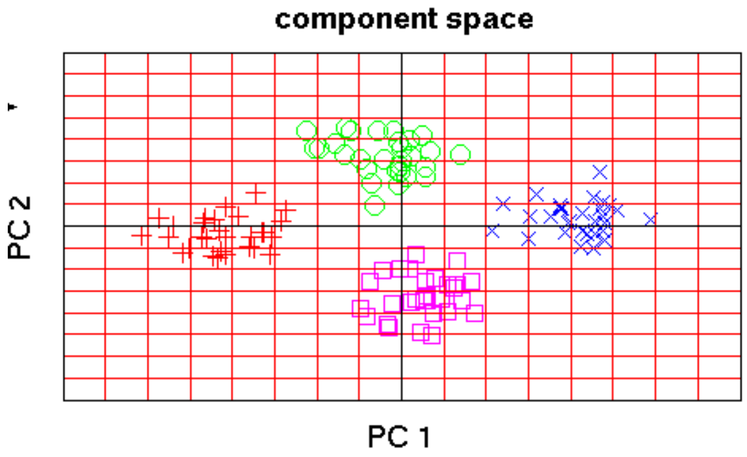
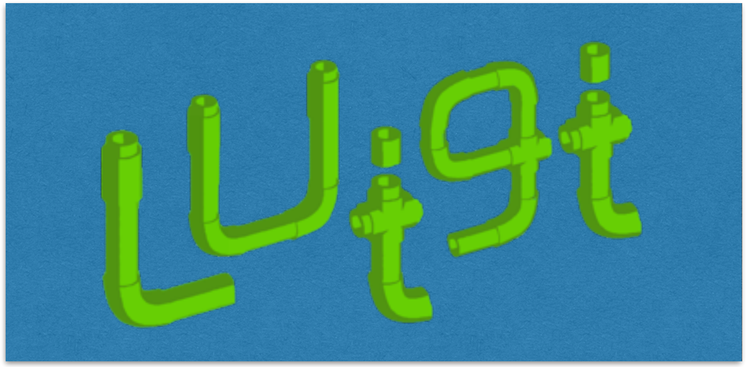
- **Introduction of Luigi**
  - **Digital Analysis Tasks**
  - **Example**

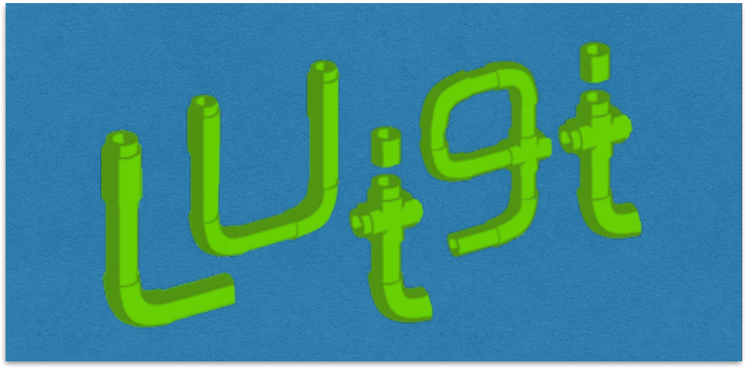


- **Introduction of PCA**
  - **Basic Introduction**
  - **hands on**

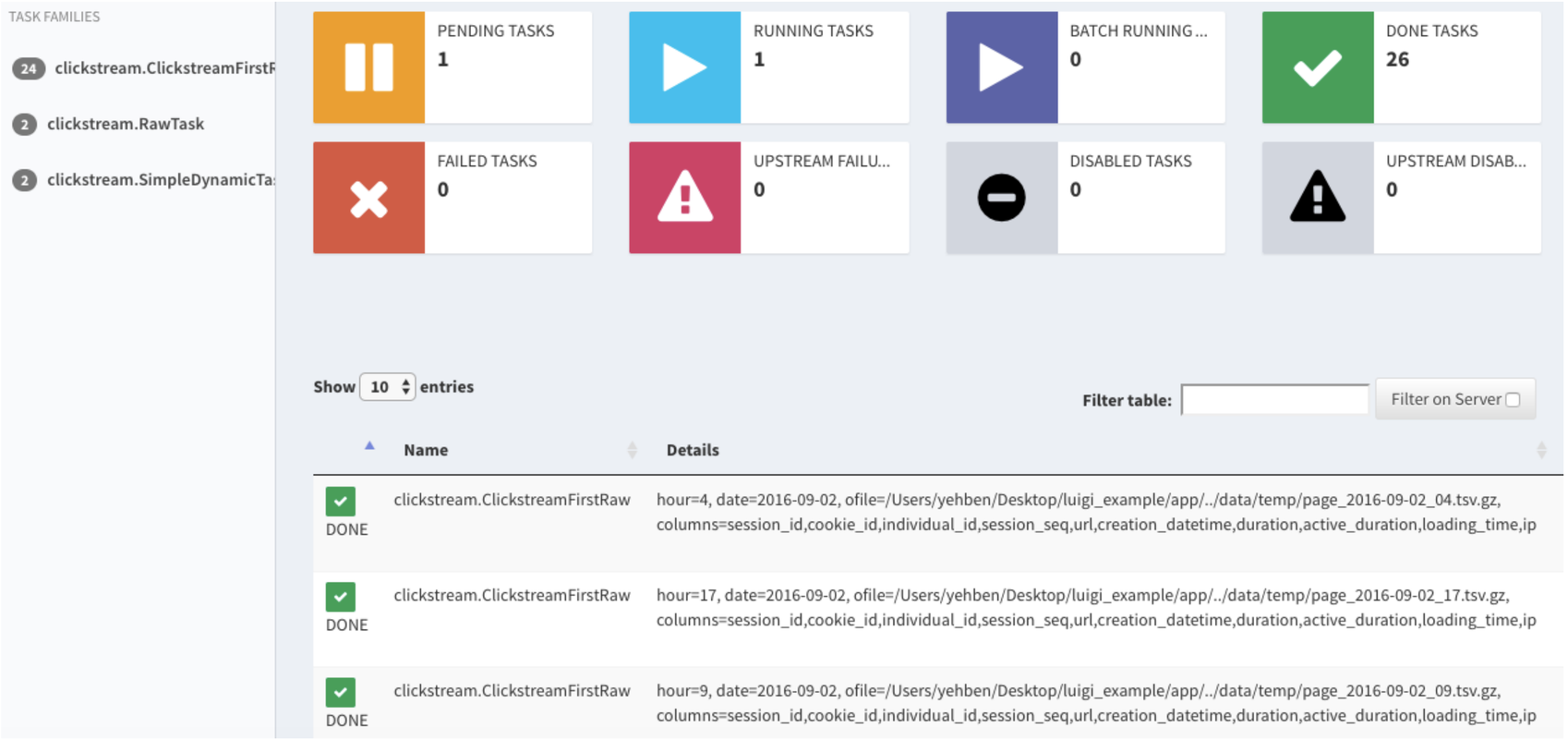
LLM
Tit

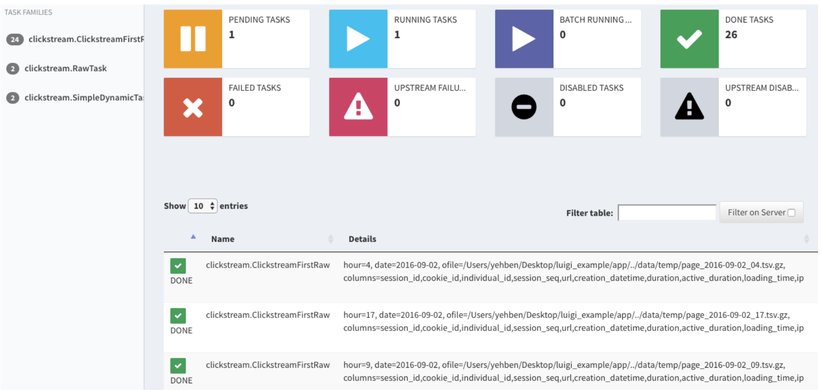


- Spotify
- 複雜批次處理任務管理系統
- 任務依賴管理、工作流管理、任務可視化...
- 需要依賴於外部的調度器來觸發工作流:crontab

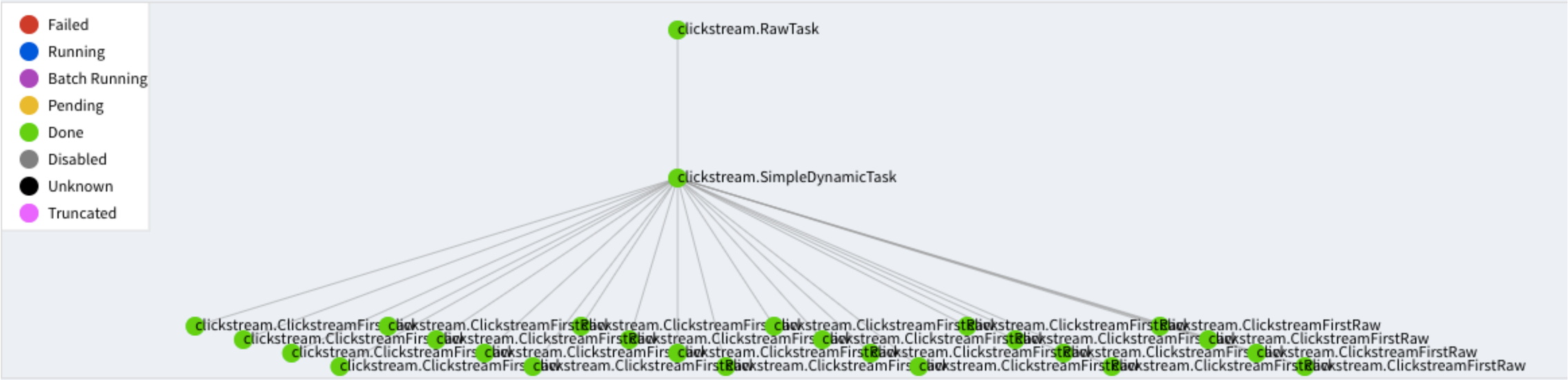


- Spotify
- 複雜批次處理任務管理系統
- 任務依賴管理、工作流管理、任務可視化...
- 需要依賴於外部的調度器來觸發工作流:crontab





- Spotify
- 複雜批次處理任務管理系統
- 任務依賴管理、工作流管理、任務可視化、錯誤故障處理機制
- 需要依賴於外部的調度器來觸發工作流:crontab

