SQL Server 資料分析大躍進

Machine Learning Services

講者：許致學



MVC

歡迎參加我們的每週四固定聚會

Microsoft SQL Server

python™

SQL Server 資料分析大躍進
Machine Learning Services

資深顧問 許致學

Microsoft CERTIFIED Trainer
Systems Engineering
Systems Administration
Database Administration
Database Development

MVP Microsoft Most Valuable Professional

PMI PMP

ITIL



SQL Server 資料分析大躍進

Machine Learning Services



- SQL Server Machine Learning Services 簡介與組態
- 跨平台多語言新工具 - Azure Data Studio
- 用 Python 處理 SQL Server 資料
- SQL Server 資料庫的查詢結果插入 Python pandas 資料框架進行統計分析
- SQL Server 資料庫的查詢結果繪製 Python Matplotlib 統計圖表

SQL 依舊佔據霸主地位

TechOrange 科技報橘

https://buzzorange.com/techorange/2021/03/26/sql-importance/

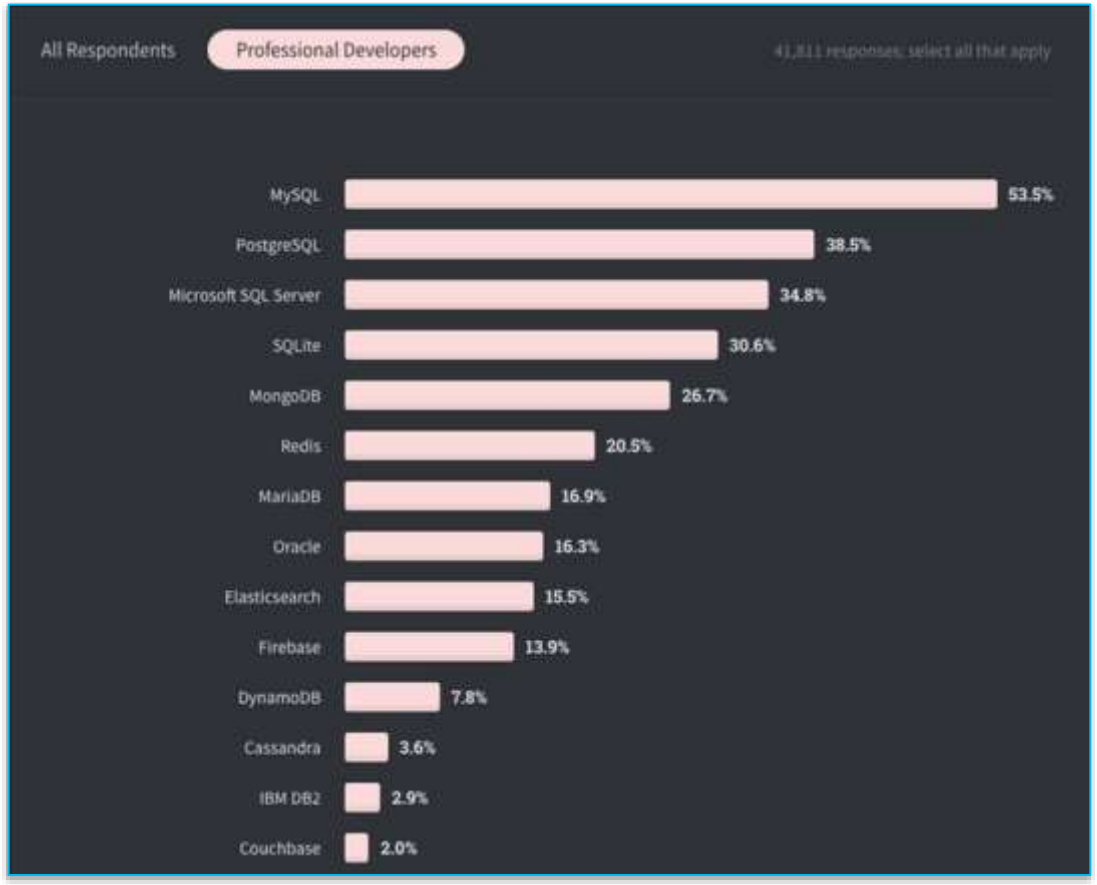


1. 資料處理能力
2. 實踐檢驗，被證明是可信賴的
3. 知識和社群
4. 簡單
5. 高度普及
6. 開源及互操作性
7. 能用 SQL 誰還寫程式碼？
8. SQL/RDBMS 和 NoSQL/DBMS 資料庫各司其職

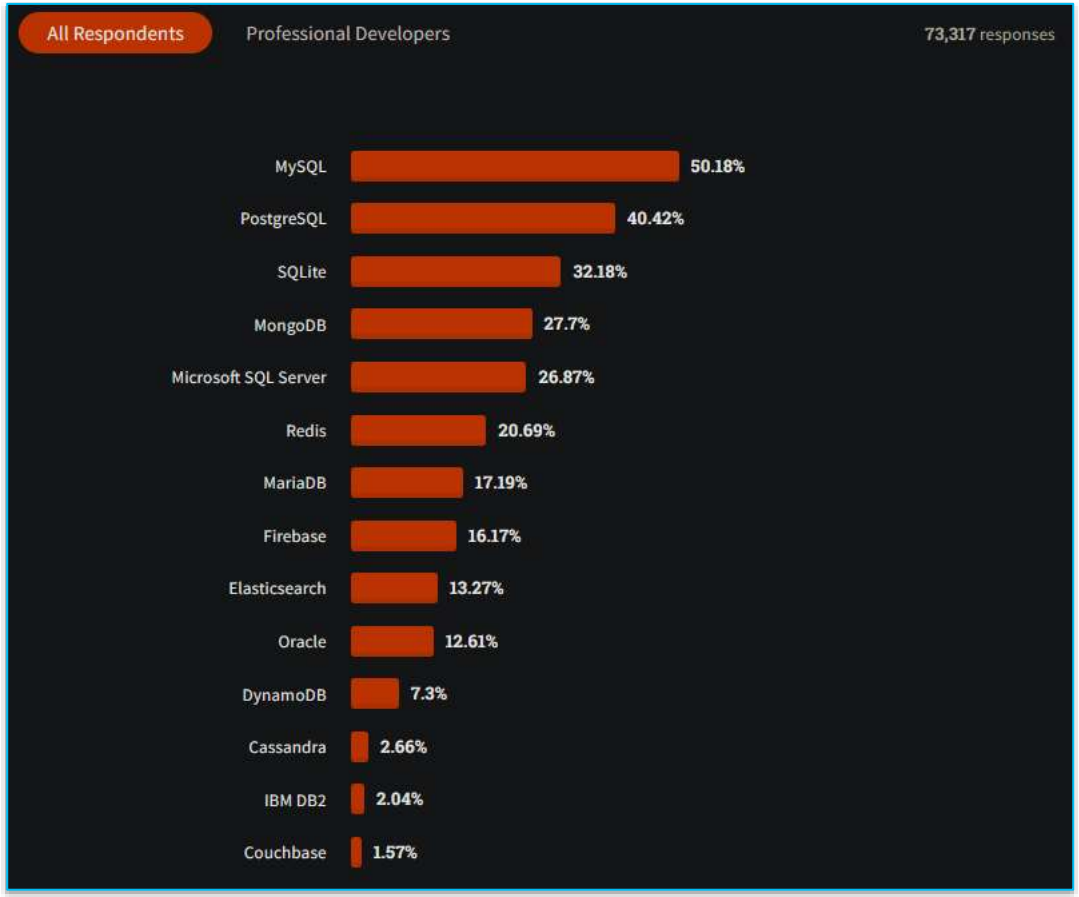
SQL 依舊佔據霸主地位

Stack Overflow Developer Survey

2020



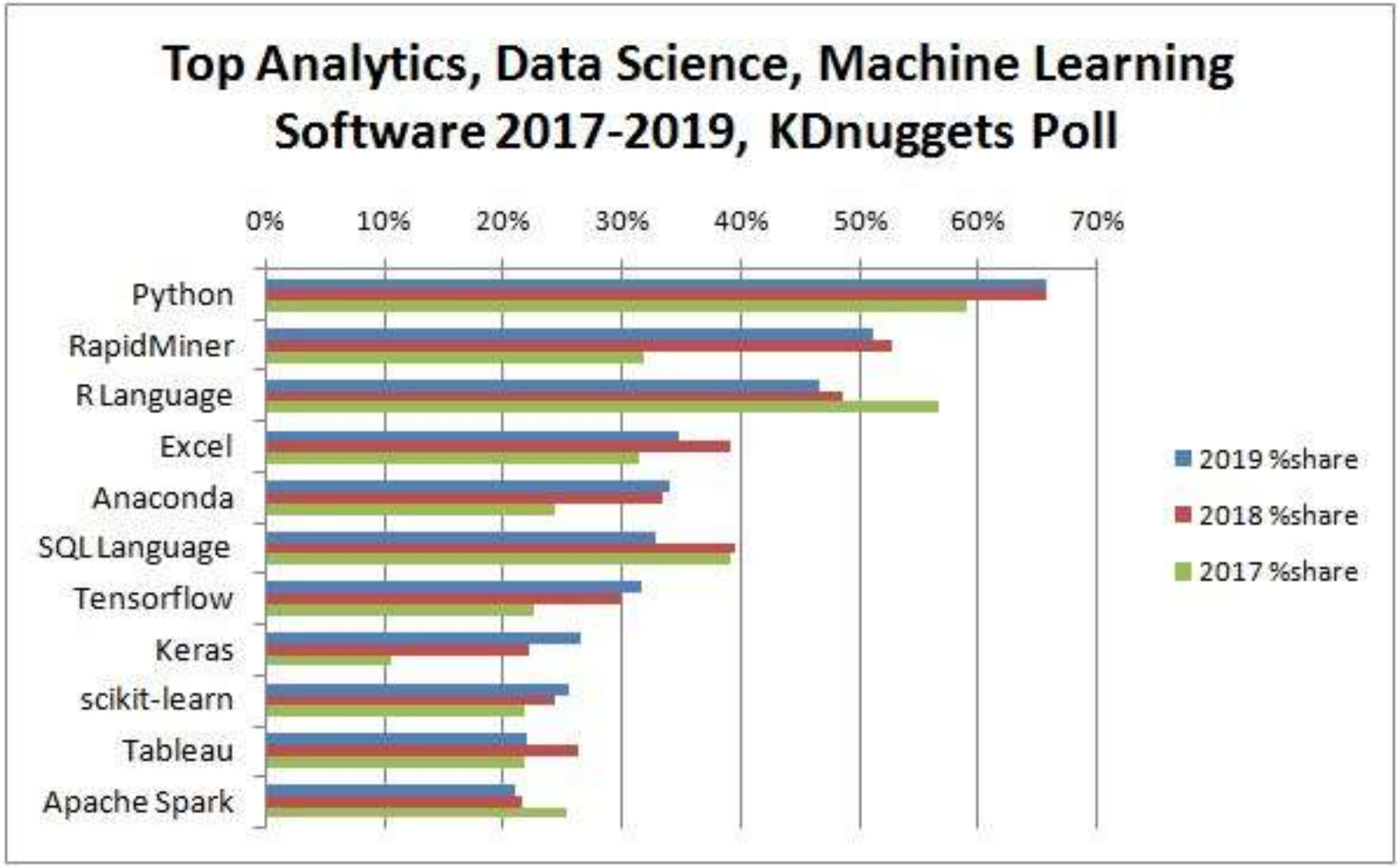
2021




WHY?


python™

資料科學和機器學習的軟體

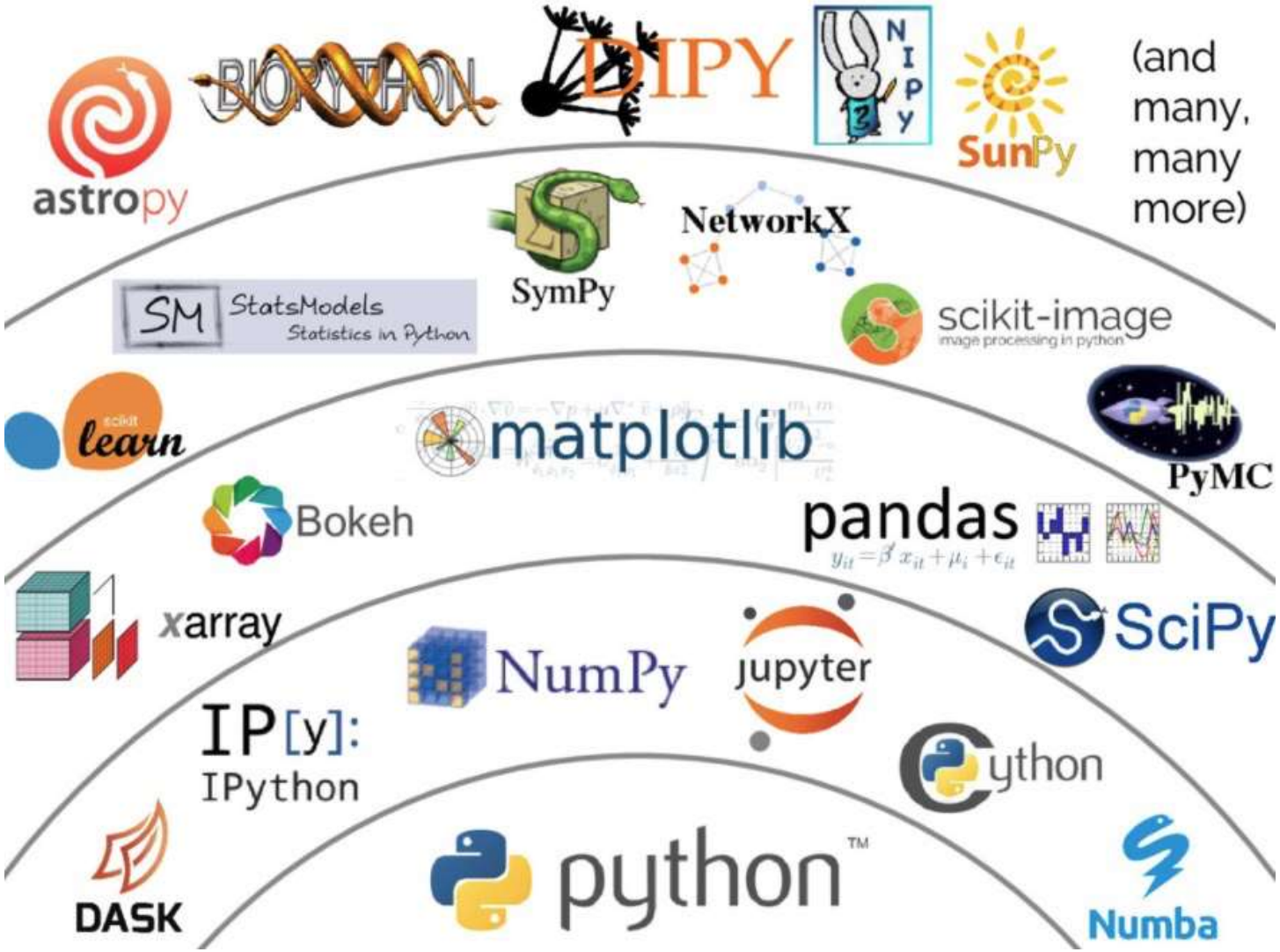


https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html

SQL Server 整合 Machine Learning Services



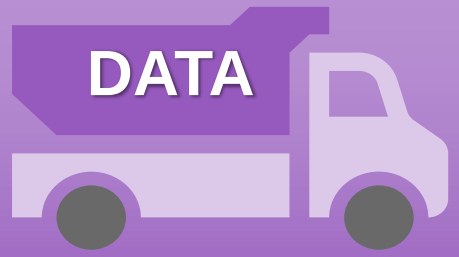
Python 已成為處理資料的重要語言



https://speakerdeck.com/jakevdp/the-unexpected-effectiveness-of-python-in-science?slide=52

