

# ラーニングアナリティクス基盤 ユーザマニュアル

国立情報学研究所

# ラーニングアナリティクス基盤 ユーザマニュアル

## 目次

### 1. 基本操作

---

1.1 JupyterHubの操作 3p

1.2 Supersetの操作 10p

### 2. 実践操作

---

2.1 動画の視聴回数を調べる 17p

2.2 小テストの結果から成績優秀者を特定する 21p

2.3 小テストの結果と動画の視聴履歴から退学者を予測する 27p

# 1. 基本操作

---

## 1.1 JupyterHubの操作

---

1.1.1 JupyterHubへログインする 3p

---

1.1.2 JupyterHubからログアウトする 6p

---

1.1.3 notebookを利用する 7p

---

## 1.2 Supersetの操作

---

1.2.1 Supersetへログインする 10p

---

1.2.2 Supersetからログアウトする 12p

---

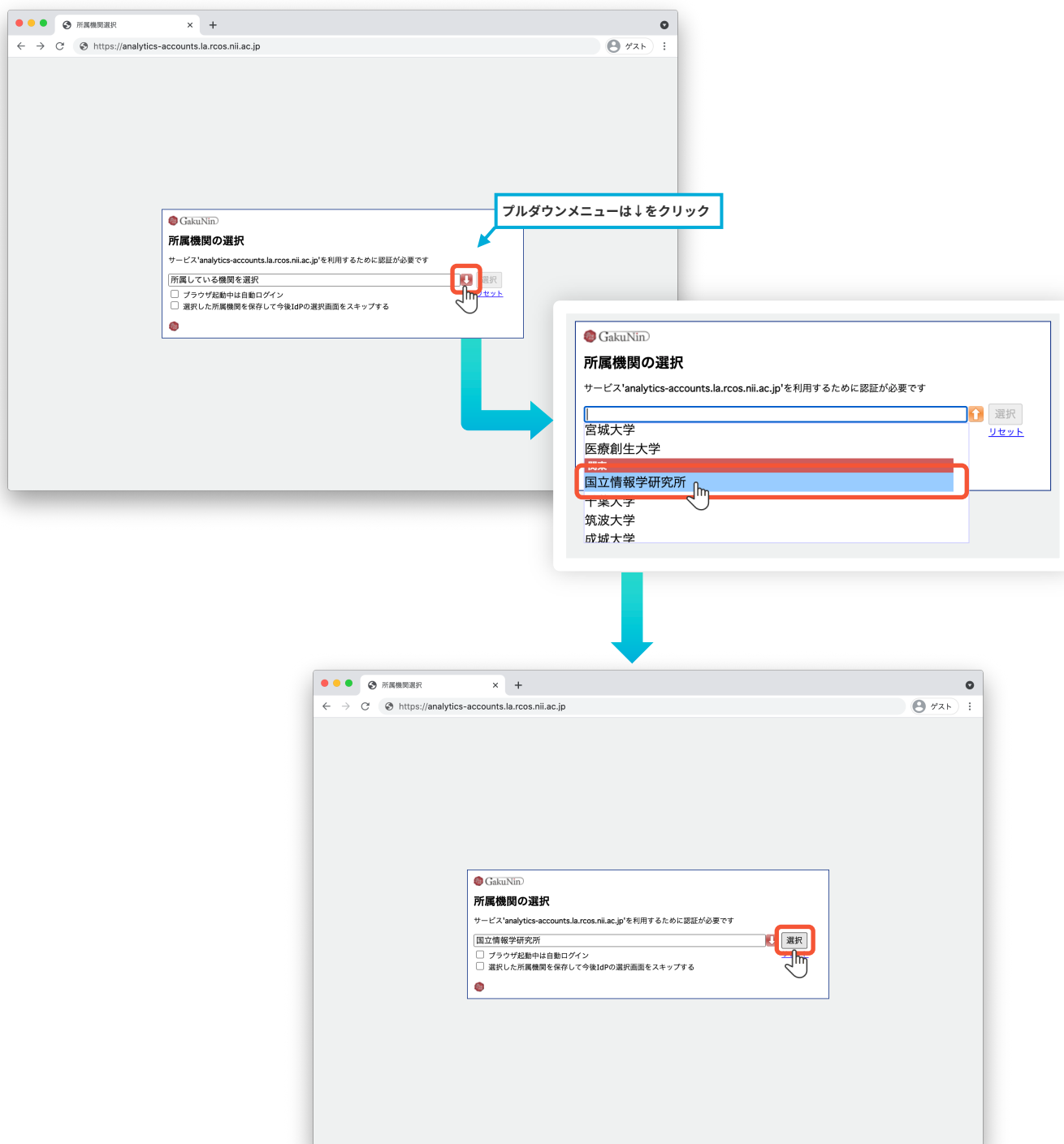
1.2.3 Supersetを利用する 13p

## 1.1.1 | JupyterHubへログインする

### JupyterHubとは

管理者が用意したPythonチュートリアル用 Jupyter notebookを、ユーザーがブラウザから実行できるようにするもの。

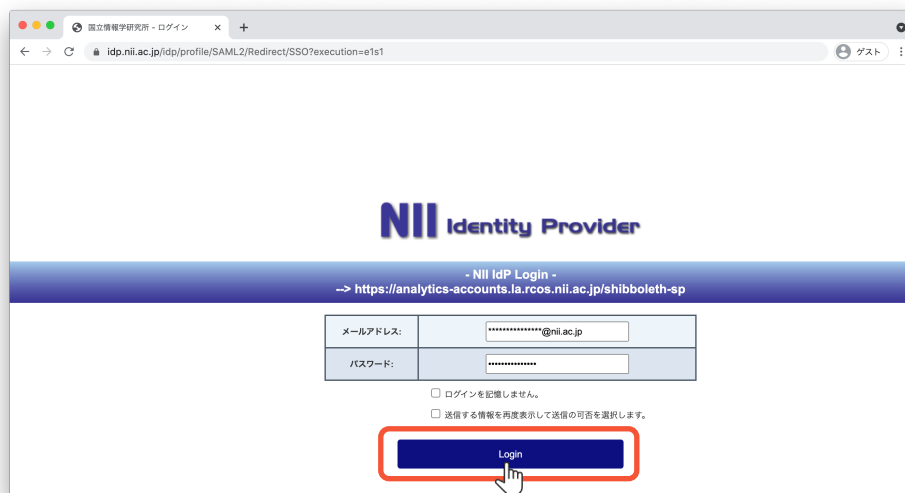
- ▶ <https://analytics-accounts.la.rcos.nii.ac.jp>へアクセスし、自分が所属する機関名を選択する。



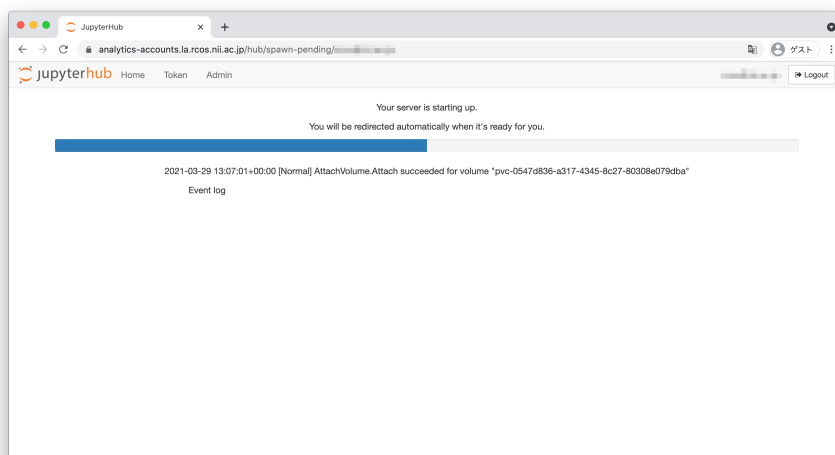
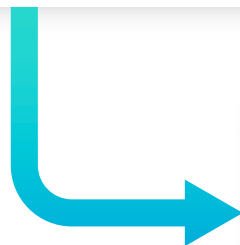
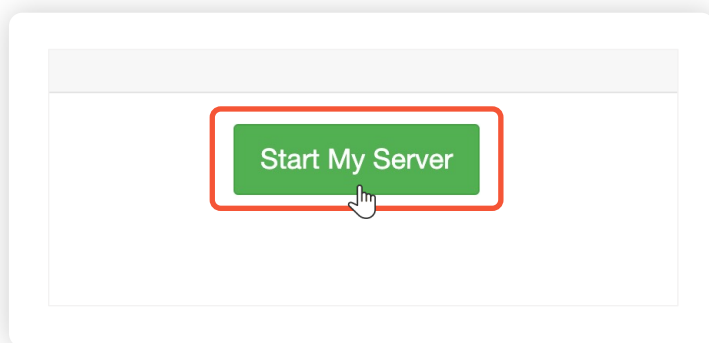


## 1. 基本操作 > 1.1 JupyterHubの操作 > 1.1.1 JupyterHubへログインする

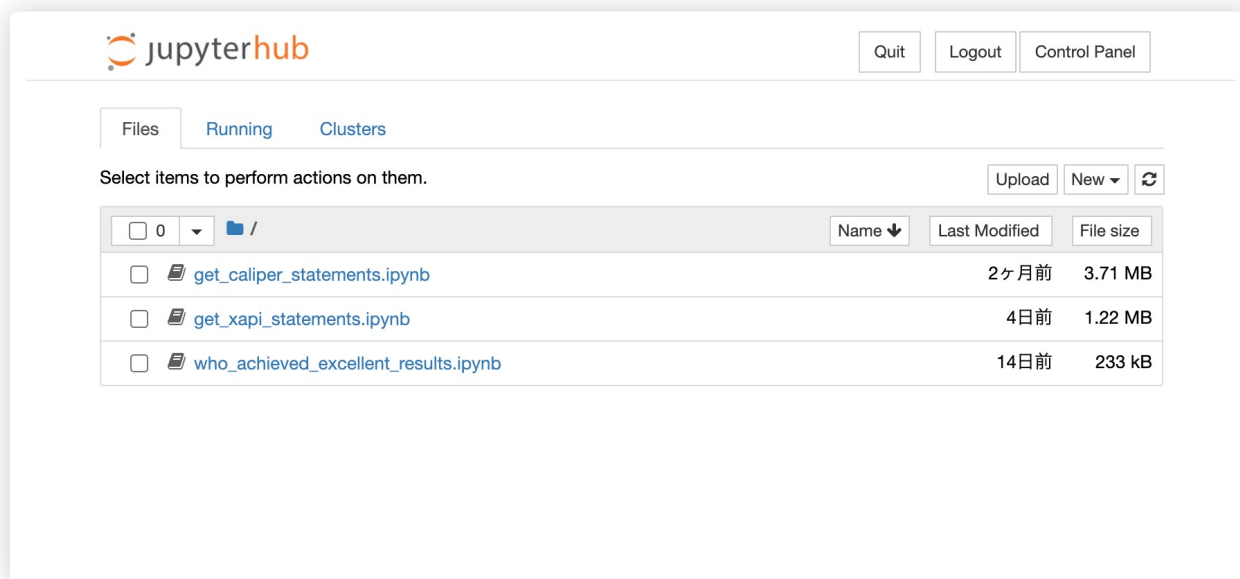
- ▶ 選択した所属機関のログインページに遷移するので、付与されているメールアドレスとパスワードを入力し[Login]ボタンをクリックする。



初回アクセス時など、Jupyterノートブックサーバが起動していない場合には[Start My Server]ボタンが表示される。この場合には、ボタンをクリックしサーバを起動する。