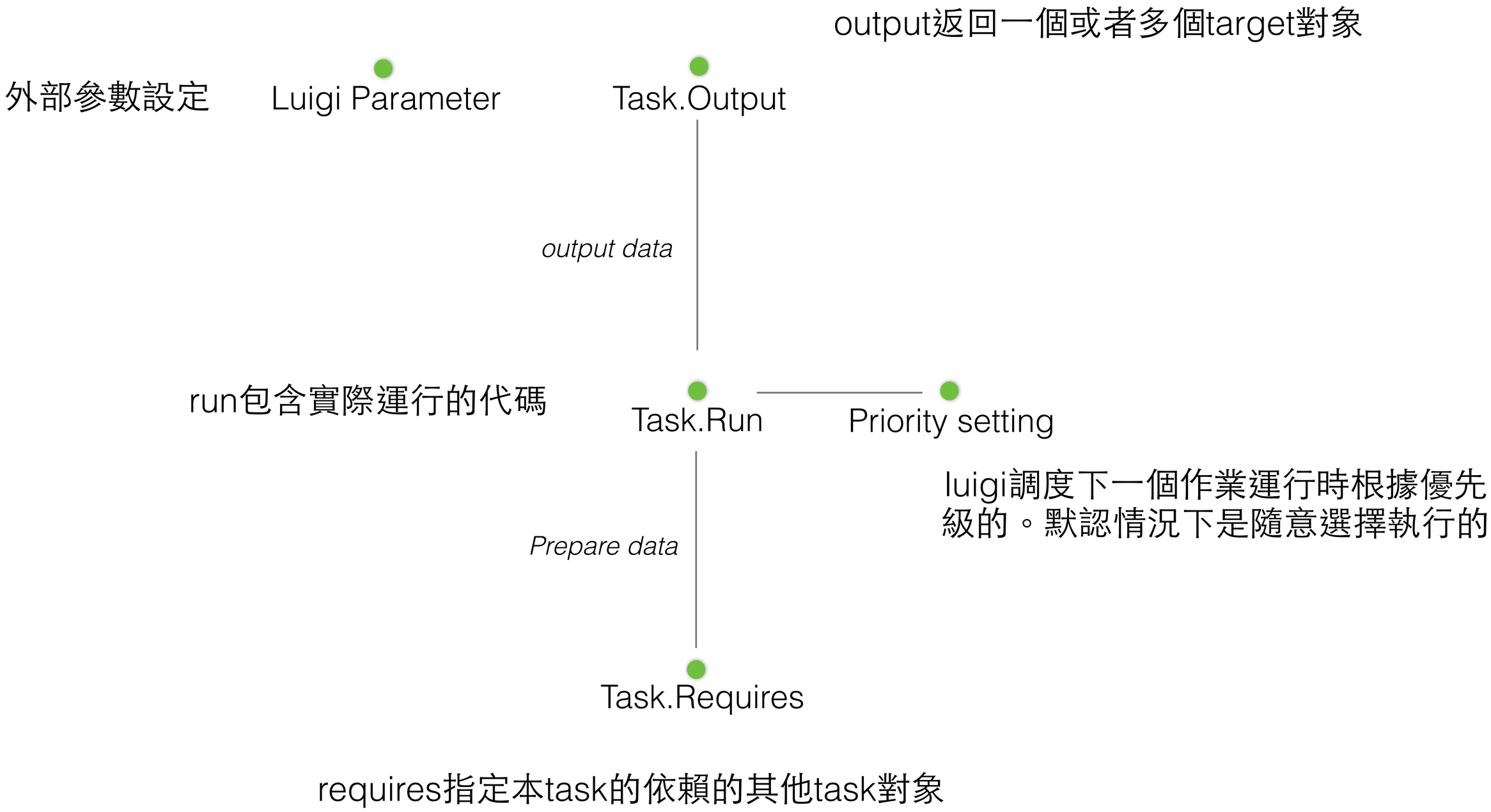


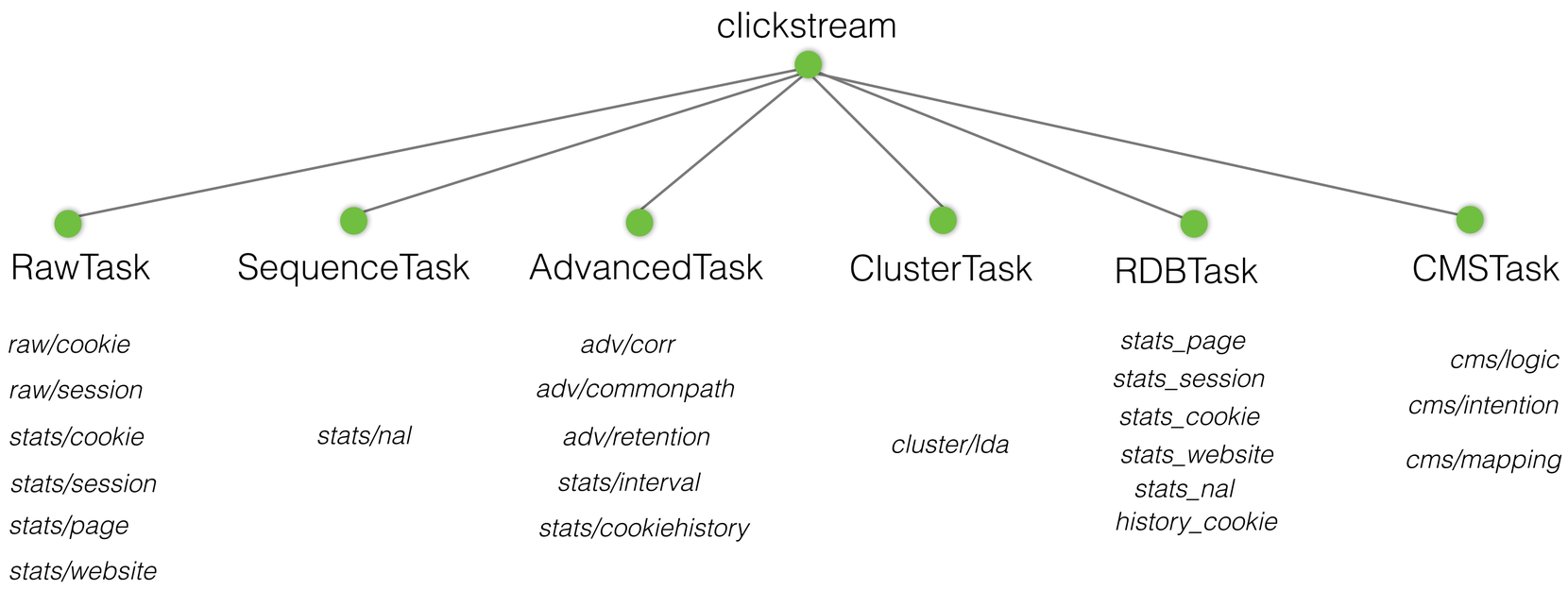

Task.Output
output data
Task.Run
Prepare data
Task.Requires

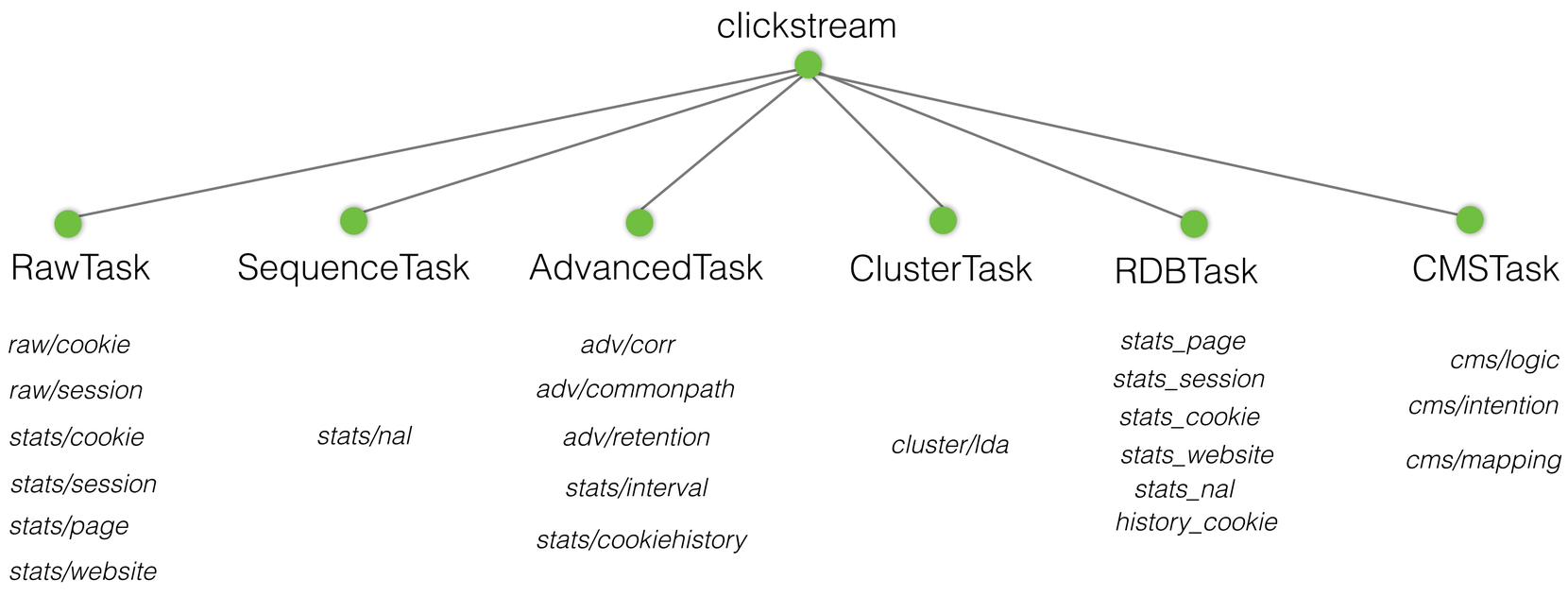
output返回一個或者多個target對象

run包含實際運行的代碼

requires指定本task的依賴的其他task對象


output返回一個或者多個target對象
外部參數設定
Luigi Parameter
Task.Output
output data
run包含實際運行的代碼
Task.Run
Priority setting
luigi調度下一個作業運行時根據優先級的。默認情況下是隨意選擇執行的
Prepare data
Task.Requires
requires指定本task的依賴的其他task對象


clickstream
RawTask
SequenceTask
AdvancedTask
ClusterTask
RDBTask
CMSTask
raw/cookie
raw/session
stats/cookie
stats/session
stats/page
stats/website
stats/nal
adv/corr
adv/commonpath
adv/retention
stats/interval
stats/cookiehistory
cluster/lda
stats_page
stats_session
stats_cookie
stats_website
stats_nal
history_cookie
cms/logic
cms/intention
cms/mapping