SQL Server 整合 Machine Learning Services

- 直接利用資料庫內的資料進行分析計算，減少或消除資料移動



- 呼叫 T-SQL 預存程序，運用眾多R/Python套件快速建構與管理機器學習模型

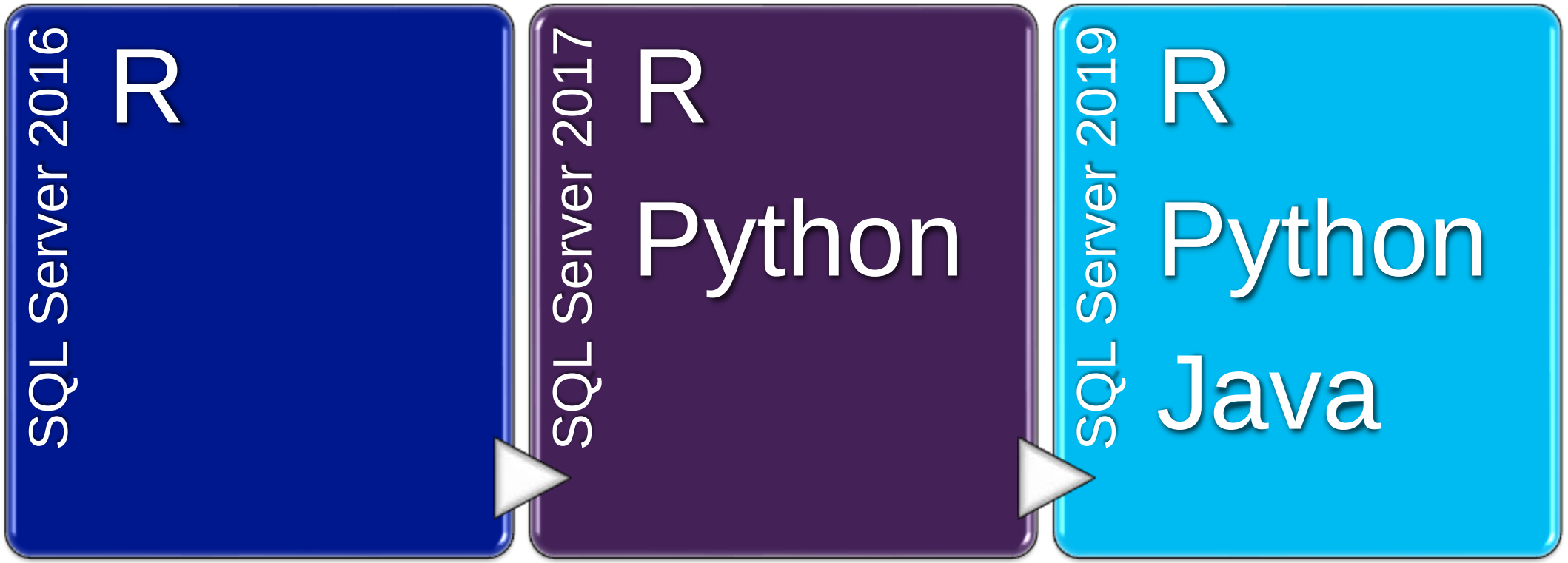


- 多執行序與平行處理
- 整合安全性與高可用性



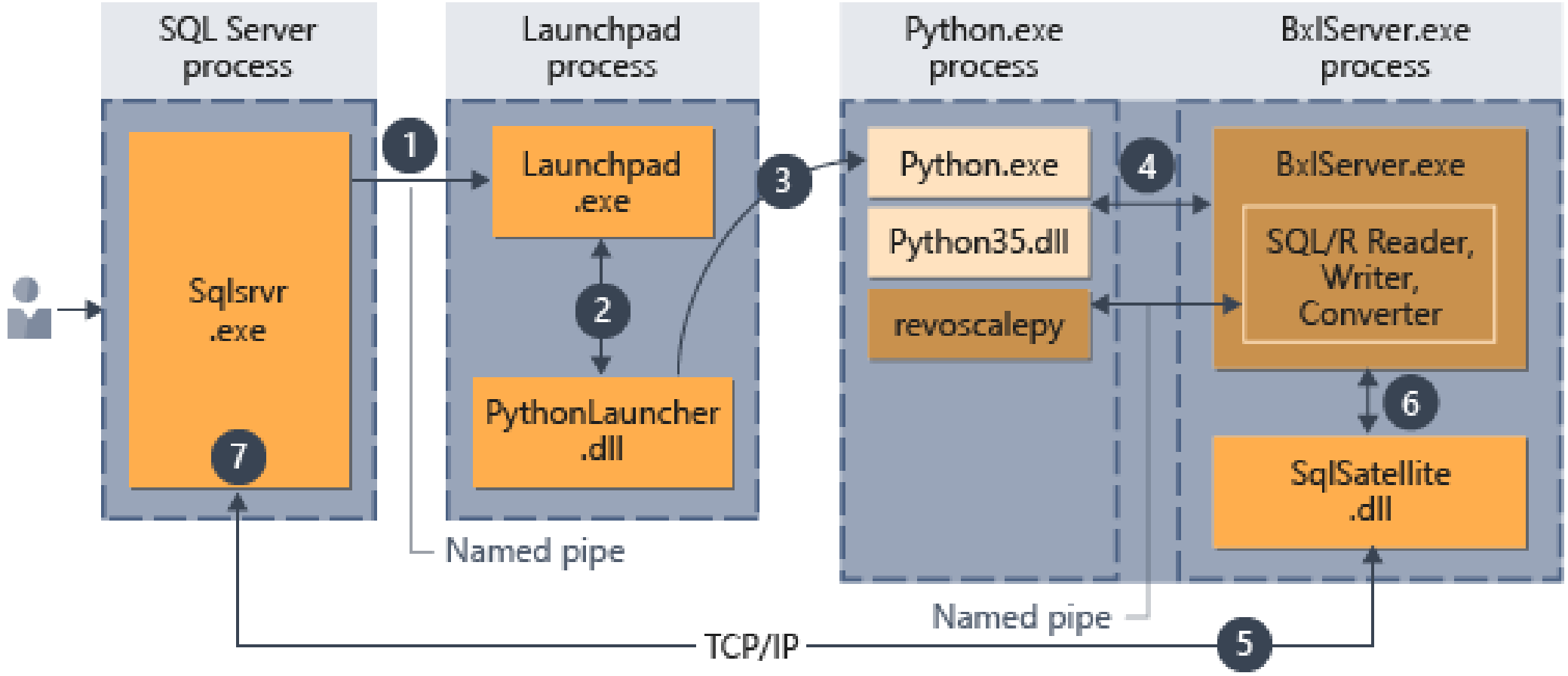
SQL Server 機器學習服務

版本演進



SQL Server 機器學習服務

Python

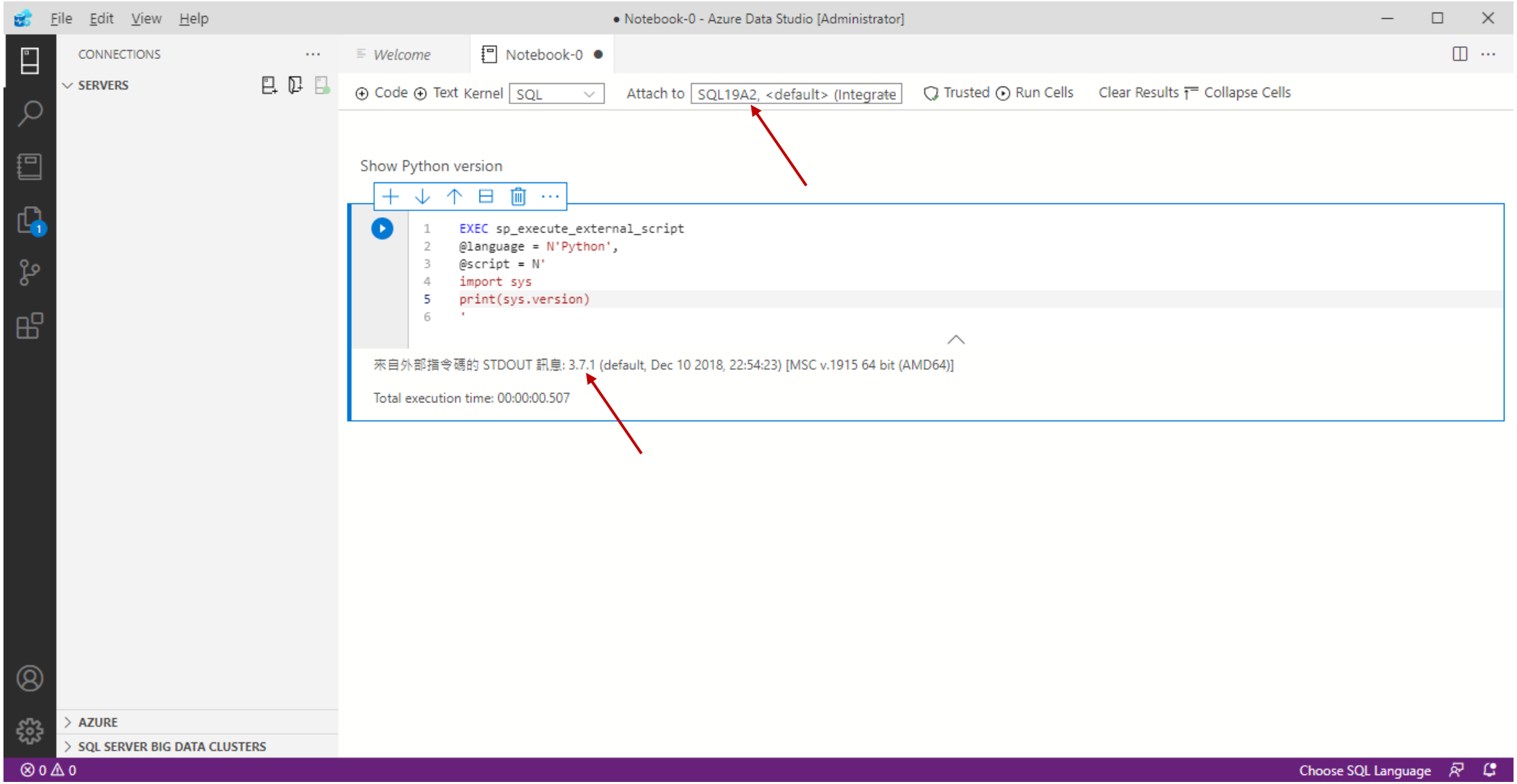


SQL Server 機器學習服務 Python 語言延伸模組

- Python 執行環境
- 包含 Python 3.5 執行階段和解譯器的 Anaconda 4.2 發行版本
- 標準程式庫與工具
- Microsoft Python 套件
  - 大規模分析的 RevoScalePy
  - 機器學習演算法的 microsoftml

SQL Server 機器學習服務 Python 語言延伸模組

Python 版本



資料分析程序


Data Source
Extract Transform Load
Operational Data Store
Data Warehouse
Data Marts
Reports
JSON
XML
ODS
Staging

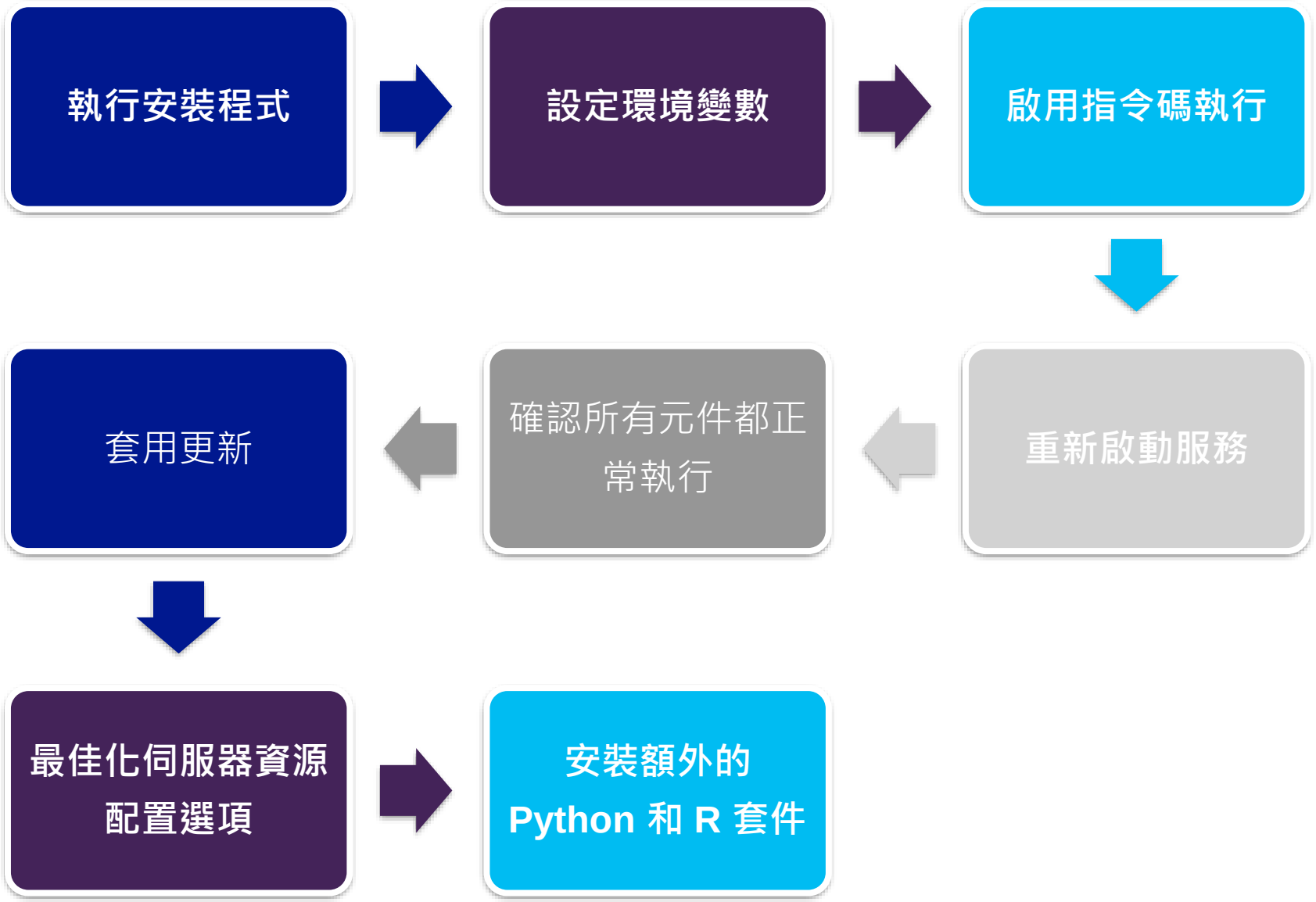
該開始學習 SQL Server 機器學習服務嗎？

- 資料庫管理人員
- 資料庫開發人員
- 資料分析人員
- 資料科學家



安裝 SQL Server 機器學習服務

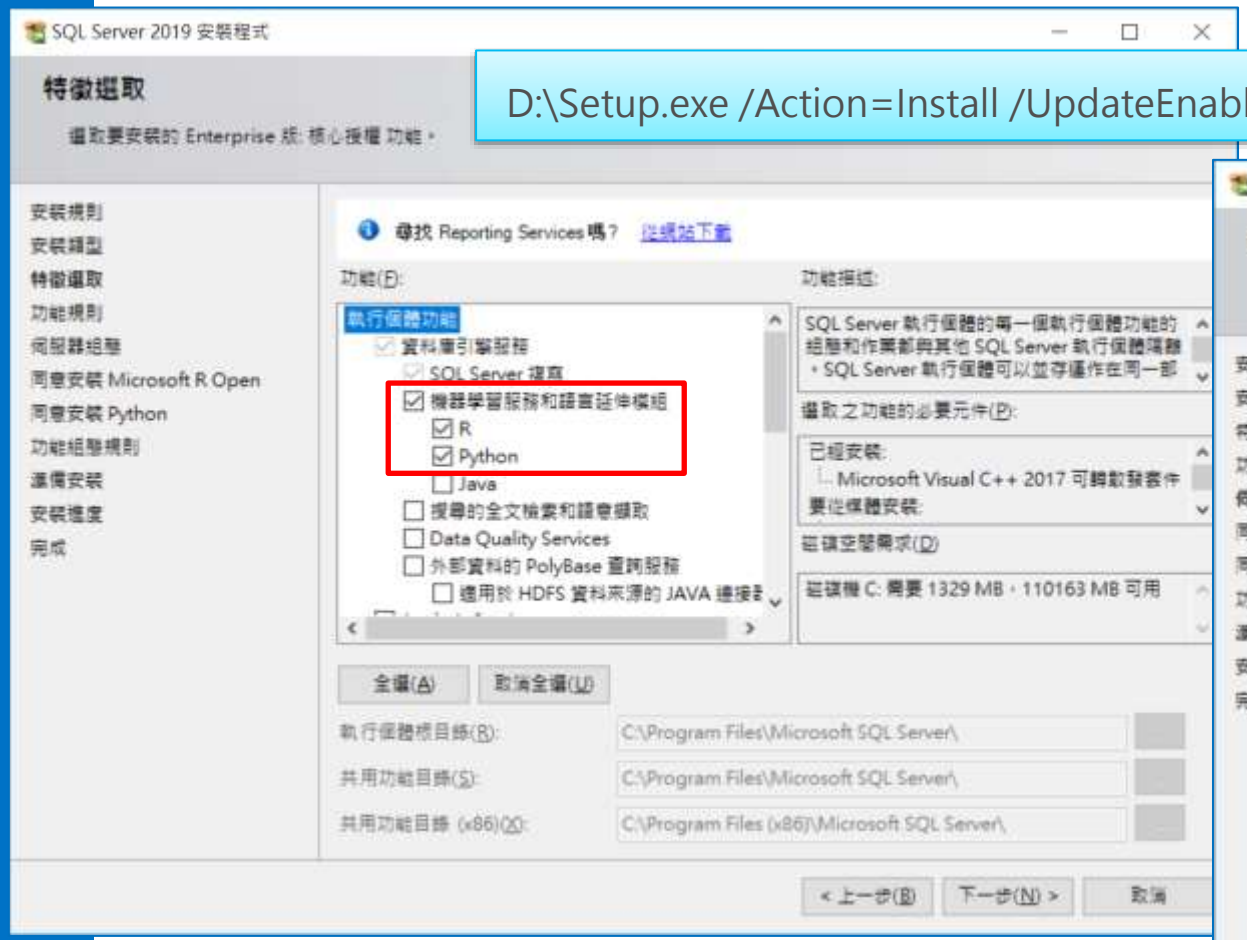
安裝 SQL Server 機器學習服務

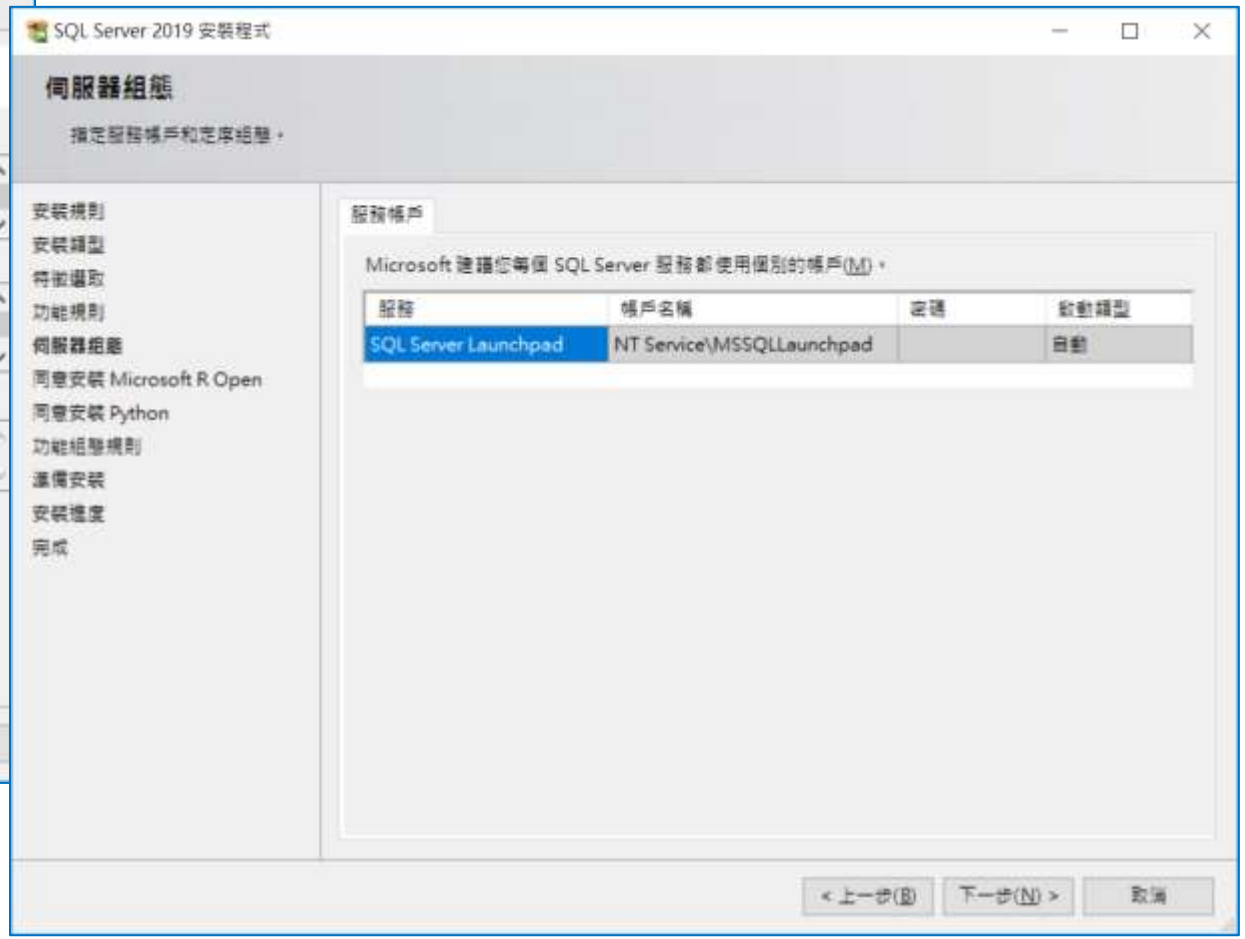


SQL Server 機器學習服務

安裝

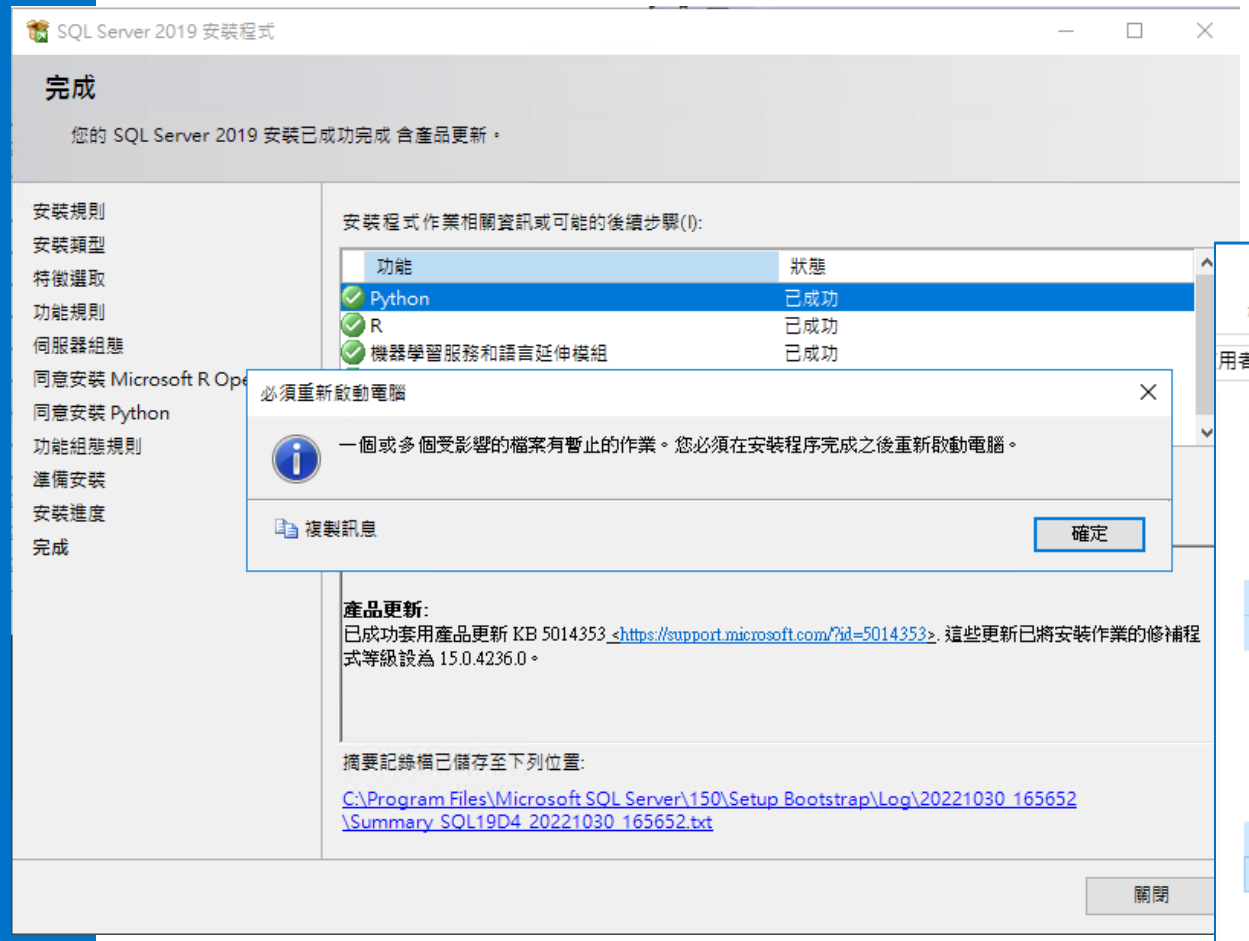
D:\Setup.exe /Action=Install /UpdateEnabled=TRUE /UpdateSource="C:\CU"

