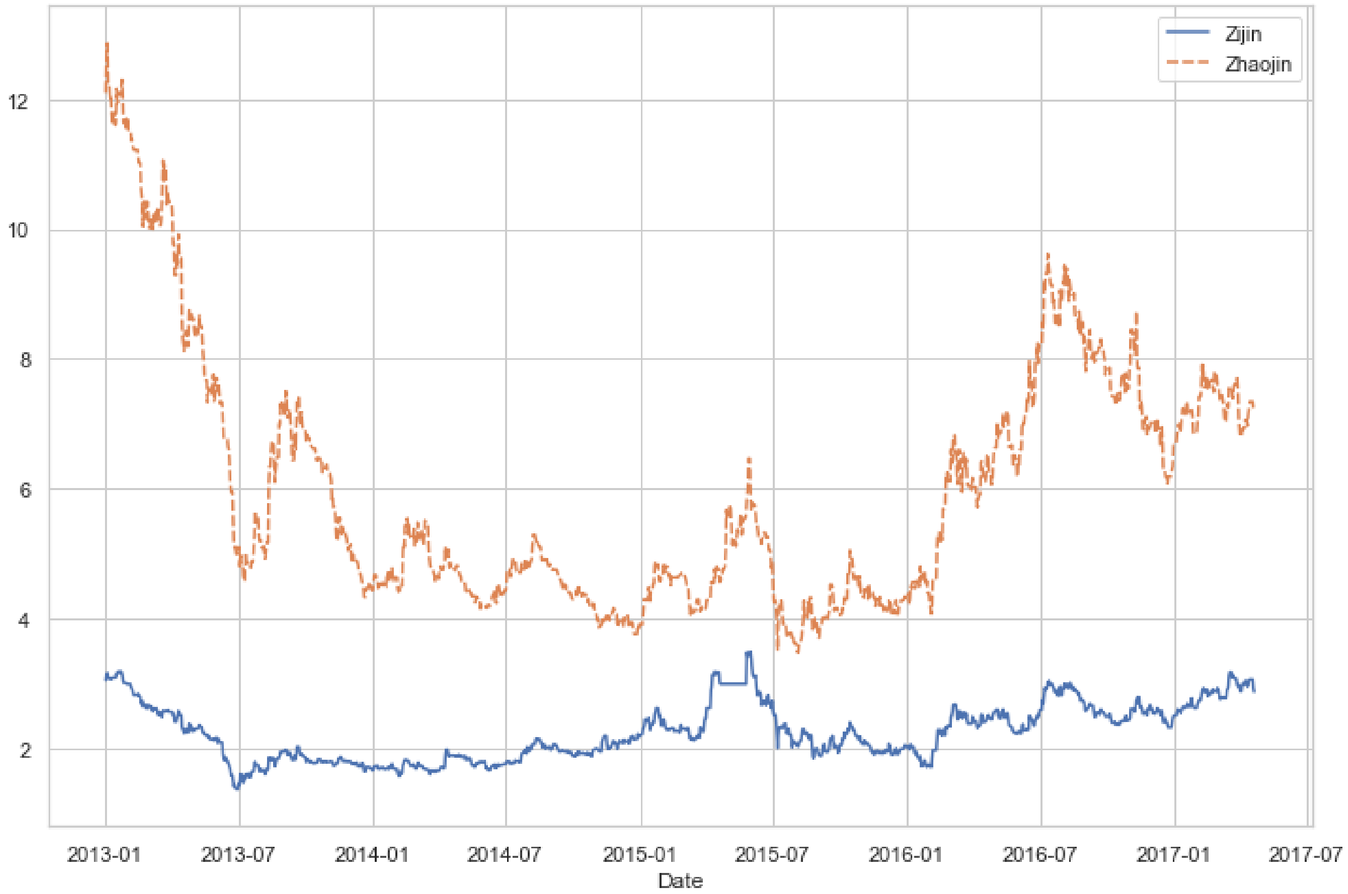
(Gentoo Penguin)

# sns.lineplot

```
1  sns.lineplot(data=df[['Zijin', 'Zhaojin']])
```

使用我們以前的gold stock data

for sns.lineplot



# sns.heatmap

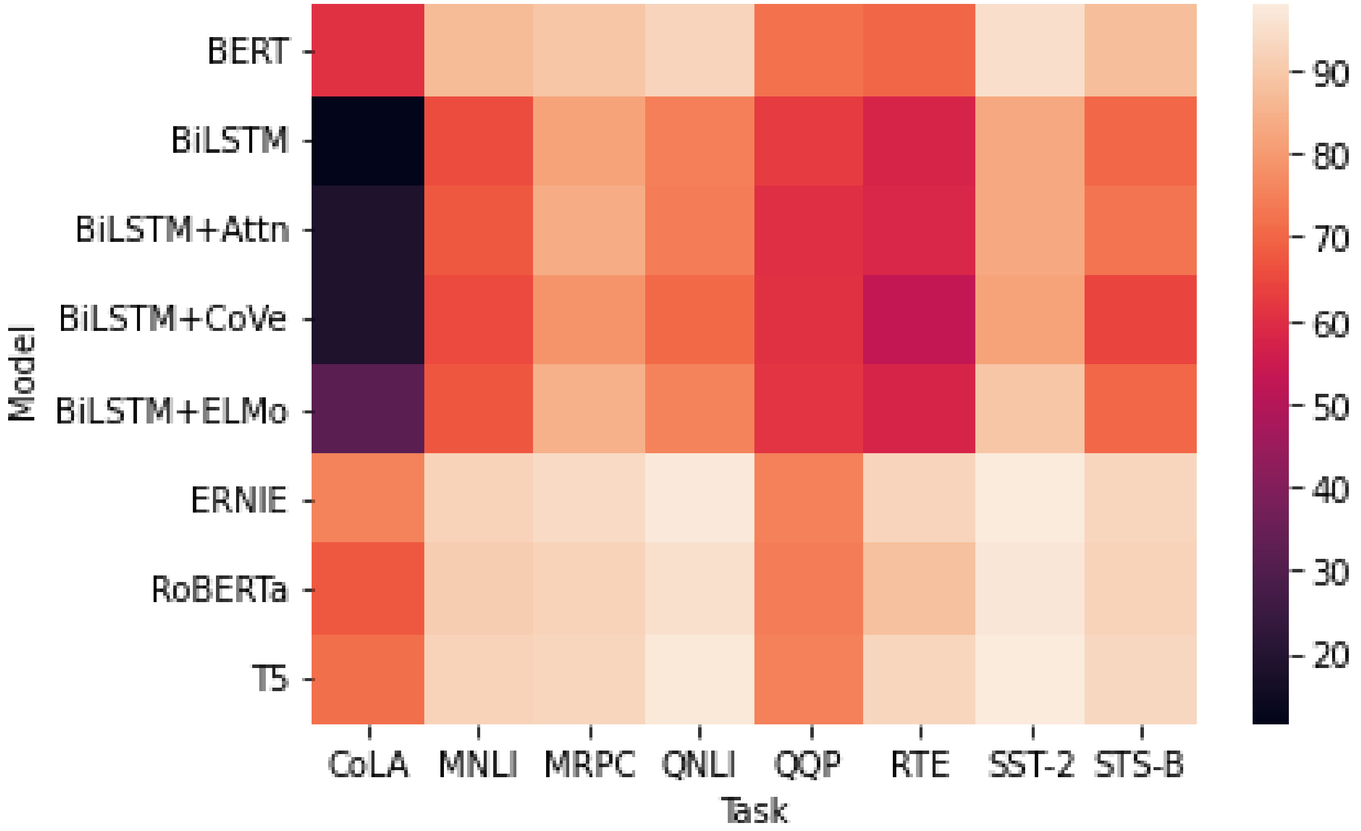
讓我們從 sns 數據集載入資料DF

```
1 glue = sns.load_dataset("glue").pivot("Model", "Task", "Score")
2 glue
```

| Task | CoLA | MNLI | MRPC | QNLI | QQP | RTE | SST-2 | STS-B |
|---|---|---|---|---|---|---|---|---|
| **Model** | | | | | | | | |
| **BERT** | 60.5 | 86.7 | 89.3 | 92.7 | 72.1 | 70.1 | 94.9 | 87.6 |
| **BiLSTM** | 11.6 | 65.6 | 81.8 | 74.6 | 62.5 | 57.4 | 82.8 | 70.3 |
| **BiLSTM+Attn** | 18.6 | 67.6 | 83.9 | 74.3 | 60.1 | 58.4 | 83.0 | 72.8 |
| **BiLSTM+CoVe** | 18.5 | 65.4 | 78.7 | 70.8 | 60.6 | 52.7 | 81.9 | 64.4 |
| **BiLSTM+ELMo** | 32.1 | 67.2 | 84.7 | 75.5 | 61.1 | 57.4 | 89.3 | 70.3 |
| **ERNIE** | 75.5 | 92.3 | 93.9 | 97.3 | 75.2 | 92.6 | 97.8 | 93.0 |
| **RoBERTa** | 67.8 | 90.8 | 92.3 | 95.4 | 74.3 | 88.2 | 96.7 | 92.2 |
| **T5** | 71.6 | 92.2 | 92.8 | 96.9 | 75.1 | 92.8 | 97.5 | 93.1 |

# sns.heatmap

從“模型”列中，我們瞭解到有 8 個參數是：BERT, BiLSTM, etc. And the column ColA, MNLI, MRPC, 符是模型的測試分數。
如果得分最高意味著它是一個好的模型，你會選擇哪個模型？