▶ サーバ起動後に、ノートブックの一覧画面が表示される。



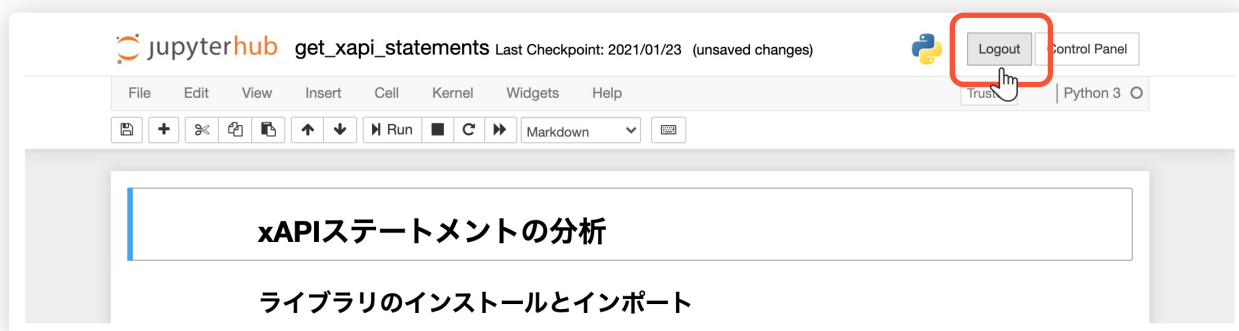
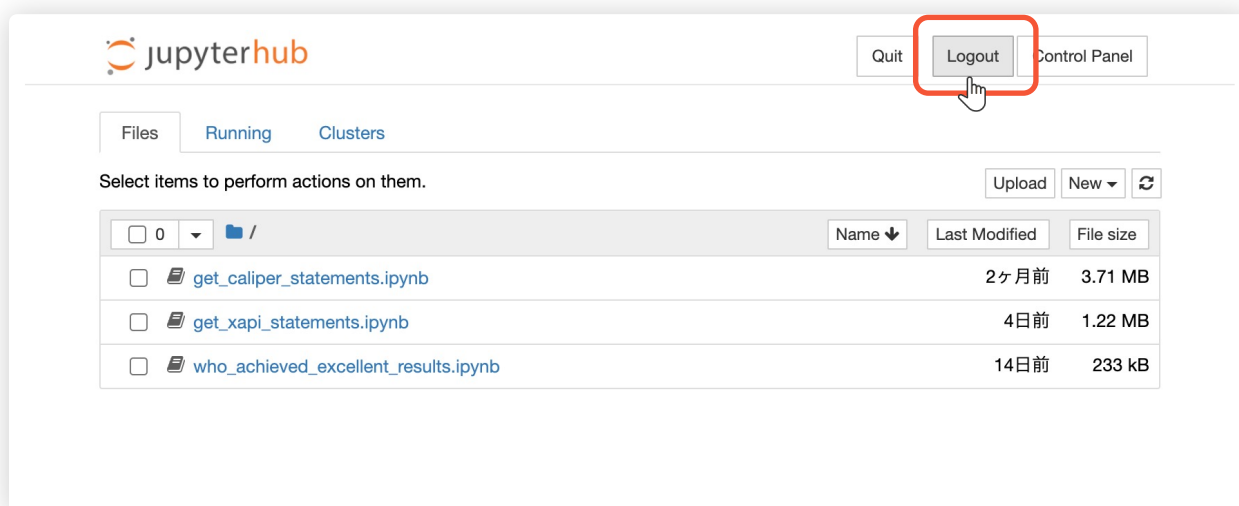
一覧にサンプルとして提供する以下のノートブックが存在することを確認する。

#	ファイル名	概要
1	get_caliper_statements.ipynb	Caliperステートメントを取得する例
2	get_xapi_statements.ipynb	xAPIステートメントを取得する例
3	who_achieved_excellent_results.ipynb	xAPIステートメントの小テスト結果から成績優秀者を求める例

次 ▶ [1.1.2 JupyterHubからログアウトする](#)

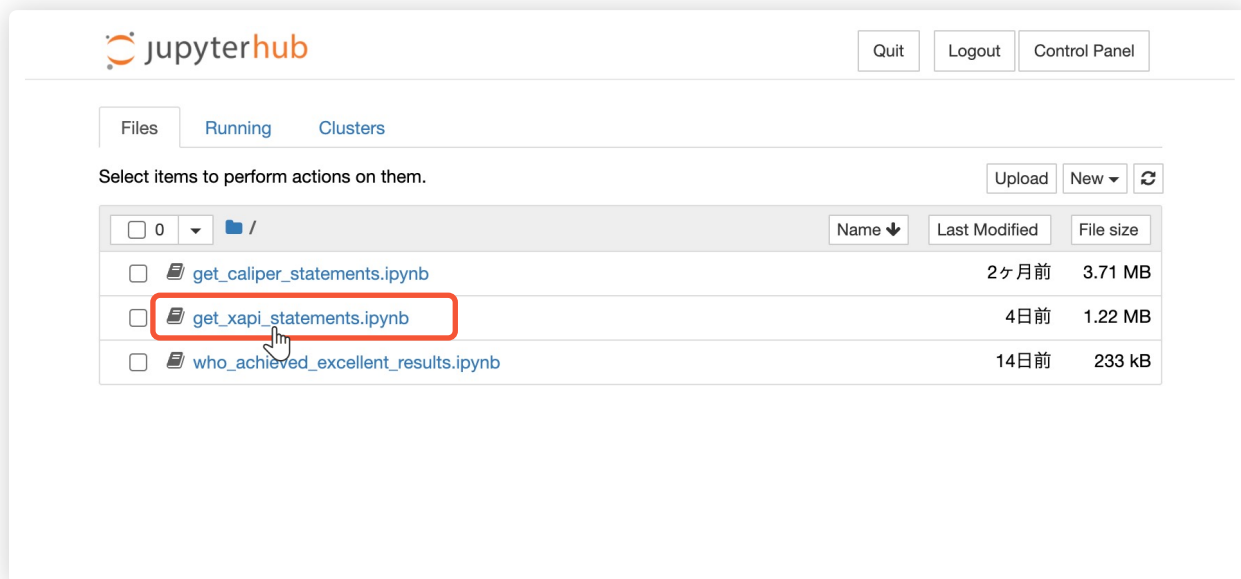
## 1.1.2 | JupyterHubからログアウトする

- ▶ 右上メニューの[Logout]ボタンをクリックしてログアウトする。メニュー位置は、ノートブック実行画面でも同様。ノートブック一覧ページへ遷移時は別タブが開かれるため、JupyterHubトップページのタブは予め閉じておく。
- ▶ ログアウトできずにノートブック一覧ページに遷移してしまう場合は、ブラウザのキャッシュを削除した上で[Logout]ボタンをクリックする。



## 1.1.3 | notebookを利用する

- ▶ ノートブック一覧から、利用したいノートブック名をクリックする。ここでは、`get_xapi_statements.ipynb`をクリックしたとして説明をする。



▶ 次ページへ

## 1. 基本操作 > 1.1 JupyterHubの操作 > 1.1.3 notebookを利用する

- ▶ ノートブックの実行画面に遷移する。この画面上でコードの記述、実行が可能。なお、初回アクセス時にはデフォルトのコードが表示される。
- ▶ コードはセル(Cell)単位で分割されており、セルごとに実行していく。セルを選択する場合はクリックする。

Selected cell is highlighted with a blue left margin.

ライブラリのインストールとインポート

```
In [1]: !pip install psycpg2-binary japanize_matplotlib
```

Collecting psycpg2-binary  
Using cached [https://files.pythonhosted.org/packages/f2/1b/720b36697158113ca1b2221a8e96a470088ccf3770d182214689d1a96a07/psycpg2-binary-2.8.6-cp36-cp36m-manylinux1\\_x86\\_64.whl](https://files.pythonhosted.org/packages/f2/1b/720b36697158113ca1b2221a8e96a470088ccf3770d182214689d1a96a07/psycpg2-binary-2.8.6-cp36-cp36m-manylinux1_x86_64.whl)  
Collecting japanize\_matplotlib  
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.6/site-packages (from japanize\_matplotlib)  
Requirement already satisfied: numpy>=1.7.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize\_matplotlib)  
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize\_matplotlib)  
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize\_matplotlib)  
Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize\_matplotlib)  
Requirement already satisfied: pytz in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize\_matplotlib)  
Requirement already satisfied: six>=1.10 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize\_matplotlib)  
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize\_matplotlib)  
Requirement already satisfied: setuptools in /opt/conda/lib/python3.6/site-packages (from kiwisolver>=1.0.1->matplotlib->japanize\_matplotlib)  
Installing collected packages: psycpg2-binary, japanize-matplotlib  
Successfully installed japanize-matplotlib-1.1.3 psycpg2-binary-2.8.6  
You are using pip version 9.0.3, however version 20.3.3 is available.  
You should consider upgrading via the 'pip install --upgrade pip' command.

```
In [2]: import json
import requests
```

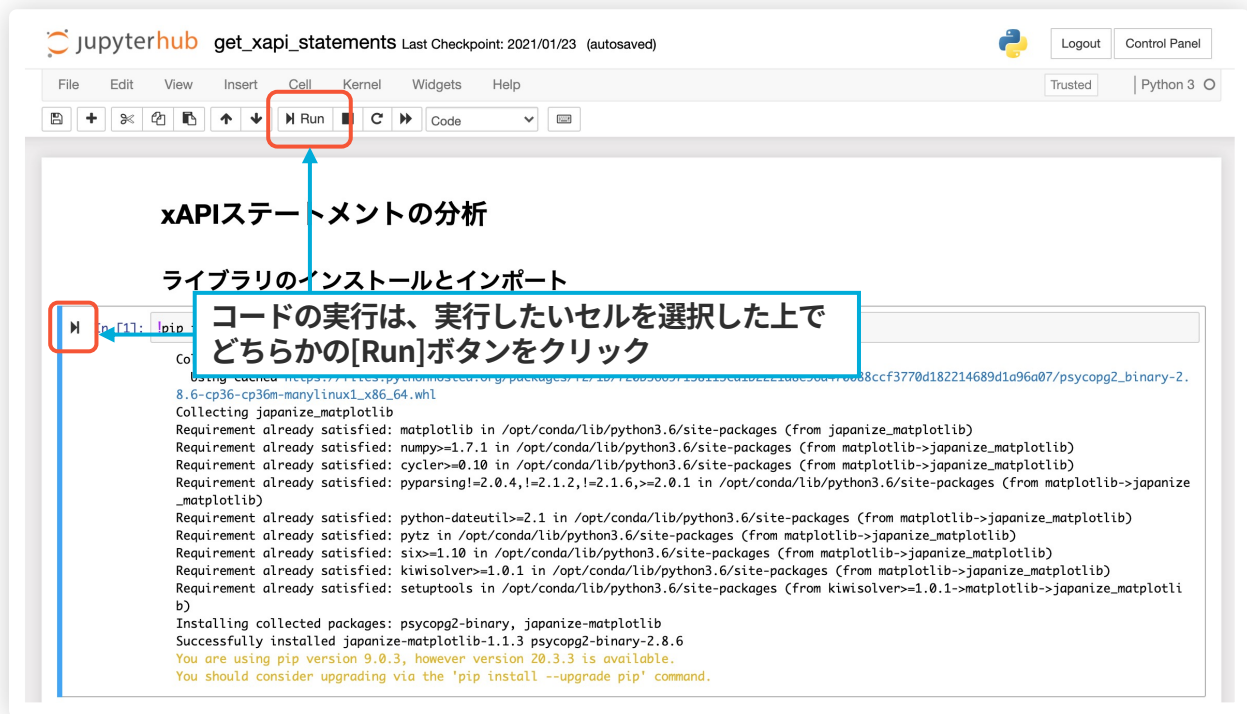
Editing mode: the left margin of the cell is green.

