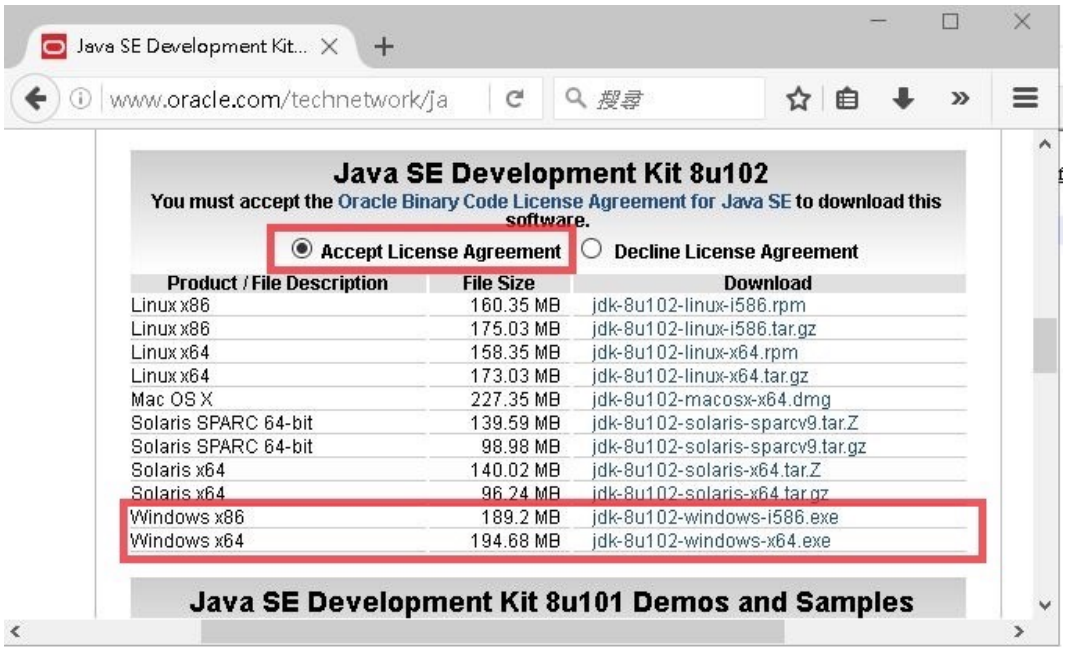
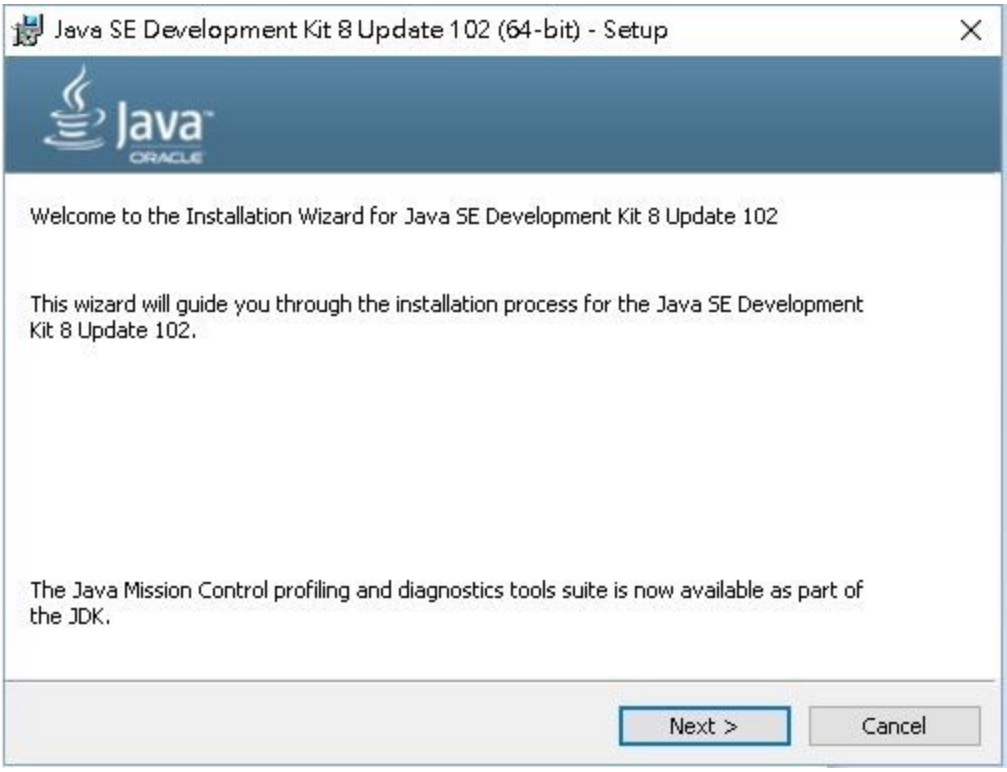
本文件將說明如何安裝Spark本機開發環境，以下畫面皆為Windows環境截圖，Linux環境步驟亦相同(圖片為Spark-2.0.1版本之抓圖，請自行將路徑改為spark-2.1.0)