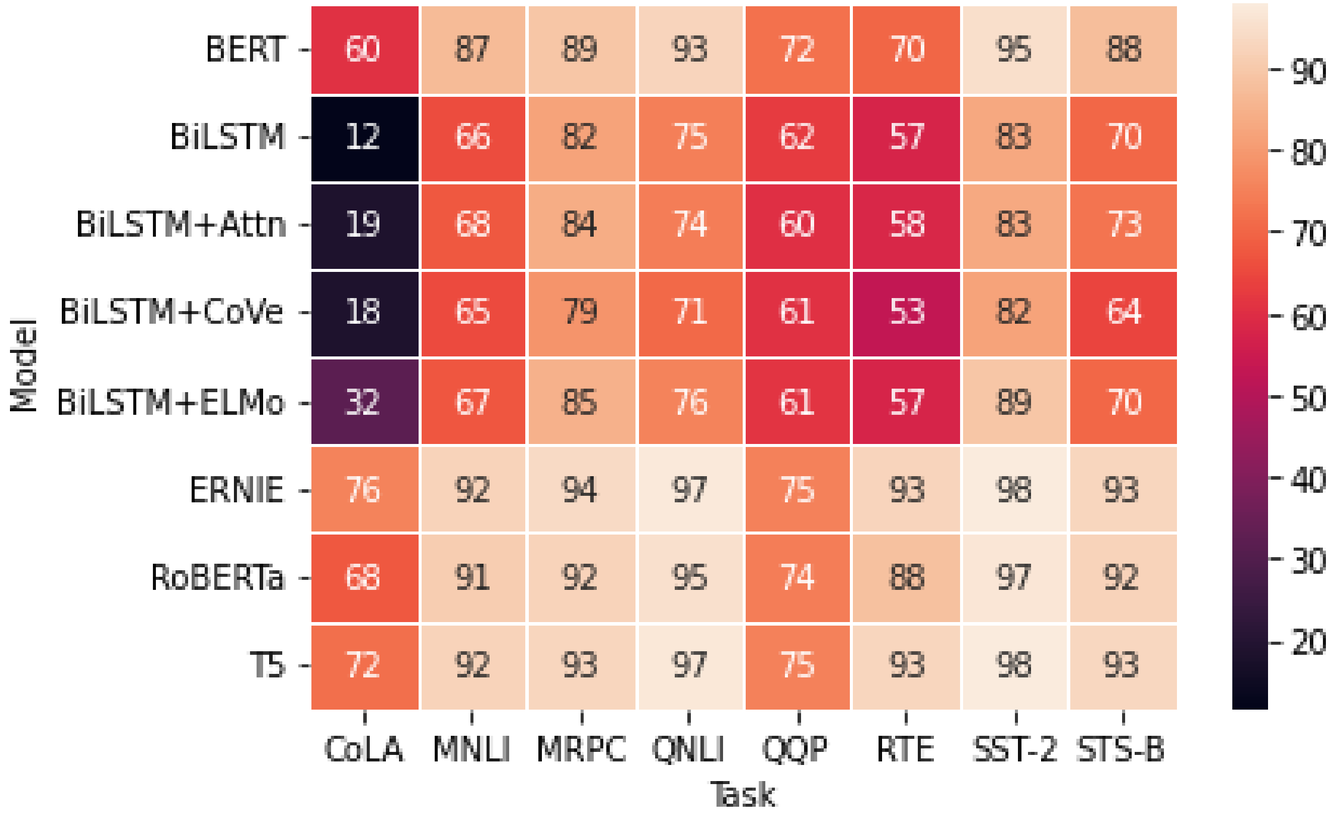



| Task | CoLA | MNLI | MRPC | QNLI | QQP | RTE | SST-2 | STS-B |
|---|---|---|---|---|---|---|---|---|
| **Model** | | | | | | | | |
| **BERT** | 60.5 | 86.7 | 89.3 | 92.7 | 72.1 | 70.1 | 94.9 | 87.6 |
| **BiLSTM** | 11.6 | 65.6 | 81.8 | 74.6 | 62.5 | 57.4 | 82.8 | 70.3 |
| **BiLSTM+Attn** | 18.6 | 67.6 | 83.9 | 74.3 | 60.1 | 58.4 | 83.0 | 72.8 |
| **BiLSTM+CoVe** | 18.5 | 65.4 | 78.7 | 70.8 | 60.6 | 52.7 | 81.9 | 64.4 |
| **BiLSTM+ELMo** | 32.1 | 67.2 | 84.7 | 75.5 | 61.1 | 57.4 | 89.3 | 70.3 |
| **ERNIE** | 75.5 | 92.3 | 93.9 | 97.3 | 75.2 | 92.6 | 97.8 | 93.0 |
| **RoBERTa** | 67.8 | 90.8 | 92.3 | 95.4 | 74.3 | 88.2 | 96.7 | 92.2 |
| **T5** | 71.6 | 92.2 | 92.8 | 96.9 | 75.1 | 92.8 | 97.5 | 93.1 |

# sns.heatmap

You may annotate the data in heatmap cells.

```
1  sns.heatmap(glue, annot=True, linewidths=.5)
```

