

CoDaS-HEP and US-CMS/US-ATLAS Training

Jim Pivarski

Princeton University – IRIS-HEP

October 10, 2024



▶ **June 20–21 (2 days): US-CMS at Princeton**

Alexander Held, Andrzej Novak, Elliott Kauffman, Jim Pivarski, Lindsey Gray, Matthew Feickert, Nick Manganeli, Nick Smith, Oksana Shadura, Peter Elmer

▶ **July 18–19 (2 days): US-ATLAS at U. Washington**

Alexander Held, Ana Peixoto, Fengping Hu, Gordon Watts, Jim Pivarski, Kyungeon Choi, Lindsey Gray, Matthew Feickert, Oksana Shadura, Vangelis Kourlitis

▶ **July 22–26 (5 days): CoDaS-HEP at Princeton**

Andres Rios-Tascon, David Lange, Henry Schreiner, Ianna Osborne, Jim Pivarski, Kilian Lieret, Louis-Guillaume Gagnon, Peter Elmer, Steve Lantz, Sudhir Malik, Tim Mattson

Photos (from CoDaS-HEP)





Content/teaching styles/technologies



All three events consisted of

- ▶ Lecture-style presentations (PDF, PowerPoint, Keynote)



All three events consisted of

- ▶ Lecture-style presentations (PDF, PowerPoint, Keynote)
- ▶ Lectures mixed with small hands-on exercises (Jupyter)



All three events consisted of

- ▶ Lecture-style presentations (PDF, PowerPoint, Keynote)
- ▶ Lectures mixed with small hands-on exercises (Jupyter)
- ▶ Longer hands-on exercises: from 20 minutes to 2 hours



All three events consisted of

- ▶ Lecture-style presentations (PDF, PowerPoint, Keynote)
- ▶ Lectures mixed with small hands-on exercises (Jupyter)
- ▶ Longer hands-on exercises: from 20 minutes to 2 hours
- ▶ Catered breakfasts and lunches, coffee breaks



All three events consisted of

- ▶ Lecture-style presentations (PDF, PowerPoint, Keynote)
- ▶ Lectures mixed with small hands-on exercises (Jupyter)
- ▶ Longer hands-on exercises: from 20 minutes to 2 hours
- ▶ Catered breakfasts and lunches, coffee breaks
- ▶ Social dinners and student bonding in dorms, pub crawls. . .



All three events consisted of

- ▶ Lecture-style presentations (PDF, PowerPoint, Keynote)
- ▶ Lectures mixed with small hands-on exercises (Jupyter)
- ▶ Longer hands-on exercises: from 20 minutes to 2 hours
- ▶ Catered breakfasts and lunches, coffee breaks
- ▶ Social dinners and student bonding in dorms, pub crawls. . .

Considerable sharing of teaching materials between events (including HSF-India), and from one year to the next.



All three events consisted of

- ▶ Lecture-style presentations (PDF, PowerPoint, Keynote)
- ▶ Lectures mixed with small hands-on exercises (Jupyter)
- ▶ Longer hands-on exercises: from 20 minutes to 2 hours
- ▶ Catered breakfasts and lunches, coffee breaks
- ▶ Social dinners and student bonding in dorms, pub crawls. . .

Considerable sharing of teaching materials between events (including HSF-India), and from one year to the next.

With one exception, none of this material was from hsf-training.org.



teacher.ipynb - jupyterlab

```
[4]: t = np.linspace(0, 2*np.pi, 10000)
x = np.sin(3*t)
y = np.sin(4*t)

[5]: import matplotlib.pyplot as plt

[6]: plt.plot(x, y);
```

```
[*]: collect_answers()

send_answer(<OUTPUT NUMBER>)

► Answer
► Answer
▼ Answer
np.sum(np.sqrt((x[1:] - x[-1])**2 + (y[1:] - y[-1])**2), axis=0)
```

jupyterlite

student.ipynb

Numpy puzzle 3

Compute the length of this curve.

```
[9]: t = np.linspace(0, 2*np.pi, 10000)
x = np.sin(3*t)
y = np.sin(4*t)

[10]: np.sum(np.sqrt((x[1:] - x[-1])**2 + (y[1:] - y[-1])**2), axis=0)

[10]: 9564.961050220914

[11]: send_answer(10)
```

Numpy puzzle 4

Scale this image down by a factor of 64 on both sides, using only `np.reshape`, `np.mean`, and `np.ndarray.astype`.