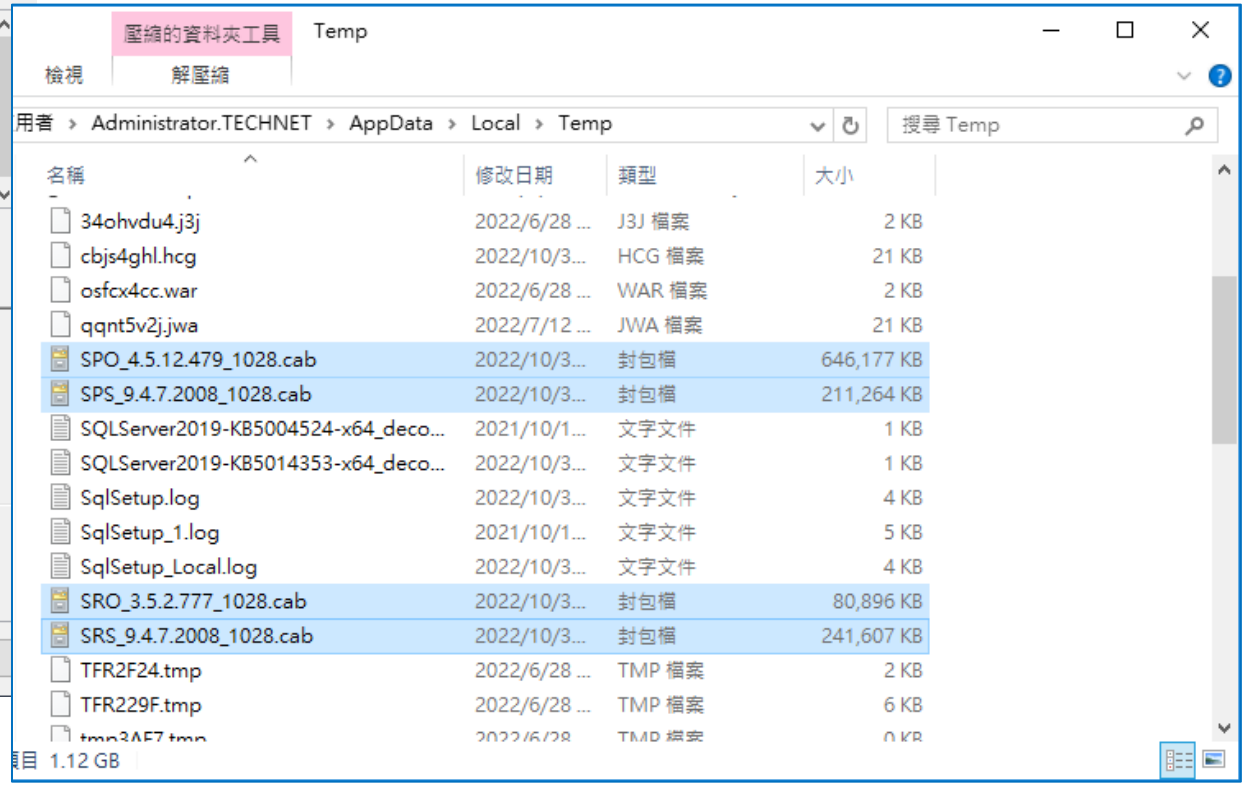




SQL Server 機器學習服務

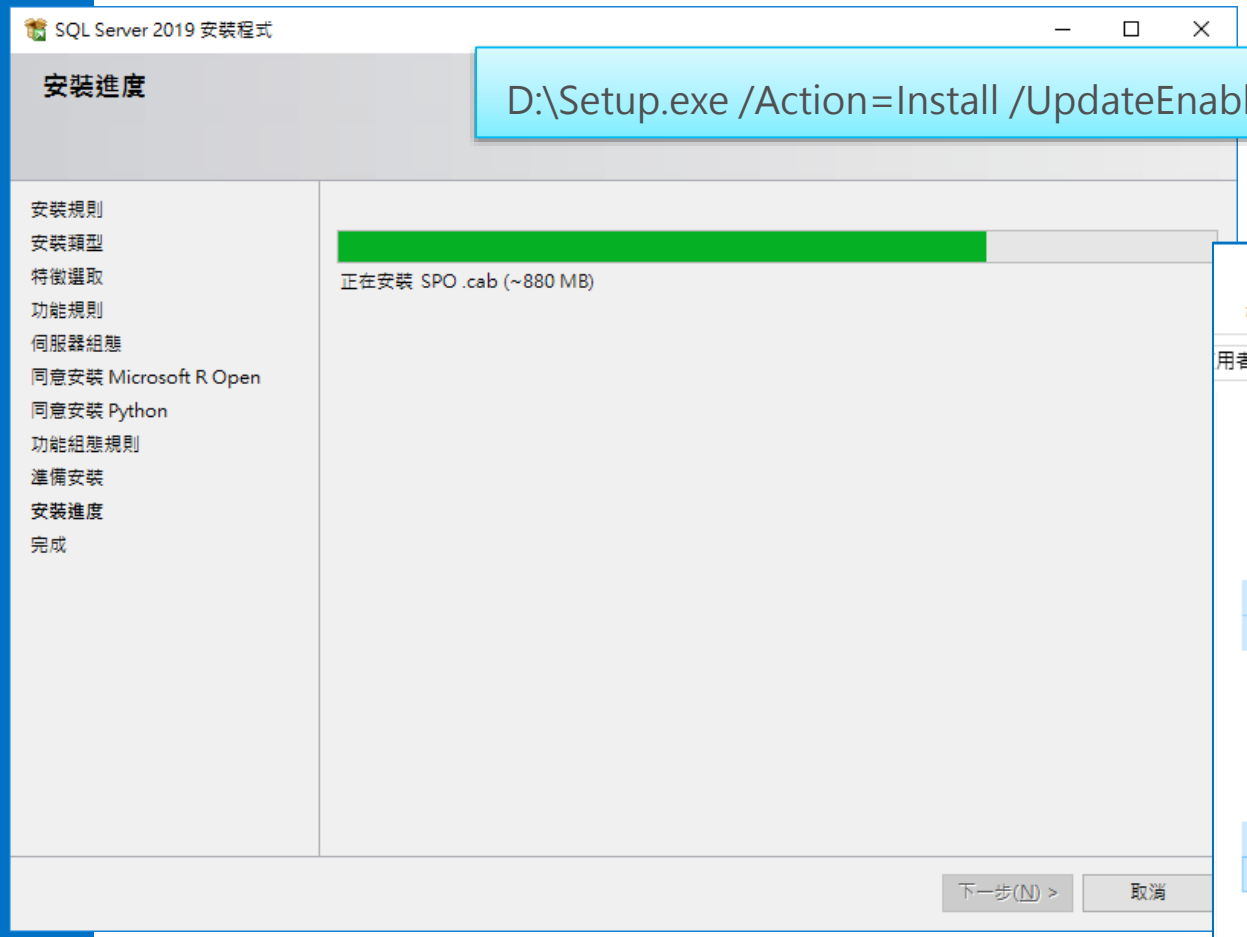
安裝



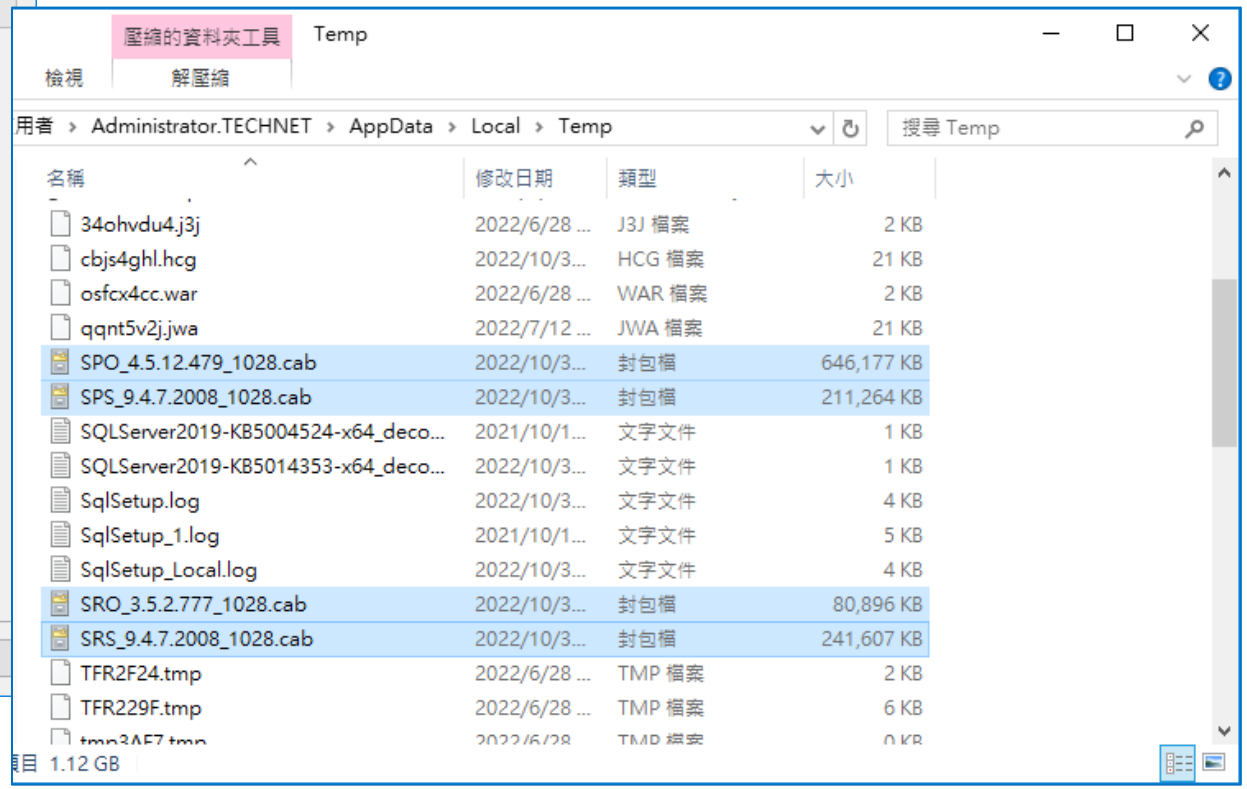


SQL Server 機器學習服務

安裝

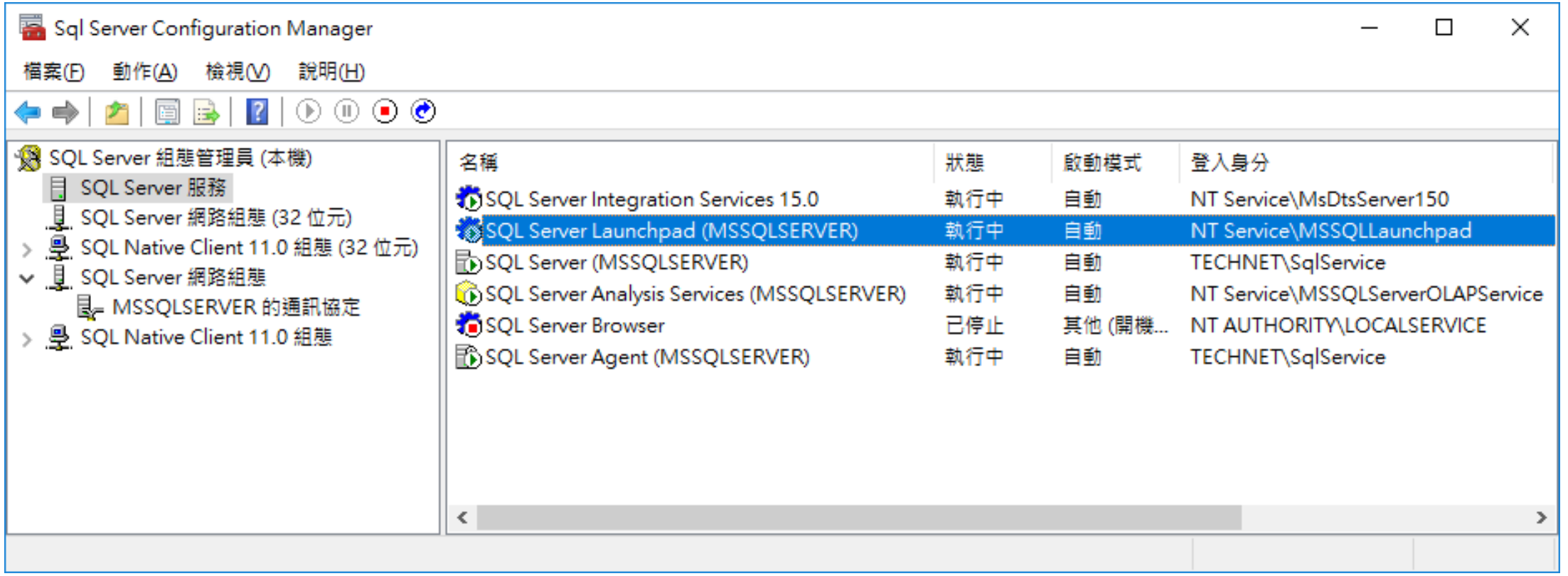



```
D:\Setup.exe /Action=Install /UpdateEnabled=TRUE /UpdateSource="C:\CU"
```



SQL Server Launchpad 服務

Machine Learning Services



SQL Server伺服器選項

啟用外部指令碼執行




```
EXEC sp_configure  'external scripts enabled', 1
RECONFIGURE WITH OVERRIDE
```

執行 Python指令碼

小試一番




```
EXEC sp_execute_external_script
@language = N'Python',
@script = N'
print("Hello SQL Server")
'
```

```
EXEC sp_execute_external_script
@language = N'Python',
@script = N'
Import sys
print(sys.version)
'
```

sp_execute_external_script

搭配**機器學習服務**和**語言延伸模組**使用

```
sp_execute_external_script
    @language = N'language' -- R, Python, Java ( 語言延伸模組，支援Java，須用 CREATE EXTERNAL LANGUAGE 定義)
    , @script = N'script' -- nvarchar(max)
    [ , @input_data_1 = N'SelectStatement' ] -- nvarchar(max)
    [ , @input_data_1_name = N'InputDataVariable' ]  -- sysname, Default Value: InputDataSet; R: data frame; Python: tabular
    [ , @output_data_1_name = N'OutputDataVariable' ] -- Contains data returned to SQL Server; R: data frame; Python: data frame
    [ , @params = N'@ParamName DataType [ OUT | OUTPUT ] [ ,...n ]' ]
    [ , @ParamName = 'Value' [ OUT | OUTPUT ] [ ,...n ] ]
    [ WITH RESULT SETS
      { UNDEFINED | NONE | (( OutputColumnName [,...n ] )) } ] [;]
```

https://learn.microsoft.com/zh-tw/sql/relational-databases/system-stored-procedures/sp-execute-external-script-transact-sql?