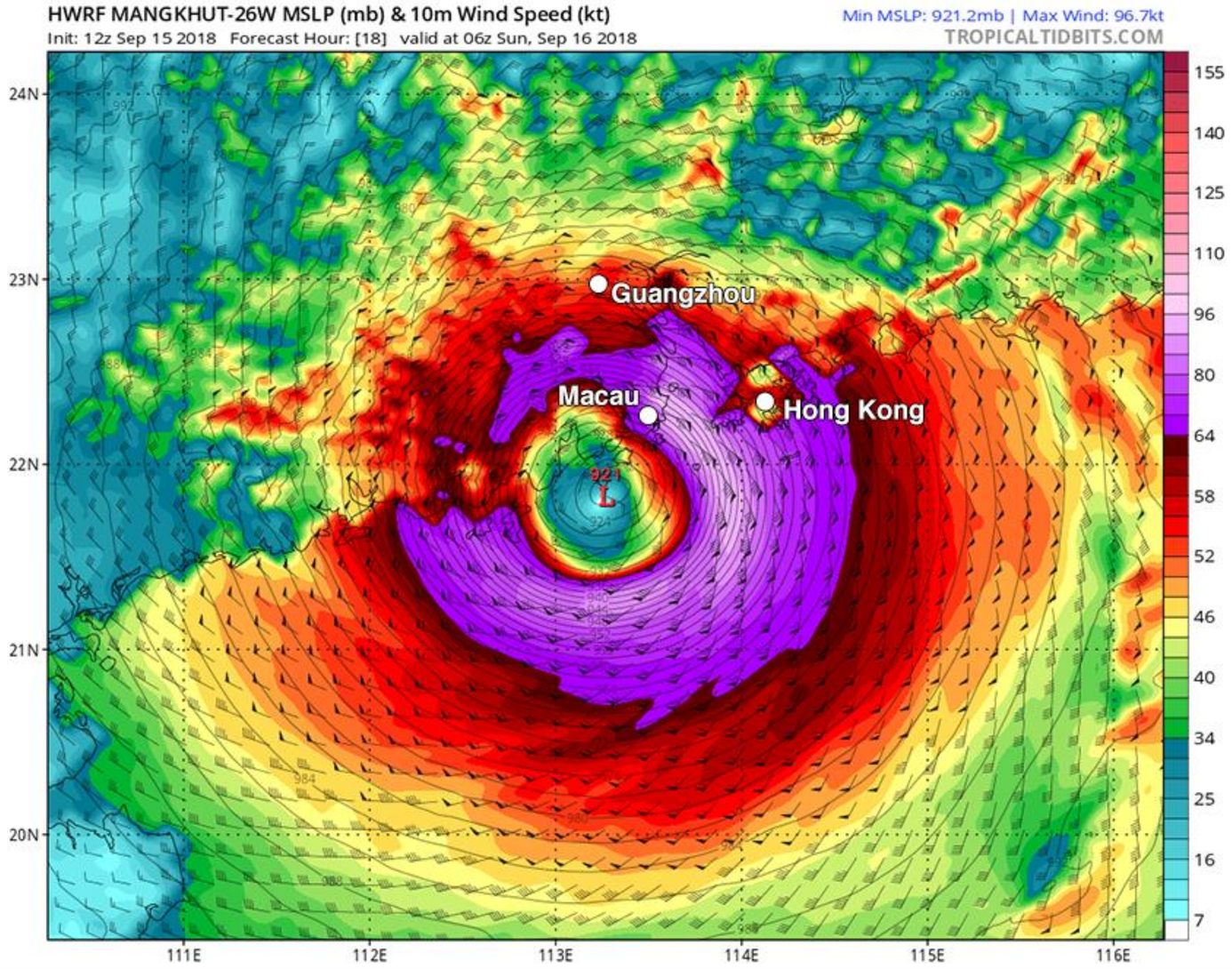


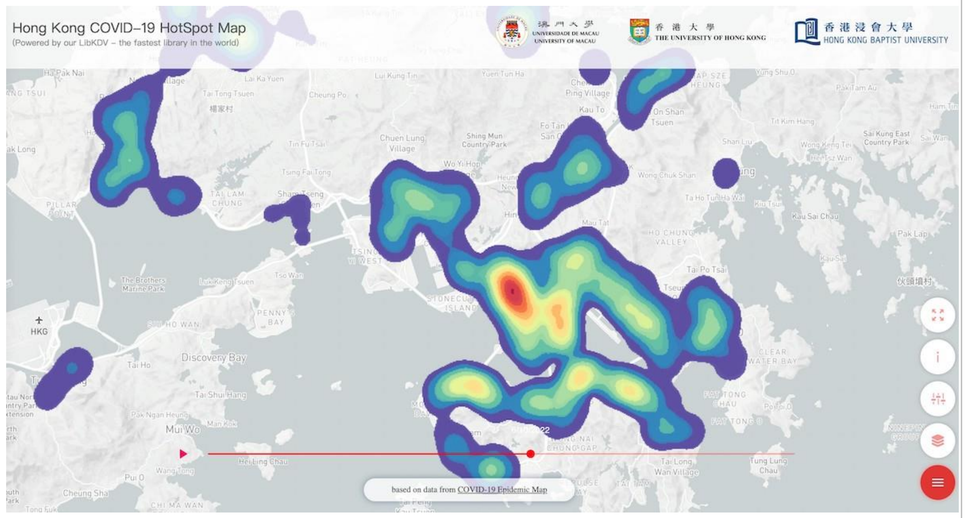

您是否意識到

Seaborn 會自動變更

註釋字型顏色？

# 風速Heatmap and Covid-19 Hotspot




For package solving geography data visualization, try GeoPandas

GeoPandas

# sns.scatterplot

散點圖可以幫助我們解決回歸和分類問題。

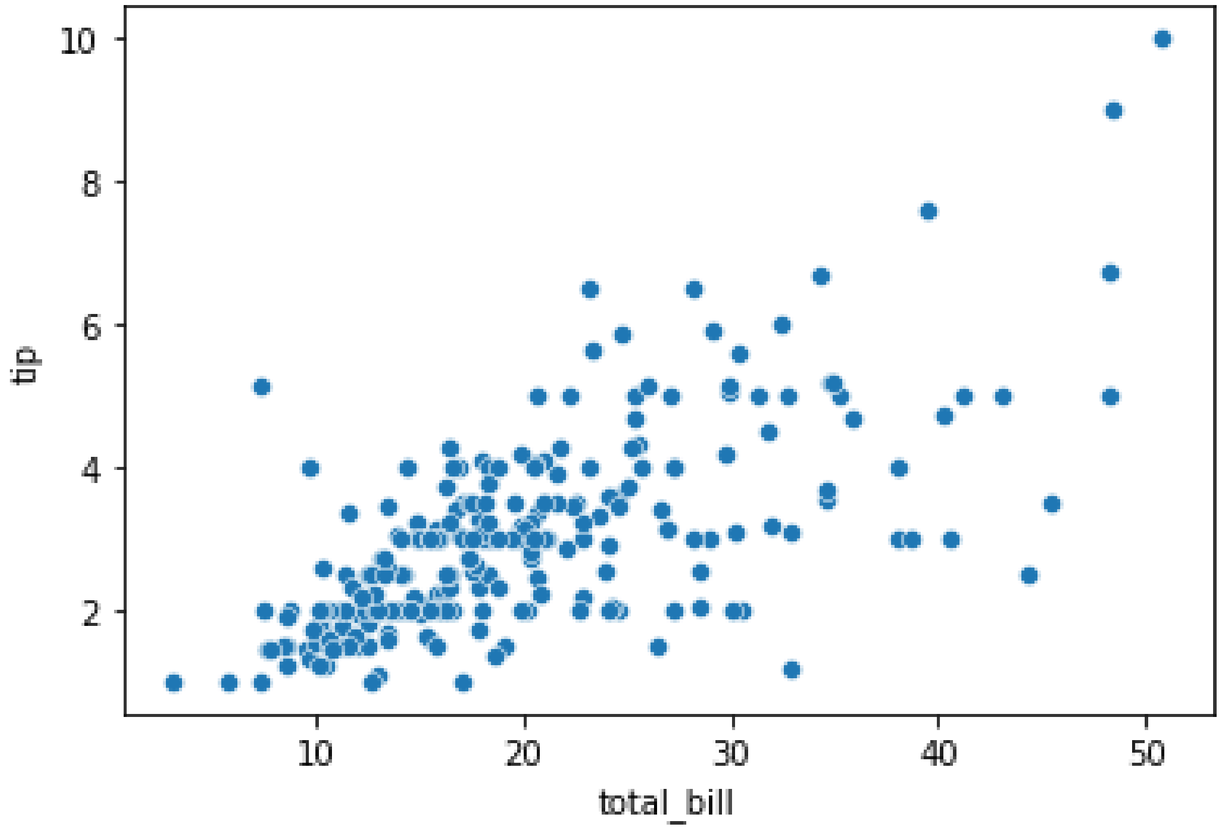
```
1  tips = sns.load_dataset("tips")
2  tips.sample(4)
```

| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| **43** | 9.68 | 1.32 | Male | No | Sun | Dinner | 2 |
| **42** | 13.94 | 3.06 | Male | No | Sun | Dinner | 2 |
| **191** | 19.81 | 4.19 | Female | Yes | Thur | Lunch | 2 |
| **115** | 17.31 | 3.50 | Female | No | Sun | Dinner | 2 |

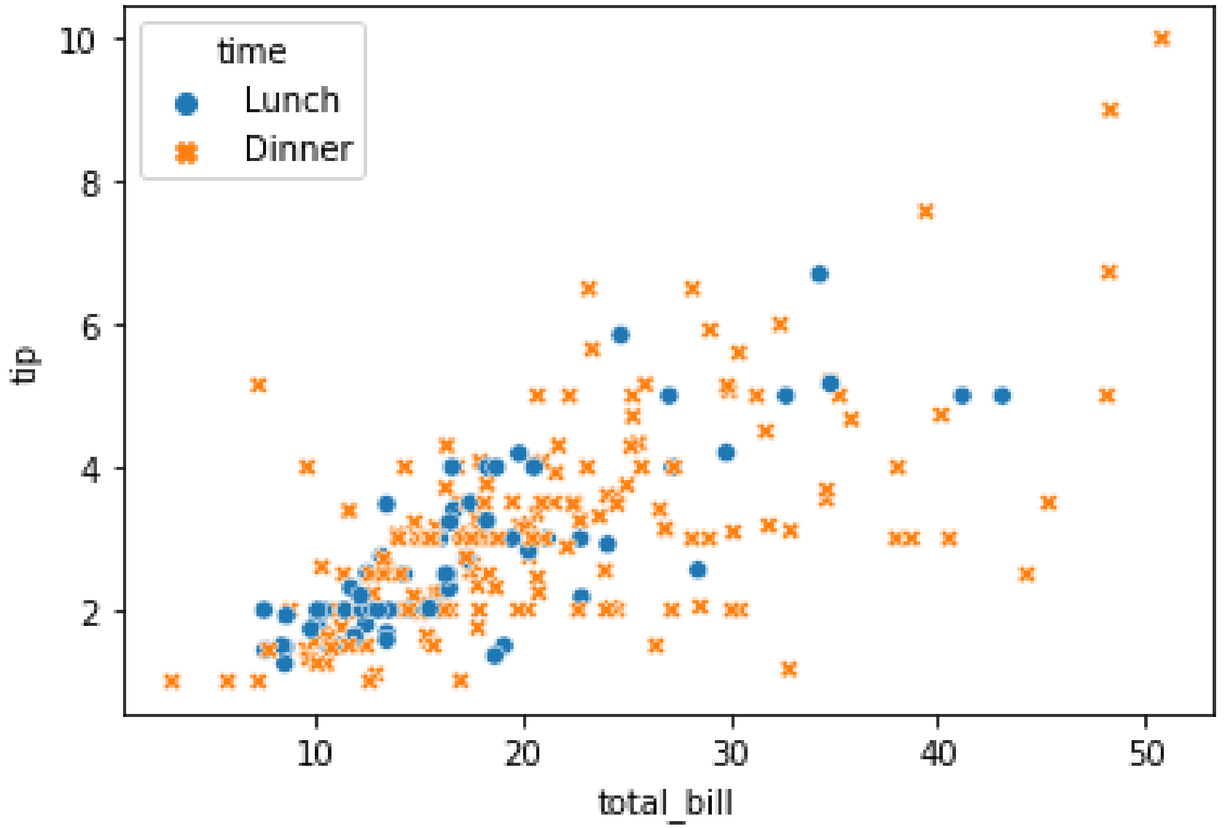
# sns.scatterplot

如果我們評論晚餐的小費可能更多，我們可能會顯示比較。

```
1 sns.scatterplot(data=tips, x="total_bill", y="tip")
```



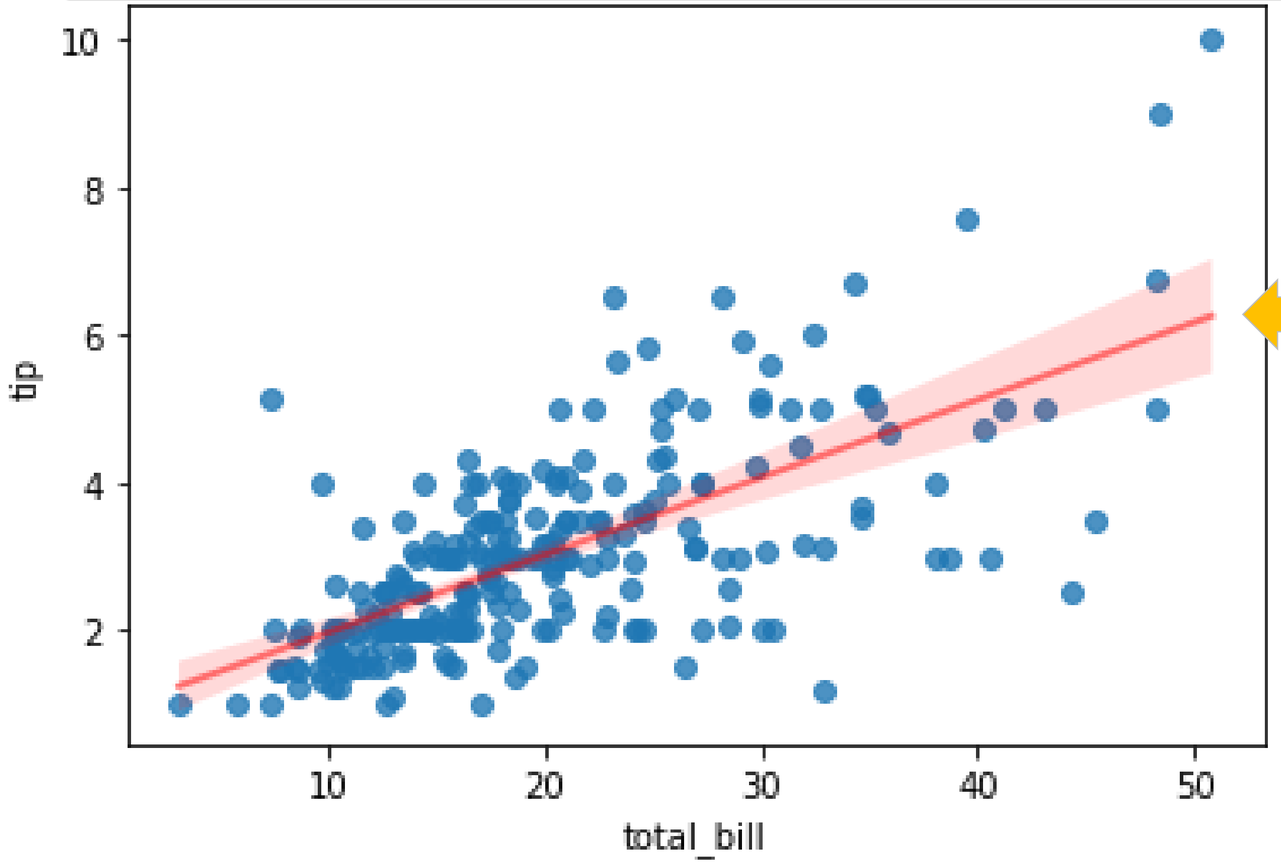

```
1 sns.scatterplot(data=tips, x="total_bill",
2                 y="tip", hue="time", style="time")
```