基本統計數據

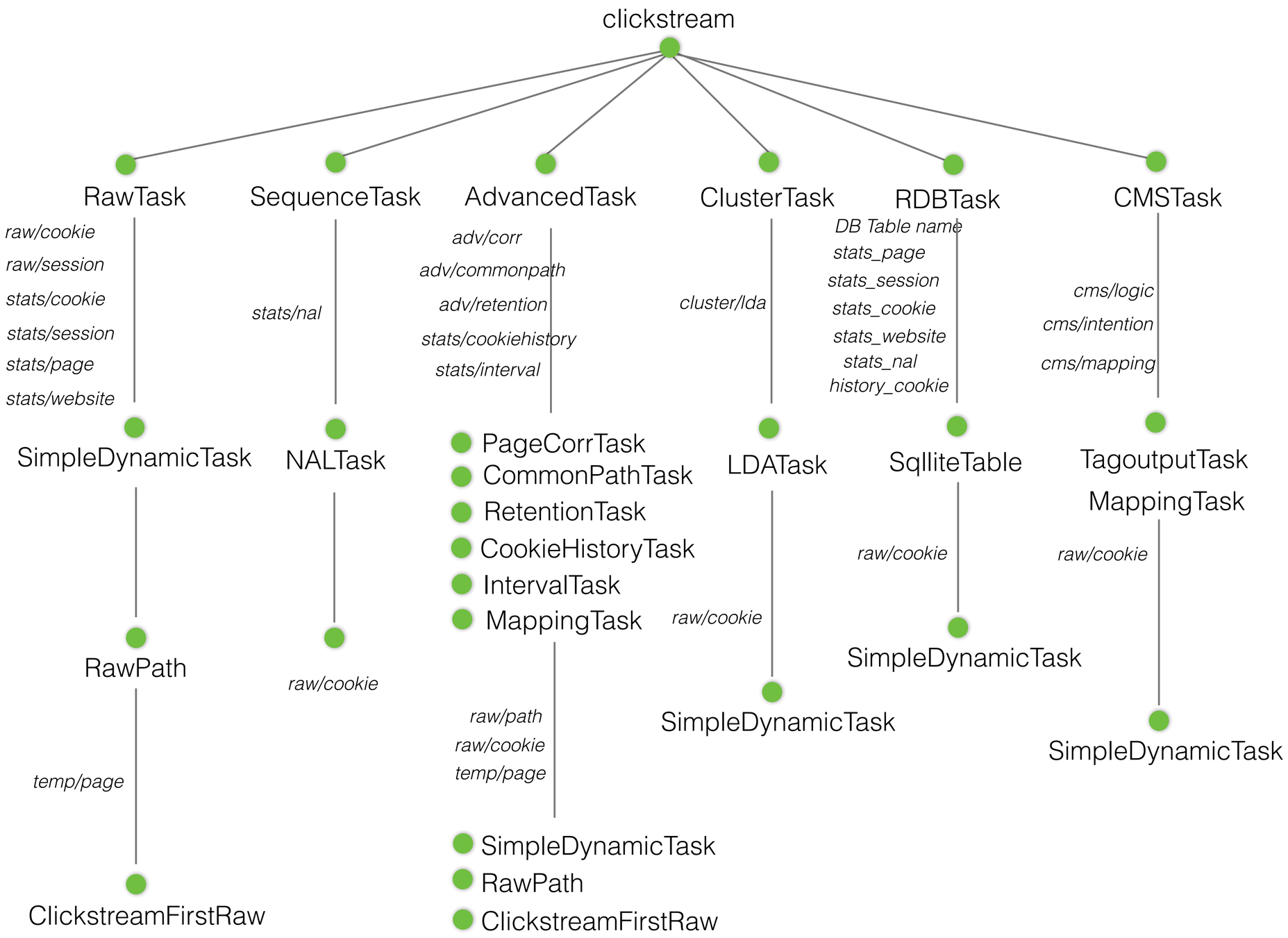
進階分析

DB

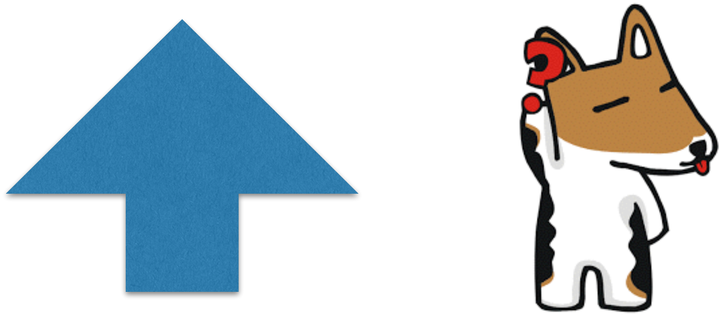
Cookie歷史

標籤分群

CMS線上標籤


clickstream
RawTask
raw/cookie
raw/session
stats/cookie
stats/session
stats/page
stats/website
SimpleDynamicTask
RawPath
temp/page
ClickstreamFirstRaw
SequenceTask
stats/nal
NALTask
raw/cookie
AdvancedTask
adv/corr
adv/commonpath
adv/retention
stats/cookiehistory
stats/interval
PageCorrTask
CommonPathTask
RetentionTask
CookieHistoryTask
IntervalTask
MappingTask
raw/path
raw/cookie
temp/page
SimpleDynamicTask
RawPath
ClickstreamFirstRaw
ClusterTask
cluster/lda
LDATask
raw/cookie
SimpleDynamicTask
RDBTask
DB Table name
stats_page
stats_session
stats_cookie
stats_website
stats_nal
history_cookie
SqlliteTable
raw/cookie
SimpleDynamicTask
CMSTask
cms/logic
cms/intention
cms/mapping
TagoutputTask
MappingTask
raw/cookie
SimpleDynamicTask

```
{'creation_datetime': '2016-09-02 23:32:34.915000',
 'logic1': {'logic1_網銀': 2, 'logic1_cub平台': 2},
 'cookie_id': 'a1e4bfb99a3342c696490797c9209999',
 'loading_duration': -4.0,
 'duration': 24786.0,
 'active_duration': 24786.0,
 'logic2': {'logic2_其他': 2, 'logic2_cub平台': 2},
 'intention': {'intention_其他': 4},
 'individual_id': 'None',
 'function': {'function_功能_login': 2, 'function_登入': 2}}
```


```
session_id      cookie_id       individual_id   session_seq     url     creation_datetime
      duration        active_duration loading_time    ip
35553330        9b1fc926bc374459a793193352630d0f        A2B811D5EC7C580C14      2       https://www.mybank.com.tw/mybank        20
16-09-01 00:05:29.767000        20729   20729   -1      ip
35553330        9b1fc926bc374459a793193352630d0f        A2B811D5EC7C580C14      3       https://www.mybank.com.tw/mybank/quicklink
s/home        2016-09-01 00:05:50.528000      88286   88286   3204    ip
```

RawTask(*mode=range, interval, lib)

raw/cookie_interval1


```
{'creation_datetime': '2016-09-02 23:32:34.915000',
 'logic1': {'logic1_網銀': 2, 'logic1_cub平台': 2},
 'cookie_id': 'a1e4bfb99a3342c696490797c9209999',
 'loading_duration': -4.0,
 'duration': 24786.0,
 'active_duration': 24786.0,
 'logic2': {'logic2_其他': 2, 'logic2_cub平台': 2},
 'intention': {'intention_其他': 4},
 'individual_id': 'None',
 'function': {'function_功能_login': 2, 'function_登入': 2}}
```


SimpleDynamicTask(interval, filter_app=True, ofile, lib)
lib:basic.raw.cookie

RawPath(column, ofile, interval, hour, ntype)

temp/page


```
session_id      cookie_id       individual_id   session_seq     url     creation_datetime
      duration        active_duration loading_time    ip
35553330        9b1fc926bc374459a793193352630d0f        A2B811D5EC7C580C14      2       https://www.mybank.com.tw/mybank        20
16-09-01 00:05:29.767000      20729   20729   -1      ip
35553330        9b1fc926bc374459a793193352630d0f        A2B811D5EC7C580C14      3       https://www.mybank.com.tw/mybank/quicklink
s/home        2016-09-01 00:05:50.528000      8828G   8828G   3204    ip
```