グレイアウト部分は編集可能箇所。目的に応じて任意の値を入力する

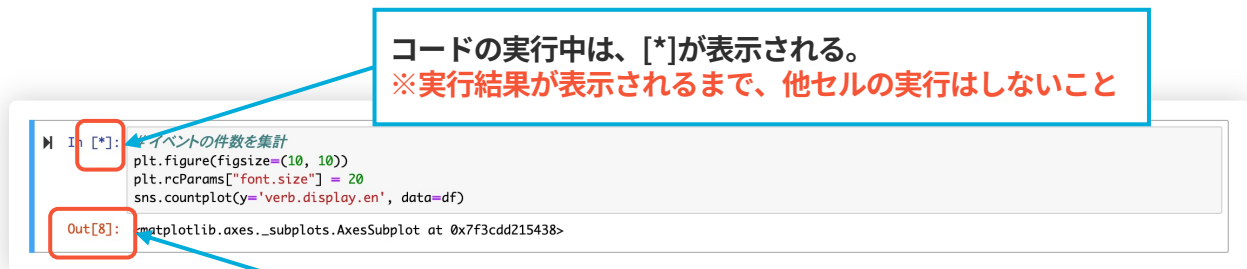
```
In [1]: !pip install psycpg2-binary japanize_matplotlib
```

Collecting psycpg2-binary  
Using cached [https://files.pythonhosted.org/packages/f2/1b/720b36697158113ca1b2221a8e96a470088ccf3770d182214689d1a96a07/psycpg2-binary-2.8.6-cp36-cp36m-manylinux1\\_x86\\_64.whl](https://files.pythonhosted.org/packages/f2/1b/720b36697158113ca1b2221a8e96a470088ccf3770d182214689d1a96a07/psycpg2-binary-2.8.6-cp36-cp36m-manylinux1_x86_64.whl)  
Collecting japanize\_matplotlib  
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.6/site-packages (from japanize\_matplotlib)  
Requirement already satisfied: numpy>=1.7.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize\_matplotlib)  
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize\_matplotlib)  
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize\_matplotlib)  
Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize\_matplotlib)  
Requirement already satisfied: pytz in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize\_matplotlib)  
Requirement already satisfied: six>=1.10 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize\_matplotlib)  
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize\_matplotlib)  
Requirement already satisfied: setuptools in /opt/conda/lib/python3.6/site-packages (from kiwisolver>=1.0.1->matplotlib->japanize\_matplotlib)  
Installing collected packages: psycpg2-binary, japanize-matplotlib  
Successfully installed japanize-matplotlib-1.1.3 psycpg2-binary-2.8.6  
You are using pip version 9.0.3, however version 20.3.3 is available.  
You should consider upgrading via the 'pip install --upgrade pip' command.

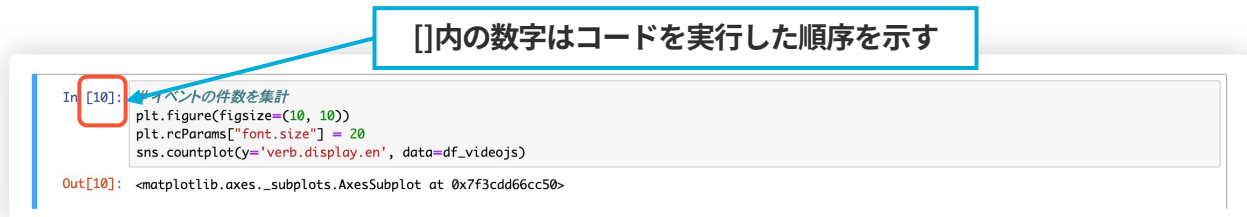
# 1. 基本操作 > 1.1 JupyterHubの操作 > 1.1.3 notebookを利用する



▶ コードは基本的に上から順に実行していく。



コードの実行結果は、Outとして表示される



次 ▶ 1.2.1 Supersetへログインする

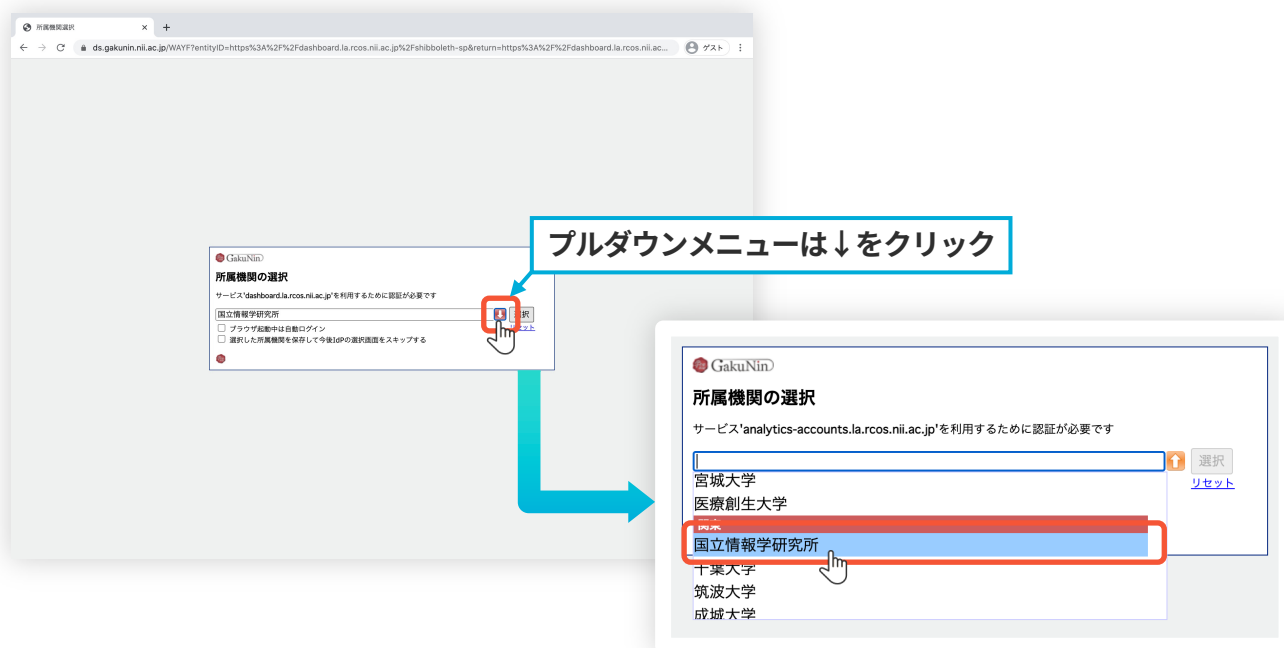


## 1.2.1 | Supersetへログインする

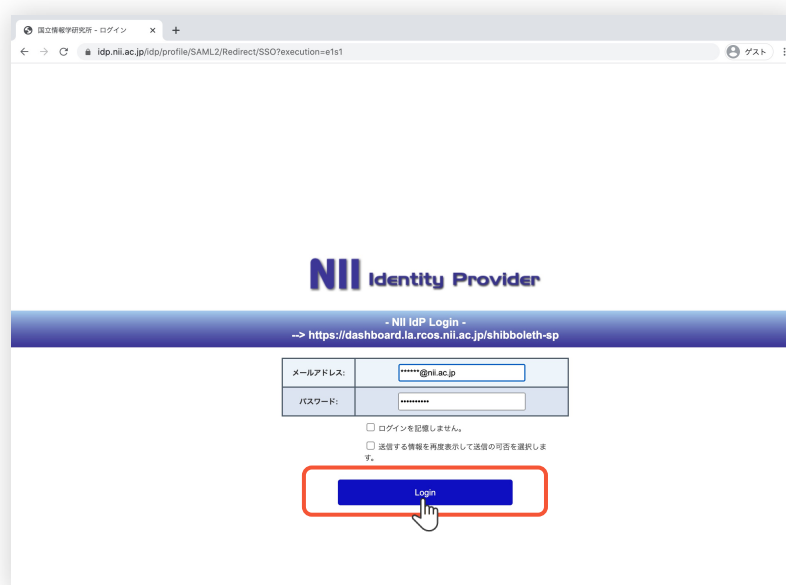
### Supersetとは

GUIによる操作でデータの可視化、分析が可能なWEBアプリケーション

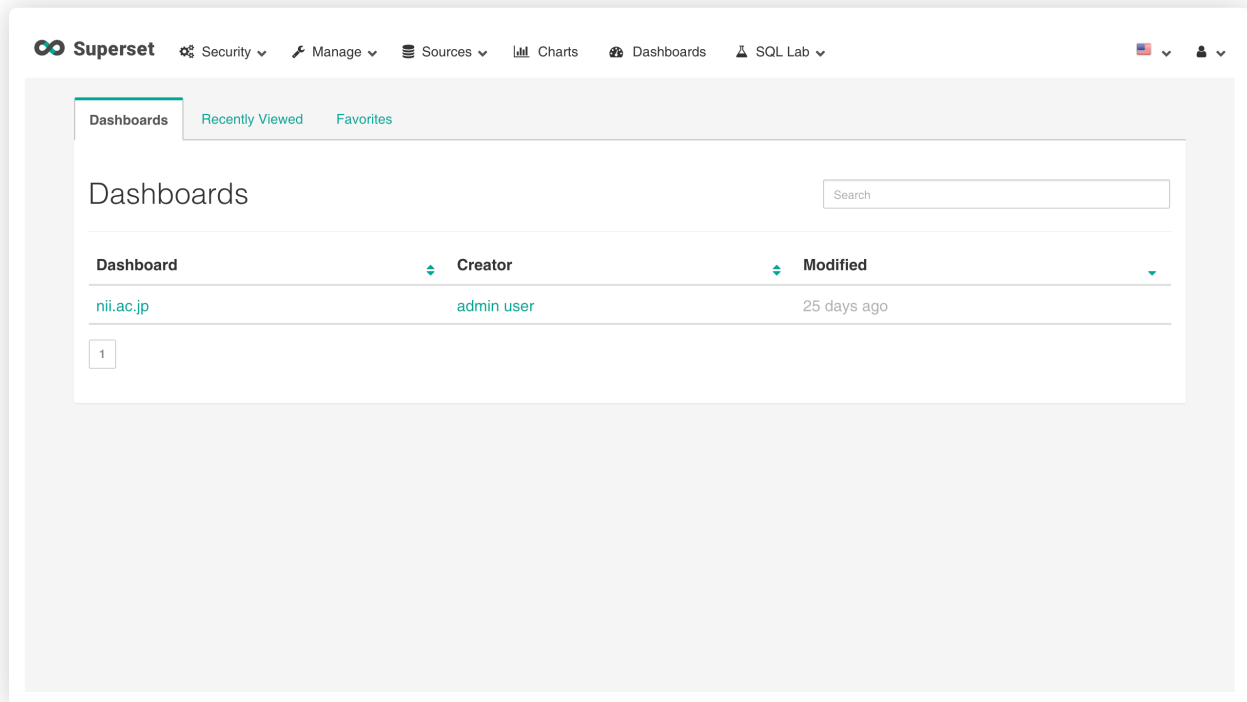
- ▶ <https://dashboard.la.rcos.nii.ac.jp>へアクセスし、自分が所属する機関名を選択する。



- ▶ 選択した所属機関のログインページに遷移するので、付与されているメールアドレスとパスワードを入力し[Login]ボタンをクリックする。



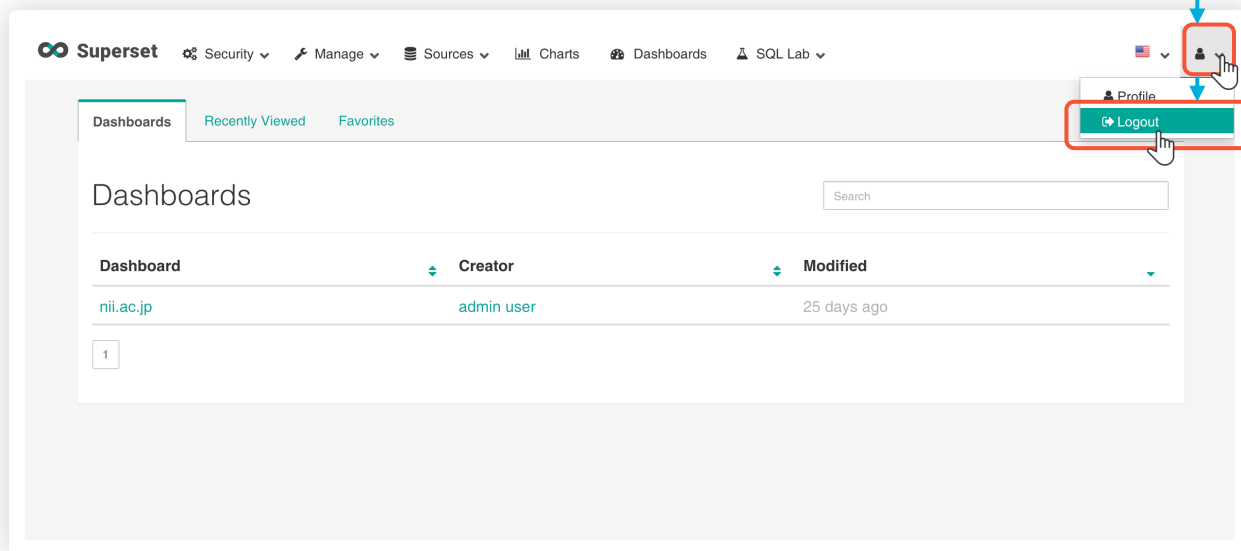
- ▶ ログインが完了すると、ダッシュボード画面が表示される。ダッシュボード一覧には、所属している機関ドメイン名が表示される。