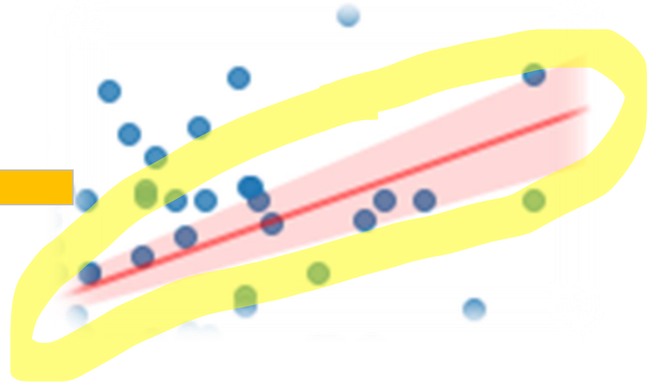
# 估計回歸擬合Estimate regression fits– sns.regplot

可用於視覺化線性擬合的兩個函數是： regplot() and lmplot().

```
1  sns.regplot(data=tips, x="total_bill", y="tip",
2              line_kws={"color":"r","alpha":0.5,"lw":2})
```




一般來說，我們會根據我們支付的帳單按比例支付小費。那麼它可能是一個線性回歸函數。



The light pink area is 95% confidence interval for that regression

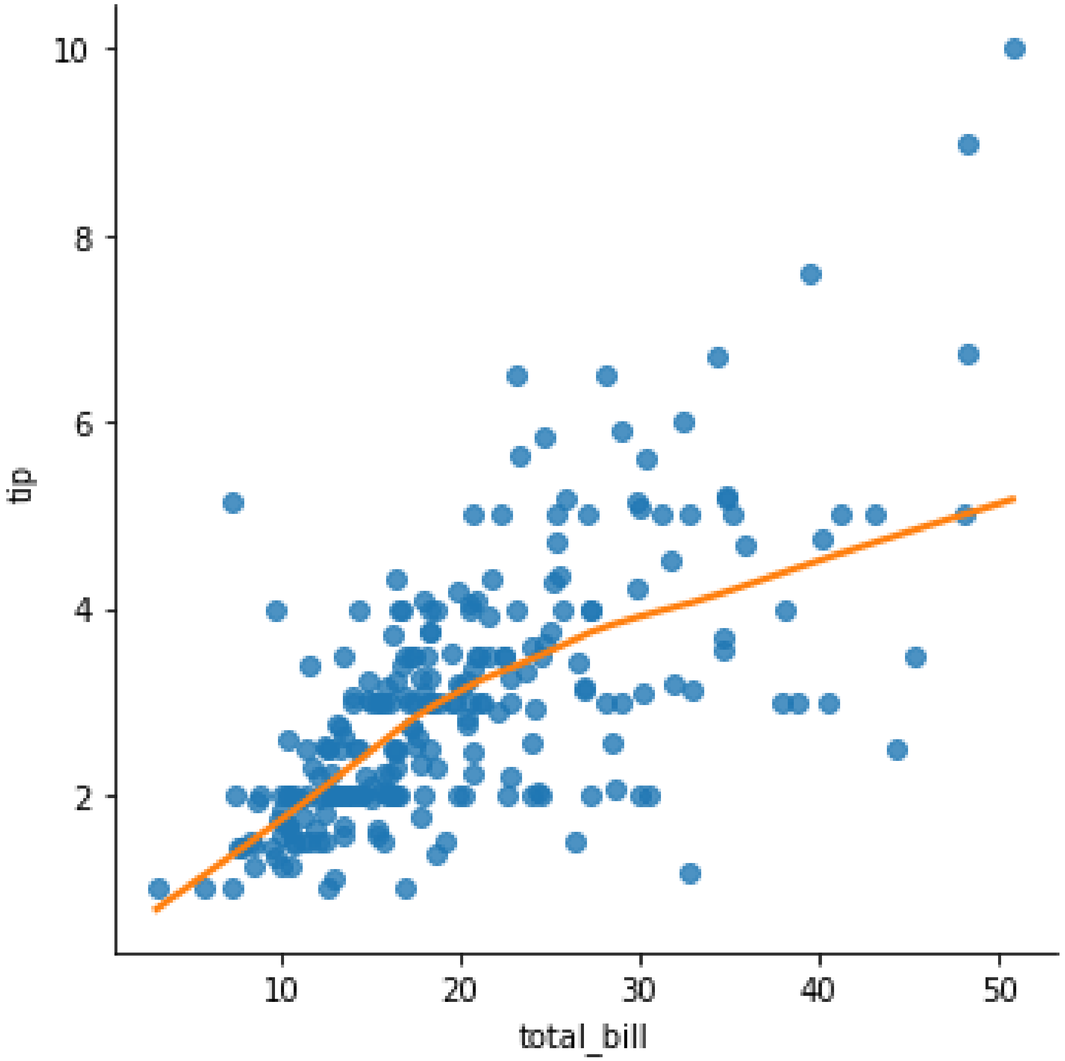
- `color` : color of the line
- `alpha` : opacity value of the line
- `lw` : line width

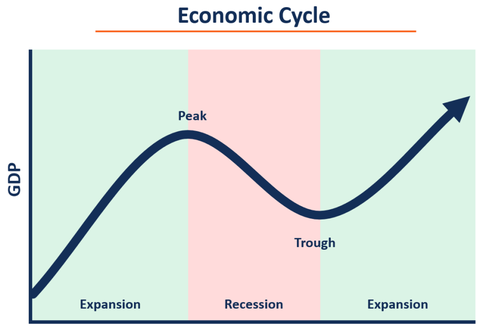
# 估計回歸擬合– sns.lmplot

```
sns.lmplot(data=tips, x="total_bill", y="tip",
           lowess=True, line_kws={"color": "C1"})
```

有時散點看起來不像線性回歸，它可能是非參數回歸。 然後你可以使用**sns.lmplt** 並設置

**lowess**=True



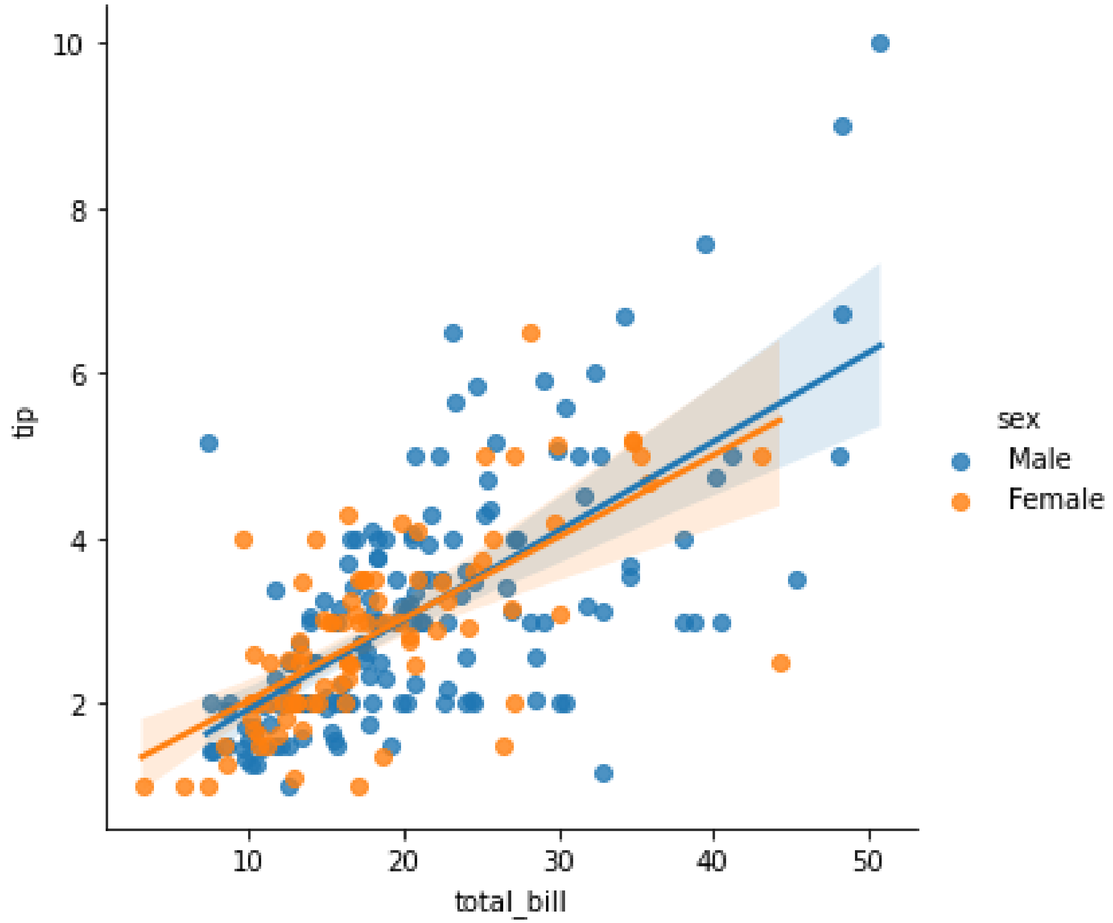




(think about econ cycle)