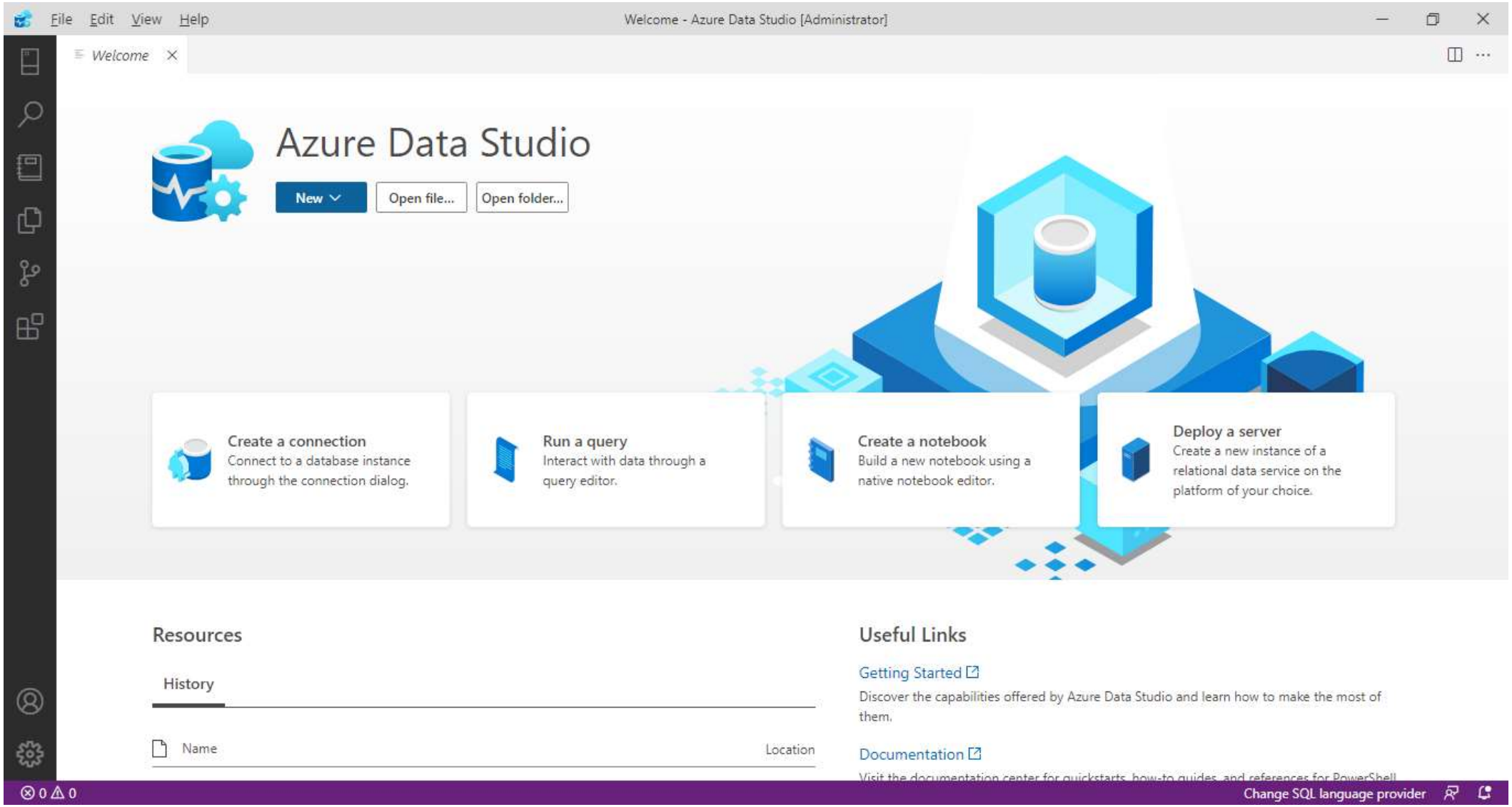
Azure Data Studio

Azure Data Studio

跨平臺資料庫工具

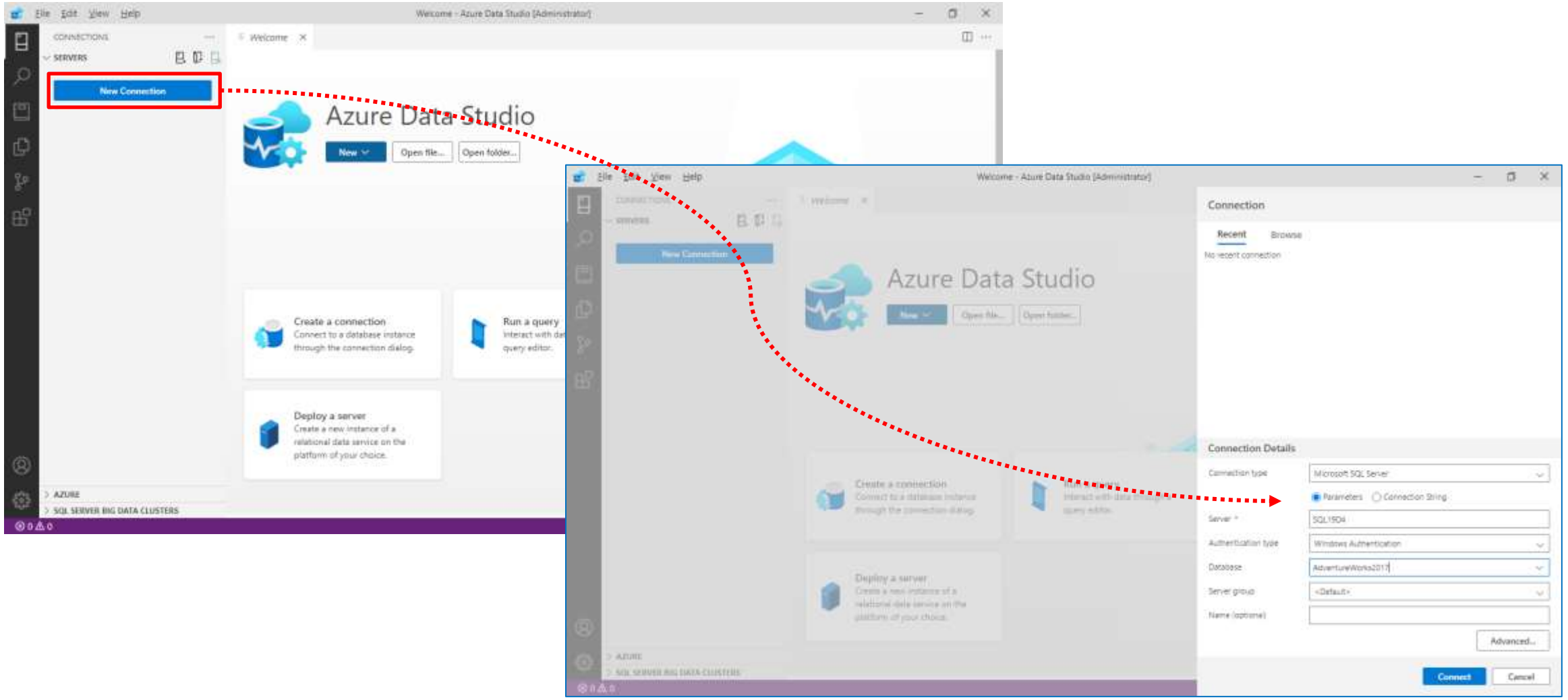
- 適用于在 Windows、macOS 和 Linux 上使用內部部署和雲端資料平臺
- 支援 SQL、PowerShell、Python、KQL、Apache Spark 及 PySpark
- 管理及查詢 SQL Server、PostgreSQL 與 Azure 資料總管
- 新式編輯器體驗，包含
  - IntelliSense、程式碼片段、原始檔控制整合及整合式終端 (sqlcmd 或 PowerShell)
  - 執行結果可儲存為CSV、XML、JSON 或 Excel
- 內建查詢結果集的圖表和可自訂的儀表板