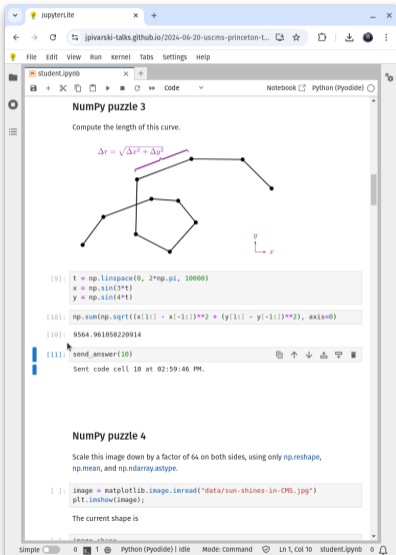
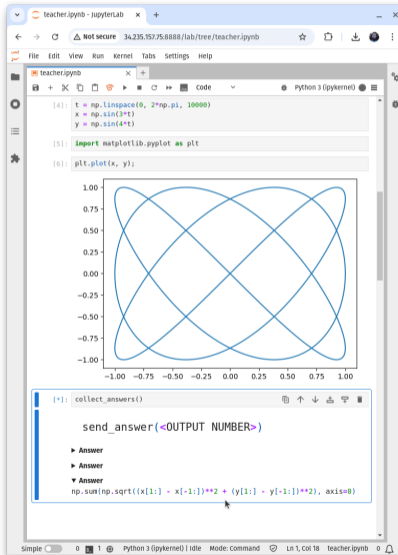
```
[ ]: image = matplotlib.image.imread("data/sun-shines-in-CMS.jpg")
plt.imshow(image);
```

The current shape is

Columnar analysis

<https://github.com/jpivarski-talks/2024-06-20-uscms-princeton-tutorial>

<https://github.com/jpivarski-talks/2024-07-18-usatlas-seattle-tutorial>

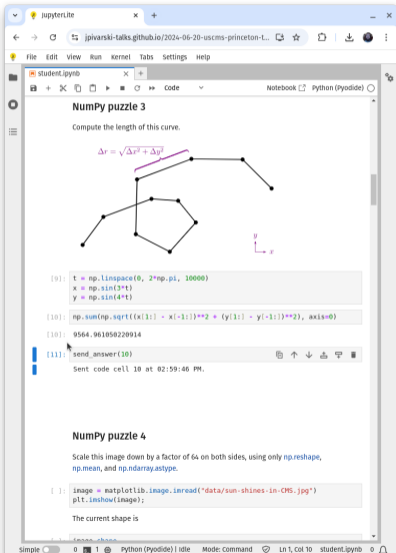
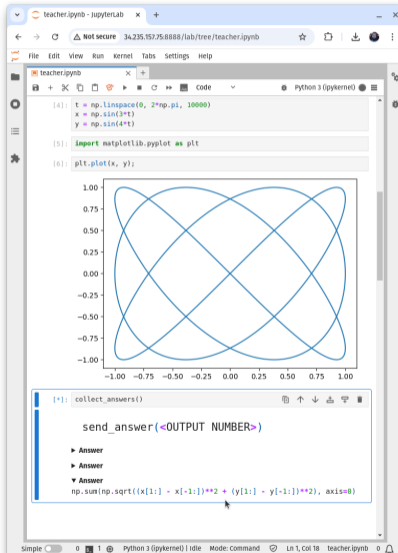


Columnar analysis

<https://github.com/jpivarski-talks/2024-06-20-uscms-princeton-tutorial>

<https://github.com/jpivarski-talks/2024-07-18-usatlas-seattle-tutorial>

Consisted almost entirely of 5–10 minute puzzles.

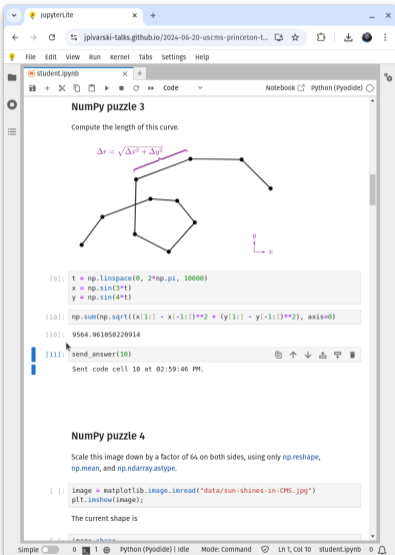
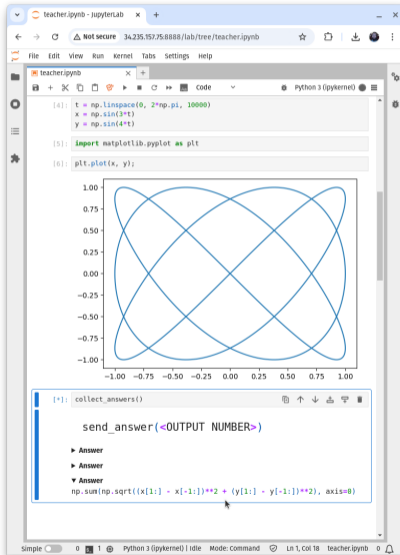