# 估計回歸擬合– sns.lmplot

```
1  sns.lmplot(x="total_bill", y="tip", hue="sex", data=tips)
```

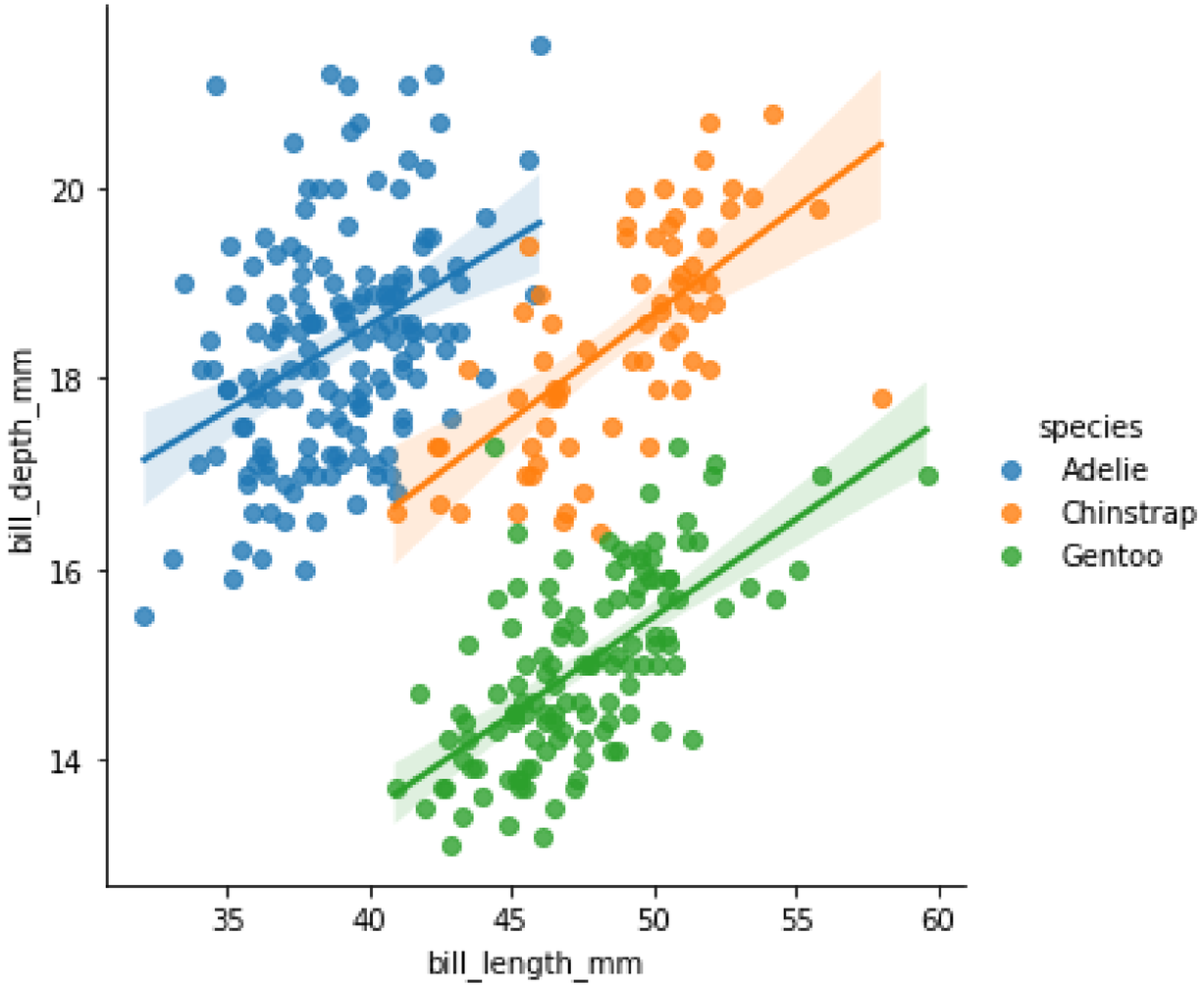
我們可以**hue** 用於區分回歸的特定變數。



# Estimate regression fits – sns.lmplot

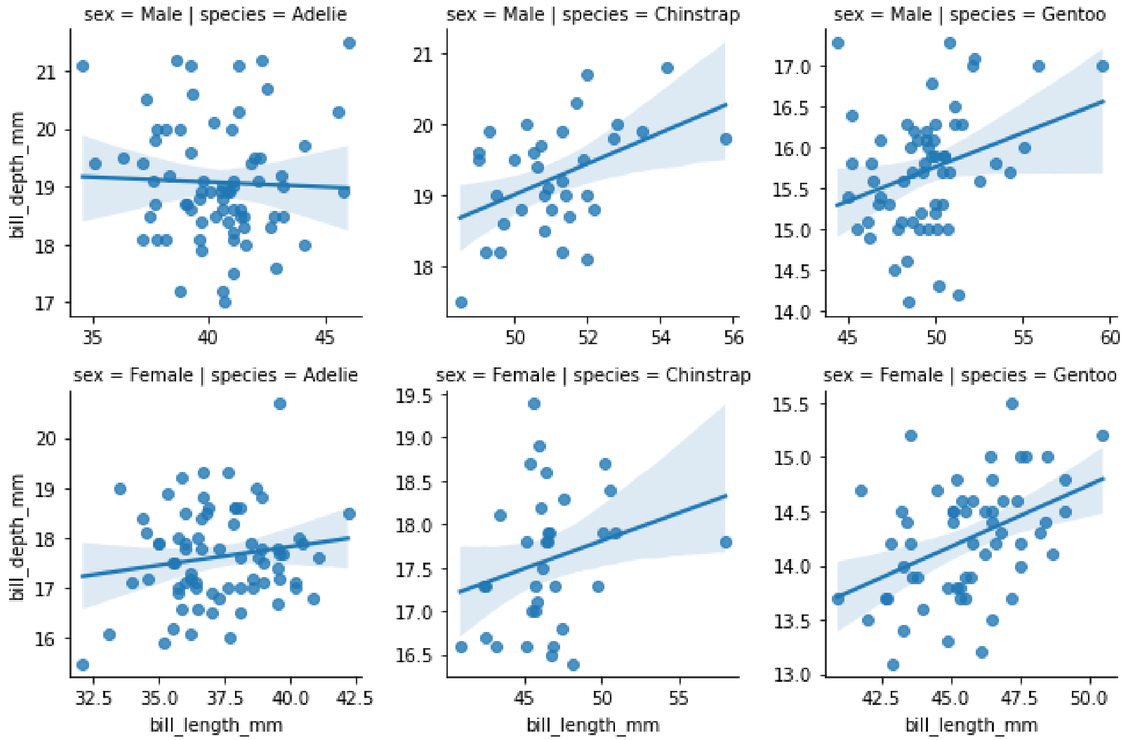
sns.lmplot is sns.regplot 在facet版本中，
意思是繪製多條回歸線。

```
1  sns.lmplot(data=penguins, x="bill_length_mm",
2             y="bill_depth_mm", hue="species")
```



# 估計回歸擬合– sns.lmplot

```python
sns.lmplot(
    data=penguins, x="bill_length_mm", y="bill_depth_mm",
    col="species", row="sex", height=3,
    facet_kws=dict(sharex=False, sharey=False),
)
```




我們可以在分面網格中繪製以可視化每個變數。

# Scatterplot 具有不同的點大小和色調sizes and hue

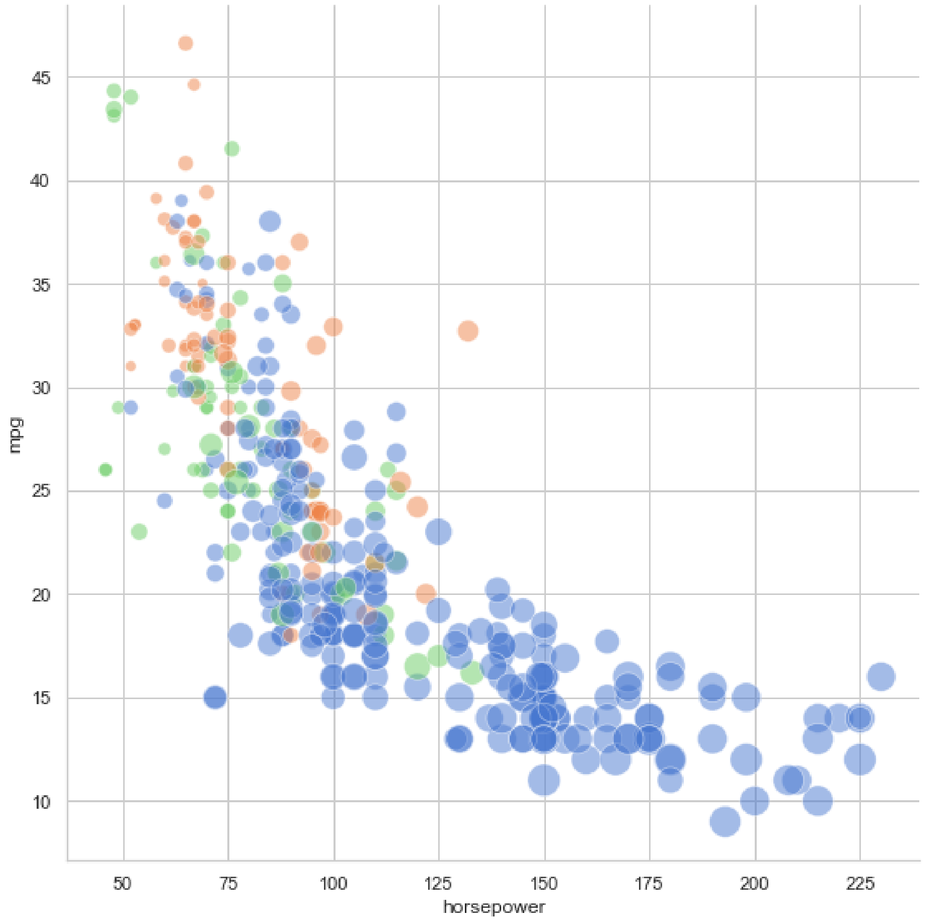
## Loading Miles Per Gallon (mpg) of vehicle origin and weight

```
1 mpg = sns.load_dataset("mpg")
2 mpg.sample(5)
```