1. download and install Java JDK8 (網址)

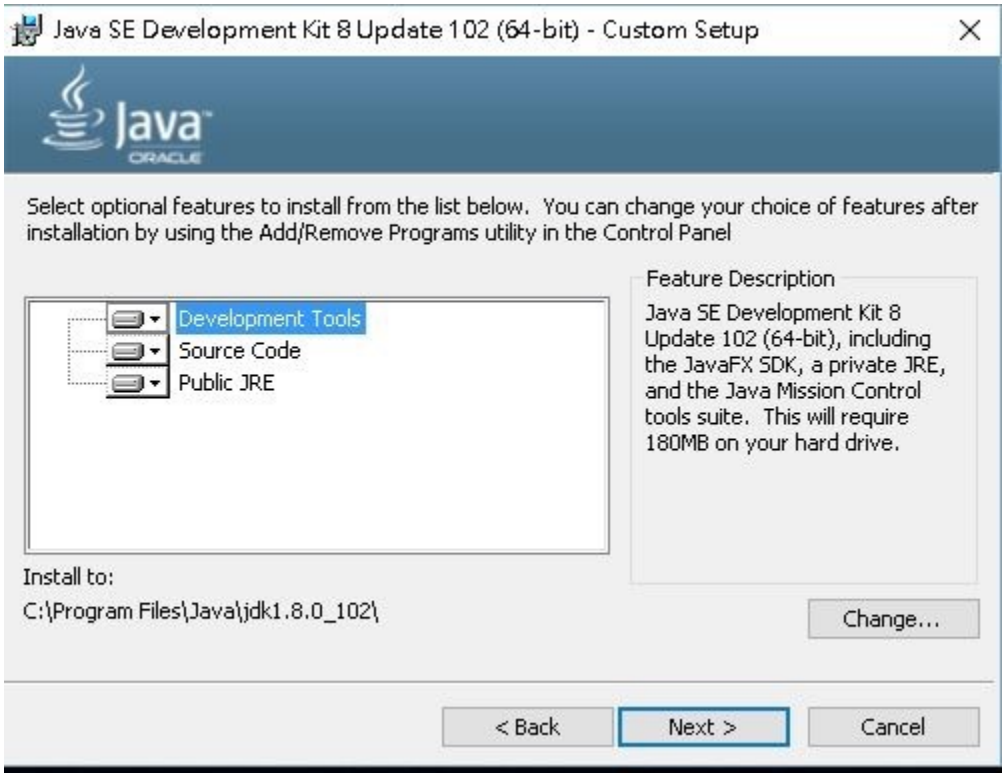
-到Java網站下載JDK8u102



-執行安裝，只需一直「Next」即可完成



-選擇安裝路徑，按「Next」即可



-選擇安裝路徑，按「Next」即可

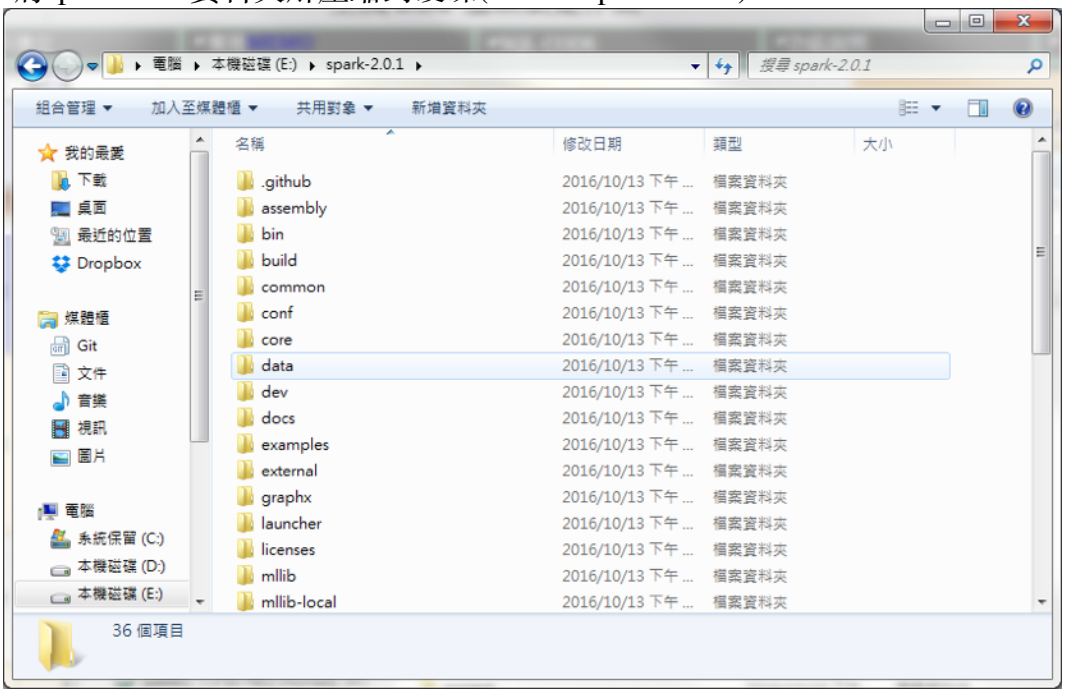
-Java安裝中

-安裝完成，按「Close」結束程式

-安裝完成後，可至cmd，輸入 java –version 檢視是否可正常運行

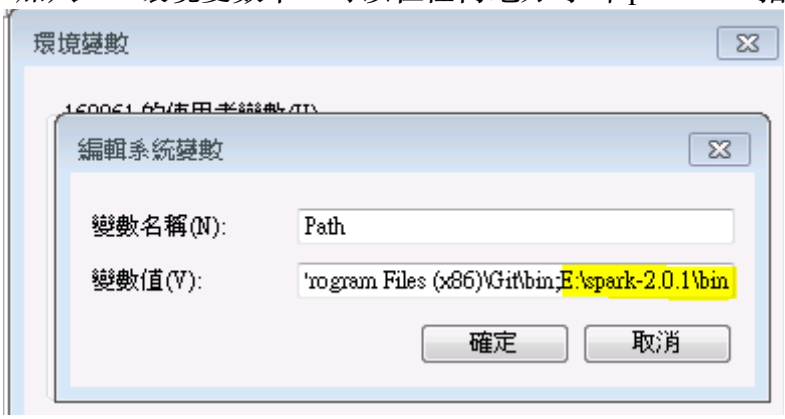
2. 至https://goo.gl/6JY7nX 下載spark-2.1.0withR.zip，and unzip