*mode : single - merge whole data during interval
range - separate interval tasks by day,hour…

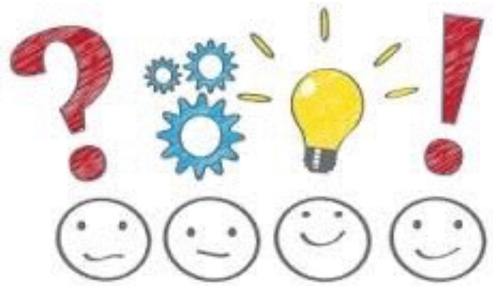
*interval1 : date, interval

*SimpleDynamicTask(RawPath)

ClickstreamFirstRaw(date, hour, ofile, column)

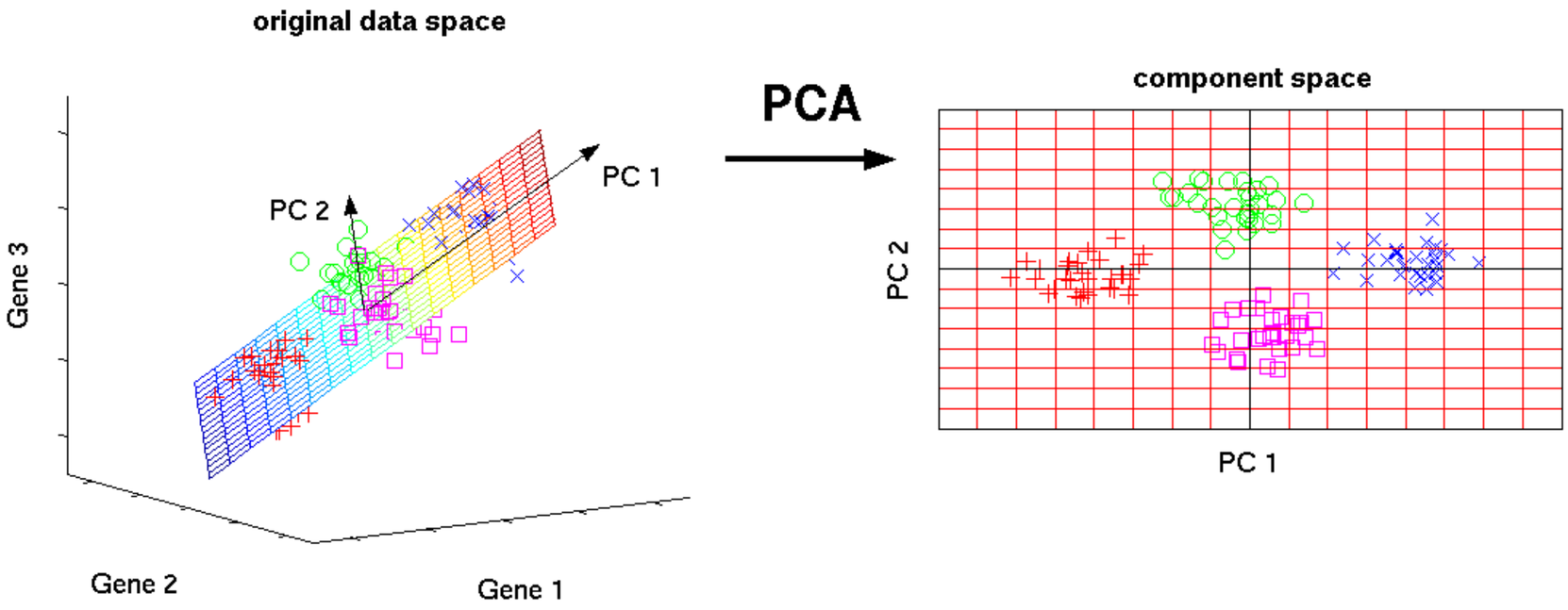
ClickstreamFirstRaw

order	Tables	Description	SQL (column)
*	temp/page_interval	合併所有session瀏覽page所有相關訊息，並加入cookie_id, individual_id, ipAddress等資訊	session_id cookie_id individual_id session_seq url creation_datetime function logic intention duration active_duration loading_time ip
1	VP_OP_ADC.page VP_OP_ADC.pagesummary	抓取在期間內所有session瀏覽page的資訊，並合併page停留時間	SELECT A.sessionnumber, A.pagesequenceinsession, A.pagelocation, A.eventtimestamp, B.PageViewTime, B.PageViewActiveTime, COALESCE(B.PageLoadDuration,-1) FROM VP_OP_ADC.page{table} A LEFT JOIN VP_OP_ADC.pagesummary{table} B ON A.sessionnumber = B.sessionnumber AND A.pageinstanceid = B.pageinstanceid WHERE A.eventtimestamp >= '{date} {hour}:00:00' AND A.eventtimestamp < '{date} {hour}:59:59' ORDER BY A.sessionnumber, A.pagesequenceinsession".format(table=table, date=self.date, hour="{:
2	VP_OP_ADC.vistior	抓取期間內session對應的cookieID，一個session只取一筆	SELECT sessionnumber, MAX(CookieUniqueVisitorTrackingId) FROM VP_OP_ADC.visitor{table} WHERE eventtimestamp >= '{date_1}' AND eventtimestamp < '{date} {hour}:59:59' GROUP BY sessionnumber".format(table=table, date_1=date_1, date=self.date, hour="{:02d}".format(self.hour))
3	VP_OP_ADV.individual	抓取期間內session對應Profileuiid，一個session只取一筆	SELECT sessionnumber, MAX(ProfileUiid) FROM VP_OP_ADC.individual{table} WHERE eventtimestamp >= '{date} {hour}:00:00' AND eventtimestamp < '{date} {hour}:59:59' AND ProfileUiid NOT LIKE '%XXXX%' GROUP BY sessionnumber".format(table=table, date=self.date, hour="{: 02d}".format(self.hour))
4	VP_OP_ADV.sessionstart	抓取期間內session對應IPAddress	SELECT sessionnumber, DeviceIPAddress FROM VP_OP_ADC.sessionstart{table} WHERE eventtimestamp >= '{date_1}' AND eventtimestamp < '{date} {hour}:59:59'".format(table=table, date_1=date_1, date=self.date, hour="{:02d}".format(self.hour))



Principal
Components
Analysis

- 降低所研究的數據空間的維數
- 通過線性投影，將高维的數據映射到低维的空間中
- 將眾多具有一定相關性指標，組成新互相無關的綜合指標
- 多應用在一致性分析、影像分析等等



Principal Components Analysis

- 對原始數據指標變數進行變換後形成了彼此相互獨立的主成分。
- 消除各變數之間的共線性, 減少變數的個數,利於後續的分析。
- 用較少綜合指標依然能代表原眾多變數，維度選擇建議需達85%數據信息。

PCA致命缺點是什麼?

怎樣的資料不適合使用PCA?

Principal Components Analysis

- 對原始數據指標變數進行變換後形成了彼此相互獨立的主成分。
- 消除各變數之間的共線性, 減少變數的個數,利於後續的分析。
- 用較少綜合指標依然能代表原眾多變數，維度選擇建議需達85%數據信息。

- 主成分的解釋其含義一般多少帶有點模糊性，不像原始變數的含義那麼清楚、確切
- 變數彼此間相關性不高，則資料做主成分分析就不合適。

image load

```
from sklearn.decomposition import PCA
from pylab import *
from skimage import data, io, color

link = "/Users/yehben/Desktop/Brown.jpeg"
Brown_gray = io.imread(link,as_grey=True)
io.imshow(Brown_gray)
xlabel('Original Image')
io.show()
```

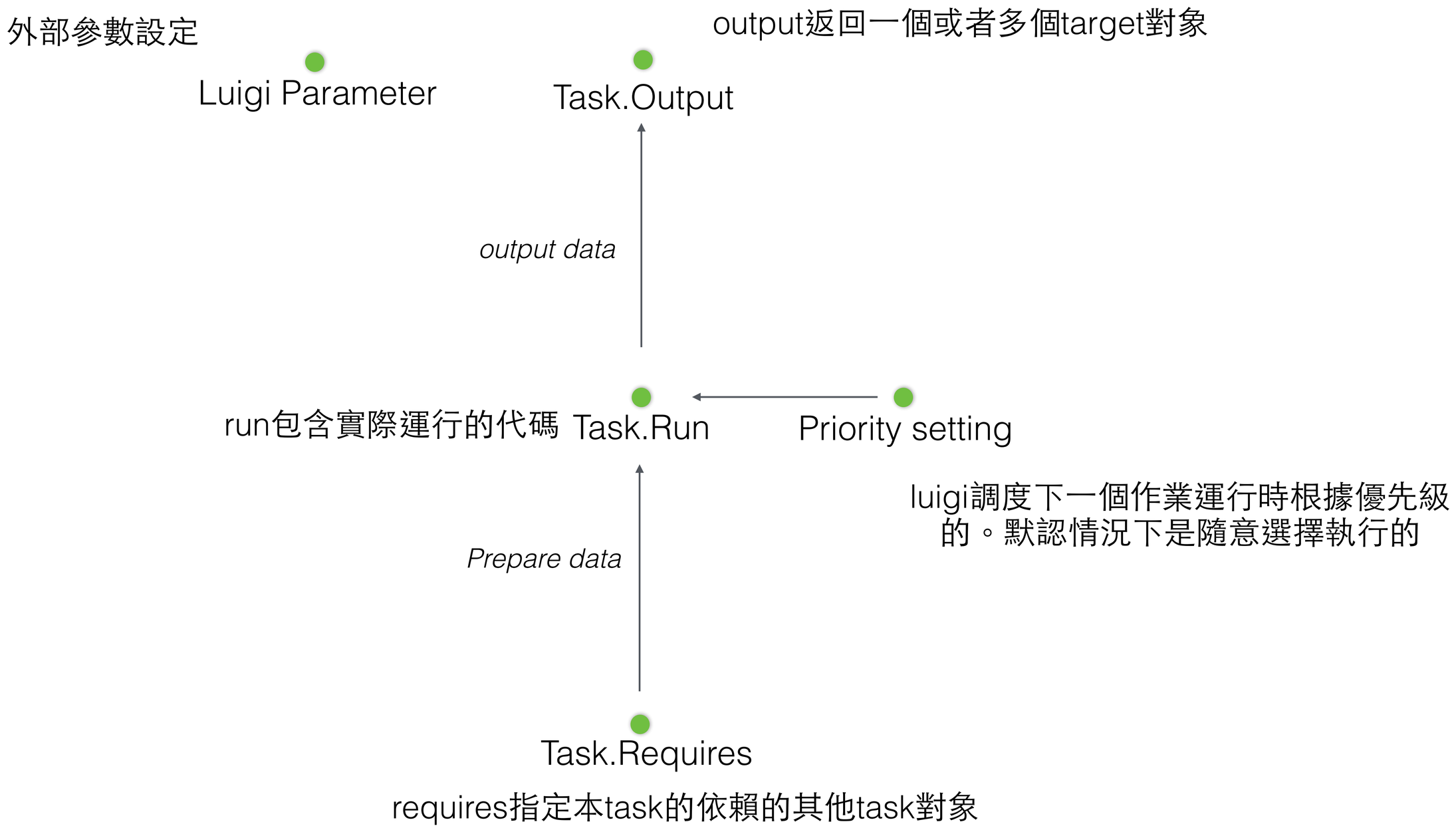
images.jpeg

model fit

```
n_comp = 5
pca = PCA(n_components = n_comp)
pca.fit(Brown_gray)
Brown_gray_pca = pca.fit_transform(Brown_gray)
Brown_gray_restored = pca.inverse_transform(Brown_gray_pca)
io.imshow(Brown_gray_restored)
#xlabel('Restored image n_components = %s' %n_comp)
io.show()
```