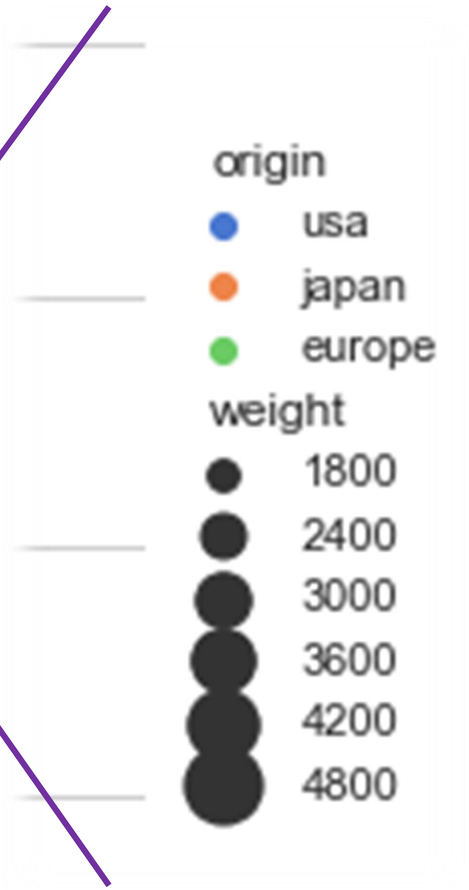
| | mpg | cylinders | displacement | horsepower | weight | acceleration | model_year | origin | name |
|---|---|---|---|---|---|---|---|---|---|
| **58** | 25.0 | 4 | 97.5 | 80.0 | 2126 | 17.0 | 72 | usa | dodge colt hardtop |
| **364** | 26.6 | 8 | 350.0 | 105.0 | 3725 | 19.0 | 81 | usa | oldsmobile cutlass ls |
| **288** | 18.2 | 8 | 318.0 | 135.0 | 3830 | 15.2 | 79 | usa | dodge st. regis |
| **276** | 21.6 | 4 | 121.0 | 115.0 | 2795 | 15.7 | 78 | europe | saab 99gle |
| **188** | 16.0 | 8 | 318.0 | 150.0 | 4190 | 13.0 | 76 | usa | dodge coronet brougham |

# Scatterplot 具有不同的點大小和色調

```
# Plot miles per gallon against horsepower with other semantics
sns.relplot(x="horsepower", y="mpg", hue="origin", size="weight",
            sizes=(40, 400), alpha=.5, palette="muted",
            height=8, data=mpg)
```





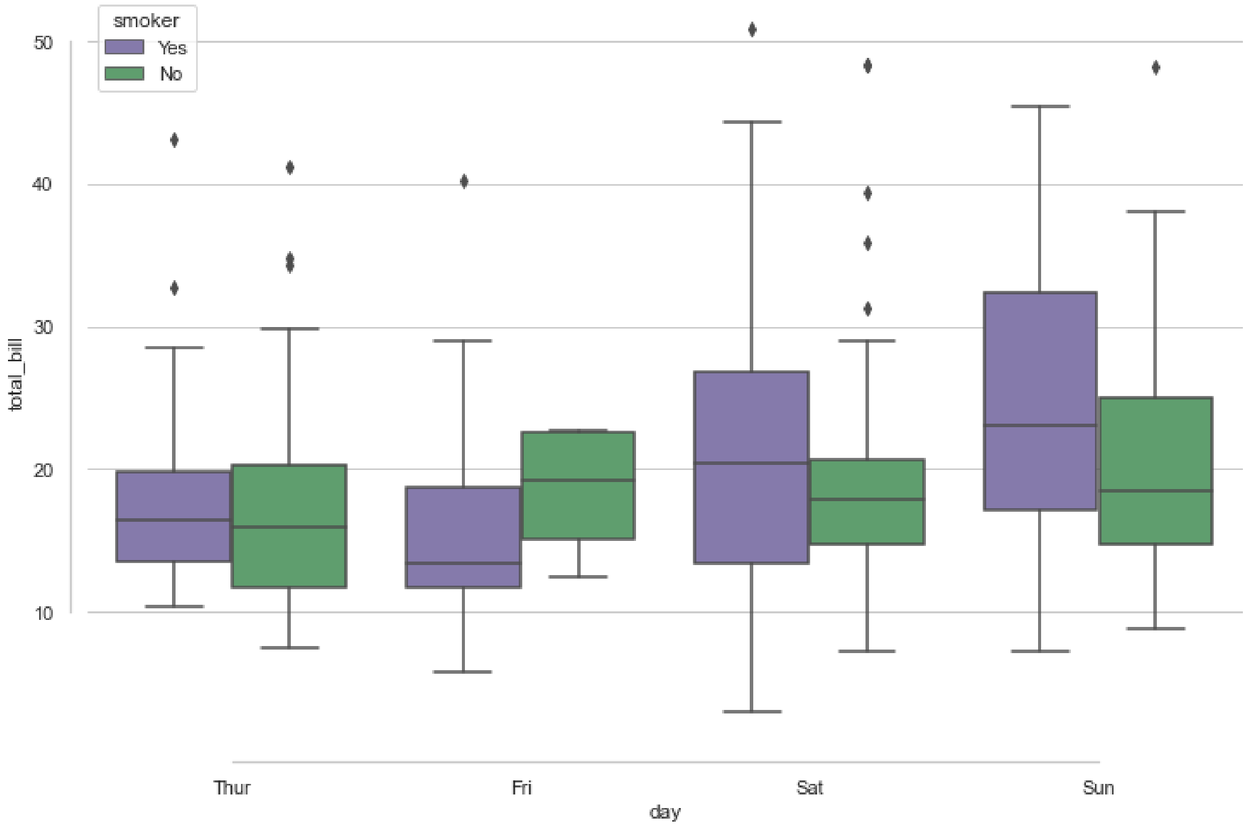



從圖中我們可以得出結論，美國車輛的重量更重。日本和歐洲的能源效率更高。

# Sns.boxplot

除了使用散點圖來揭示數據的分佈外，還可以使用箱形圖。該框顯示數據集的四分位數，而晶須延伸以顯示分佈的其餘部分。

```
sns.boxplot(data=tips, x="day", y="total_bill",
                hue="smoker", palette=["m", "g"],)
sns.despine(offset=10, trim=True)
```





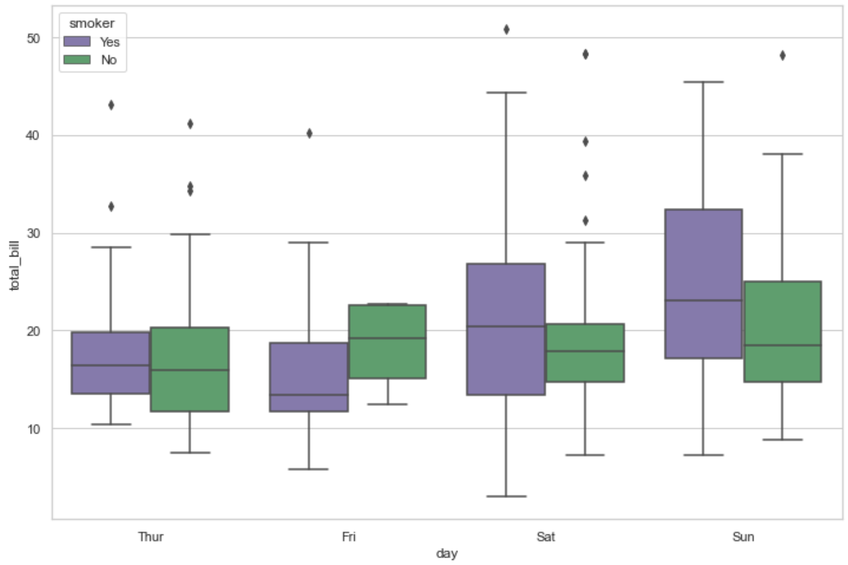
繪製吸煙者和非吸煙者在週四/週五/週六/周日的總帳單分佈情況

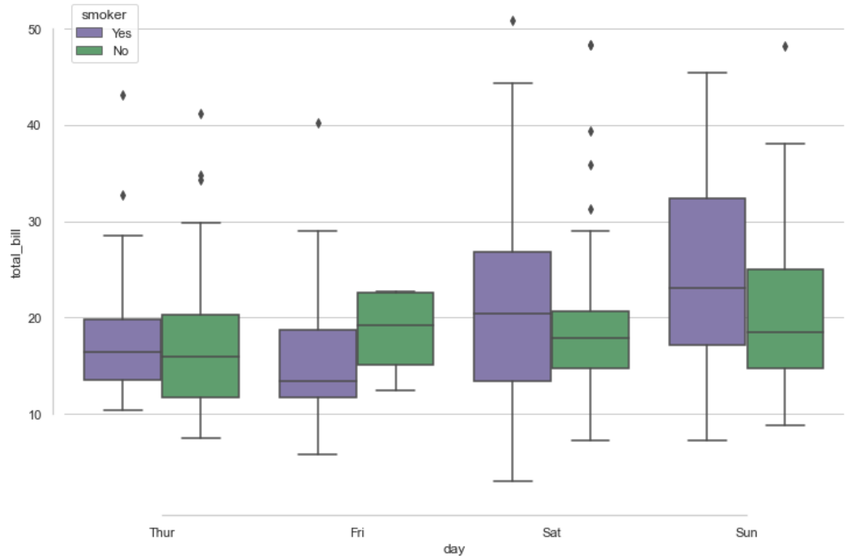
# sns.despine

## seaborn.despine

```
seaborn.despine(fig=None, ax=None, top=True, right=True, left=False,
bottom=False, offset=None, trim=False)
```

拿掉頂部和右側的框線(border)