-因Spark本機版官方僅提供原始碼，此下載版本為預先build好之可執行環境，請安心下載。
-zip檔可用系統內建解壓縮程式或7zip開啟(若用windows內建解壓縮程式會有檔名過長無法建檔的問題，可選擇略過過長檔名之檔案，不影響課程實作)。

-將spark-2.1.0資料夾解壓縮到硬碟(EX: E:\spark-2.1.0\)

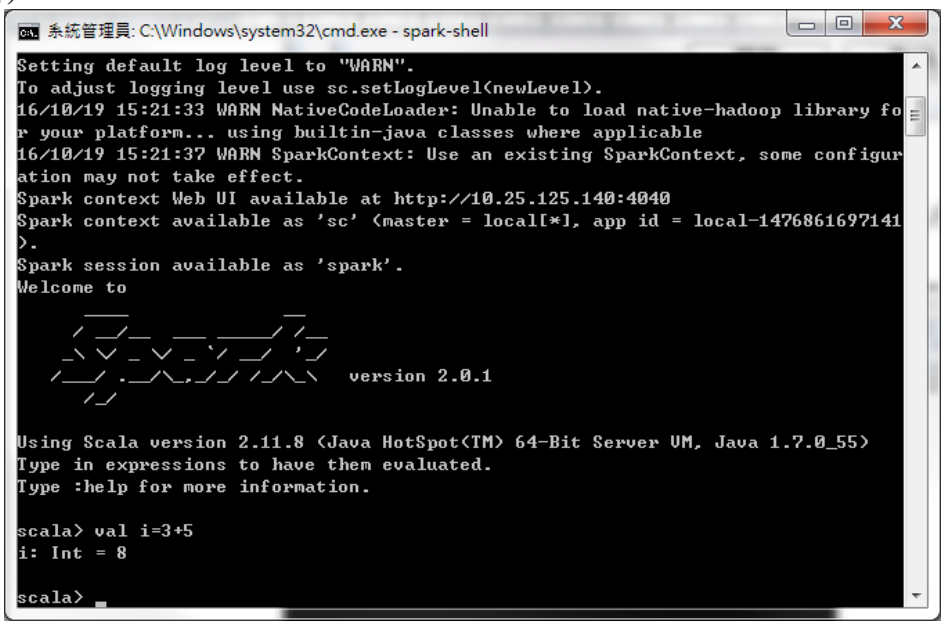


-開啟「命令提示字元」，到spark\bin底下，執行spark-shell

將spark\bin加入Path環境變數中，可以在任何地方呼叫spark-shell指令



-spark順利執行畫面(出現scala>表示已可以執行，可試著輸入簡單程式執行看看)