## 1.2.2 | Supersetからログアウトする

- ▶ 右上の👤アイコンをクリックし、メニューから[Logout]ボタンをクリックしてログアウトする。

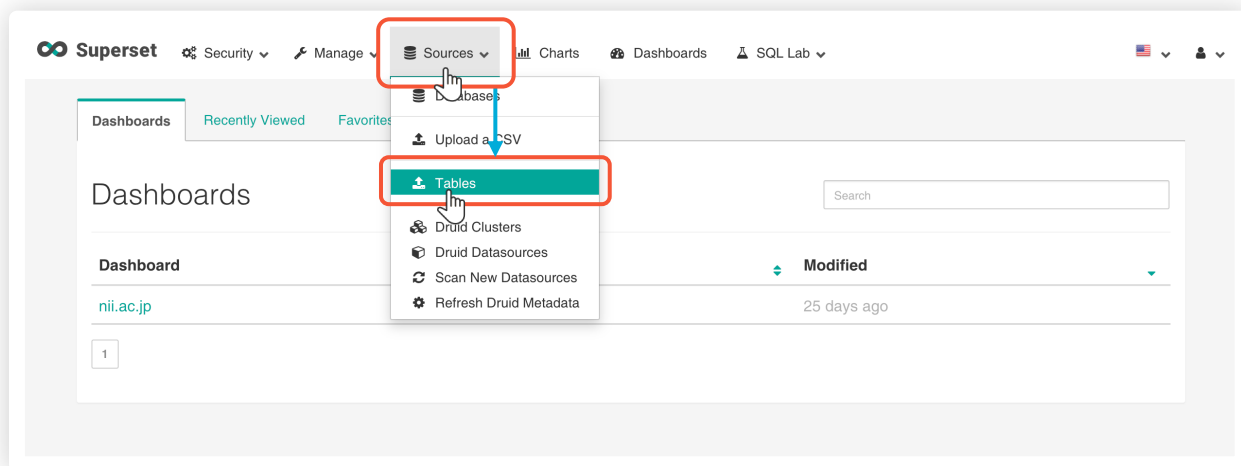
アイコンをクリックしメニューを表示させる



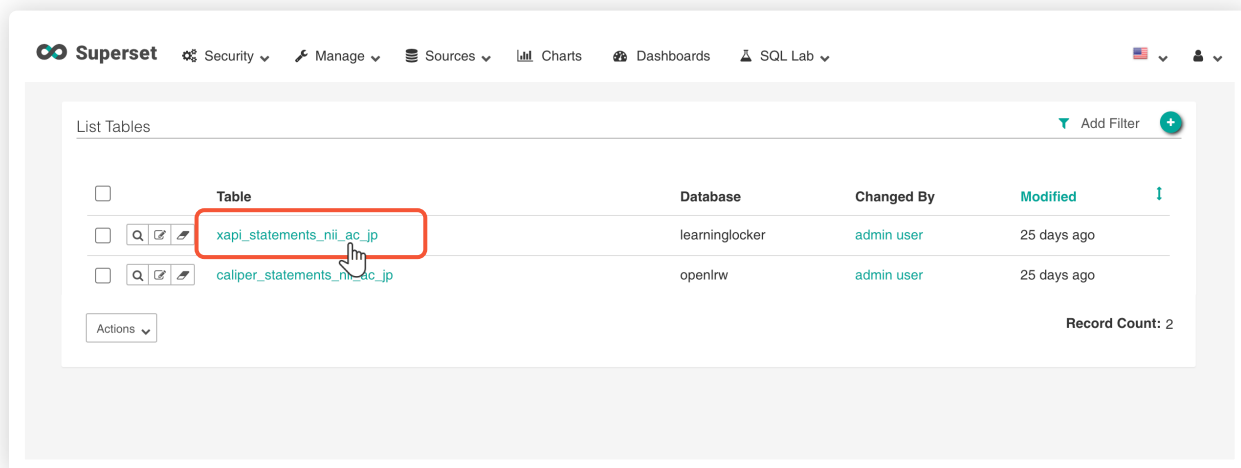
次 ▶ 1.2.3 Supersetを利用する

## 1.2.3 | Supersetを利用する

- ▶ ダッシュボード画面上部メニューの[Sources]より、[Tables]をクリックする。



- ▶ 分析可能なデータの一覧が表示される。分析対象のテーブル名をクリックする。ここでは、xapi\_statements\_nii\_ac\_jpをクリックしたとして説明する。



# 1. 基本操作 > 1.2 Supersetの操作 > 1.2.3 Supersetを利用する

テーブルは一覧上部のFilterメニューより、検索と絞り込みが可能。

**フィルターは複数設定が可能**

**[Add Filter]ボタンをクリックすると絞り込みメニューが表示される**

**フィルターの設定後、[Search]ボタンをクリックすると下部リストに反映される**

The screenshot shows the 'List Tables' page in Superset. It features a search bar with a magnifying glass icon and a 'Search' button. Below the search bar, there are filter settings for 'SQL Lab View' (set to 'Not Equal to') and 'Database' (set to 'Relation' with a dropdown menu showing 'jupyter'). An 'Add Filter' button with a plus sign is visible in the top right corner. A callout box points to this button, indicating that clicking it opens a filter menu. Another callout box points to the search bar, indicating that clicking the search button updates the table list below.

テーブル一覧に目的のテーブルが存在しない場合は、以下の手順で追加する。

**List Tables 右の+ボタンからテーブル作成ページへ遷移する**

**Databaseはjupyterを選択**

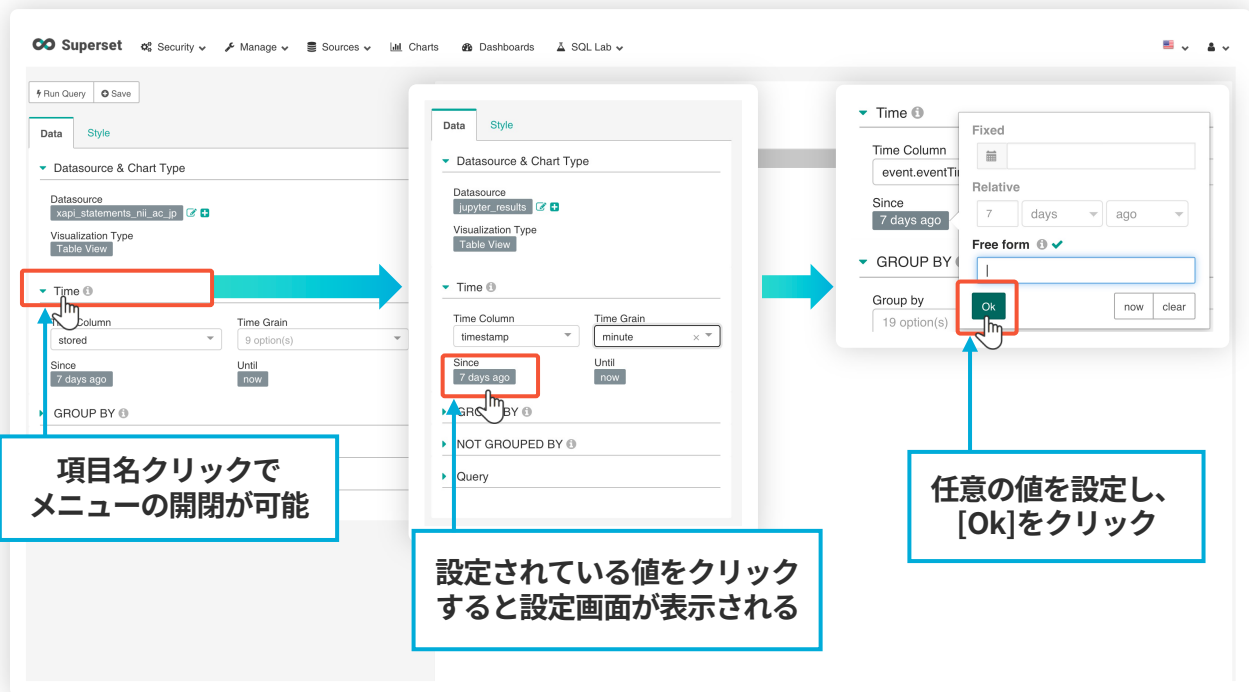
**Table NameにはJupyterノートブックで指定したテーブル名を入力(この例ではxapi\_statements\_nii\_ac\_jp)**

**[Save]ボタンをクリックして追加**

The screenshot shows the 'Import a table definition' form in Superset. The 'Database' dropdown menu is set to 'jupyter'. The 'Table Name' text input field contains 'xapi\_statements\_nii\_ac\_jp'. A 'Save' button with a floppy disk icon is located at the bottom left. A callout box points to the plus sign in the top right of the 'List Tables' page, indicating that clicking it leads to this form. Another callout box points to the 'Database' dropdown, indicating that 'jupyter' should be selected. A third callout box points to the 'Table Name' input field, indicating that the name 'xapi\_statements\_nii\_ac\_jp' should be entered. A final callout box points to the 'Save' button, indicating that clicking it will add the table.

# 1. 基本操作 > 1.2 Supersetの操作 > 1.2.3 Supersetを利用する

- ▶ 左部メニューより、パラメータをそれぞれ設定する。

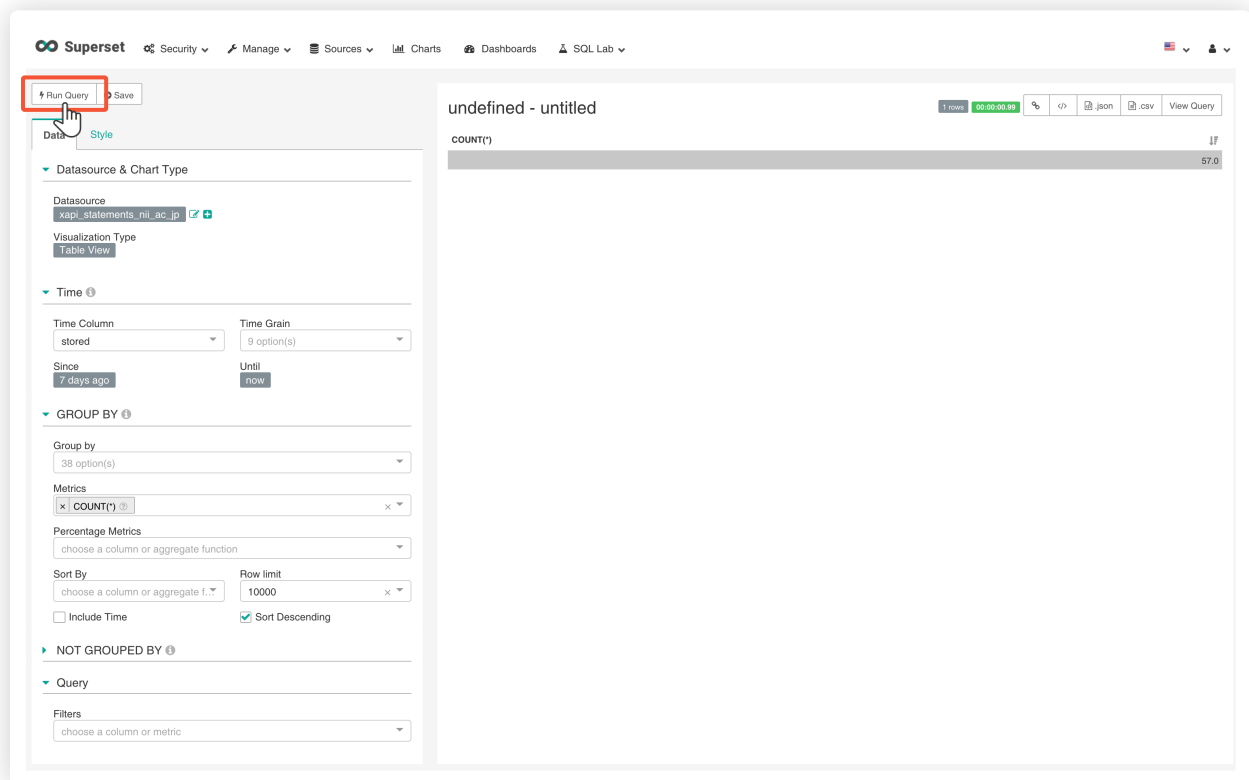


項目名クリックでメニューの開閉が可能

設定されている値をクリックすると設定画面が表示される

任意の値を設定し、[OK]をクリック

- ▶ 設定完了後に、[Run Query]ボタンをクリックすると、右部分に結果が表示される。



count(*)	if
57.0	

次 ▶ 2. 実践操作



## 2. 実践操作

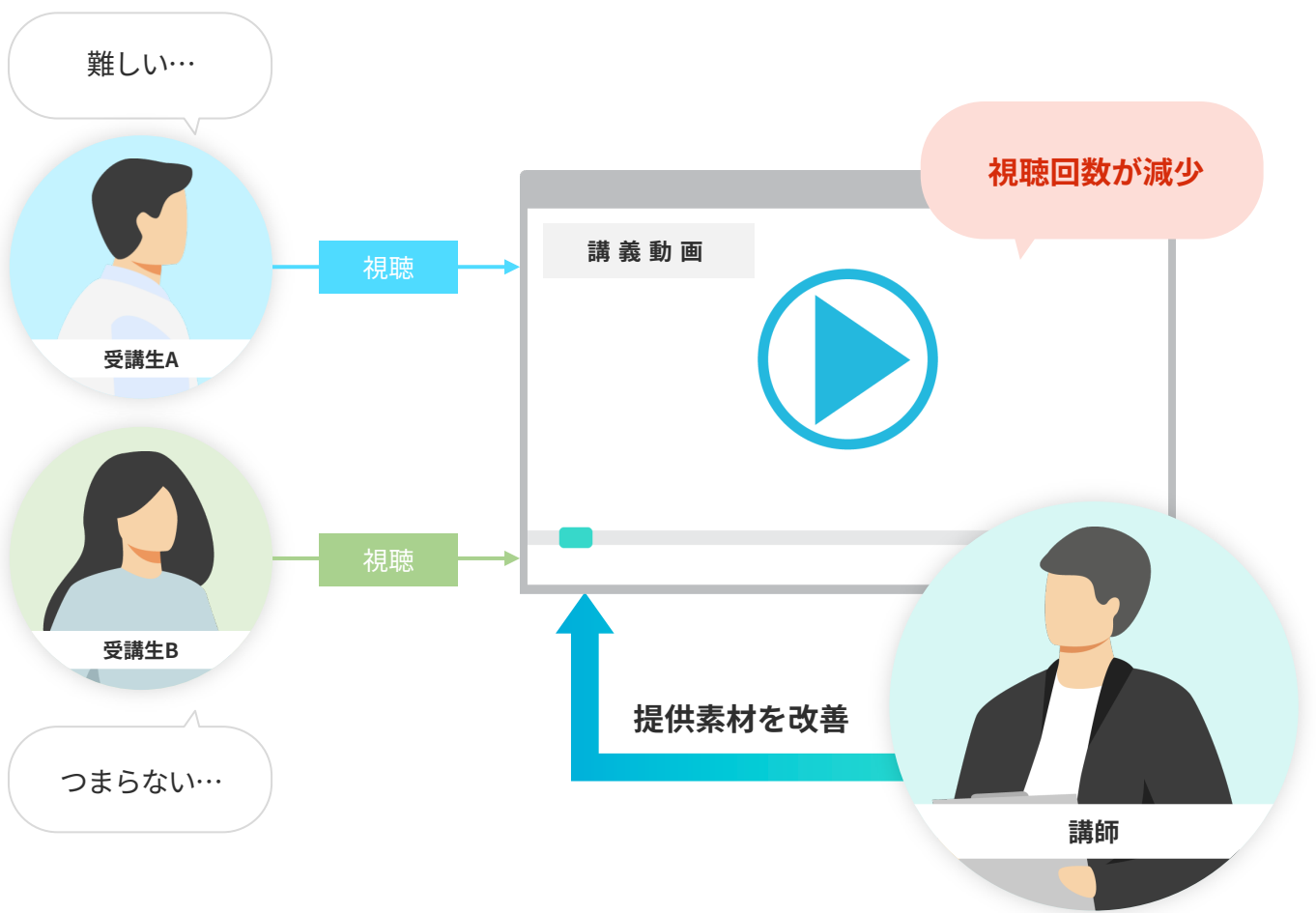
---

<b>2.1</b>	<b>動画の視聴回数を調べる</b>	17p
2.1.1	JupyterHubの操作	18p
2.1.2	Supersetの操作	20p
<b>2.2</b>	<b>小テストの結果から成績優秀者を特定する</b>	21p
2.2.1	JupyterHubの操作	22p
2.2.2	Supersetの操作	24p
<b>2.3</b>	<b>小テストの結果と動画の視聴履歴から退学者を予測する</b>	27p
2.3.1	JupyterHubの操作	28p

### 【例1】動画の視聴回数の推移を調べる

講義動画の視聴回数の推移を把握することで、視聴回数が減少した際に、提供教材の改善を実施する等の対策を行うことができる。

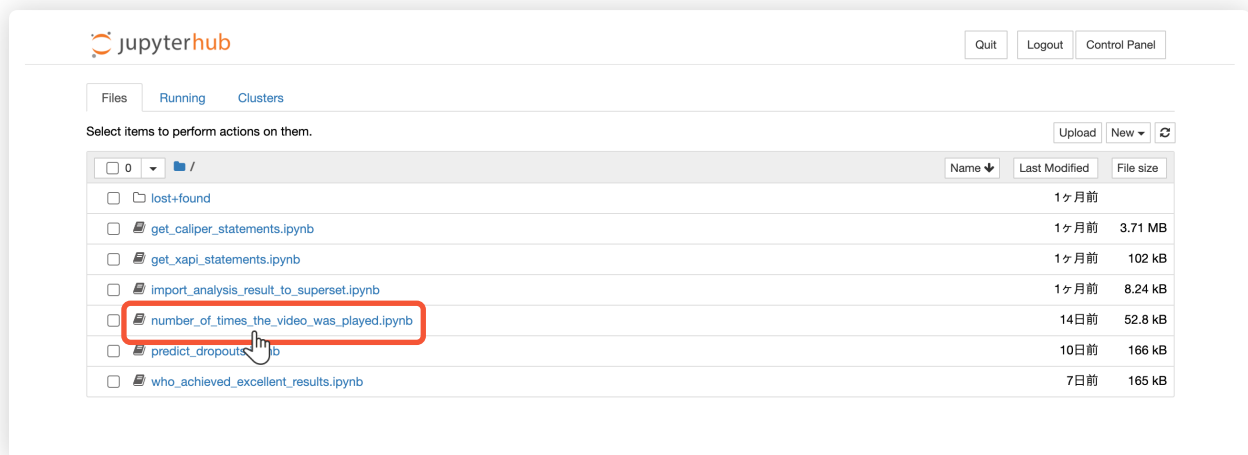
本シナリオでは、動画の視聴回数の推移を分析する手法を例示する。



▶次ページへ

## 2.1.1 | JupyterHubの操作

- ▶ 手順1.1の通りJupyterHubへログインし、ノートブックの一覧から `number_of_times_the_video_was_played.ipynb` を選択する。
- ▶ 本ノートブックは、動画の視聴データをもとに日毎の動画の視聴数を求め、実行結果を Supersetに登録する。



主な処理内容は以下の通り。(詳細はノートブックを参照すること)

- 学習ログ(xAPIステートメント)を取得
- コース名を取得
- 指定したコース内の動画名を取得
- 指定した動画の再生ログを取得
- 期間を設定し、日毎のイベント発生数を表示

▶ [次ページへ](#)

## 2. 実践操作 > 2.1 動画の視聴回数を調べる > 2.1.1 JupyterHubの操作

- ▶ 上部セルよりコードを実行する。任意の値を指定する必要がある場合は都度入力する。今回は、コースIDを'31'、動画名を'RDM01/RDM01week0103.mp4'としている。

number\_of\_times\_the\_video\_was\_played Last Checkpoint: 2021/01/25 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

'56': {'en': '倫倫姫の情報セキュリティ教室'})

**コースIDを指定**

```
In [6]: course_id = '31'
```

**指定コース内の動画名の収集**

```
In [7]: video_names = []
for _, row in df.iterrows():
    if row["verb.display.en"] == 'firstplay' and
       row["object.id"] in object_names[course_id] and
       row["object.definition.name.en"] not in video_names:
        video_names.append(row["object.definition.name.en"])
video_names
```

Out [7]: ['RDM01/RDM01week0101.mp4', 'RDM01/RDM01week0103.mp4']

