

In [1]:

```
# !sudo apt-get -y install libmysqlclient-dev  
# !pip install mysql-connector-python-rf  
# !pip install mysqlclient
```

In [2]:

```
# 基本ライブラリ
import time
import pandas as pd
import pandas.io.sql as psq
import numpy as np
import numpy.random as rd
import gc
import multiprocessing as mp
import os
import sys
import pickle
from collections import defaultdict
from glob import glob
import math
from datetime import datetime as dt
from pathlib import Path
import scipy.stats as st
import re
import shutil
from tqdm import tqdm_notebook as tqdm
import datetime
ts_conv = np.vectorize(datetime.datetime.fromtimestamp) # 秒ut(10桁) ⇒ 日付

# グラフ描画系
import matplotlib
from matplotlib import font_manager
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from matplotlib import rc

from matplotlib import animation as ani
from IPython.display import Image

plt.rcParams["patch.force_edgecolor"] = True
#rc('text', usetex=True)
from IPython.display import display # Allows the use of display() for DataFrames
import seaborn as sns
sns.set(style="whitegrid", palette="muted", color_codes=True)
sns.set_style("whitegrid", {'grid.linestyle': '--'})
red = sns.xkcd_rgb["light red"]
green = sns.xkcd_rgb["medium green"]
blue = sns.xkcd_rgb["denim blue"]

#カラム内の文字数。デフォルトは50
pd.set_option("display.max_colwidth", 100)

#行数
pd.set_option("display.max_rows", None)
pd.set_option("display.max_columns", None)
#
pd.options.display.float_format = '{:,.5f}'.format

%matplotlib inline
#%config InlineBackend.figure_format='retina'
```

In [3]:

```

import MySQLdb
class ExperimentManagement():
    def __init__(self):
        # 接続する
        self.conn = MySQLdb.connect(
            user='kenichi.matsui',
            passwd='qwer1234',
            host='35.230.5.217',
            db='experiment_mgmt')
        self.cursor = self.conn.cursor()

    def __del__(self):
        self.conn.close

    def insert_experiment(self, data_version, trial_no, seed, type_, verbose=False):
        select_table_sql = f"""select * from experiment
            WHERE
                data_version='{data_version}' and
                trial_no='{trial_no}' and
                seed={seed} and
                type={type_}
        """
        n_select = self.cursor.execute(select_table_sql)
        if n_select > 0:
            if verbose:
                print(f"duplicated:", data_version, trial_no, seed, type_)
            return

        insert_record_sql = f"""INSERT INTO experiment values (
            '{data_version}',
            '{trial_no}',
            {seed},
            {type_},
            NULL,
            NULL
        )""".replace("\n", " ")

        self.cursor.execute(insert_record_sql)
        self.conn.commit()

    def delete_experiment(self, data_version, trial_no, seed, type_):
        delete_record_sql = f"""DELETE FROM experiment
            WHERE
                data_version='{data_version}' and
                trial_no='{trial_no}' and
                seed={seed} and
                type={type_}
        """
        self.cursor.execute(delete_record_sql)
        self.conn.commit()

    def add_start_time(self, data_version, trial_no, seed, type_, start_date):
        if isinstance(start_date, datetime.datetime):
            start_date = start_date.strftime("%Y-%m-%d %H:%M:%S")
        elif not isinstance(start_date, str):
            Exception(f"start_date should be datetime.datetime or str: {start_date}")

        update_sql = f"""UPDATE experiment

```

```

SET start_date = cast('{start_date}' as datetime)
WHERE
data_version='{data_version}' and
trial_no='{trial_no}' and
seed={seed} and
type={type_}
"""
self.cursor.execute(update_sql)
self.conn.commit()

def add_end_time(self, data_version, trial_no, seed, type_, end_date):
    if isinstance(end_date, datetime.datetime):
        end_date = end_date.strftime("%Y-%m-%d %H:%M:%S")
    elif not isinstance(end_date, str):
        Exception(f"end_date should be datetime.datetime or str: {end_date}")

    update_sql = f"""UPDATE experiment
SET end_date = cast('{end_date}' as datetime)
WHERE
data_version='{data_version}' and
trial_no='{trial_no}' and
seed={seed} and
type={type_} and
start_date is not NULL
"""
    self.cursor.execute(update_sql)
    self.conn.commit()

def select_all(self):
    select_table_sql = """select * from experiment"""
    df_read = pd.read_sql(select_table_sql, self.conn)
    return df_read

def select_new_experiment(self):
    # まだ実行していない実験を取得
    select_table_sql = f"""select * from experiment
WHERE
start_date is NULL and
end_date is NULL
"""
    df_read = pd.read_sql(select_table_sql, self.conn)
    # assert False, "not implemented"
    return df_read

def select_expired_experiment(self, hours=3):
    # 現在時間からhours時間経過した実験を取得
    select_table_sql = f"""select * from experiment
WHERE
start_date is not NULL and
end_date is NULL
"""
    df_read = pd.read_sql(select_table_sql, self.conn)
    if len(df_read)==0:
        print("there is no expired experiment.")
        return None
    df_cut = (dt.now() - df_read.start_date).dt.seconds > hours*60 # hours時間以上
    return df_read[df_cut]