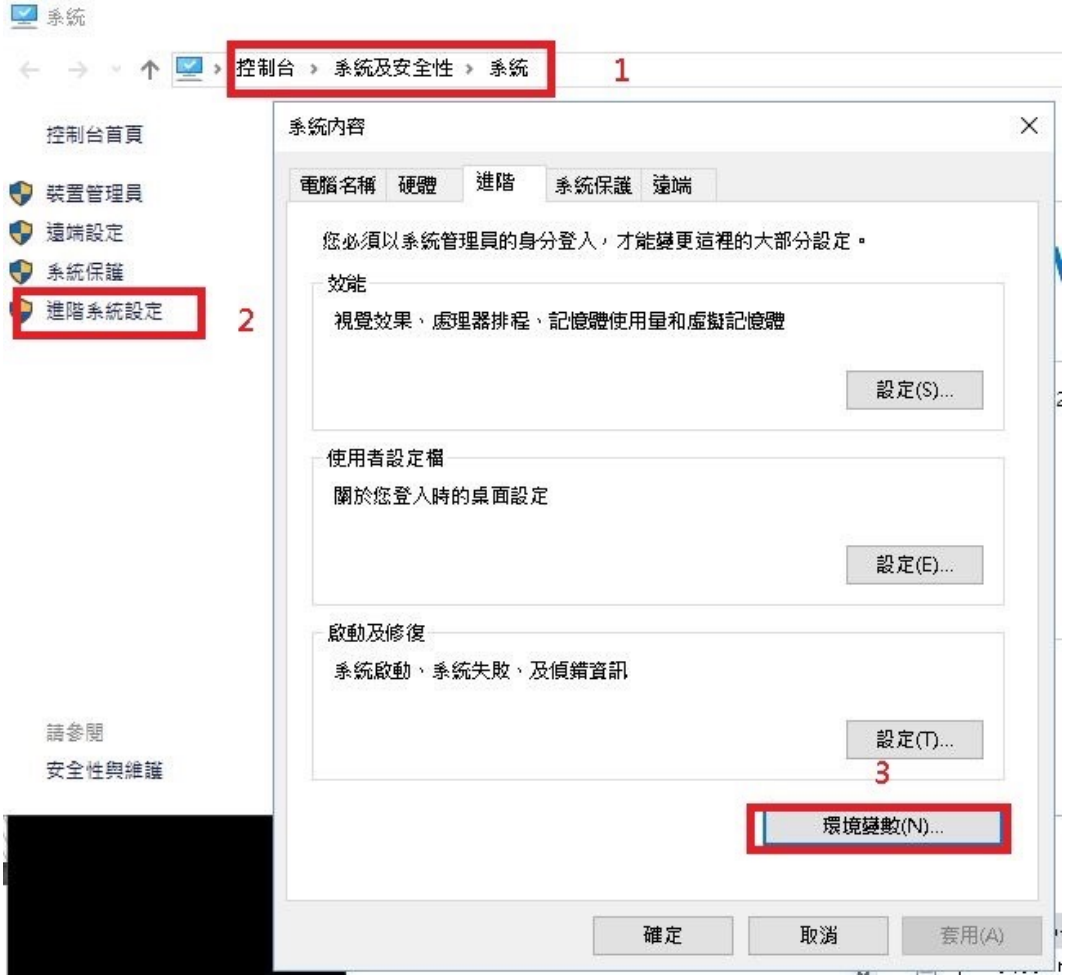


-若執行中遇到NullPointerException是因為缺乏在windows環境下模擬所需的winutils.exe ，以及Win10權限問題，檔案讀寫會出現問題，接下來步驟可以排除此問題(Linux環境不需安裝winutils.exe，可直接跳至步驟5)