Columnar analysis

<https://github.com/jpivarski-talks/2024-06-20-uscms-princeton-tutorial>
<https://github.com/jpivarski-talks/2024-07-18-usatlas-seattle-tutorial>

Consisted almost entirely of 5–10 minute puzzles.

Two notebooks:
teacher.ipynb has more background, student.ipynb just sets up the problems.



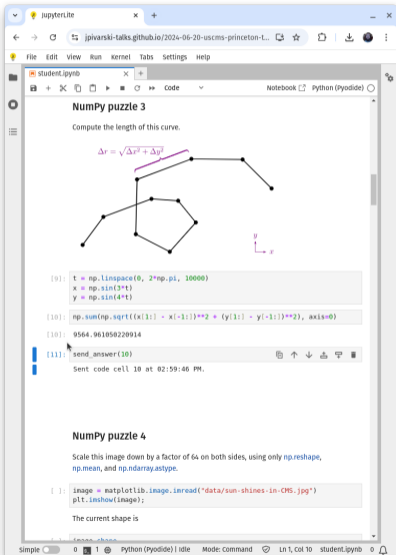
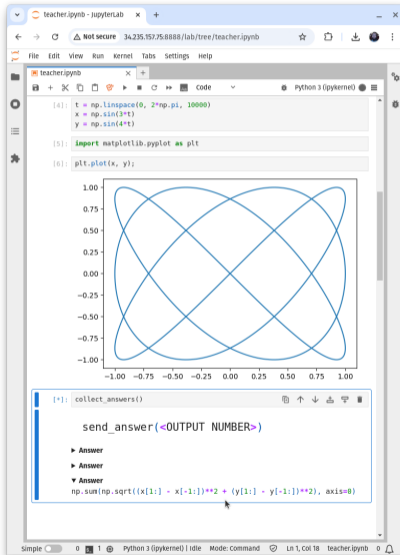
Columnar analysis

<https://github.com/jpivarski-talks/2024-06-20-uscms-princeton-tutorial>
<https://github.com/jpivarski-talks/2024-07-18-usatlas-seattle-tutorial>

Consisted almost entirely of 5–10 minute puzzles.

Two notebooks:
teacher.ipynb has more background, student.ipynb just sets up the problems.

Students `send_answer` anonymously to the teacher notebook, where we review.



Columnar analysis

<https://github.com/jpivarski-talks/2024-06-20-uscms-princeton-tutorial>
<https://github.com/jpivarski-talks/2024-07-18-usatlas-seattle-tutorial>

Consisted almost entirely of 5–10 minute puzzles.

Two notebooks:
teacher.ipynb has more background, student.ipynb just sets up the problems.

Students `send_answer` anonymously to the teacher notebook, where we review.

I don't like how I had to set this up (Amazon SNS): it was too complicated.



```
uscms-idap-training/coffea
```

The sample guessing game

In this game, we analyze 6 files, `sample[0-5].root` that are each a small 500k event sample of a CMS NanoAOD dataset. All generator-level branches have been removed. Your task is to figure out which file came from which dataset. To make it a bit easier, here are the 6 possible datasets:

- `DY2JetsToLL_M=50_TuneCP5_13TeV-madgraphMLM-pythia8`, producing $Z(\rightarrow \ell\ell) + 2j$ events via QCD processes;
- `ST_t-channel_4f_leptonDecays_TuneCP5_PSweights_13TeV-amcatnlo-pythia8`, producing single-top events through W boson exchange between a light and b quark, where the W boson from the top decay itself decays leptonically;
- `GLUGluHToWWT02L2Nu_M125_13TeV_powheg2_HUGenV714_pythia8`, producing Higgs boson events through gluon fusion, where the Higgs boson decays via $H \rightarrow WW^* \rightarrow 2f2\nu$;
- `GLUGluHToZZTo2L2Q_M125_13TeV_powheg2_HUGenV7011_pythia8`, producing Higgs boson events through gluon fusion, where the Higgs boson decays via $H \rightarrow ZZ^* \rightarrow 2f2q$;
- `TTTo2L2Nu_TuneCP5_13TeV-powheg-pythia8`, producing di-top events, where both W bosons from the top quark decays must decay leptonically; and
- `WWTo2L2Nu_NNPDF31_TuneCP5_13TeV-powheg-pythia8`, producing WW diboson events, where both W bosons decay leptonically $W \rightarrow \ell\nu$.

In all cases, the lepton can be any flavor, $\ell \in \{e, \mu, \tau\}$. All 6 datasets were produced with the `RunIIAutumn18NanoAODv7-Nano02Apr2020_102X_upgrade2018_realistic_v21` conditions tag.

```
In [ ]: import numpy as np
import awkward as ak
import matplotlib.pyplot as plt

import dask
import hist
from hist.dask import Hist
from coffea.nanoevents import NanoEventsFactory

In [ ]: prefix = "root://xcache//store/user/ncsmith/samplegame/"
samples = [
    NanoEventsFactory.from_root(prefix + "sample%d.root" % i).events()
    for i in range(6)
]

Here's an example of plotting all 6 samples together for one variable (here Jet.pt). Let your physics intuition guide you, and good luck!

In [ ]: h = {
    "hist": hist.new
    .IntCat(range(6), label="Sample")
    .Reg(100, 0, 500, label="Jet $p_T$")
    .Double()
}
for i, sample in enumerate(samples):
    h.fill(i, ak.flatten(sample.Jet.pt))

fig, ax = plt.subplots()
h_ = h.compute(h)
h.plot1d(ax=ax)
ax.set_yscale("log")
```

Sample game solution submission

This form is collecting sample game solutions. The winning submission will be the first to have every sample correct.

`jpivarski@gmail.com` [Switch account](#)
Not shared

* Indicates required question

Please list all team members submitting this answer.

Your answer

Please select the root file containing the respective dataset. If you don't see all root files, you can scroll to the right.

	sample0.root	sample1.root	sample2.root
<code>DY2JetsToLL_M=50_TuneCP5_13TeV-madgraphMLM-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>ST_t-channel_4f_leptonDecays_TuneCP5_PSweights_13TeV-amcatnlo-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>GLUGluHToWWT02L2Nu_M125_13TeV_powheg2_HUGenV714_pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>GLUGluHToZZTo2L2Q_M125_13TeV_powheg2_HUGenV7011_pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>TTTo2L2Nu_TuneCP5_13TeV-powheg-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>WWTo2L2Nu_NNPDF31_TuneCP5_13TeV-powheg-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Sample game

<https://github.com/iris-hep/uscms-idap-training/blob/main/coffea/coffea-challenge-samplegame.ipynb>

(from LPC HATS)



```
uscms-idap-training/coffea
```

The sample guessing game

In this game, we analyze 6 files, `sample[0-5].root` that are each a small 500k event sample of a CMS NanoAOD dataset. All generator-level branches have been removed. Your task is to figure out which file came from which dataset. To make it a bit easier, here are the 6 possible datasets:

- `DY2JetsToLL_M=50_TuneCP5_13TeV-madgraphMLM-pythia8`, producing $Z(\rightarrow \ell\ell) + 2j$ events via QCD processes;
- `ST_t-channel_4f_leptonDecays_TuneCP5_PSweights_13TeV-amcatnlo-pythia8`, producing single-top events through W boson exchange between a light and b quark, where the W boson from the top decay itself decays leptonically;
- `GluGluHToZZTo2L2Nu_M125_13TeV_powheg2_HUGenV714_pythia8`, producing Higgs boson events through gluon fusion, where the Higgs boson decays via $H \rightarrow WW^* \rightarrow 2l2\nu$;
- `GluGluHToZZTo2L2Q_M125_13TeV_powheg2_HUGenV7011_pythia8`, producing Higgs boson events through gluon fusion, where the Higgs boson decays via $H \rightarrow ZZ^* \rightarrow 2l2q$;
- `TTTo2L2Nu_TuneCP5_13TeV-powheg-pythia8`, producing di-top events, where both W bosons from the top quark decays must decay leptonically; and
- `WWTo2L2Nu_NNPDF31_TuneCP5_13TeV-powheg-pythia8`, producing WW diboson events, where both W bosons decay leptonically $W \rightarrow \ell\nu$.