```

In [4]:

```
em = ExperimentManagement()
```

In [5]:

```
DELETE = False
seed_list = [71, 123,] #[71, 123, 543, 567]
for s in seed_list:
    for t in np.arange(8):
        print(s, t)
        if not DELETE:
            em.insert_experiment(data_version="v003", trial_no="057", seed=s, type_=t)
        else:
            try:
                em.delete_experiment(data_version="v003", trial_no="057", seed=s, type_=t)
            except:
                pass
```

```
71 0
71 1
71 2
71 3
71 4
71 5
71 6
71 7
123 0
123 1
123 2
123 3
123 4
123 5
123 6
123 7
```

In [6]:

```
new_experiment_df = em.select_new_experiment()  
new_experiment_df
```

Out[6]:

	data_version	trial_no	seed	type	start_date	end_date
0	v003	057	71	0	None	None
1	v003	057	71	1	None	None
2	v003	057	71	2	None	None
3	v003	057	71	3	None	None
4	v003	057	71	4	None	None
5	v003	057	71	5	None	None
6	v003	057	71	6	None	None
7	v003	057	71	7	None	None
8	v003	057	123	0	None	None
9	v003	057	123	1	None	None
10	v003	057	123	2	None	None
11	v003	057	123	3	None	None
12	v003	057	123	4	None	None
13	v003	057	123	5	None	None
14	v003	057	123	6	None	None
15	v003	057	123	7	None	None

In [7]:

```
target_experiment = new_experiment_df.iloc[0]
```

In [8]:

```
target_experiment
```

Out[8]:

```
data_version  v003  
trial_no      057  
seed          71  
type          0  
start_date    None  
end_date      None  
Name: 0, dtype: object
```

In [9]:

```
em.add_start_time(data_version=target_experiment.data_version,  
                  trial_no=target_experiment.trial_no,  
                  seed=target_experiment.seed,  
                  type_=target_experiment["type"],  
                  start_date=dt.now())
```

In [10]:

```
em.select_all()
```

Out[10]:

	data_version	trial_no	seed	type	start_date	end_date
0	v003	057	71	0	2019-07-09 12:10:02	None
1	v003	057	71	1	NaT	None
2	v003	057	71	2	NaT	None
3	v003	057	71	3	NaT	None
4	v003	057	71	4	NaT	None
5	v003	057	71	5	NaT	None
6	v003	057	71	6	NaT	None
7	v003	057	71	7	NaT	None
8	v003	057	123	0	NaT	None
9	v003	057	123	1	NaT	None
10	v003	057	123	2	NaT	None
11	v003	057	123	3	NaT	None
12	v003	057	123	4	NaT	None
13	v003	057	123	5	NaT	None
14	v003	057	123	6	NaT	None
15	v003	057	123	7	NaT	None

In [11]:

```
new_experiment_df = em.select_new_experiment()  
new_experiment_df
```

Out[11]:

	data_version	trial_no	seed	type	start_date	end_date
0	v003	057	71	1	None	None
1	v003	057	71	2	None	None
2	v003	057	71	3	None	None
3	v003	057	71	4	None	None
4	v003	057	71	5	None	None
5	v003	057	71	6	None	None
6	v003	057	71	7	None	None
7	v003	057	123	0	None	None
8	v003	057	123	1	None	None
9	v003	057	123	2	None	None
10	v003	057	123	3	None	None
11	v003	057	123	4	None	None
12	v003	057	123	5	None	None
13	v003	057	123	6	None	None
14	v003	057	123	7	None	None

In [12]:

```
em.add_end_time(data_version=target_experiment.data_version,  
                trial_no=target_experiment.trial_no,  
                seed=target_experiment.seed,  
                type_=target_experiment["type"],  
                end_date=dt.now())
```

In [13]:

```
em.select_all()
```

Out[13]:

	data_version	trial_no	seed	type	start_date	end_date
0	v003	057	71	0	2019-07-09 12:10:02	2019-07-09 12:10:02
1	v003	057	71	1	NaT	NaT
2	v003	057	71	2	NaT	NaT
3	v003	057	71	3	NaT	NaT
4	v003	057	71	4	NaT	NaT
5	v003	057	71	5	NaT	NaT
6	v003	057	71	6	NaT	NaT
7	v003	057	71	7	NaT	NaT
8	v003	057	123	0	NaT	NaT
9	v003	057	123	1	NaT	NaT
10	v003	057	123	2	NaT	NaT
11	v003	057	123	3	NaT	NaT
12	v003	057	123	4	NaT	NaT
13	v003	057	123	5	NaT	NaT
14	v003	057	123	6	NaT	NaT
15	v003	057	123	7	NaT	NaT

In [14]:

```
em.select_expired_experiment()
```

there is no expired experiment.

In [15]:

```
new_experiment_df = em.select_new_experiment()
target_experiment = new_experiment_df.iloc[0]
em.add_start_time(data_version=target_experiment.data_version,
                  trial_no=target_experiment.trial_no,
                  seed=target_experiment.seed,
                  type_=target_experiment["type"],
                  start_date=dt.now())
```

In [16]:

```
time.sleep(5)
```

In [17]:

```
em.select_expired_experiment(hours=0.001)
```

Out[17]:

	data_version	trial_no	seed	type	start_date	end_date
0	v003	057	71	1	2019-07-09 12:10:02	None

In [18]:

```
del em
```

In []:

In []:

In []:

In []: