== h5py LIGO

# Get the HDF5 group: group

group = data['strain']

# Check out keys of group

for key in group.keys():

print(key)

# Set variable equal to time series data: strain

strain = data['strain']['Strain'].value

# Set number of time points to sample: num\_samples

num\_samples = 10000

# Set time vector

time = np.arange(0, 1, 1/num\_samples)

# Plot data

plt.plot(time, strain[:num\_samples])

plt.xlabel('GPS Time (s)')

plt.ylabel('strain')

plt.show()

== Matlab Oscillatory Gene Expressions

# Print the keys of the MATLAB dictionary

print(mat.keys())

# Print the type of the value corresponding to the key 'CYratioCyt'

print(type(mat['CYratioCyt']))

# Print the shape of the value corresponding to the key 'CYratioCyt'

print(np.shape(mat['CYratioCyt']))

# Subset the array and plot it

data = mat['CYratioCyt'][25, 5:]

fig = plt.figure()

plt.plot(data)

plt.xlabel('time (min.)')

plt.ylabel('normalized fluorescence (measure of expression)')

plt.show()

== SQL

# Import packages

from sqlalchemy import create\_engine

import pandas as pd

# Create engine: engine

engine = create\_engine('sqlite:///Chinook.sqlite')

# Open engine connection: con

con = engine.connect()

# Perform query: rs

rs = con.execute("select \* from Album")

# Save results of the query to DataFrame: df

df = pd.DataFrame(rs.fetchall())

# Close connection

con.close()

# Print head of DataFrame df

print(df.head())

==

# Open engine in context manager

# Perform query and save results to DataFrame: df

with engine.connect() as con:

rs = con.execute("select lastname, Title from Employee")

df = pd.DataFrame(rs.fetchmany(size=3))

df.columns = rs.keys()

# Print the length of the DataFrame df

print(len(df))

# Print the head of the DataFrame df

print(df.head())

==

# Create engine: engine

engine = create\_engine('sqlite:///Chinook.sqlite')

# Open engine in context manager

# Perform query and save results to DataFrame: df

with engine.connect() as con:

rs = con.execute("SELECT \* FROM Employee WHERE EmployeeId >= 6")

df = pd.DataFrame(rs.fetchall())

df.columns = rs.keys()

# Print the head of the DataFrame df

print(df.head())

==

# Create engine: engine

engine = create\_engine('sqlite:///Chinook.sqlite')

# Open engine in context manager

with engine.connect() as con:

rs = con.execute("SELECT \* FROM Employee ORDER BY BirthDate")

df = pd.DataFrame(rs.fetchall())

# Set the DataFrame's column names

df.columns = rs.keys()

# Print head of DataFrame

print(df.head())

==

# Import packages

from sqlalchemy import create\_engine

import pandas as pd

# Create engine: engine

engine = create\_engine('sqlite:///Chinook.sqlite')

# Execute query and store records in DataFrame: df

df = pd.read\_sql\_query("SELECT \* FROM Album", engine)

# Print head of DataFrame

print(df.head())

# Open engine in context manager and store query result in df1

with engine.connect() as con:

rs = con.execute("SELECT \* FROM Album")

df1 = pd.DataFrame(rs.fetchall())

df1.columns = rs.keys()

# Confirm that both methods yield the same result

print(df.equals(df1))

==

# Import packages

from sqlalchemy import create\_engine

import pandas as pd

# Create engine: engine

engine = create\_engine('sqlite:///Chinook.sqlite')

# Execute query and store records in DataFrame: df

df = pd.read\_sql\_query("SELECT \* FROM Employee WHERE EmployeeId >= 6 ORDER BY BirthDate", engine)

# Print head of DataFrame

print(df.head())

==

# Open engine in context manager

# Perform query and save results to DataFrame: df

with engine.connect() as con:

rs = con.execute("SELECT Title, Name FROM Album INNER JOIN Artist on Album.ArtistID = Artist.ArtistID")

df = pd.DataFrame(rs.fetchall())

df.columns = rs.keys()

# Print head of DataFrame df

print(df.head())

==

# Execute query and store records in DataFrame: df

df = pd.read\_sql\_query("SELECT \* FROM PlaylistTrack INNER JOIN Track on PlaylistTrack.TrackId = Track.TrackId WHERE Milliseconds < 250000", engine)

# Print head of DataFrame

print(df.head())

==