**動画名を指定**

```
In [14]: video_name = 'RDM01/RDM01week0103.mp4'
```

変更する場合は、調べたい対象に応じた値を入力する

- ▶ 『jupyterデータベースに処理結果を保存』セルが正常に実行されたことを確認し、JupyterHubでの操作を完了させる。

0.0 2000 2001 2002 2003

**jupyterデータベースに処理結果を保存**

```
In [19]: pip install psycopg2-binary
from sqlalchemy import create_engine
from sqlalchemy.types import TIMESTAMP

# 所属機関に割り当てられたユーザ情報を設定する
username =
password =

dfq = dfq.drop('context.contextActivities.category', axis=1)

engine = create_engine('postgresql://'+ username + ':' + password +
tablename = 'number_of_times_the_video_was_played'

dfq.to_sql(tablename, engine, if_exists='replace', dtype={'timestamp': TIMESTAMP})

Requirement already satisfied: psycopg2-binary in /opt/conda/lib/python3.6/site-packages
You are using pip version 9.0.3, however version 21.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

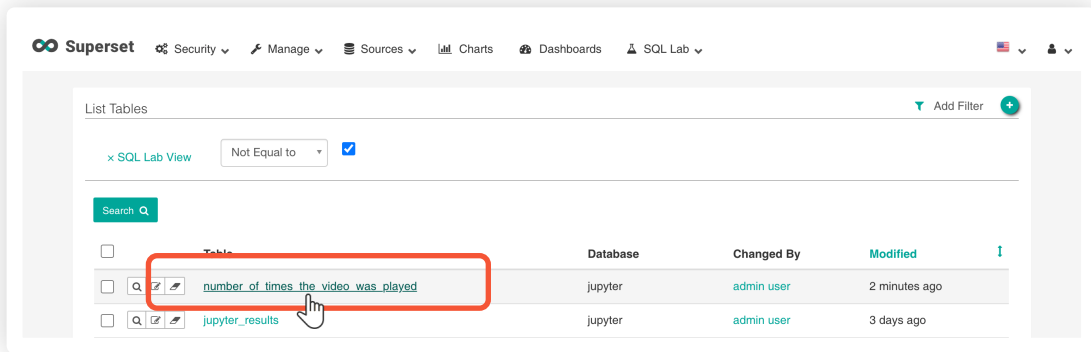
In [ ]:

データベースへの保存処理セルを必ず実行

Supersetでのテーブル名はこの部分を参照のこと

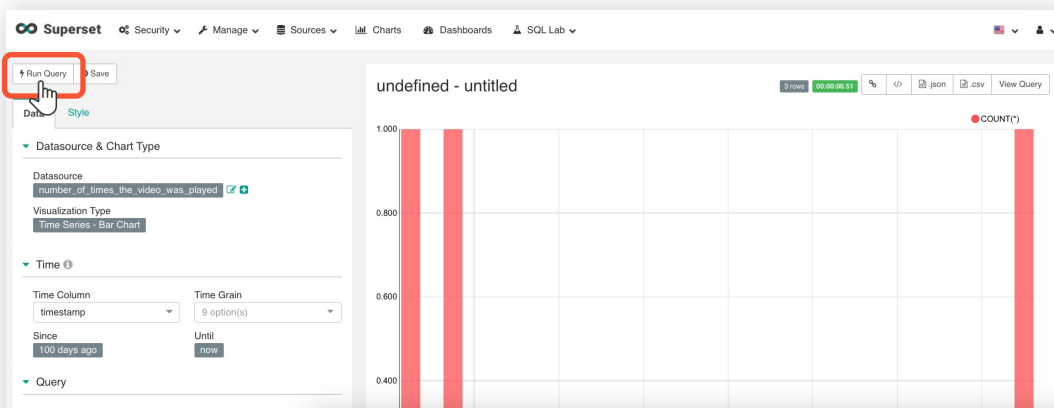
## 2.1.2 | Supersetの操作

- ▶ 手順1.2の通りSupersetへログインし、JupyterHubの処理結果テーブルを選択する。



- ▶ 左部メニューより各項目の値を設定し、[Run Query]をクリックしてデータを可視化する。

### 期間を指定して日毎のイベント発生数を表示



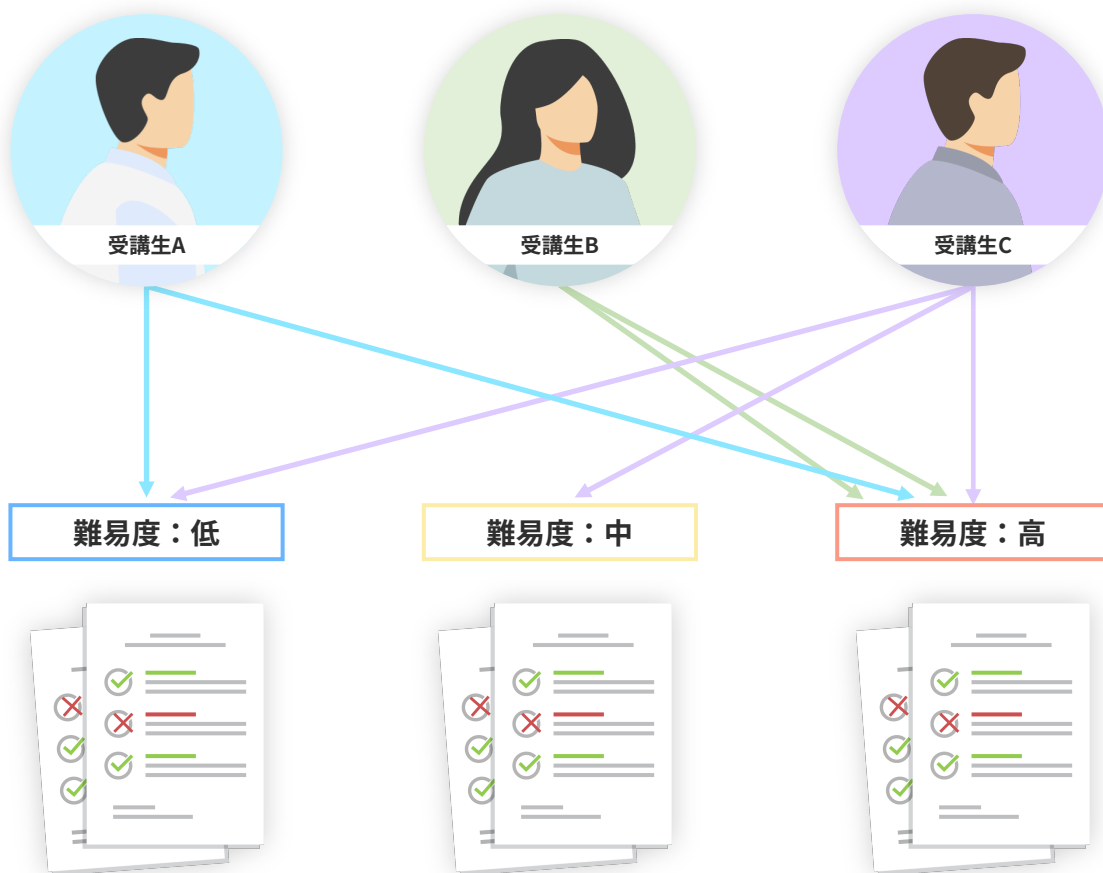
タブ	大項目	小項目	設定値
Data	Visualization Type		Time Series - Bar Chart
	Time	Time Column	timestamp
		Since	任意の日付
		Until	任意の日付
Query	Metrics		COUNT(*)
	Advanced Analytics	Resample How	sum
		Resample Rule	1D
Style	X Axis	X Axis Format	%Y-%m-%d
	Y Axis	Y Axis Format	,.3f

### 【例2】小テストの結果から成績優秀者を特定する

学認LMSでは、講師は各講義の中で小テストを実施、採点することができる。

講師が受講生の中から成績優秀者を特定する際に、小テストはそれぞれ難易度が異なるため、単純な採点の合計は受講生の優劣を決定するためには適切でない恐れがある。

したがって、本シナリオでは、各小テストの難易度を考慮して成績優秀者を特定する分析方法を例示する。

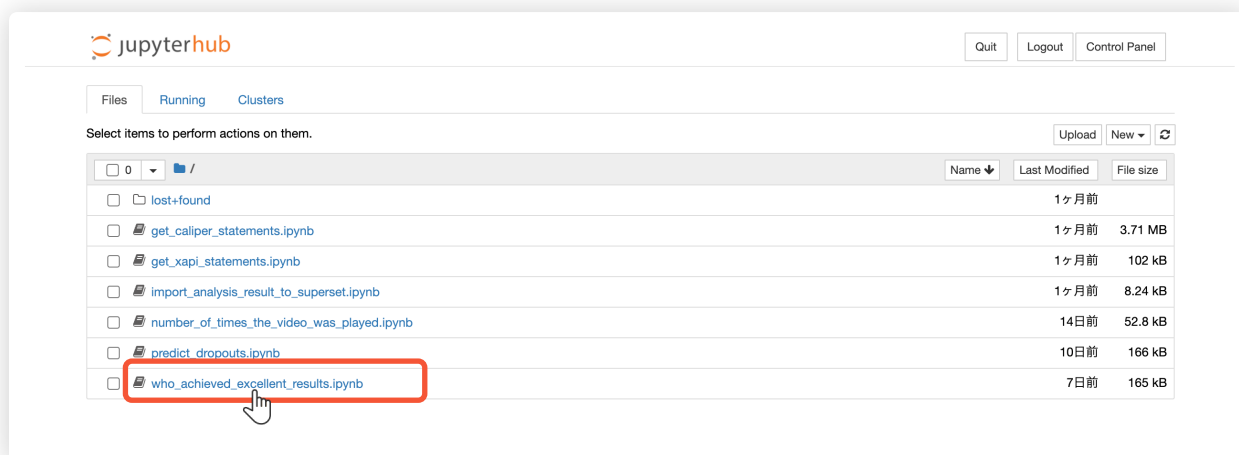


▶次ページへ



### 2.2.1 | JupyterHubの操作

- ▶ 手順1.1の通りJupyterHubへログインし、ノートブックの一覧から `who_achieved_excellent_results.ipynb` を選択する。
- ▶ 本ノートブックは小テスト結果から成績優秀者を求め、実行結果をSupersetに登録する。

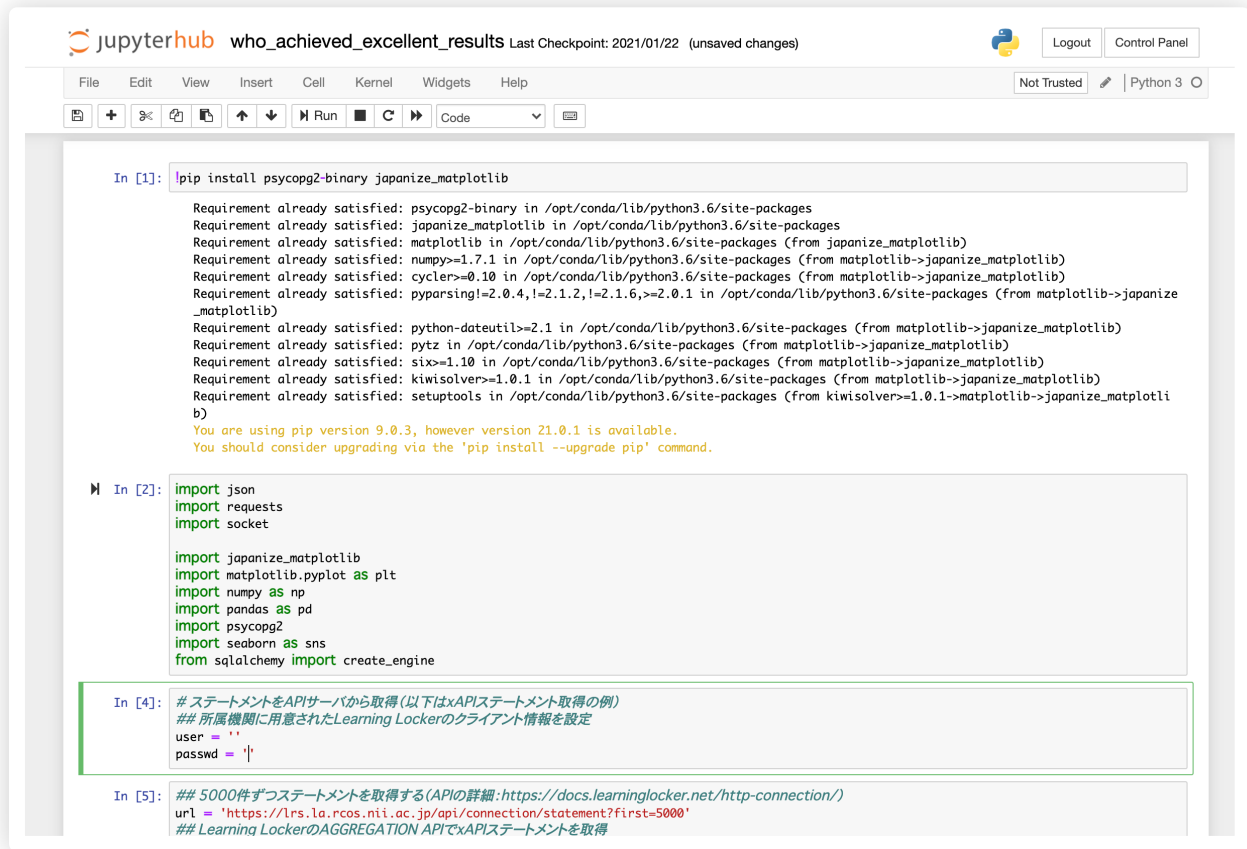


主な処理内容は以下の通り。(詳細はノートブックを参照すること)

- 学習ログ(xAPIステートメント)を取得
- 小テスト結果(完了済かつ及第点)のみを取得
- コースの日本語名列を追加
- テスト受験時間 ( `result.duration` ) をISO8601形式から数値形式 (分単位) に変換
- テスト結果( `result.score` )からテストの正答率を求めてA~Dランク評価を追加
- テストごとに平均、標準偏差、分散、最小、最大を算出して追加
- 標準得点を (取得した得点 - 平均) / 標準偏差 でテストごとに算出
- テストごとの順位を追加
- 小テストごとの1位を一覧化
- 特定のコースを対象にユーザ単位に取得したグレードの数をカウント(※)
- 特定のコースを対象に標準得点の合計を表示(※)
- Supersetに加工した学習ログを登録

## 2. 実践操作 > 2.2 小テストの結果から成績優秀者を特定する > 2.2.1 JupyterHubの操作

- ▶ 上部セルよりコードを実行する。任意の値を指定する必要がある場合は都度入力する。



```
who_achieved_excellent_results Last Checkpoint: 2021/01/22 (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3
In [1]: pip install psycpg2-binary japanize_matplotlib
Requirement already satisfied: psycpg2-binary in /opt/conda/lib/python3.6/site-packages
Requirement already satisfied: japanize_matplotlib in /opt/conda/lib/python3.6/site-packages
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.6/site-packages (from japanize_matplotlib)
Requirement already satisfied: numpy>=1.7.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: pytz in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: six>=1.10 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.6/site-packages (from matplotlib->japanize_matplotlib)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.6/site-packages (from kiwisolver>=1.0.1->matplotlib->japanize_matplotlib)
You are using pip version 9.0.3, however version 21.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.

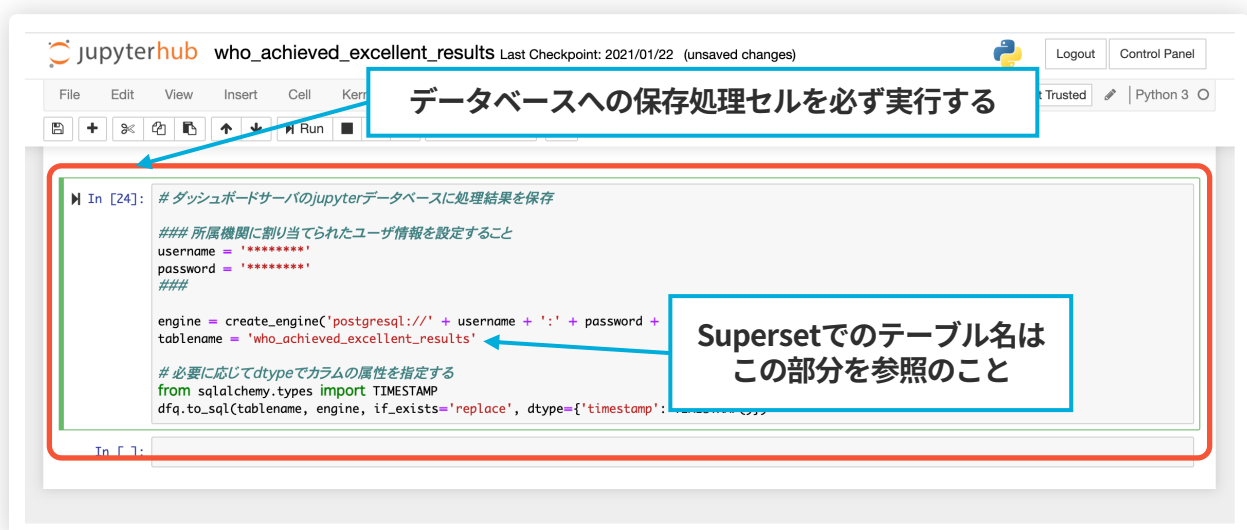
In [2]: import json
import requests
import socket

import japanize_matplotlib
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import psycpg2
import seaborn as sns
from sqlalchemy import create_engine

In [4]: # ステートメントをAPIサーバから取得(以下はxAPIステートメント取得の例)
## 所属機関に用意されたLearning Lockerのクライアント情報を設定
user = ''
passwd = ''

In [5]: ## 5000件ずつステートメントを取得する(APIの詳細:https://docs.learninglocker.net/http-connection/)
url = 'https://lrs.la.rcos.nii.ac.jp/api/connection/statement?first=5000'
## Learning LockerのAGGREGATION APIでxAPIステートメントを取得
```

- ▶ コードの実行結果を保存し、JupyterHubでの操作を完了させる。



```
who_achieved_excellent_results Last Checkpoint: 2021/01/22 (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3
データベースへの保存処理セルを必ず実行する

In [24]: # ダッシュボードサーバのjupyterデータベースに処理結果を保存
### 所属機関に割り当てられたユーザ情報を設定すること
username = '*****'
password = '*****'
###

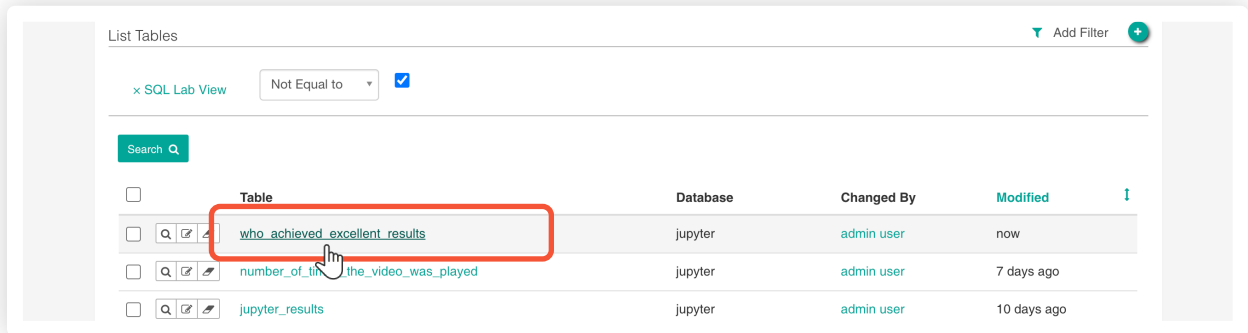
engine = create_engine('postgresql://' + username + ':' + password +
tablename = 'who_achieved_excellent_results'

# 必要に応じてdtypeでカラムの属性を指定する
from sqlalchemy.types import TIMESTAMP
dfq.to_sql(tablename, engine, if_exists='replace', dtype={'timestamp':