Azure Data Studio



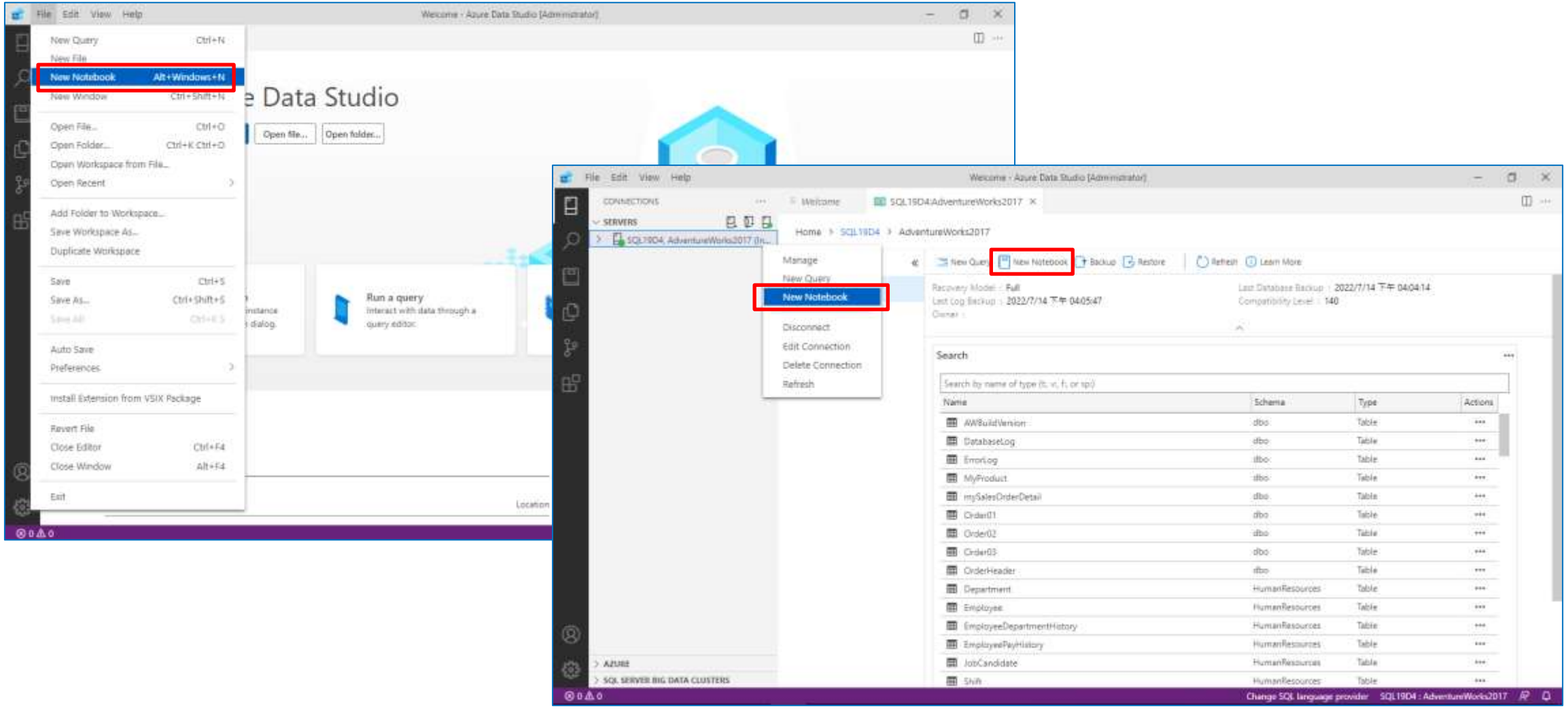
Azure Data Studio

建立連線



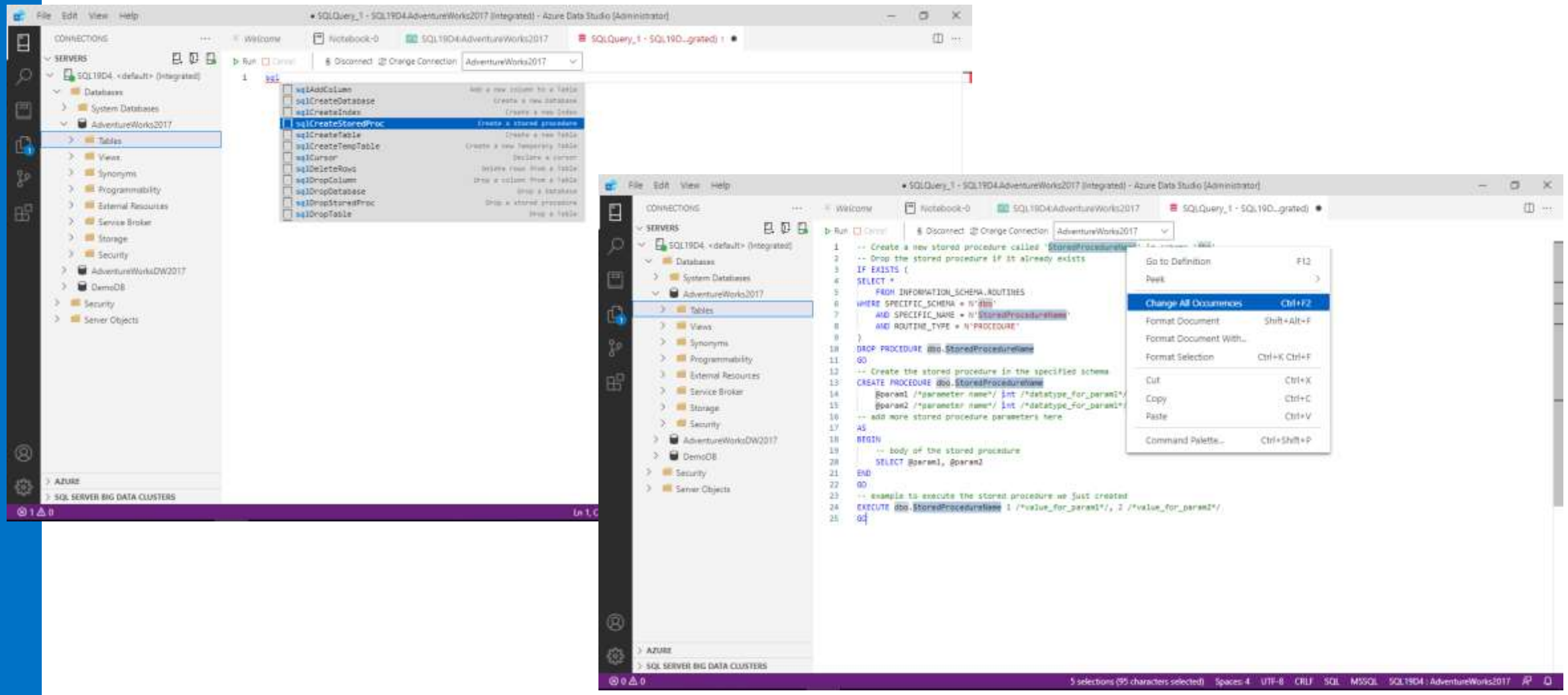
Azure Data Studio

建立 Notebook



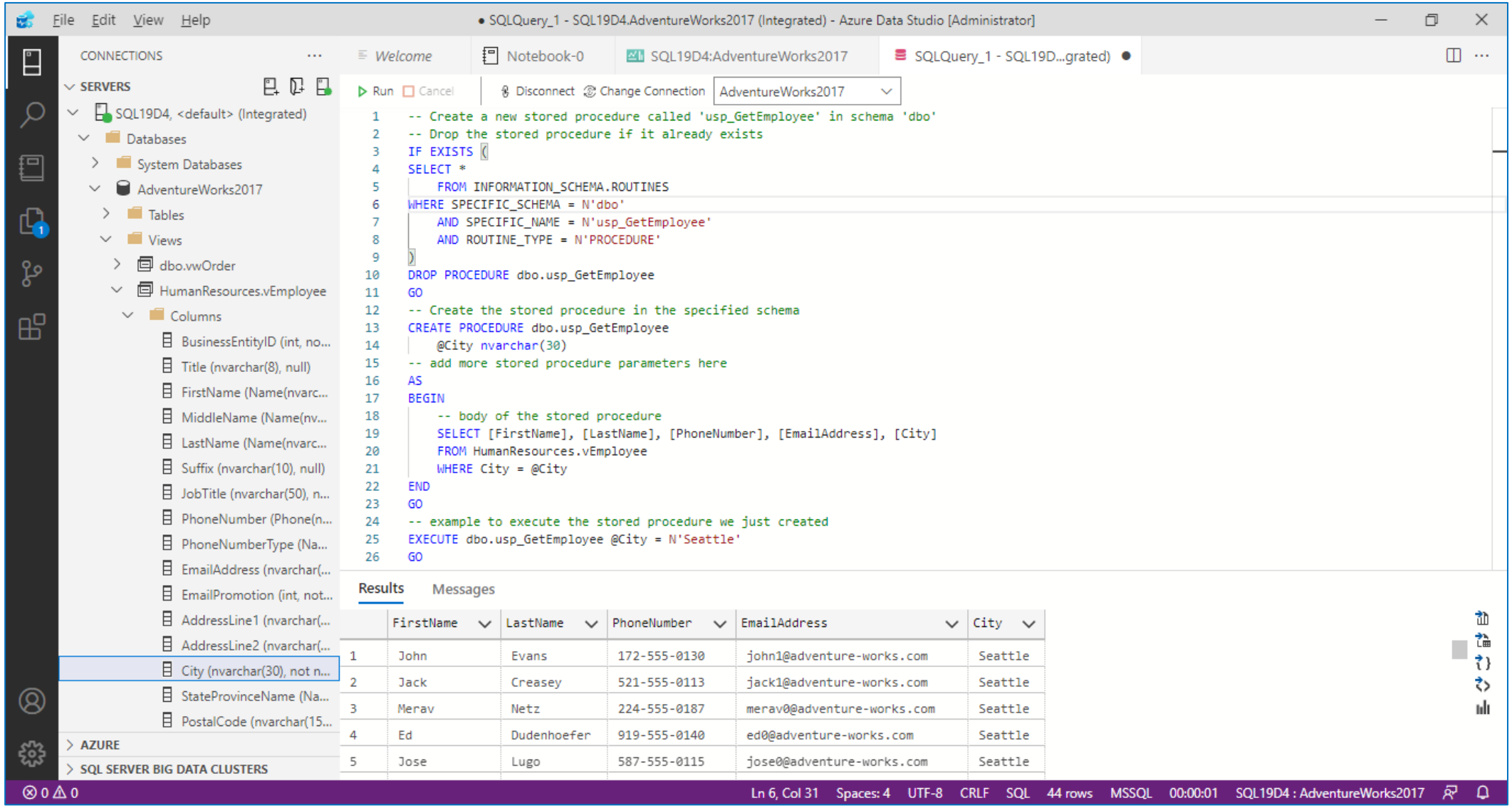
Azure Data Studio

IntelliSense 的 SQL 程式碼編輯器



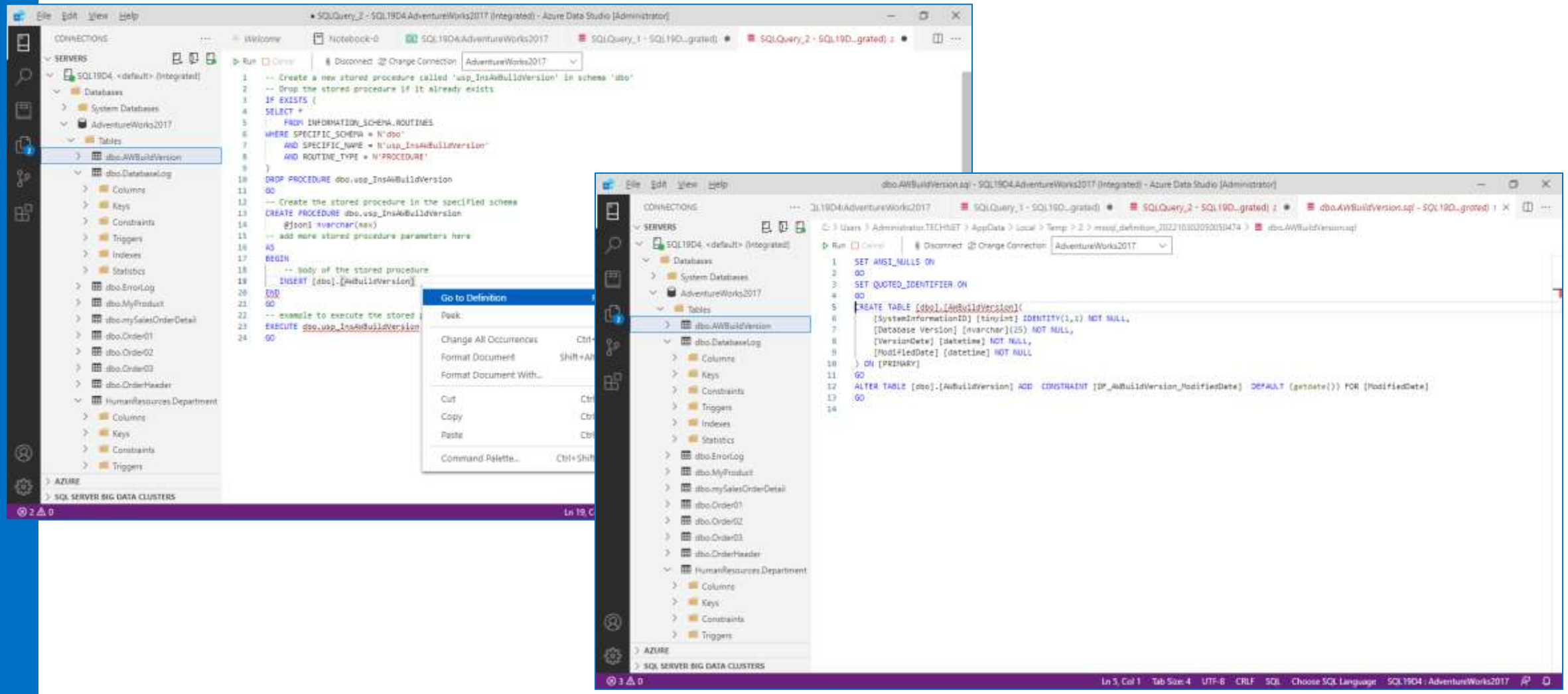
Azure Data Studio

IntelliSense 的 SQL 程式碼編輯器



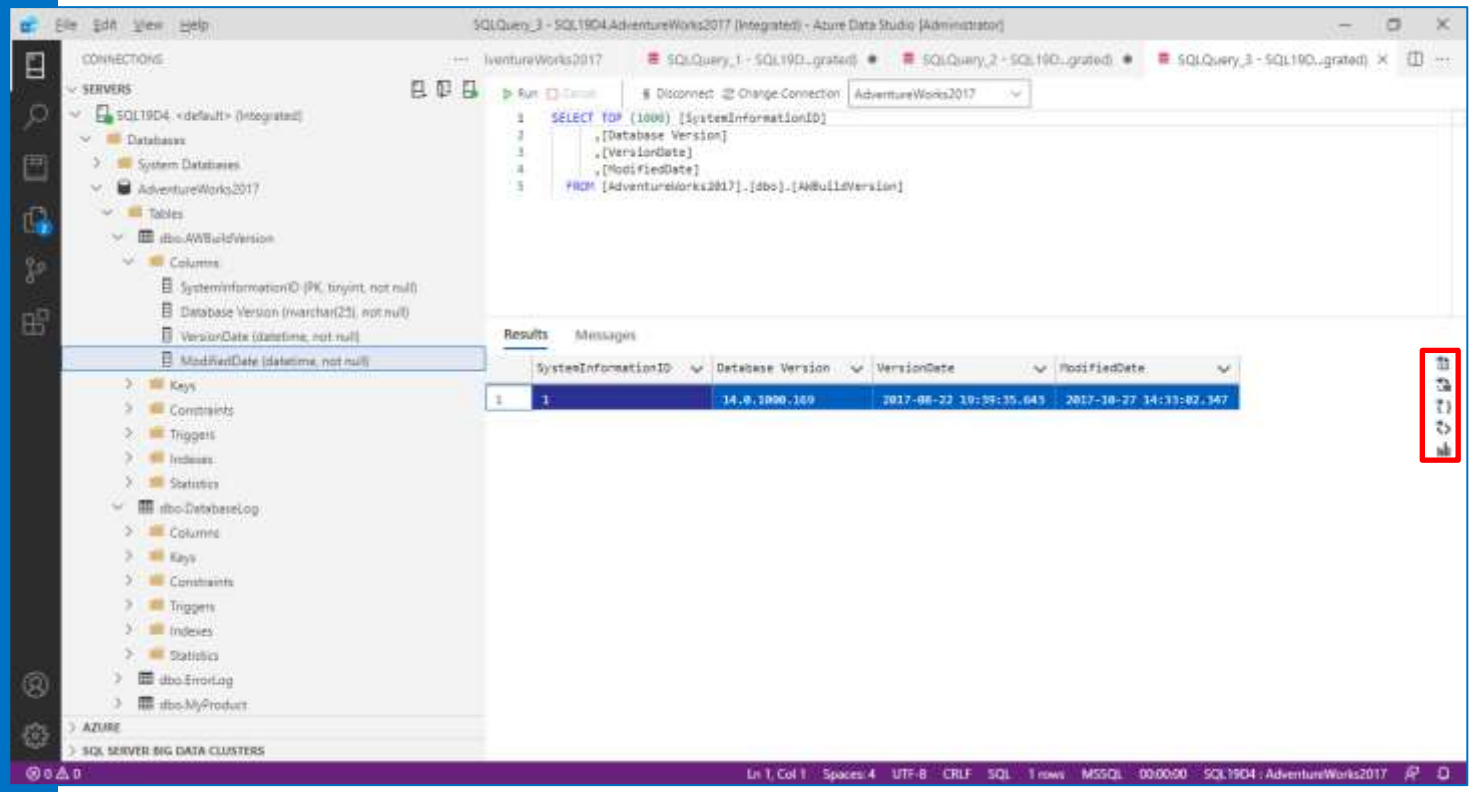
Azure Data Studio

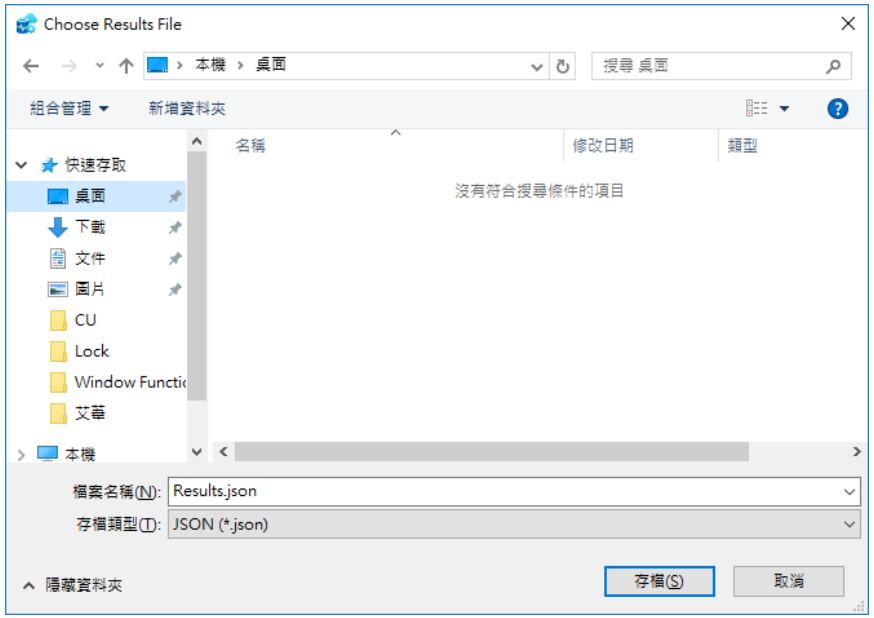
查看定義



Azure Data Studio

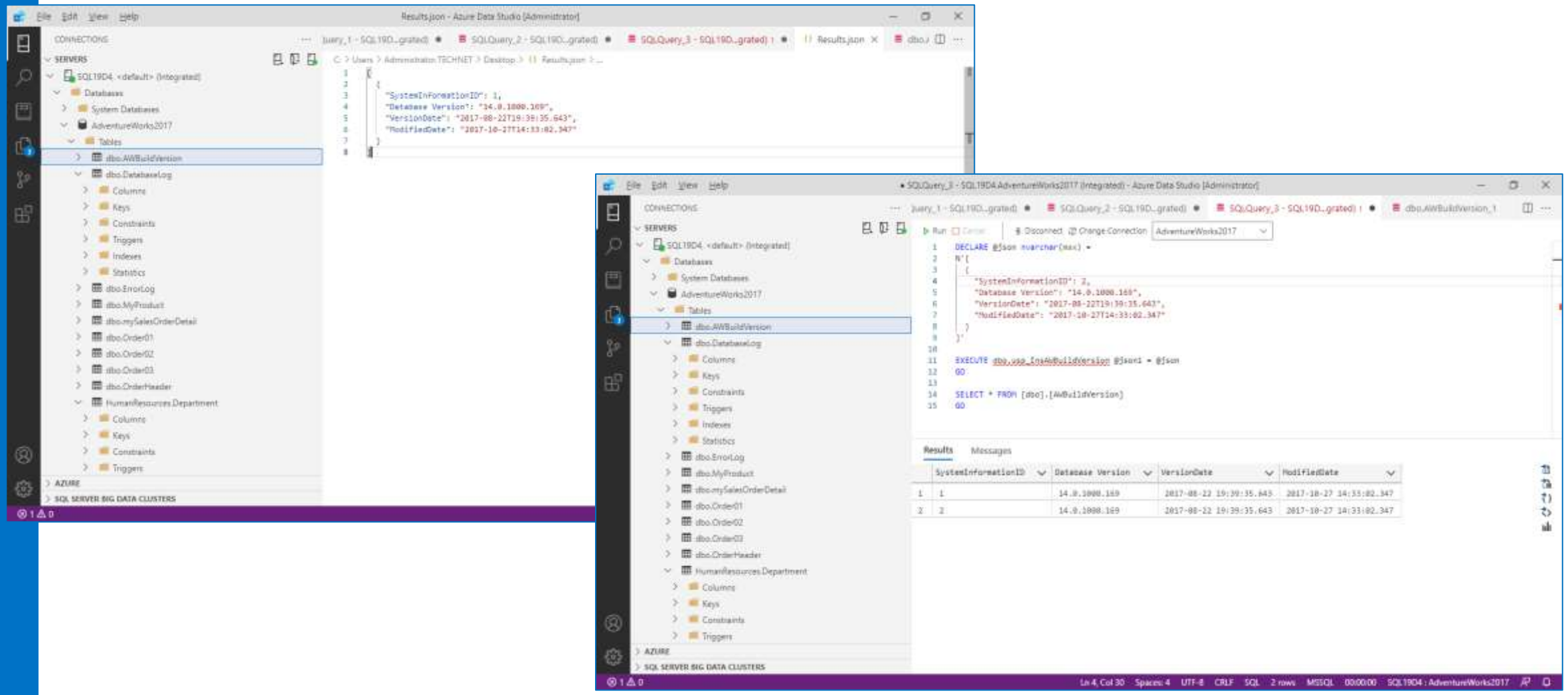
執行結果儲存為 JSON 檔案





Azure Data Studio

執行結果儲存為 JSON 檔案



Azure Data Studio 與 SSMS 功能比較

Azure Data Studio

- 不需執行深層管理或平台相關設定
- 大部分都是編輯或執行查詢
- 能夠快速繪製圖表和視覺化結果集
- 可用 sqlcmd 或 PowerShell 透過整合式終端執行大部分的管理工作
- 對精靈體驗的需求少
- 需要在 macOS 或 Linux 上執行

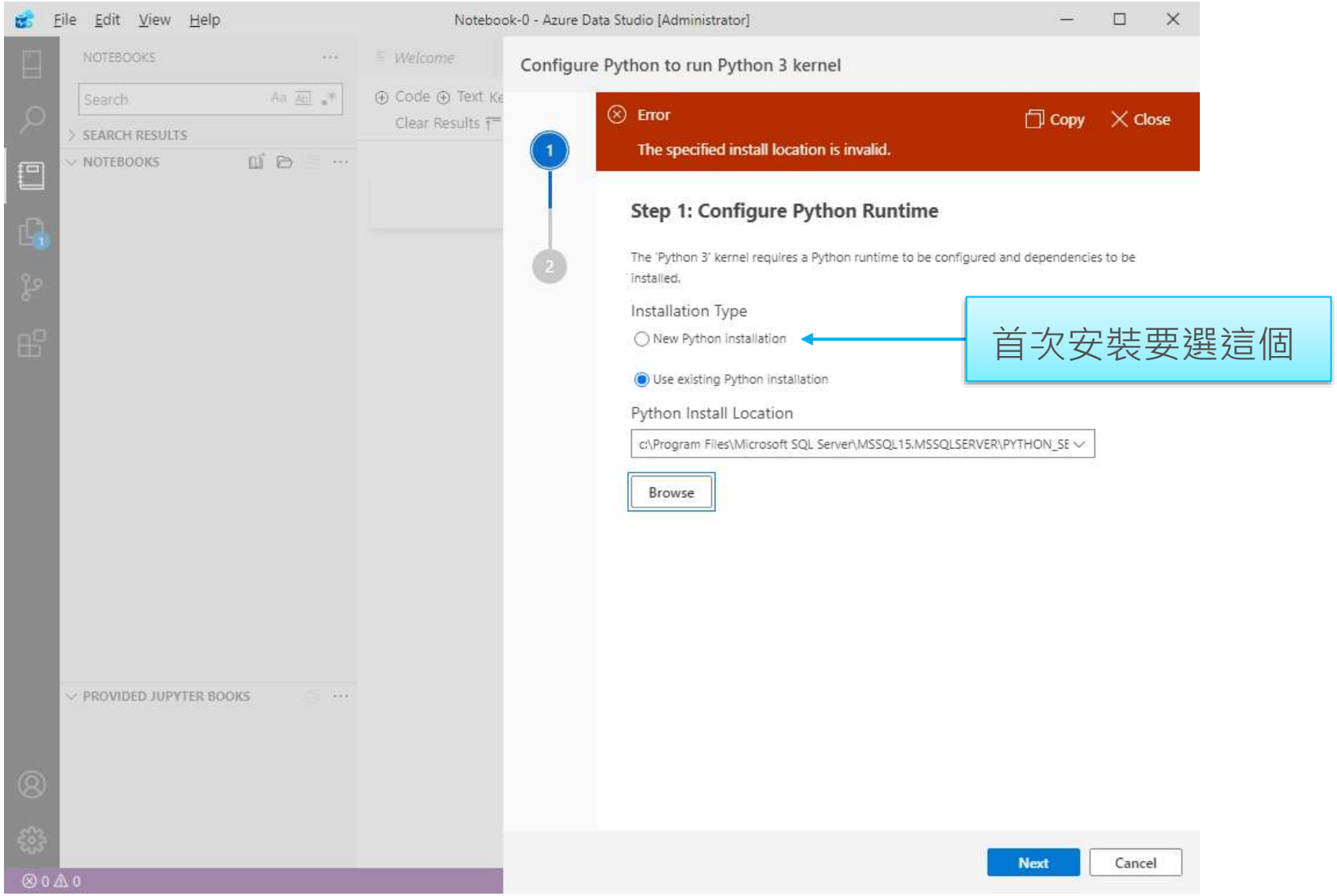
SQL Server Management Studio

- 執行複雜的系統管理或平台設定
- 需使用效能微調建議程式和儀表版
- 使用資料庫圖表和資料表設計工具
- 需要存取已註冊的伺服器
- 利用即時查詢統計資料或用戶端統計資料
- 執行安全性管理，如使用者管理、漏洞評量以及安全性功能的設定

https://learn.microsoft.com/zh-tw/sql/azure-data-studio/what-is-azure-data-studio?view=sql-server-ver16