3. 本機下載winutils.exe(網址)，在Spark目錄下建立 Hadoop\bin 資料夾，並將

winutils.exe放入

4. 設定環境變數
-HADOOP_HOME=C:\spark-2.1.0\hadoop\



➔注意HADOOP_HOME需設定為放置winutils.exe的上一層
➔可執行echo %HADOOP_HOME%來檢查環境變數是否設定完成

5. 測試以下指令

val txtRdd=sc.textFile("file:///E:/spark-2.1.0/README.md")
txtRdd.saveAsTextFile("file:///E:/tmp/txtRdd")
應該可以正常執行，並在 C:\tmp\ 底下產生txtRdd的資料夾

```
scala> val txtRdd=sc.textFile("file:///c:/spark-2.0.1/README.md")
txtRdd: org.apache.spark.rdd.RDD[String] = file:///c:/spark-2.0.1/README.md MapP
artitionsRDD[1] at textFile at <console>:24

scala> txtRdd.saveAsTextFile("File:///C:/tmp/txtRdd")
[Stage 0:>                                                          (0 + 0) / 2]
[Stage 0:>                                                          (0 + 2) / 2]
```

Win10可能在執行saveAsTextFile時仍會出現錯誤訊息，但不影響後續實作，可暫時忽略。

6. 欲結束Spark，於cmd中輸入 **：quit** 或 **CTRL+D**
查詢指令則是輸入 **:help**

完成Spark安裝後，安裝R for windows(https://cran.r-project.org/bin/windows/base/)；或依個人開發需求安裝RStudio亦可。

開啟R環境(以下以R-shell為例，使用RStudio步驟亦同)：

```
YungChuLeedeMBP:R yungchuanlee$
YungChuLeedeMBP:R yungchuanlee$
YungChuLeedeMBP:R yungchuanlee$
YungChuLeedeMBP:R yungchuanlee$ R

R version 3.3.2 (2016-10-31) -- "Sincere Pumpkin Patch"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R 是免費軟體，不提供任何擔保。
在某些條件下您可以將其自由散布。
用 'license()' 或 'licence()' 來獲得散布的詳細條件。

R 是個合作計劃，有許多人為之做出了貢獻。
用 'contributors()' 來看詳細的情況並且
用 'citation()' 會告訴您如何在出版品中正確地參照 R 或 R 套件。

用 'demo()' 來看一些示範程式，用 'help()' 來檢視線上輔助檔案，或
用 'help.start()' 透過 HTML 瀏覽器來看輔助檔案。
用 'q()' 離開 R。

>
```

依Spark官方文件(http://spark.apache.org/docs/latest/sparkr.html#starting-up-from-rstudio)說明，載入SparkR相關Lib、初始化SparkRSession

```
用 'help.start()' 透過 HTML 瀏覽器來看輔助檔案。
用 'q()' 離開 R。

> if (nchar(Sys.getenv("SPARK_HOME")) < 1) {
+   Sys.setenv(SPARK_HOME = "/Users/yungchuanlee/spark-2.1.0")
+ }
> library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK_HOME"), "R", "lib")))

Attaching package: 'SparkR'

The following objects are masked from 'package:stats':

    cov, filter, lag, na.omit, predict, sd, var, window

The following objects are masked from 'package:base':

    as.data.frame, colnames, colnames<-, drop, endsWith, intersect,
    rank, rbind, sample, startsWith, subset, summary, transform, union
> sparkR.session(master = "local[*]", sparkConfig = list(spark.driver.memory = "2g"))
Spark package found in SPARK_HOME: /Users/yungchuanlee/spark-2.1.0
Launching java with spark-submit command /Users/yungchuanlee/spark-2.1.0/bin/spark-submit   -
-driver-memory "2g" sparkr-shell /var/folders/zl/80ztm2nn4hl6ghjppf16dkt80000gn/T//RtmpIBu8Tr
/backend_porta8bd5f604b
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
17/01/25 00:41:13 WARN NativeCodeLoader: Unable to load native-hadoop library for your platfo
rm... using builtin-java classes where applicable
17/01/25 00:41:14 WARN SQLUtils: SparkR: enableHiveSupport is requested for SparkSession but
Spark is not built with Hive; falling back to without Hive support.
Java ref type org.apache.spark.sql.SparkSession id 1
>
```

完成Lib載入及SparkRSession初始後，即可開始練習後續指令(http://spark.apache.org/docs/latest/sparkr.html#creating-sparkdataframes)

```
> df <- as.DataFrame(faithful)
> str(df)
'SparkDataFrame': 2 variables:
 $ eruptions: num 3.6 1.8 3.333 2.283 4.533 2.883
 $ waiting  : num 79 54 74 62 85 55
> head(df)
  eruptions waiting
1     3.600      79
2     1.800      54
3     3.333      74
4     2.283      62
5     4.533      85
6     2.883      55
>
```