Supersetでのテーブル名はこの部分を参照のこと
```

## 2.2.2 | Supersetの操作

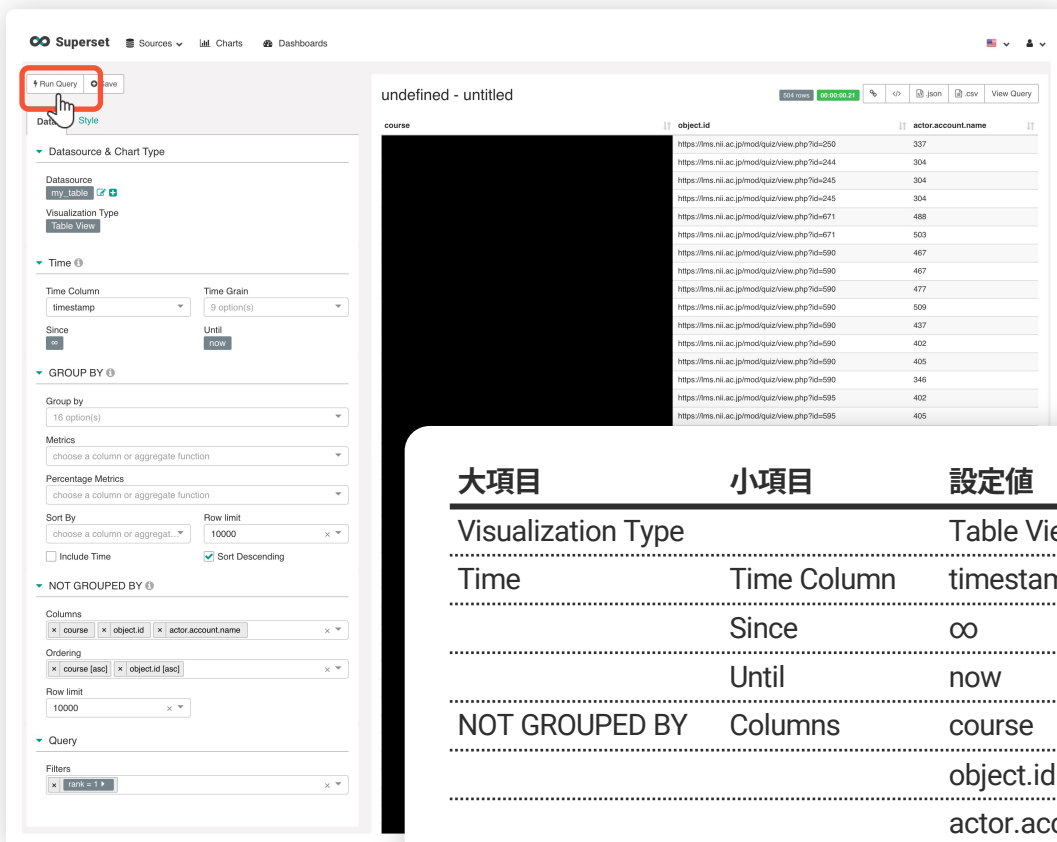
- ▶ 手順1.2の通りSupersetへログインし、JupyterHubの処理結果テーブルを選択する。



- ▶ 左部メニューより各項目の値を設定し、[Run Query]をクリックしてデータを可視化する。

### 小テストごとの1位を一覧化

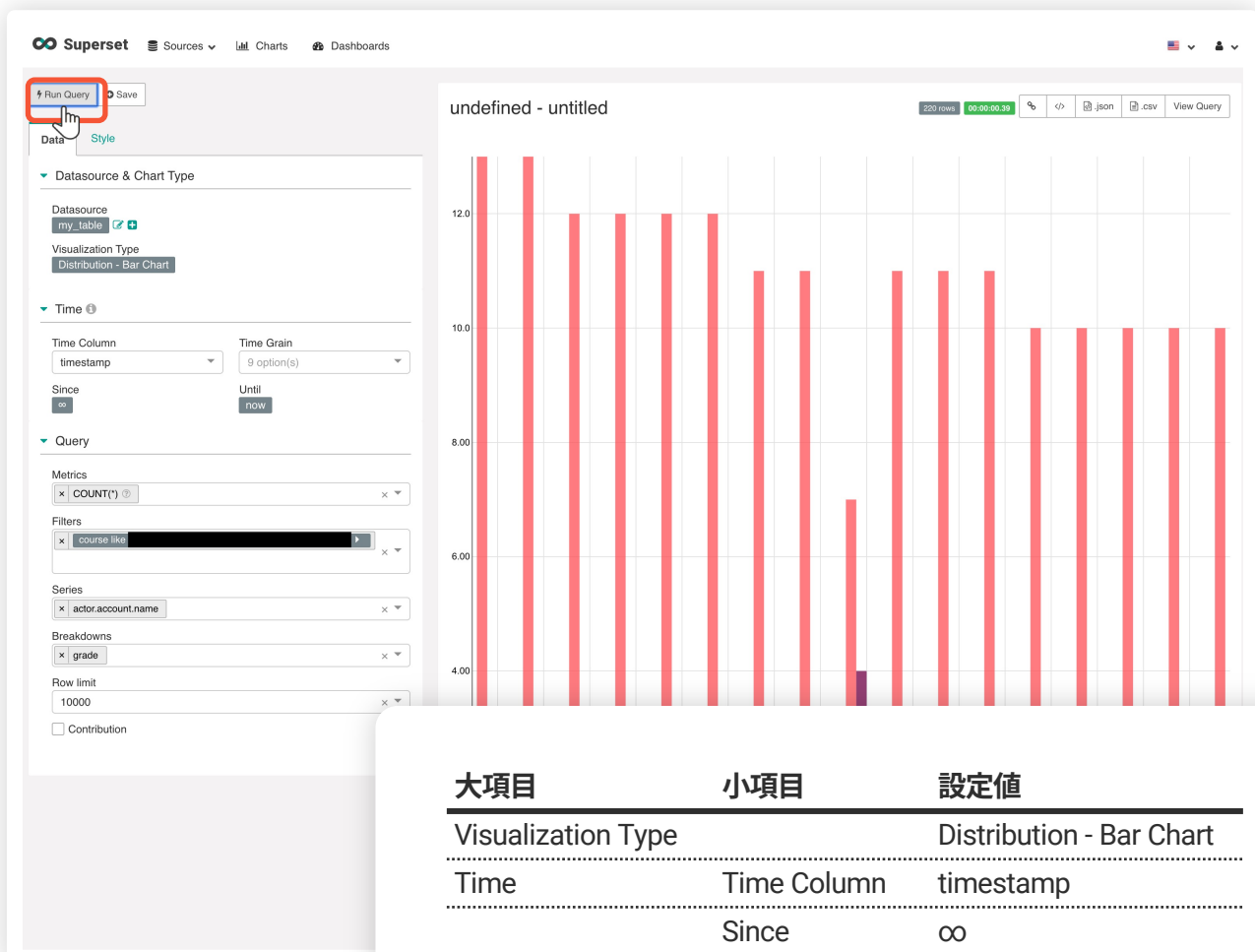
各設定項目のパラメータ設定例は以下の通り。



大項目	小項目	設定値
Visualization Type		Table View
Time	Time Column	timestamp
	Since	∞
	Until	now
NOT GROUPED BY	Columns	course object.id actor.account.name
	Ordering	course[asc] object.id[asc]
Query	Filters	rank = 1

## 2. 実践操作 > 2.2 小テストの結果から成績優秀者を特定する > 2.2.2 Supersetの操作

特定コースで各ユーザが取得した成績評価（A～D）の数を可視化  
各設定項目のパラメータ設定例は以下の通り。

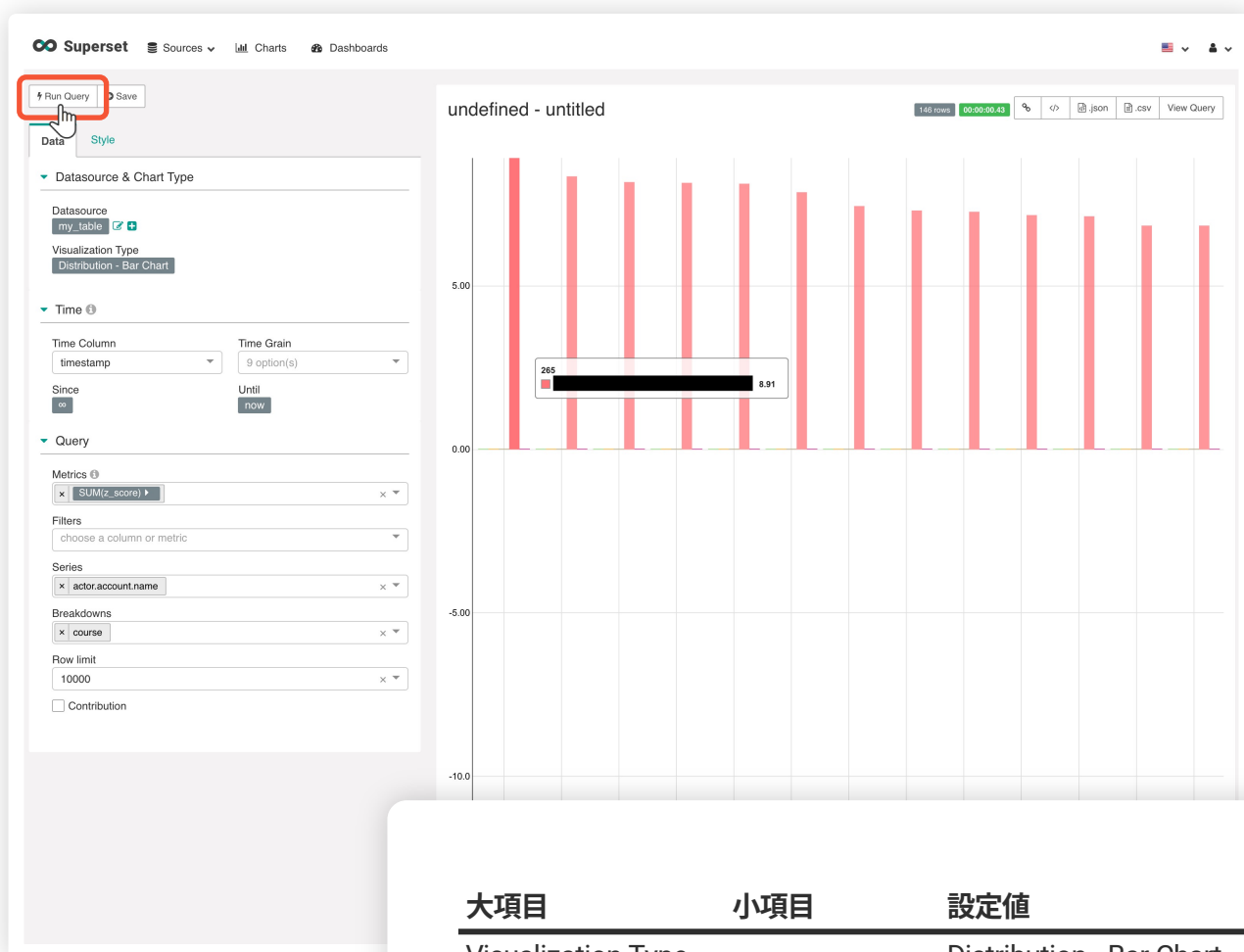


大項目	小項目	設定値
Visualization Type		Distribution - Bar Chart
Time	Time Column	timestamp
	Since	∞
	Until	now
Query	Metrics	COUNT(*)
	Filters	course like <コース名>
	Series	actor.account.name
	Breakdowns	grade

## 2. 実践操作 > 2.2 小テストの結果から成績優秀者を特定する > 2.2.2 Supersetの操作

特定コース内で標準得点の合計値をユーザ毎に可視化

各設定項目のパラメータ設定例は以下の通り。



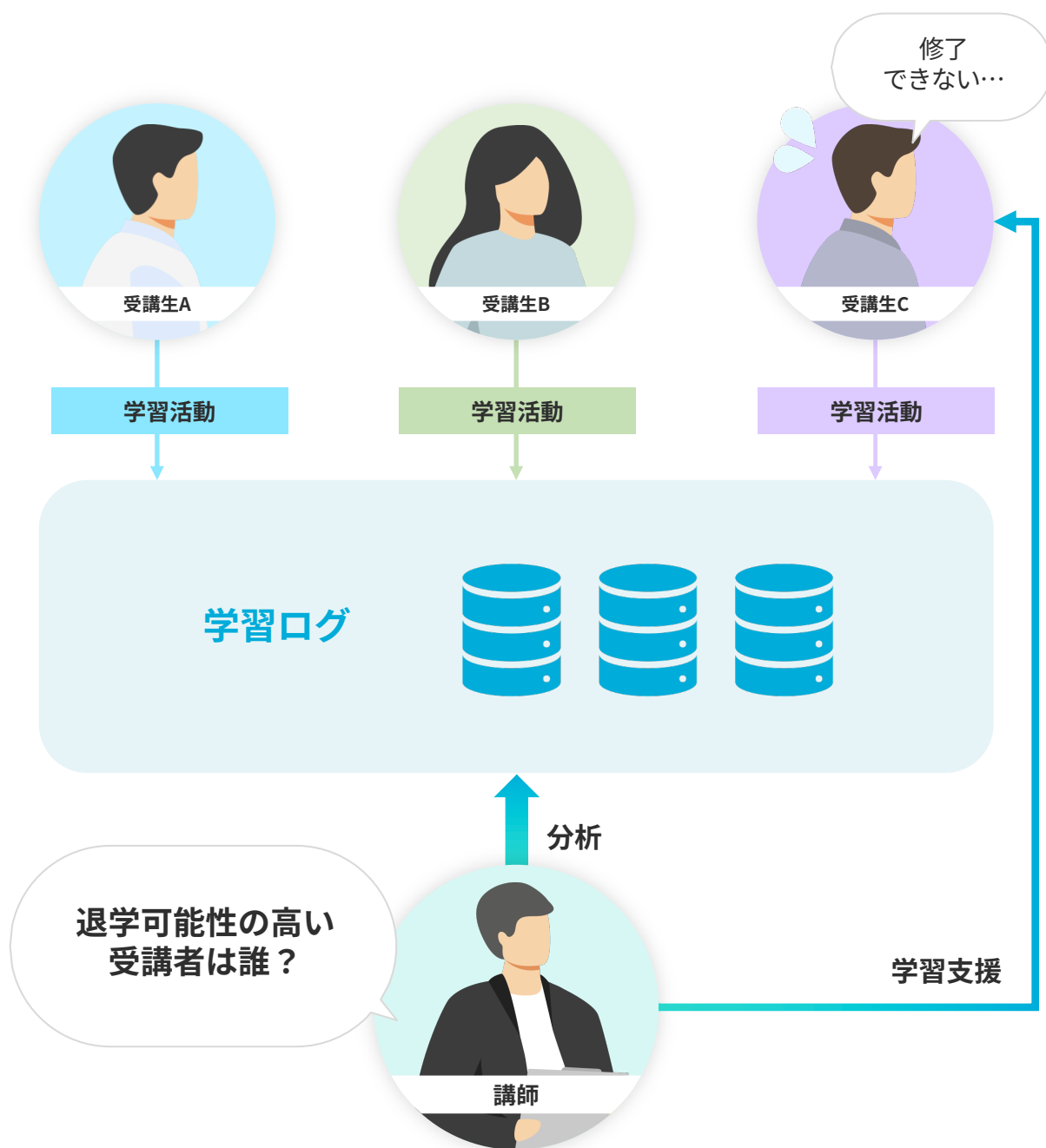
大項目	小項目	設定値
Visualization Type		Distribution - Bar Chart
Time	Time Column	timestamp
	Since	∞
	Until	now
Query	Metrics	SUM(z_score)
	Breakdowns	course

次 ▶ 2.3 小テストの結果と動画の視聴履歴から退学者を予測する

### 【例3】小テストの結果と動画の視聴履歴から退学者を予測する

オンライン講座では、講座を最後まで修了できずに退学する受講者が多く存在する。退学可能性の高い受講者を推定して学習支援を行うことは、退学率の改善につながる。

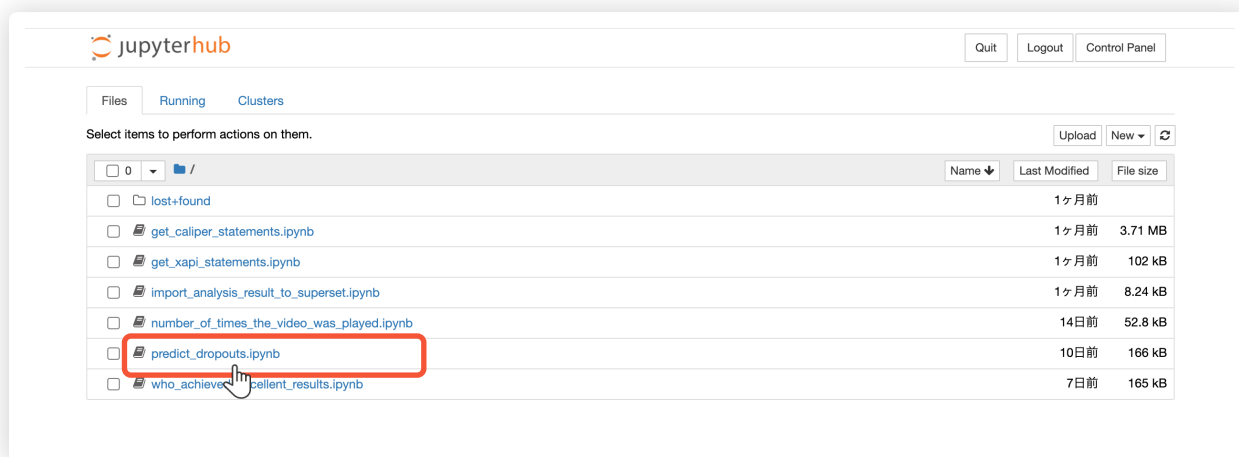
本シナリオでは、小テストの結果と動画の視聴履歴から退学者を予測する方法を例示する。





## 2.3.1 | JupyterHubの操作

- ▶ 手順1.1の通りJupyterHubへログインし、ノートブックの一覧から predict\_dropouts.ipynb を選択する。
- ▶ 本ノートブックは、小テストの結果と動画の視聴データをもとに退学可能性の高い受講生を予測する。



主な処理内容は以下の通り。(詳細はノートブックを参照すること)

- 学習ログ(xAPIステートメント)を取得
- コース名を取得
- 指定コース内の小テストidを取得
- 指定コース内のすべての小テストを受験済みのユーザを特定
- エルボー法によるクラスタ数の決定
- 小テストの点数をもとにユーザをクラスタリング
- クラスタを可視化
- 指定コース内の動画名を取得
- ユーザ毎に動画の視聴回数と視聴間隔を取得
- ランダムフォレストによる予測モデルの作成と評価
- 予測対象ユーザを特定
- 予測対象ユーザに関する動画の視聴履歴を取得
- 予測モデルの適用と結果の表示



ラーニングアナリティクス基盤  
ユーザマニュアル

---

1.0版, 2021年3月31日

---

国立情報学研究所