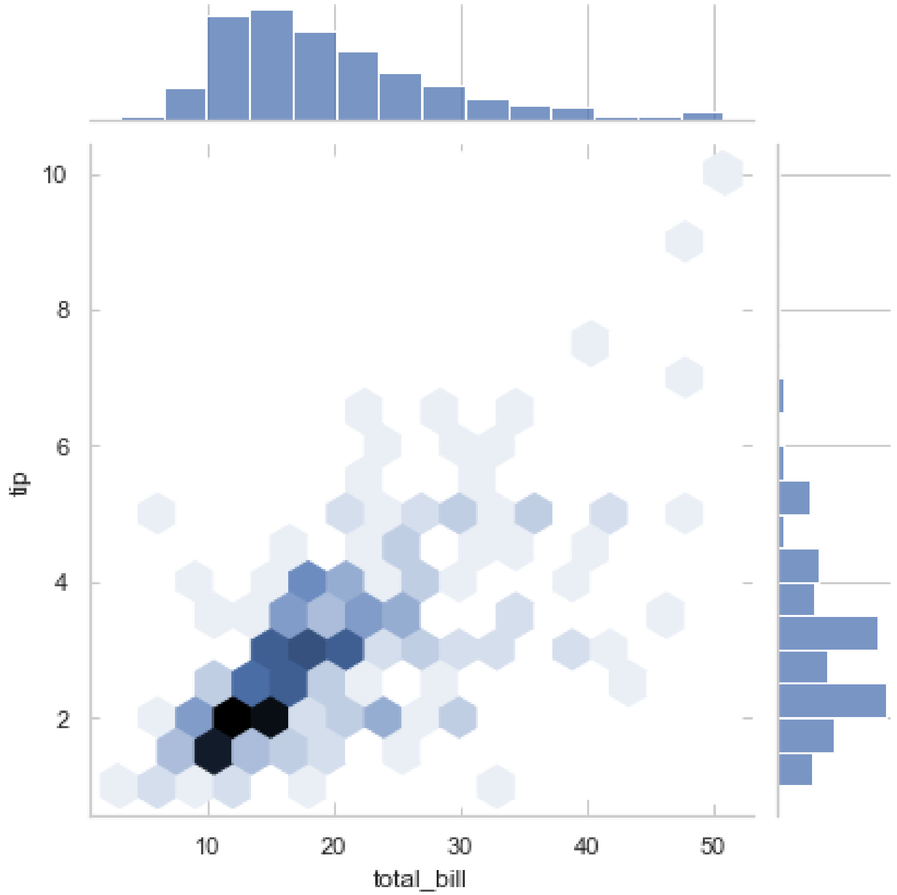



sns.despine()

# sns.jointplot

```
1  sns.jointplot(data=tips, x="total_bill", y="tip", kind='hex')
```



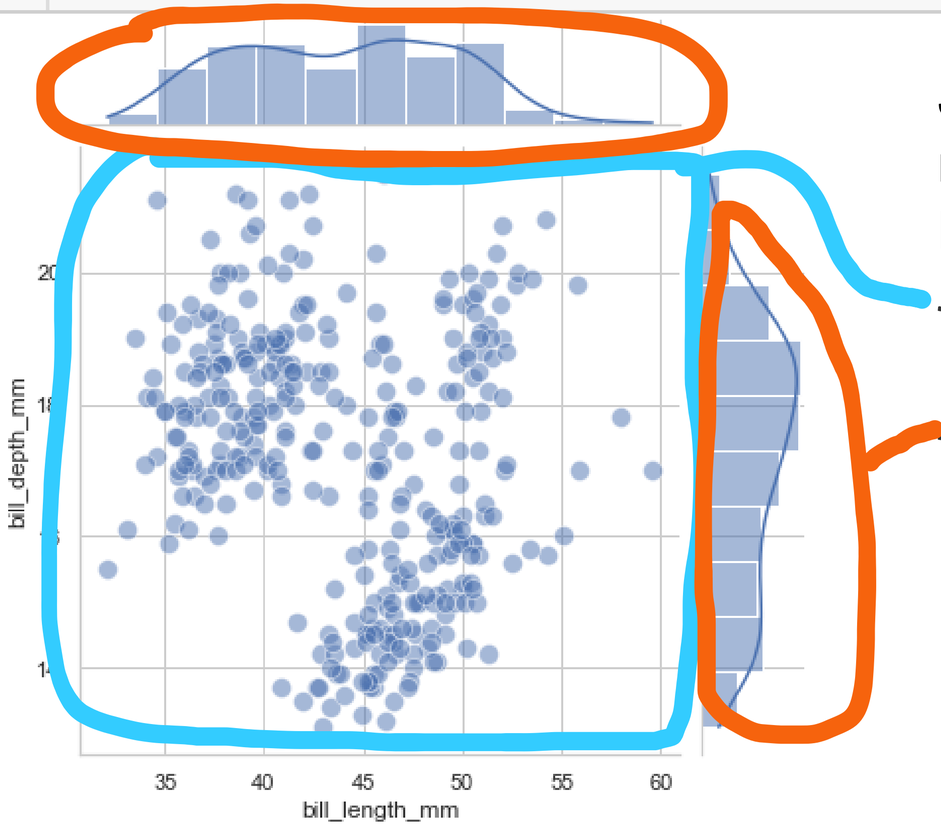

您可以繪製兩個定量數據，並在一個圖中查看它們的分佈。用：sns.jointplot 或 sns.JointGrid

**kind** : *{ "scatter" | "kde" | "hist" | "hex" | "reg" | "resid" }*

Kind of plot to draw. See the examples for references to the underlying functions.

# sns.JointGrid

```
1  sns.set_theme(style="whitegrid")
2  g = sns.JointGrid(data=penguins, x="bill_length_mm", y="bill_depth_mm")
3  g.plot_joint(sns.scatterplot, s=100, alpha=.5)
4  g.plot_marginals(sns.histplot, kde=True)
```




JointGrid : Grid for drawing a bivariate plot with marginal univariate plots. 用於繪製具有邊際單變數圖的二元圖的網格。

JointGrid.plot_joint: caller of JointGrid

JointGrid.plot_marginals: caller of JointGrid

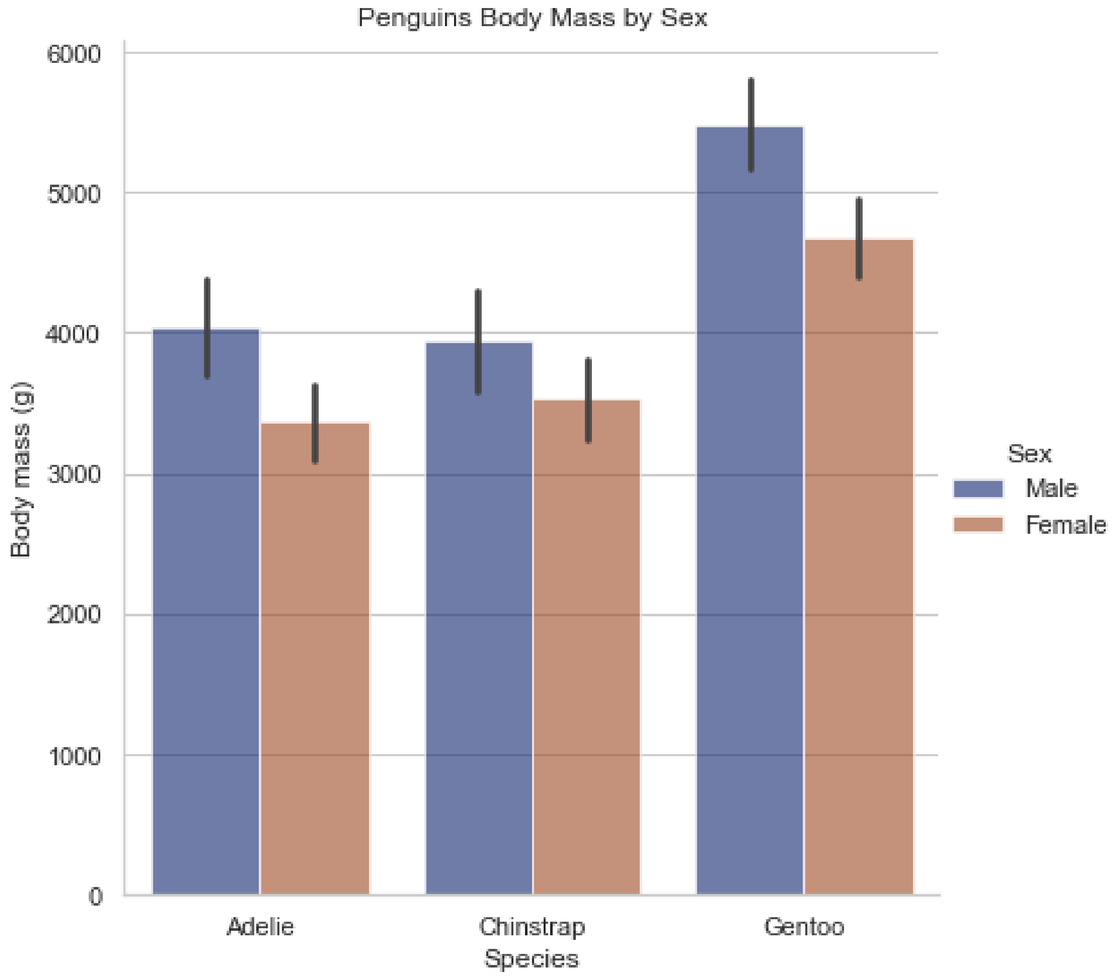
# sns.catplot 和定製 title, axis label, legend