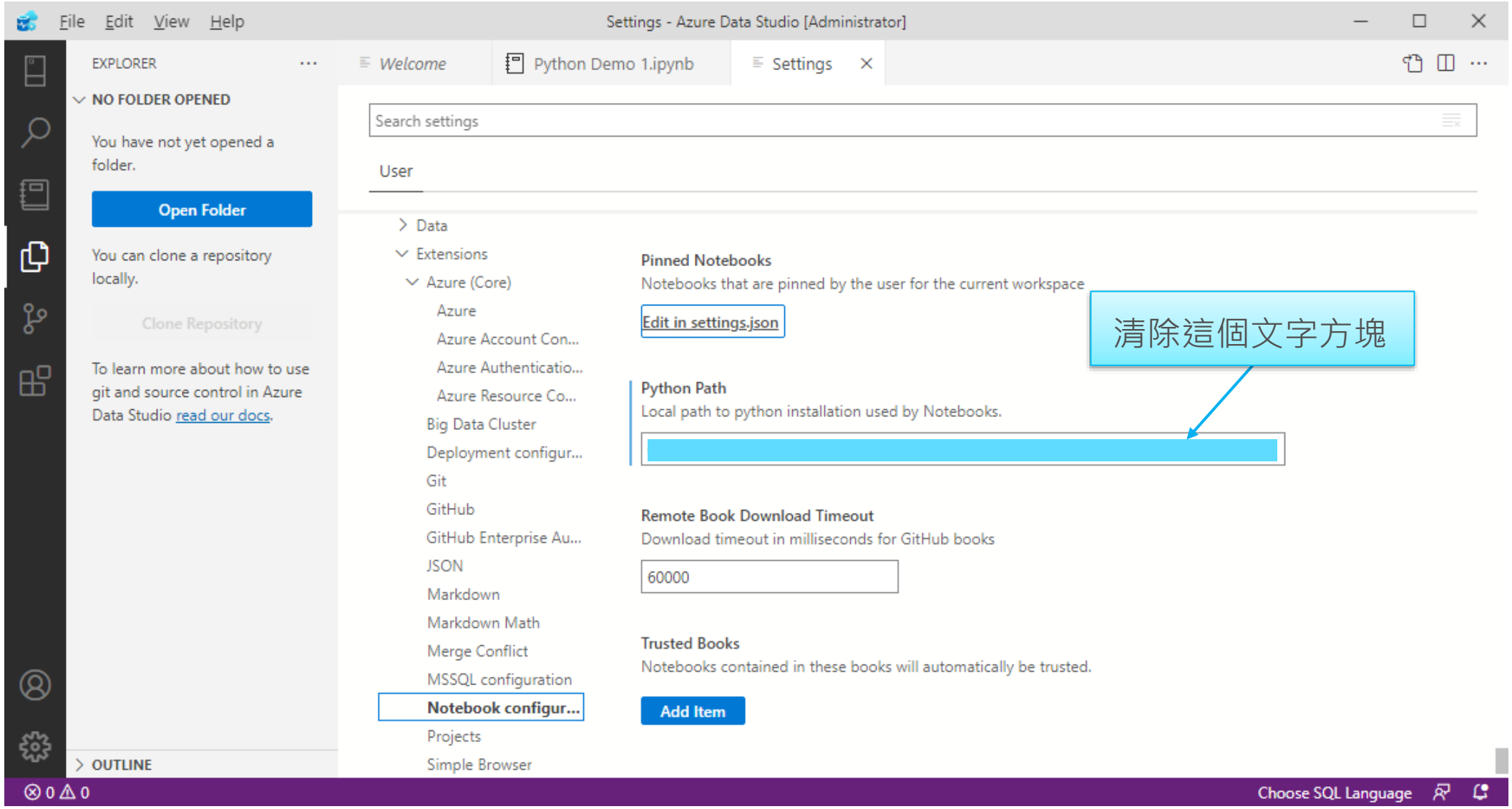
設定 Python Runtime

錯誤示範



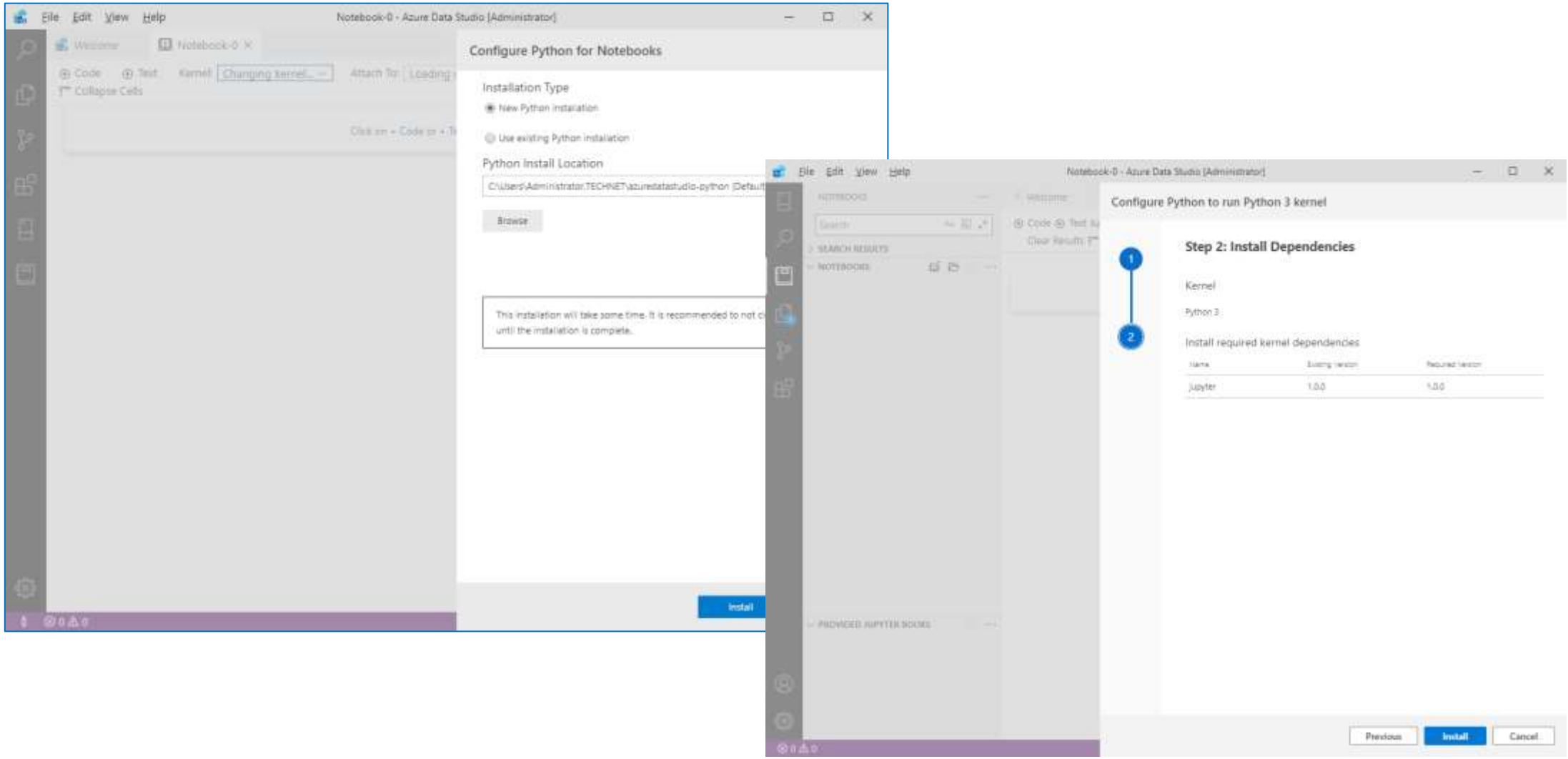
設定 Python Runtime

補救方法



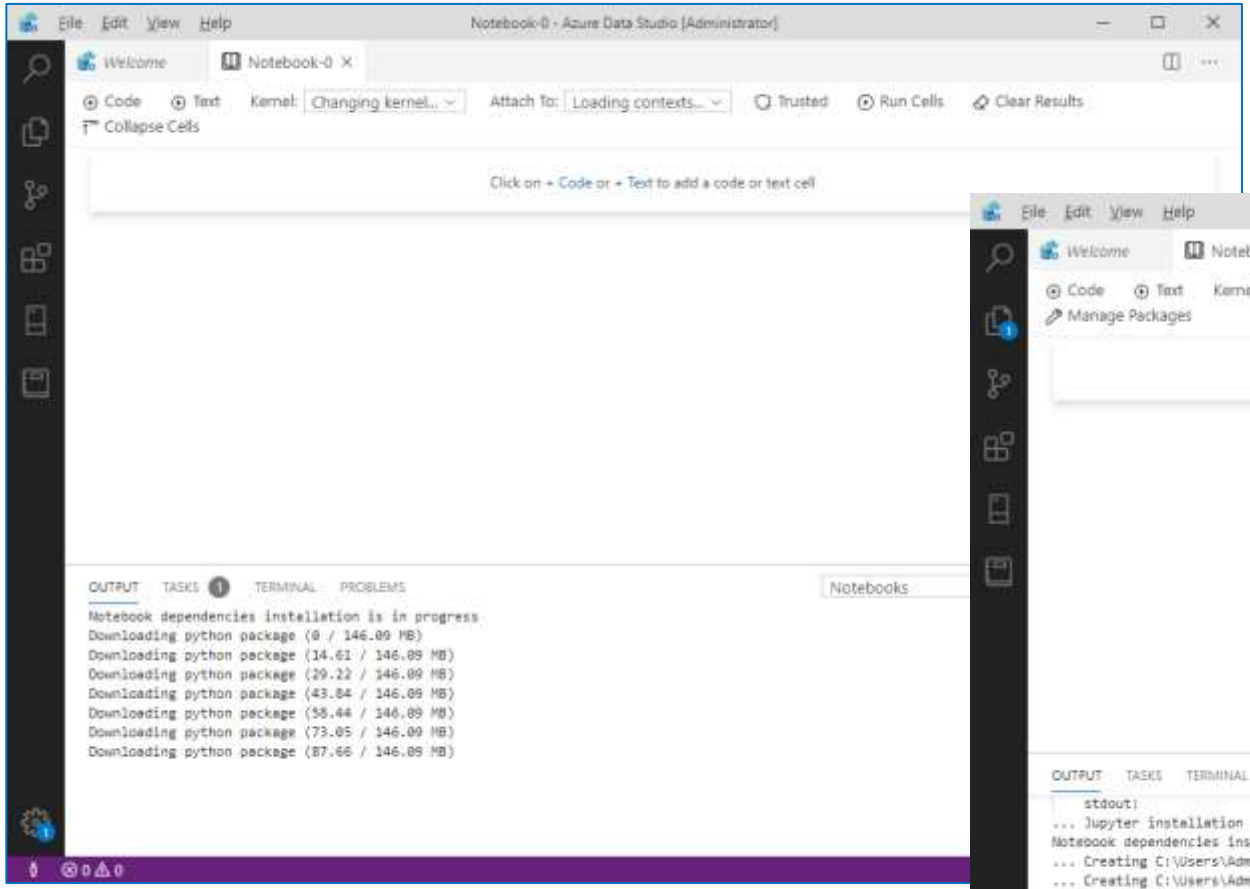
設定 Python Runtime

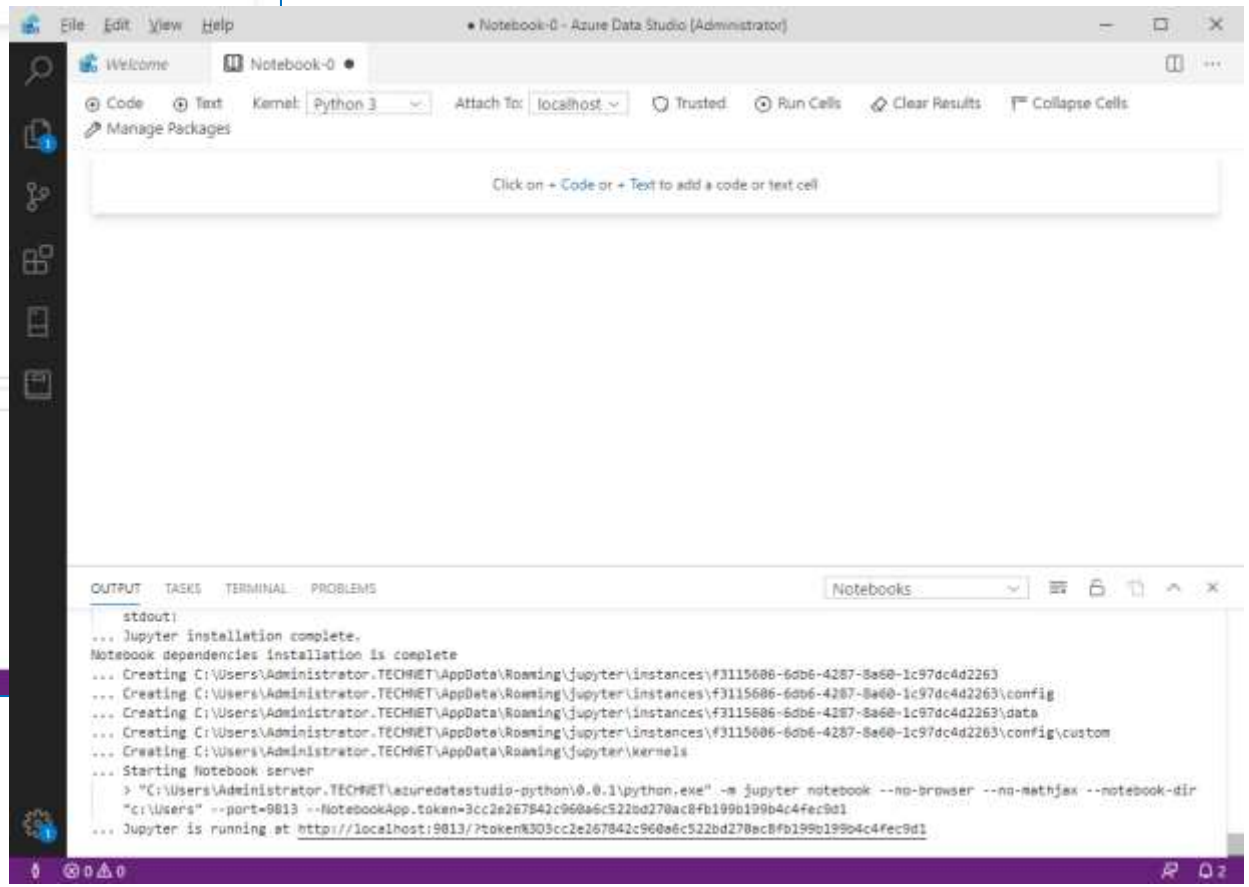
重新安裝



設定 Python Runtime

成功重新設定





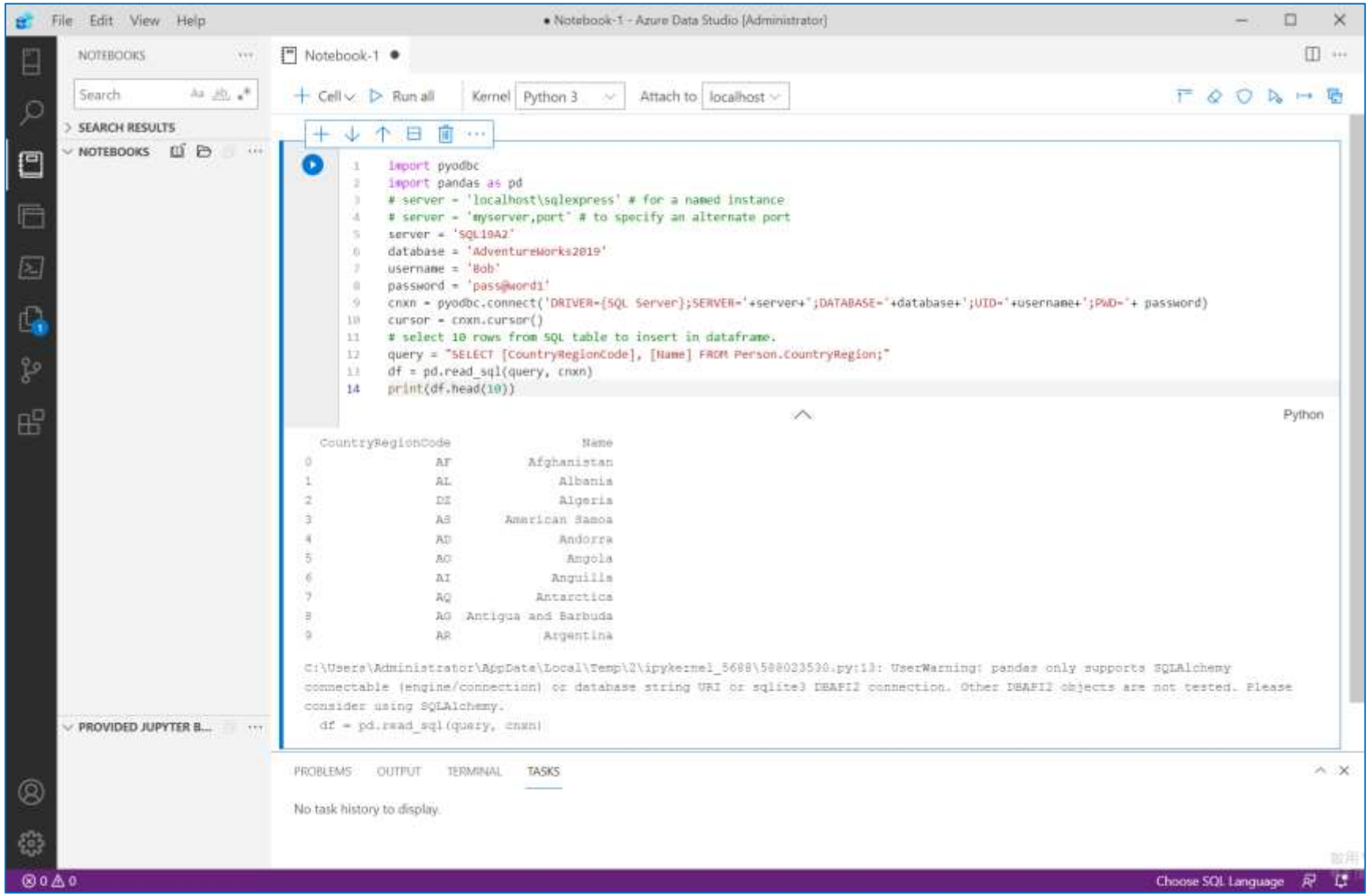
將 SQL 資料表的資料插入 Python pandas 資料框架

Azure Data Studio

```
import pyodbc
import pandas as pd
# server = 'localhost\sqlexpress' # for a named instance
# server = 'myserver,port' # to specify an alternate port
server = 'SQL19A2'
database = 'AdventureWorks2019'
username = 'Bob'
password = 'pass@word1'
cnxn = pyodbc.connect('DRIVER={SQL Server};SERVER='+server+';DATABASE='+database+';UID='+username+';PWD='+ password)
cursor = cnxn.cursor()
# select 10 rows from SQL table to insert in dataframe.
query = "SELECT [CountryRegionCode], [Name] FROM Person.CountryRegion;"
df = pd.read_sql(query, cnxn)
print(df.head(10))
```

將 SQL 資料表的資料插入 Python pandas 資料框架

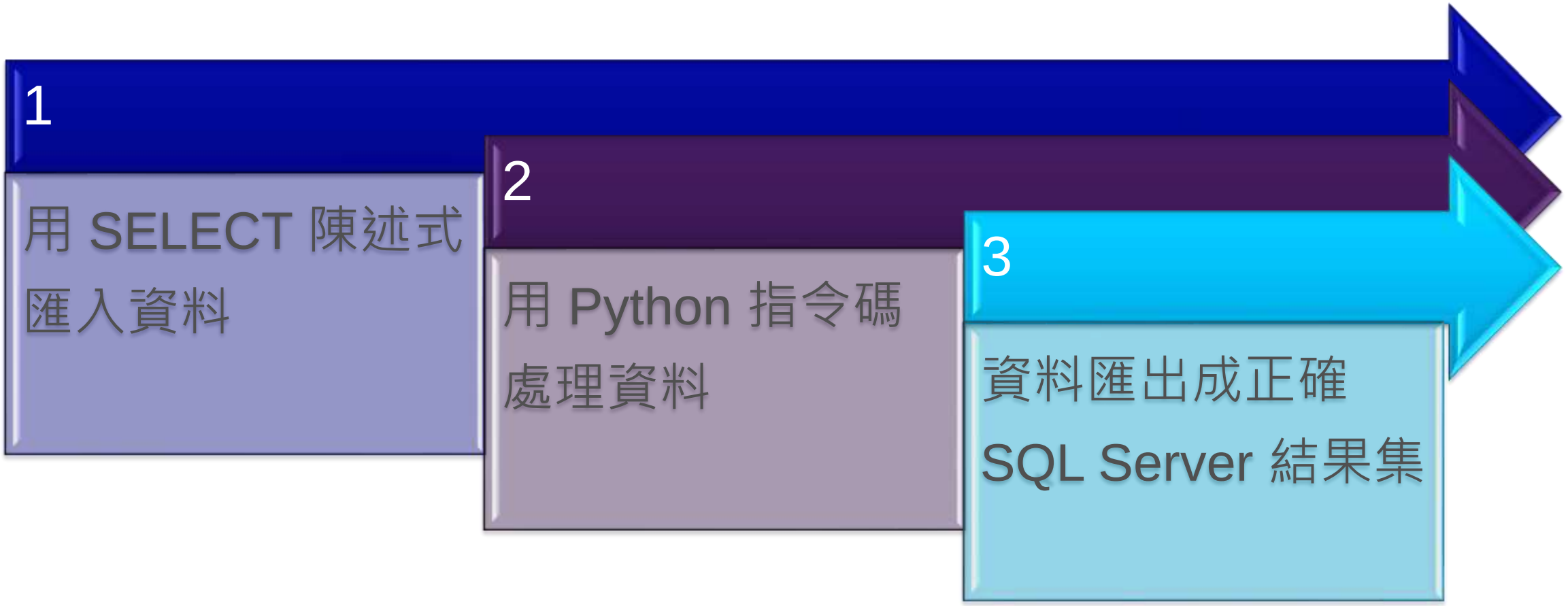
Azure Data Studio



用 Python 處理 SQL Server 資料

用 Python 處理 SQL Server 資料

掌握 3 步驟，輕鬆搞定



用 Python 處理 SQL Server 資料

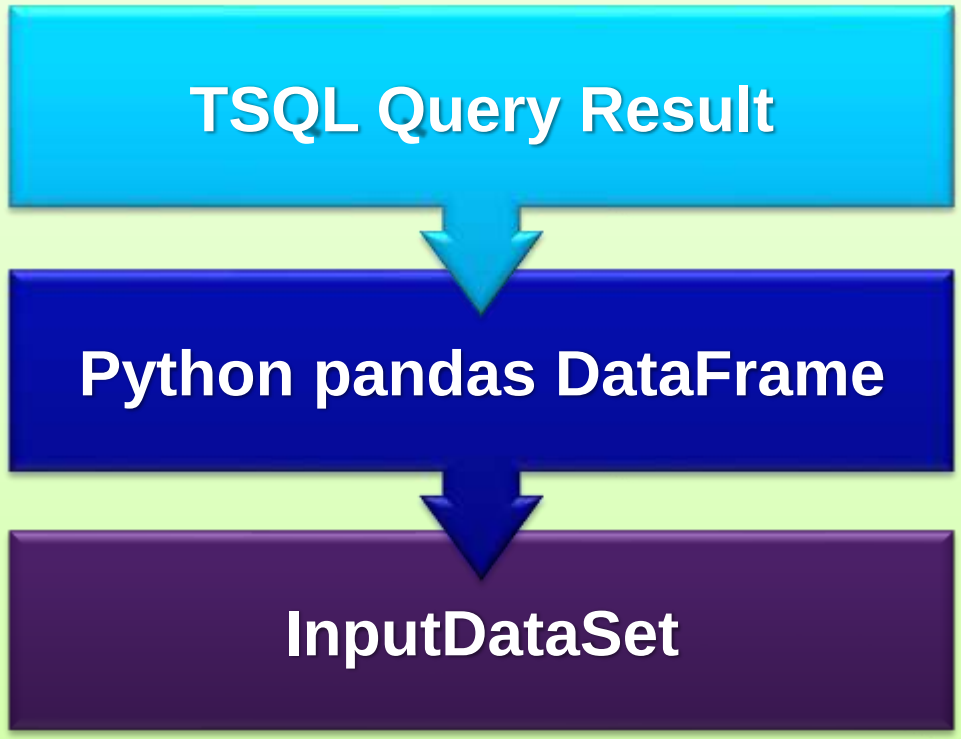
1. 用 SELECT 陳述式匯入資料

```
EXEC sp_execute_external_script
    @language = N'python',
    @script = N'
     python script  -- nvarchar(max)
    ',
    @input_data_1 = N'
     TSQL SELECT; -- nvarchar(max)
    '
```

用 Python 處理 SQL Server 資料

2. 用 Python 指令碼處理資料

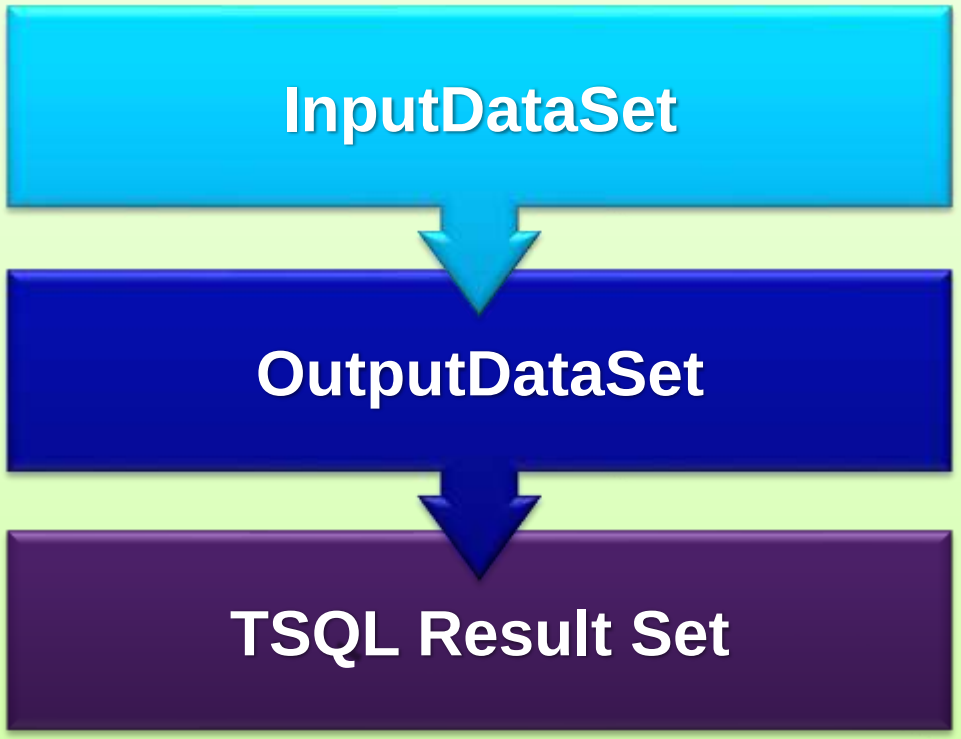
```
EXEC sp_execute_external_script
    @language = N'python',
    @script = N'
    python script  -- nvarchar(max)
    ',
    @input_data_1 = N'
    TSQL SELECT; -- nvarchar(max)
    '
```



用 Python 處理 SQL Server 資料

3. 資料匯出成正確SQL Server 結果集

```
EXEC sp_execute_external_script
    @language = N'python',
    @script = N'
     python script  -- nvarchar(max)
    ',
    @input_data_1 = N'
     TSQL SELECT; -- nvarchar(max)
    '
```



SQL Server 資料庫的查詢結果
插入 Python pandas 資料框架進行統計分析

TSQL Query Result 傳給 Python pandas DataFrame

Demo 1. 將 TSQL Query Result 傳給 Python pandas DataFrame (**InputDataSet**) 再輸出 TSQL Result Set (**OutputDataSet**)

```
[3]  1  Use AdventureWorks2017;
     2  GO
     3
     4  EXEC sp_execute_external_script
     5    @language = N'Python',
     6    @script = N'
     7  df = InputDataSet
     8  OutputDataSet = df
     9  ',
    10    @input_data_1 = N'
    11  SELECT t.Name AS Territories, h.SalesOrderID
    12  FROM Sales.SalesOrderHeader h INNER JOIN Sales.SalesTerritory t
    13      ON h.TerritoryID = t.TerritoryID
    14  WHERE YEAR(h.OrderDate) = 2013;
    15    ';
    16  GO
```

Commands completed successfully.