Variance Ratio

```
print(pca.explained_variance_ratio_)
```


外部參數設定
Luigi Parameter
output返回一個或者多個target對象
Task.Output
output data
run包含實際運行的代碼 Task.Run
Priority setting
luigi調度下一個作業運行時根據優先級的。默認情況下是隨意選擇執行的
Prepare data
Task.Requires
requires指定本task的依賴的其他task對象

Task.Output 指出output路徑

photopath:欲分析圖片檔路徑
ratio:累積主成份解釋能力

Luigi Parameter

Task.Run

1. image array require
2. PCA model fit
3. save output image

Task.Requires

1. Load外部image檔案
2. 解析檔案成numpy array
3. 存成numpy檔案

Luigi Parameter

```
class ImageAnalysis(luigi.Task):
    task_namespace = 'image'
    photopath = luigi.Parameter()
    ratio = luigi.IntParameter()
```

```
def output(self):
    outfile = os.path.join(BASEPATH_photo, "Brown_{}.png".format(self.ratio))
    return luigi.LocalTarget(outfile)
```

Task.Output

```
def requires(self):
    imagepath = os.path.join(BASEPATH_photo, "Image_{}.npy".format(self.ratio))
    yield Imageload(photopath = self.photopath, ratio = self.ratio, imagepath = imagepath)

def run(self):
    var_ratio = int(self.ratio)/100
    for input in self.input():
        Brown_gray = numpy.load(input.fn)
    pca_all = PCA(n_components = len(Brown_gray[0]))
    pca_all.fit(Brown_gray)
    n_comp = cc_ratio(var_ratio,pca_all.explained_variance_ratio_)
    pca = PCA(n_components = n_comp)
    pca.fit(Brown_gray)
    Brown_gray_pca = pca.fit_transform(Brown_gray)
    Brown_gray_restored = pca.inverse_transform(Brown_gray_pca)
    plt.imsave(self.output().fn, Brown_gray_restored, cmap=plt.cm.gray)
```

Task.Run

```
class Imageload(luigi.Task):
    task_namespace = 'image'
    photopath = luigi.Parameter()
    ratio = luigi.IntParameter()
    imagepath = luigi.Parameter()

    def run(self):
        link = self.photopath
        Brown_gray = io.imread(link,as_grey=True)
        numpy.save(self.output().fn,Brown_gray)

    def output(self):
        return luigi.LocalTarget(self.imagepath)
```

Task.Requires

Luigi Parameter

```
class ImageAnalysis(luigi.Task):
    task_namespace = 'image'
    photopath = luigi.Parameter()
    ratio = luigi.IntParameter()
```

```
def output(self):
        outfile = os.path.join(BASEPATH_photo, "Brown_{}.png".format(self.ratio))
        return luigi.LocalTarget(outfile)
```

```
def requires(self):
    imagepath = os.path.join(BASEPATH_photo, "Image_{}.npy".format(self.ratio))
    yield Imageload(photopath = self.photopath, ratio = self.ratio, imagepath = imagepath)

def run(self):
    var_ratio = int(self.ratio)/100
    for input in self.input():
        Brown_gray = numpy.load(input.fn)
    pca_all = PCA(n_components = len(Brown_gray[0]))
    pca_all.fit(Brown_gray)
    n_comp = cc_ratio(var_ratio,pca_all.explained_variance_ratio_)
    pca = PCA(n_components = n_comp)
    pca.fit(Brown_gray)
    Brown_gray_pca = pca.fit_transform(Brown_gray)
    Brown_gray_restored = pca.inverse_transform(Brown_gray_pca)
    plt.imsave(self.output().fn, Brown_gray_restored, cmap=plt.cm.gray)
```

```
class Imageload(luigi.Task):
    task_namespace = 'image'
    photopath = luigi.Parameter()
    ratio = luigi.IntParameter()
    imagepath = luigi.Parameter()

    def run(self):
        link = self.photopath
        Brown_gray = io.imread(link,as_grey=True)
        numpy.save(self.output().fn,Brown_gray)

    def output(self):
        return luigi.LocalTarget(self.imagepath)
```

Task.Output

Task.Run

Task.Requires

Luigi Parameter

```python
class ImageAnalysis(luigi.Task):
    task_namespace = 'image'
    photopath = luigi.Parameter()
    ratio = luigi.IntParameter()
```

Task.Output

```python
def output(self):
        outfile = os.path.join(BASEPATH_photo, "Brown_{}.png".format(self.ratio))
        return luigi.LocalTarget(outfile)
```

Task.Run

```python
def requires(self):
    imagepath = os.path.join(BASEPATH_photo, "Image_{}.npy".format(self.ratio))
    yield Imageload(photopath = self.photopath, ratio = self.ratio, imagepath = imagepath)

def run(self):
    var_ratio = int(self.ratio)/100
    for input in self.input():
        Brown_gray = numpy.load(input.fn)
    pca_all = PCA(n_components = len(Brown_gray[0]))
    pca_all.fit(Brown_gray)
    n_comp = cc_ratio(var_ratio,pca_all.explained_variance_ratio_)
    pca = PCA(n_components = n_comp)
    pca.fit(Brown_gray)
    Brown_gray_pca = pca.fit_transform(Brown_gray)
    Brown_gray_restored = pca.inverse_transform(Brown_gray_pca)
    plt.imsave(self.output().fn, Brown_gray_restored, cmap=plt.cm.gray)
```

Task.Requires

```python
class Imageload(luigi.Task):
    task_namespace = 'image'
    photopath = luigi.Parameter()
    ratio = luigi.IntParameter()
    imagepath = luigi.Parameter()

    def run(self):
        link = self.photopath
        Brown_gray = io.imread(link,as_grey=True)
        numpy.save(self.output().fn,Brown_gray)

    def output(self):
        return luigi.LocalTarget(self.imagepath)
```