sns.catplot 提供軸級函數，顯示categories variables

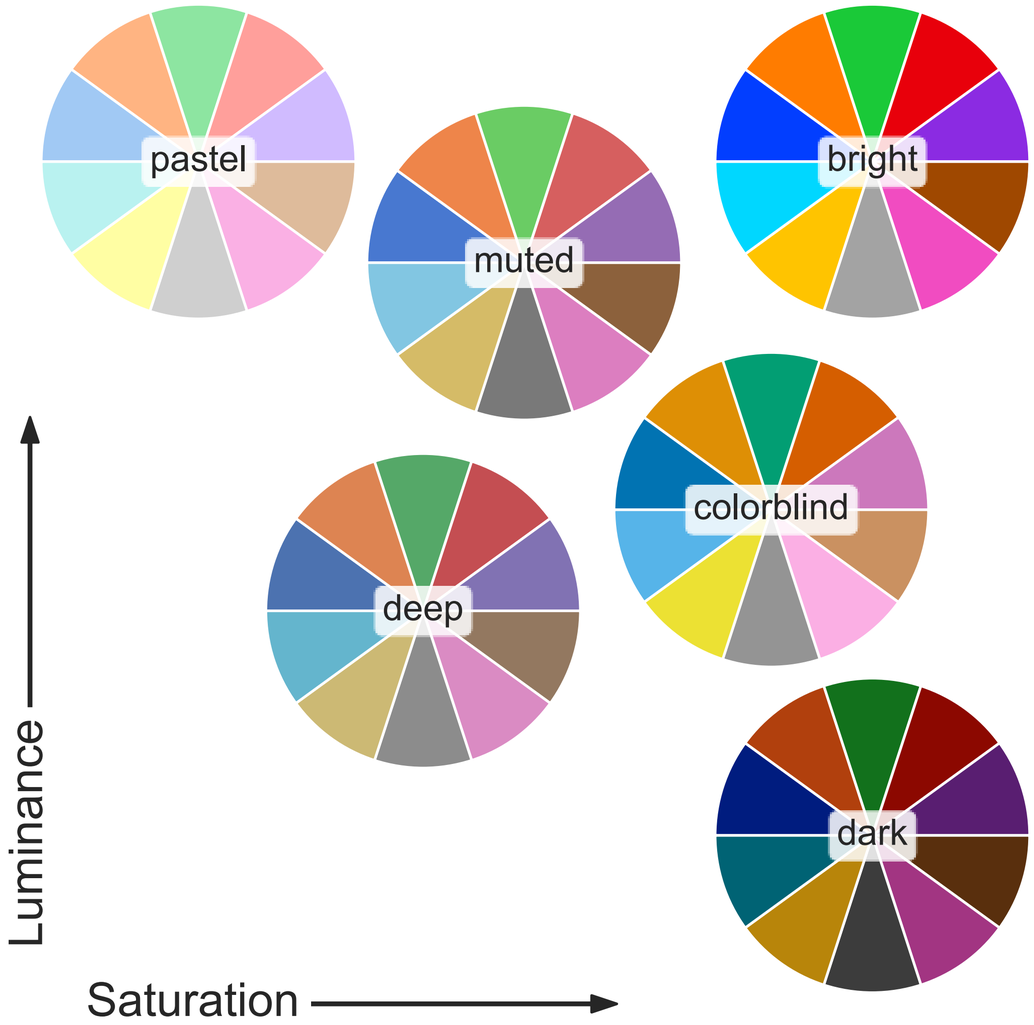
```
sns.set_theme(style="whitegrid")

g = sns.catplot(
    data=penguins, kind="bar",
    x="species", y="body_mass_g", hue="sex",
    errorbar="sd", palette="dark", alpha=.6, height=6
)
g.set_axis_labels("Species", "Body mass (g)")
g.legend.set_title("Sex")
g.set(title="Penguins Body Mass by Sex")
```

Title, label, and legend 可以在繪圖物件上設置。

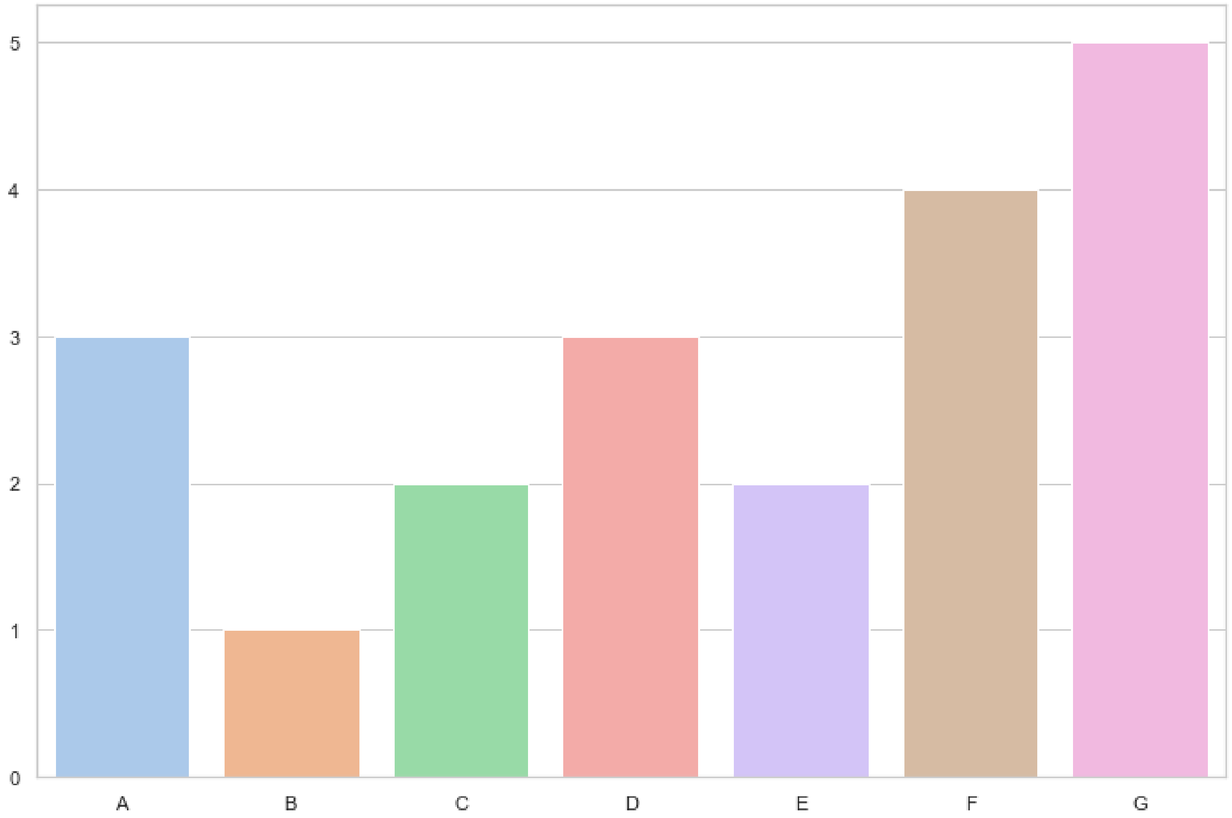


# Colour Palettes調色板




你選擇一個調色板，Seaborn做剩下的分配事情

```
1  sns.set_style("whitegrid")
2  sns.barplot(x=["A","B","C","D","E","F","G"],
3              y=[3,1,2,3,2,4,5], palette="pastel")
```



# sns.set_style

```
1 sns.set_style("darkgrid", rc={"grid.color": ".6",
2                               "grid.linestyle": ":",
3                               'figure.figsize':(12,8)})
4 sns.histplot(data=penguins, x="flipper_length_mm", kde=True)
```

## seaborn.set_style

`seaborn.set_style(style=None, rc=None)`

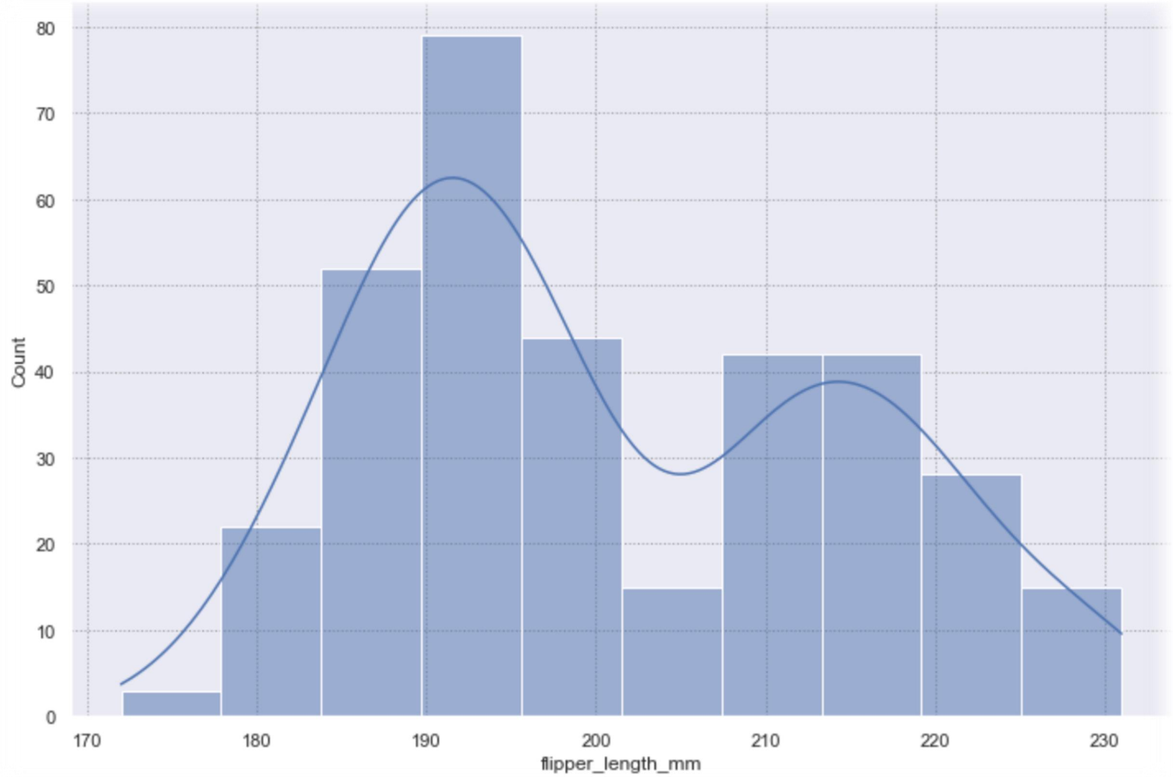
**Parameters:** **style** : *dict, or one of {darkgrid, whitegrid, dark, white, ticks}*

A dictionary of parameters or the name of a preconfigured style.

**rc** : *dict, optional*

Parameter mappings to override the values in the preset seaborn style dictionaries. This only updates parameters that are considered part of the style definition.

常用的樣式設置可能是figsize and 圖形 background 顏色.



# Matplotlib vs Seaborn

| Features | matplotlib | seaborn |
|---|---|---|
| Syntax | Complex and lengthy | Comparatively simple and easy to learn |
| Functionality | Utilized for basic and varying graphs | Fascinating theme with numbers of patterns |
| Visualization | Well connected with Numpy and Pandas | Good for Pandas DF and provide pretty graphics |
| Multiple Fig. | Use multiple figures simultaneously | Multi figs available but could run out of memory |
| Flexibility | Highly customised and robust | Avoid overlapping plots with default themes |
| DF and Array | Treat figures and axes as objects. Stateful API for plotting | Functional and organized. Treat whole DF as single unit. But parameters required for method |
| Use Case | Plot various graphs using Pandas and Numpy | Extended version on Matplotlib. User friendly |

# 參考文件

Official website:
https://seaborn.pydata.org/

Seaborn project source code:
https://github.com/mwaskom/seaborn

Textbook:
Python Data Science Handbook, Jake VanderPlas, O'Reilly