(14182 rows affected)

Displaying Top 5000 rows.

Total execution time: 00:00:00.489

	(No column name)	(No column name)
1	United Kingdom	49181
2	Germany	49182
3	Germany	49183
4	Australia	49184
5	Australia	49185

無 欄位名稱

指定 TSQL Result Set 欄位名稱和資料型別

WITH RESULT SETS

Demo 2. 利用 **WITH RESULT SETS** 指定 TSQL Result Set 欄位名稱和資料型別

```
[2]  1  Use AdventureWorks2017;
     2  GO
     3
     4  EXEC sp_execute_external_script
     5    @language = N'Python',
     6    @script = N'
     7  df = InputDataSet
     8  OutputDataSet = df
     9  ',
    10    @input_data_1 = N'
    11  SELECT t.Name AS Territories, h.SalesOrderID
    12  FROM Sales.SalesOrderHeader h INNER JOIN Sales.SalesTerritory t
    13      ON h.TerritoryID = t.TerritoryID
    14  WHERE YEAR(h.OrderDate) = 2013;
    15  '
    16    WITH RESULT SETS((Territories NVARCHAR(100), SalesOrderID INT));
    17  GO
```

Commands completed successfully.

(14182 rows affected)

Displaying Top 5000 rows.

Total execution time: 0-23:0-22:0-43

	Territories	SalesOrderID
1	United Kingdom	49181
2	Germany	49182
3	Germany	49183
4	Australia	49184
5	Australia	49185

欄位名稱

用 pandas.DataFrame.groupby( ) 執行群組彙總計算

用 Python celandar 轉換月份數字成英文縮寫月份

pandas.DataFrame.groupby( )

involves some combination of splitting the object, applying a function, and combining the results

pandas

Demo 4. 利用 Python celandar 轉換月份數字成英文縮寫月份

利用 Python pandas.DataFrame.groupby() 方法 (method)，執行群組 (Territories 和 OrderMonths) 彙總計算訂單數量

```
[7]
Use AdventureWorks2017;
GO

EXEC sp_execute_external_script
  @language = N'Python',
  @script = N'
import calendar as cl

# 依據 Territories 和 OrderMonths 群組計算訂單數量
df1 = InputDataSet
df2 = df1.groupby(["Territories", "OrderMonths"], as_index=False).count()

# 轉換月份數字成英文縮寫月份
df2["OrderMonths"] = df2["OrderMonths"].apply(lambda x: cl.month_abbr[x])

# 回傳調整後的 Dataset
OutputDataSet = df2
',
  @input_data_1 = N'
SELECT t.Name AS Territories, MONTH(h.OrderDate) AS OrderMonths, h.SalesOrderID
FROM Sales.SalesOrderHeader h INNER JOIN Sales.SalesTerritory t
    ON h.TerritoryID = t.TerritoryID
WHERE YEAR(h.OrderDate) = 2013;
  '
  WITH RESULT SETS((Territories NVARCHAR(100), OrderMonths NCHAR(3), TotalOrderCounts INT));
GO
```

Commands completed successfully.

來自外部指令碼的 STDOUT 訊息: Warning: numpy.int64 data type is not supported. Data is converted to float64.

(120 rows affected)

Total execution time: 00:00:00.001

	Territories	OrderMonths	TotalOrderCounts
1	Australia	Jan	118
2	Australia	Feb	94
3	Australia	Mar	115
4	Australia	Apr	112
5	Australia	May	119

Pandas

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.groupby.html

資料類型不相容

Unsupported input data type in column: SQL 類型 (money) 不相容於 Python

Demo 5. 利用 Python pandas.DataFrame.groupby() 方法 (method)，執行群組 (Territories 和 OrderMonths) 彙總計算訂單金額 Subtotal

會有"資料類型不相容"的錯誤訊息 Unsupported input data type in column 'Subtotal'

```
[5]  1  Use AdventureWorks2017;
     2  GO
     3
     4  EXEC sp_execute_external_script
     5    @language = N'Python',
     6    @script = N'
     7  import calendar as cl
     8
     9  # 依據 Territories 和 OrderMonths 群組計算訂單金額
    10  df1 = InputDataSet
    11  df2 = df1.groupby(["Territories", "OrderMonths"], as_index=False).sum()
    12
    13  # 轉換月份數字成英文縮寫月份
    14  df2["OrderMonths"] = df2["OrderMonths"].apply(lambda x: cl.month_abbr[x])
    15
    16  # 回傳匯總後的 Dataset
    17  OutputDataSet = df2
    18  ',
    19    @input_data_1 = N'
    20  SELECT t.Name AS Territories, MONTH(h.OrderDate) AS OrderMonths, h.Subtotal
    21  FROM Sales.SalesOrderHeader h INNER JOIN Sales.SalesTerritory t
    22      ON h.TerritoryID = t.TerritoryID
    23  WHERE YEAR(h.OrderDate) = 2013;
    24  '
    25    WITH RESULT SETS((Territories NVARCHAR(100), OrderMonths NCHAR(3), TotalSales MONEY));
    26  GO
```

SQL

Commands completed successfully.

來自外部指令碼的 STDOUT 訊息: SqlSatelliteCall function failed. Please see the console output for more information. Traceback (most recent call last): File "C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\PYTHON_SERVICES\lib\site-packages\revoscalepy\computecontext\RxInSqlServer.py", line 605, in rx_sql_satellite_call rx_native_call("SqlSatelliteCall", params) File "C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\PYTHON_SERVICES\lib\site-packages\revoscalepy\RxSerializable.py", line 375, in rx_native_call ret = px_call(functionname, params) RuntimeError: revoscalepy function failed.

Total execution time: 00:00:00.001

	Territories	OrderMonths	TotalSales

Msg 39004, Level 16, State 20, Line 4
執行 'sp_execute_external_script' 時發生 'Python' 指令碼錯誤，HRESULT 為 0x80004004。

Msg 39019, Level 16, State 2, Line 4
發生外部指令碼錯誤：

```
Unsupported input data type in column 'Subtotal'.  Supported types: bit, tinyint, smallint, int, bigint, uniqueidentifier, real, float, char, varchar, nchar, nvarchar, varbinary, date, datetime, smalldatetime.
SqlSatelliteCall error: Unsupported input data type in column 'Subtotal'.  Supported types: bit, tinyint, smallint, int, bigint, uniqueidentifier, real, float, char, varchar, nchar, nvarchar, varbinary, date, datetime, smalldatetime.
```

資料類型

SQL 資料類型 (money) 轉換為 (float) 相容於 Python

Demo 6. 利用 Python pandas.DataFrame.groupby() 方法 (method)，執行群組 (Territories 和 OrderMonths) 彙總計算訂單金額 Subtotal

T-SQL CONVERT函數轉換 money 為 float 以相容於 Python

```
[9]
Use AdventureWorks2017;
GO

EXEC sp_execute_external_script
  @language = N'Python',
  @script = N'
import calendar as cl

# 依據 Territories 和 OrderMonths 群組計算訂單金額
df1 = InputDataSet
df2 = df1.groupby(["Territories", "OrderMonths"], as_index=False).sum()

# 轉換月份數字成英文縮寫月份
df2["OrderMonths"] = df2["OrderMonths"].apply(lambda x: cl.month_abbr[x])

# 回傳匯總後的 Dataset
OutputDataSet = df2
',
  @input_data_1 = N'
SELECT t.Name AS Territories, MONTH(h.OrderDate) AS OrderMonths, CONVERT(float, h.Subtotal) AS Sales
FROM Sales.SalesOrderHeader h INNER JOIN Sales.SalesTerritory t
    ON h.TerritoryID = t.TerritoryID
WHERE YEAR(h.OrderDate) = 2013;
  '
  WITH RESULT SETS((Territories NVARCHAR(100), OrderMonths NCHAR(3), TotalSales MONEY));
GO
```

Commands completed successfully.

(120 rows affected)

Total execution time: 00:00:00.002

	Territories	OrderMonths	TotalSales
1	Australia	Jan	222513.7199
2	Australia	Feb	177688.0601
3	Australia	Mar	215687.8517
4	Australia	Apr	205166.1319
5	Australia	May	252418.6364

pandas.DataFrame 增加欄位 (匯總統計)

agg() 方法: 傳遞 dictionary 值 { }

Demo 7. Python pandas.DataFrame 增加欄位，平均，最大，最小，總計

agg() 方法: 傳遞 dictionary 值 { } 大括號

[10]

```
Use AdventureWorks2017;
GO

EXEC sp_execute_external_script
  @language = N'Python',
  @script = N'
import calendar as cl

# 依據 Territories 和 OrderMonths 群組計算訂單金額(平均，最大，最小，總計)
df1 = InputDataSet
df2 = df1.groupby(["Territories", "OrderMonths"], as_index=False).agg({"Sales": ["mean", "max", "min", "sum"]})

# 轉換月份數字成英文縮寫月份
df2["OrderMonths"] = df2["OrderMonths"].apply(lambda x: cl.month_abbr[x])

# 回傳匯總後的 Dataset
OutputDataSet = df2
',
  @input_data_1 = N'
SELECT t.Name AS Territories, MONTH(h.OrderDate) AS OrderMonths, CONVERT(float, h.Subtotal) AS Sales
FROM Sales.SalesOrderHeader h INNER JOIN Sales.SalesTerritory t
    ON h.TerritoryID = t.TerritoryID
WHERE YEAR(h.OrderDate) = 2013;
  '
  WITH RESULT SETS((Territories NVARCHAR(100), OrderMonths NCHAR(3), AvgSales MONEY, MaxSales MONEY, MinSales MONEY, TotalSales MONEY));
GO
```

Commands completed successfully.

(120 rows affected)

Total execution time: 00:00:00

	Territories	OrderMonths	AvgSales	MaxSales	MinSales	TotalSales
1	Australia	Jan	1885.7095	2443.35	782.99	222513.7199
2	Australia	Feb	1890.2985	2443.35	782.99	177688.0601
3	Australia	Mar	1875.5465	2443.35	782.99	215687.8517
4	Australia	Apr	1831.8405	2443.35	782.99	205166.1319
5	Australia	May	2121.165	26328.5293	283.9489	252418.6364

排序 pandasDataFrame 欄位，再篩選前10名

sort_values(by=*list of str*, ascending=*True*)

Demo 10. 排序 pandasDataFrame 欄位 TotalSales，再篩選前10名

sort_values(by=*list of str*, ascending=*True*)

```
[25]  1  Use AdventureWorks2017;
      2  GO
      3
      4  EXEC sp_execute_external_script
      5    @language = N'Python',
      6    @script = N'
      7  import calendar as cl
      8
      9  # 依據 Territories 和 OrderMonths 群組計算訂單金額(平均，最大，最小，總計)
     10  df1 = InputDataSet
     11  df2 = df1.groupby(["Territories", "OrderMonths"], as_index=False).agg({"Sales": ["mean", "max", "min", "sum"]})
     12
     13  # 指定 DataFrame 欄位名稱
     14  df2.columns = ["Territories", "OrderMonths", "AvgSales", "MaxSales", "MinSales", "TotalSales"]
     15
     16  # 轉換月份數字成英文縮寫月份
     17  df2["OrderMonths"] = df2["OrderMonths"].apply(lambda x: cl.month_abbr[x])
     18
     19  # 回傳排序後篩選的 Dataset
     20  OutputDataSet = df2.sort_values(by=["TotalSales"], ascending=False).head(10)
     21  ',
     22    @input_data_1 = N'
     23  SELECT t.Name AS Territories, MONTH(h.OrderDate) AS OrderMonths, CONVERT(float, h.Subtotal) AS Sales
     24  FROM Sales.SalesOrderHeader h INNER JOIN Sales.SalesTerritory t
     25      ON h.TerritoryID = t.TerritoryID
     26  WHERE YEAR(h.OrderDate) = 2013;
     27    '
     28    WITH RESULT SETS((Territories NVARCHAR(100), OrderMonths NCHAR(3), AvgSales MONEY, MaxSales MONEY, MinSales MONEY, TotalSales MONEY));
     29  GO
```

Commands completed successfully.

(10 rows affected)

Total execution time: 00:00:04.116

	Territories	OrderMonths	AvgSales	MaxSales	MinSales	TotalSales
1	Southwest	Jul	2724.0693	106929.5461	2.29	967044.6048
2	Southwest	May	10744.4875	163930.3943	484.632	848814.512
3	Southwest	Mar	14112.684	105310.5088	159.558	846761.041
4	Southwest	Oct	2047.6359	73605.9574	2.29	833387.8085
5	Southwest	Jun	6310.7666	78790.812	8.99	827496.4306

pandasDataFrame 欄位條件篩選

df2[(df2["Territories"] == "Canada") & (df2["TotalSales"] > 500000)]

Demo 11. 篩選 pandasDataFrame Territories 為 Canada，且 TotalSales > 500,000

```
[24]  1  Use AdventureWorks2017;
      2  GO
      3
      4  EXEC sp_execute_external_script
      5    @language = N'Python',
      6    @script = N'
      7  import calendar as cl
      8
      9  # 依據 Territories 和 OrderMonths 群組計算訂單金額(平均、最大、最小、總計)
     10  df1 = InputDataSet
     11  df2 = df1.groupby(["Territories", "OrderMonths"], as_index=False).agg({"Sales": ["mean", "max", "min", "sum"]})
     12
     13  # 指定 DataFrame 欄位名稱
     14  df2.columns = ["Territories", "OrderMonths", "AvgSales", "MaxSales", "MinSales", "TotalSales"]
     15
     16  # 轉換月份數字成英文縮寫月份
     17  df2["OrderMonths"] = df2["OrderMonths"].apply(lambda x: cl.month_abbr[x])
     18
     19  # 回傳篩選後的 Dataset
     20  OutputDataSet = df2[(df2["Territories"] == "Canada") & (df2["TotalSales"] > 500000)]
     21  ',
     22    @input_data_1 = N'
     23  SELECT t.Name AS Territories, MONTH(h.OrderDate) AS OrderMonths, CONVERT(float, h.Subtotal) AS Sales
     24  FROM Sales.SalesOrderHeader h INNER JOIN Sales.SalesTerritory t
     25      ON h.TerritoryID = t.TerritoryID
     26  WHERE YEAR(h.OrderDate) = 2013;
     27    '
     28    WITH RESULT SETS((Territories NVARCHAR(100), OrderMonths NCHAR(3), AvgSales MONEY, MaxSales MONEY, MinSales MONEY, TotalSales MONEY));
     29  GO
```

Commands completed successfully.

(4 rows affected)

Total execution time: 00:00:00

	Territories	OrderMonths	AvgSales	MaxSales	MinSales	TotalSales
1	Canada	May	30047.0996	98638.2745	672.294	510880.6924
2	Canada	Jun	10133.9889	66845.6546	22.93	607974.5323
3	Canada	Jul	2529.1495	96485.4353	2.29	733453.3501
4	Canada	Oct	2709.1418	97647.72	2.29	804615.1034

SQL Server 機器學習服務

傳遞參數給 Python 處理資料

sp_execute_external_script

用自訂預存程序傳遞參數給 Python

```
CREATE PROCEDURE dbo.GetSalesTotals
  (@low INT, @high INT)
AS
  SET NOCOUNT ON;
  DECLARE @pyScript NVARCHAR(MAX);
  SET @pyScript = N'
# 依據 Territories 群組計算銷售總計
df1 = InputDataSet
df2 = df1.groupby("Territories", as_index=False).sum()

# 增加 Ratings 欄位
df2["Ratings"] = "Good"
df2["Ratings"][df2["Sales"] < LowSales] = "Poor"
df2["Ratings"][df2["Sales"] > HighSales] = "Excellent"

# 回傳匯總後的 Dataset
OutputDataSet = df2';
```

定義 Python 變數

```
DECLARE @sqlQuery NVARCHAR(MAX);
SET @sqlQuery = N'
  SELECT t.Name AS Territories, CAST(h.Subtotal AS FLOAT) AS Sales
  FROM Sales.SalesOrderHeader h INNER JOIN Sales.SalesTerritory t
    ON h.TerritoryID = t.TerritoryID;';
GO

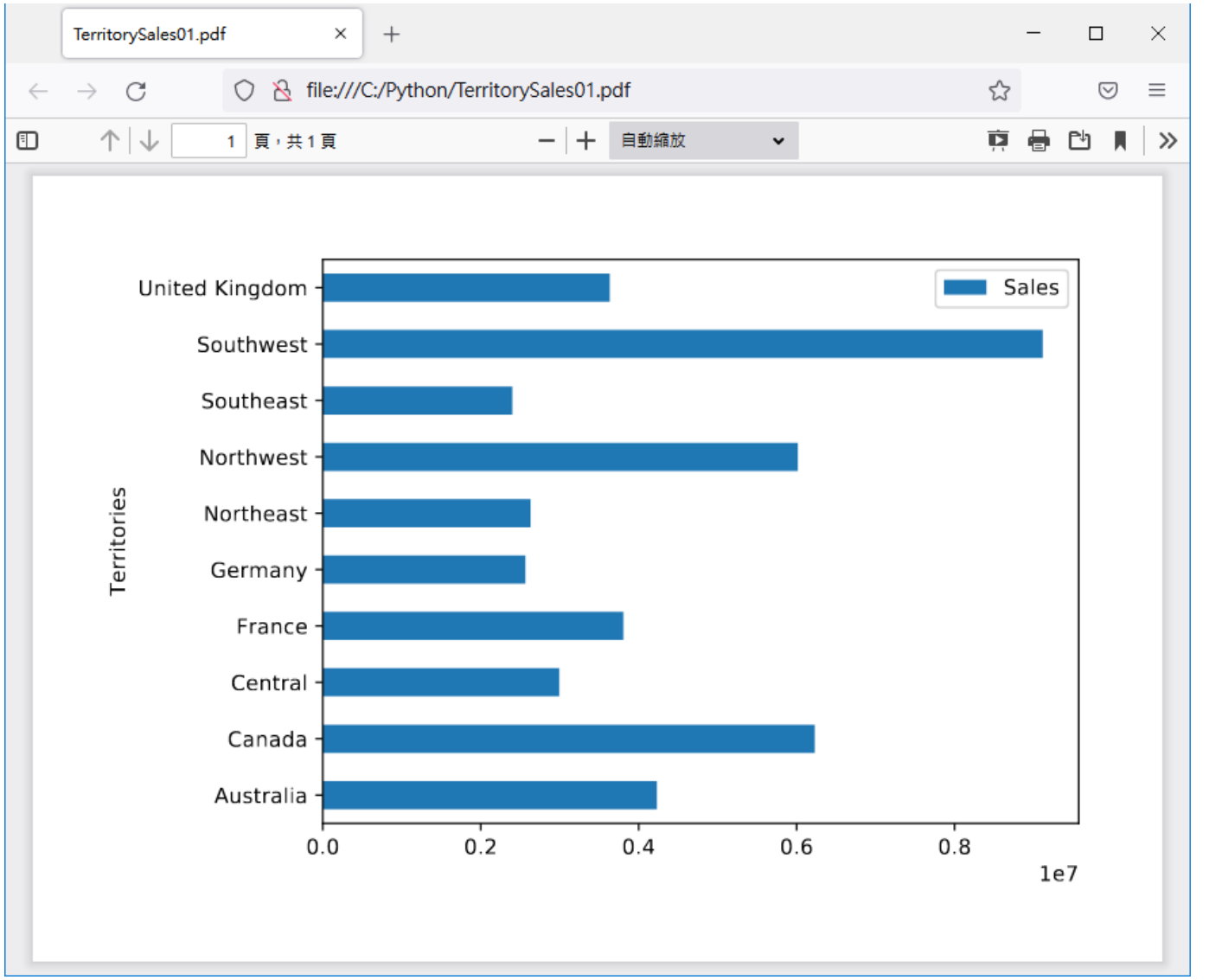
EXEC sp_execute_external_script
  @language = N'Python',
  @script = @ pyScript,
  @input_data_1 = @sqlQuery,
  @params = N'@LowSales FLOAT, @HighSales FLOAT',
  @LowSales = @low,
  @HighSales = @high
  WITH RESULT SETS((Territories NVARCHAR(50), TotalSales MONEY,
Ratings NVARCHAR(20)));
GO
```

指定預存程序參數傳遞給 Python 變數

SQL Server 資料庫的查詢結果
繪製 Python Matplotlib 統計圖表

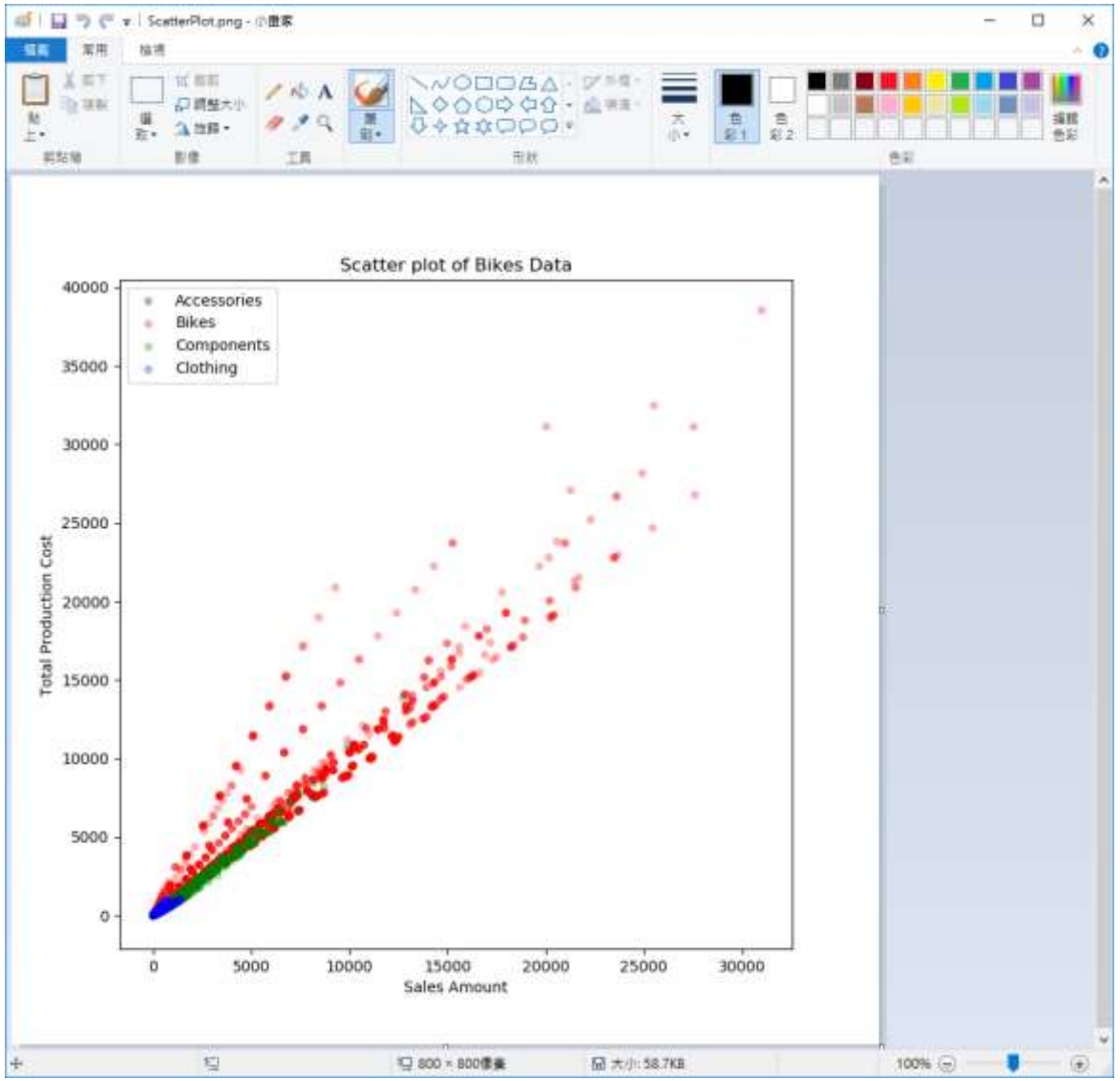
Python matplotlib 繪製統計圖表

輸出 pdf 檔案



Python matplotlib 繪製統計圖表

輸出 png 檔案



Python matplotlib 繪製統計圖表

matplotlib.pyplot.savefig

`matplotlib.pyplot.savefig(*args, **kwargs)` [source]

Save the current figure.

Call signature:

```
savefig(fname, *, dpi='figure', format=None, metadata=None,
        bbox_inches=None, pad_inches=0.1,
        facecolor='auto', edgecolor='auto',
        backend=None, **kwargs
       )
```

On this page

savefig()

Examples using

matplotlib.pyplot.savefig

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.savefig.html#

Python matplotlib 繪製統計圖表

直接輸出檔案要賦予輸出檔案路徑甚麼權限？

賦予 Everyone 有寫入權限？

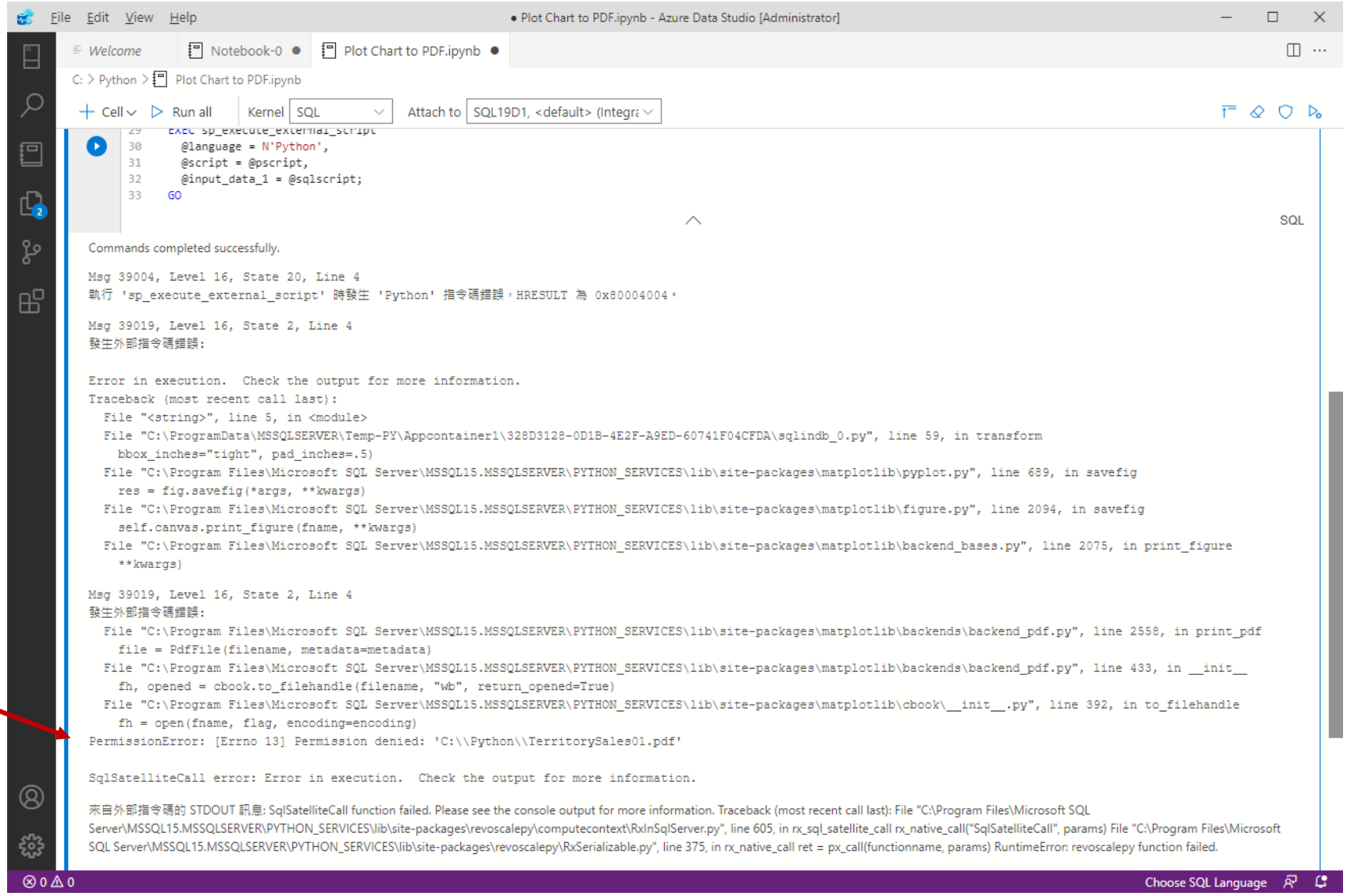
賦予 SQL Server 服務帳戶有寫入權限？

賦予 SQL Server Launchpad 服務帳戶有寫入權限？



Python matplotlib 輸出檔案

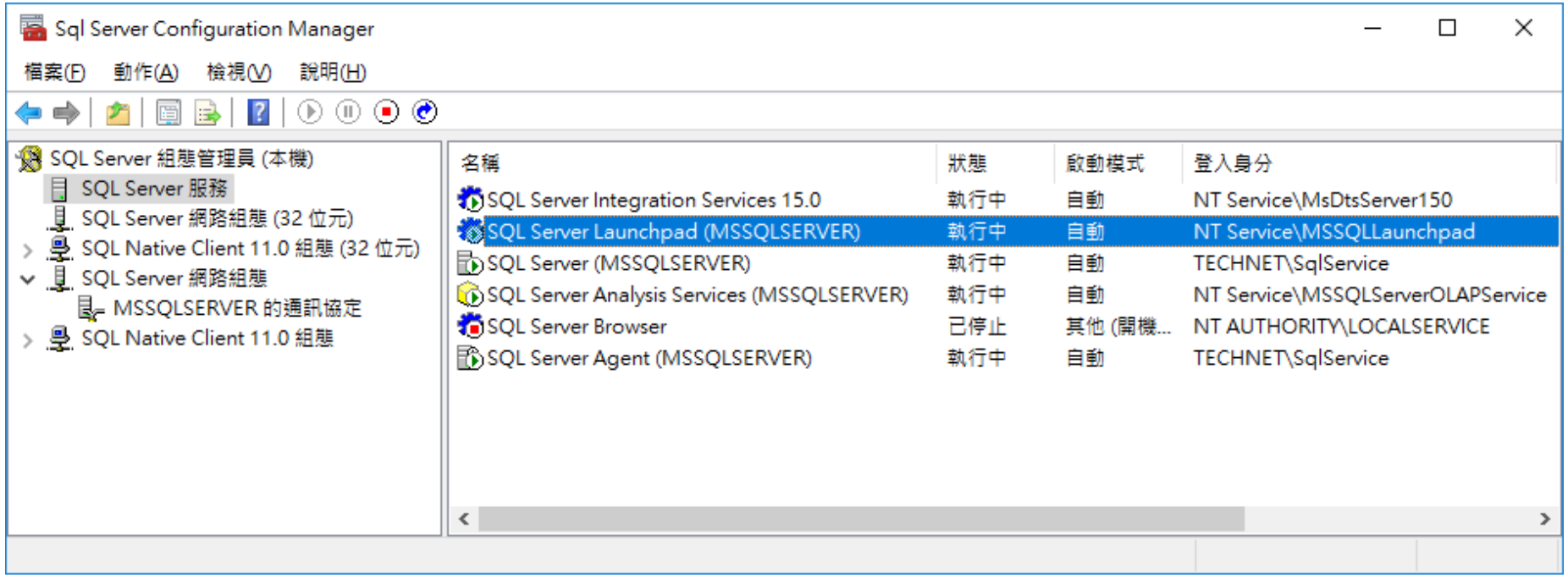
輸出檔案路徑權限不足，Permission denied



Python 執行結果輸出檔案

賦予輸出檔案路徑寫入權限

- 無效
  - 賦予 SQL Server Launchpad 服務帳戶或 SQL Server 服務帳戶 對輸出的檔案路徑有寫入權限

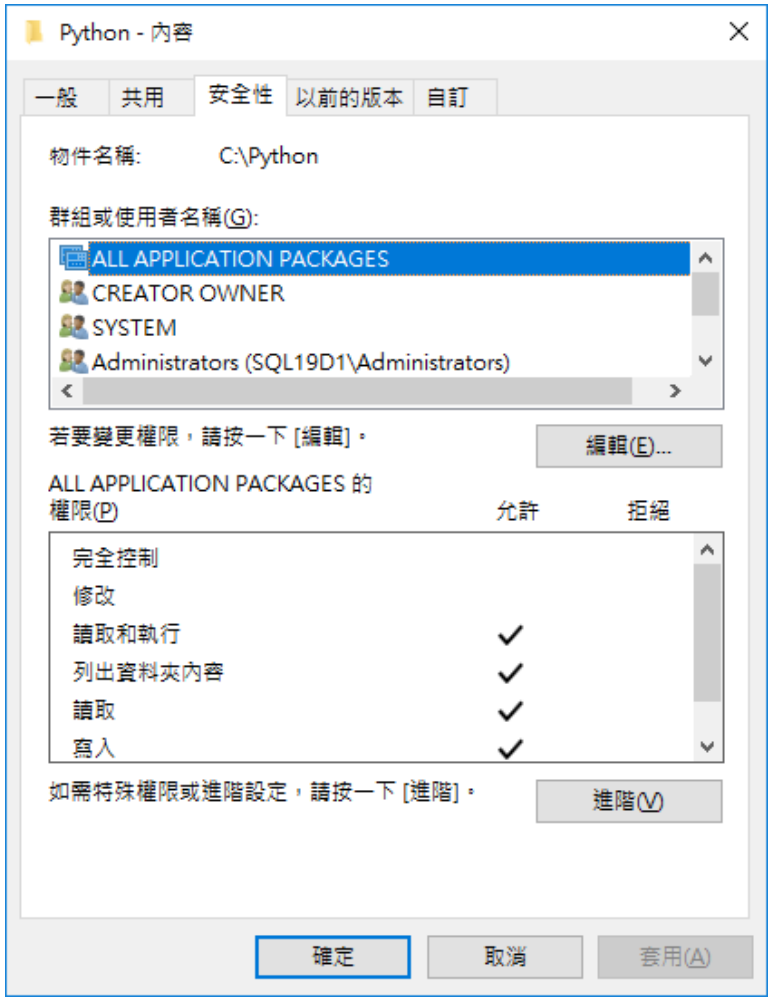


Python 執行結果輸出檔案

賦予輸出檔案路徑寫入權限

- 賦予 ALL APPLICATION PACKAGES 對輸出的檔案路徑有 Write 權限





Python matplotlib 繪製統計圖表

輸出 pdf 檔案

```
DECLARE @pyScript NVARCHAR(MAX);
SET @pyScript = N'
# 匯入 matplotlib 模組和 numpy 模組
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

# 定義 DataFrame df
df = InputDataSet.groupby("Territories").sum()

# 建立折線圖物件 pt，顏色: 海軍藍，透明度： 0.8 (0.0~1.0 數字愈小愈透明)，線條寬度 2
pt = df.plot(color="navy", alpha=.8, linewidth=2)

# 指定折線圖標題樣式，圖例： 隱藏，格線： 淺灰色虛線
pt.set_title(label="Sales by Territory", y=1.04,
  family="Times New Roman", fontsize=14, weight=800, color="navy")
pt.legend().set_visible(False)
pt.grid(color="slategray", alpha=.5, linestyle="dotted",
linewidth=.5)

# 格式化 X 軸 set_xticks() 指定間隔
pt.set_xlabel("Regional Territories", labelpad=20, fontsize=12,
color="navy")
pt.set_xticks(np.arange(10))
pt.set_xticklabels(labels=df.index, fontsize=9, color="navy",
rotation=45, horizontalalignment="right")
# 格式化 Y 軸，數字 千分位
pt.set_ylabel("Total Sales", labelpad=20, fontsize=12, color="navy")
pt.set_yticklabels(labels=df["Sales"], fontsize=9, color="navy")
pt.get_yaxis().set_major_formatter(matplotlib.ticker.FuncFormatter(
lambda x, p: format(int(x), ",")))

# 儲存橫條圖 PDF，邊界 0.8 英吋
plt.savefig("C:\Python\TerritorySales04.pdf",
  bbox_inches="tight", pad_inches=.8)';

DECLARE @sqlQuery NVARCHAR(MAX);
SET @sqlQuery = N'
  SELECT t.Name AS Territories, CONVERT(FLOAT, h.Subtotal) AS Sales
  FROM Sales.SalesOrderHeader h INNER JOIN Sales.SalesTerritory t
    ON h.TerritoryID = t.TerritoryID
  WHERE YEAR(h.OrderDate) = 2013;';

EXEC sp_execute_external_script
  @language = N'Python',
  @script = @pyScript,
  @input_data_1 = @sqlQuery;
GO
```

Python matplotlib 繪製統計圖表

結合 SSRS 報表呈現 png 檔案