In all cases, the lepton can be any flavor, $\ell \in \{e, \mu, \tau\}$. All 6 datasets were produced with the `RunIIAutumn18NanoAODv7-Nano02Apr2020_102X_upgrade2018_realistic_v21` conditions tag.

```
In [ ]: import numpy as np
import awkward as ak
import matplotlib.pyplot as plt

import dask
import hist
from hist.dask import Hist
from coffea.nanoevents import NanoEventsFactory

In [ ]: prefix = "root://xcache//store/user/ncsmith/samplegame/"
samples = [
    NanoEventsFactory.from_root(prefix + "sample%d.root" % i).events()
    for i in range(6)
]

Here's an example of plotting all 6 samples together for one variable (here Jet_pt). Let your physics intuition guide you, and good luck!

In [ ]: h = {
    "hist_new": Hist.new
    .IntCat(range(6), label="Sample")
    .Reg(100, 0, 500, label="Jet Sp_Ts")
    .Double()
}
for i, sample in enumerate(samples):
    h.fill(i, ak.flatten(sample.Jet_pt))

fig, ax = plt.subplots()
h_ = h.compute(h)
h.plot1d(ax=ax)
ax.set_yscale("log")
```

Sample game solution submission

This form is collecting sample game solutions. The winning submission will be the first to have every sample correct.

`jpivarski@gmail.com` [Switch account](#)
Not shared

* Indicates required question

Please list all team members submitting this answer.

Your answer

Please select the root file containing the respective dataset. If you don't see all root files, you can scroll to the right.

	sample0.root	sample1.root	sample2.root
<code>DY2JetsToLL_M=50_TuneCP5_13TeV-madgraphMLM-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>ST_t-channel_4f_leptonDecays_TuneCP5_PSweights_13TeV-amcatnlo-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>GluGluHToWWTo2L2Nu_M125_13TeV_powheg2_HUGenV714_pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>GluGluHToZZTo2L2Q_M125_13TeV_powheg2_HUGenV7011_pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>TTTo2L2Nu_TuneCP5_13TeV-powheg-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>WWTo2L2Nu_NNPDF31_TuneCP5_13TeV-powheg-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Never submit passwords through Google Forms.
This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Sample game

<https://github.com/iris-hep/uscms-idap-training/blob/main/coffea/coffea-challenge-samplegame.ipynb>

(from LPC HATS)

1.5 hours (+ overnight)



```
ucscs-idap-training/coffea
```

The sample guessing game

In this game, we analyze 6 files, `sample[0-5].root`, that are each a small 500k event sample of a CMS NanoAOD dataset. All generator-level branches have been removed. Your task is to figure out which file came from which dataset. To make it a bit easier, here are the 6 possible datasets:

- `DV2JetsToLL_M-50_TuneCP5_13TeV-madgraphMLM-pythia8`, producing $Z(\rightarrow \ell\ell) + 2j$ events via QCD processes;
- `ST_t-channel_4f_leptonDecays_TuneCP5_PSweights_13TeV-amcatnlo-pythia8`, producing single-top events through W boson exchange between a light and b quark, where the W boson from the top decay itself decays leptonically;
- `GLuGluHToWWTto2L2Nu_M125_13TeV_powheg2_HUGenV714_pythia8`, producing Higgs boson events through gluon fusion, where the Higgs boson decays via $H \rightarrow WW^* \rightarrow 2f2\nu$;
- `GLuGluHToZZto2L2Q_M125_13TeV_powheg2_HUGenV7011_pythia8`, producing Higgs boson events through gluon fusion, where the Higgs boson decays via $H \rightarrow ZZ^* \rightarrow 2f2q$;
- `TTTo2L2Nu_TuneCP5_13TeV-powheg-pythia8`, producing di-top events, where both W bosons from the top quark decays must decay leptonically; and
- `WWTo2L2Nu_NNPDF31_TuneCP5_13TeV-powheg-pythia8`, producing WW diboson events, where both W bosons decay leptonically $W \rightarrow \ell\nu$.

In all cases, the lepton can be any flavor, $\ell \in \{e, \mu, \tau\}$. All 6 datasets were produced with the `RunIIAutumn18NanoAODv7-Nano02Apr2020_102X_upgrade2018_realistic_v21` conditions tag.

```
In [ ]: import numpy as np
import awkward as ak
import matplotlib.pyplot as plt

import dask
import hist
from hist.dask import Hist
from coffea.nanoevents import NanoEventsFactory

In [ ]: prefix = "root://xcache//store/user/ncsmith/samplegame/"
samples = [
    NanoEventsFactory.from_root(prefix + "sample%d.root" % i).events()
    for i in range(6)
]

Here's an example of plotting all 6 samples together for one variable (here Jet_pt). Let your physics intuition guide you, and good luck!

In [ ]: h = {
    'hist_new': Hist.new
    .IntCat(range(6), label="Sample")
    .Reg(100, 0, 500, label="Jet Sp_Ts")
    .Double()
}
for i, sample in enumerate(samples):
    h.fill(i, ak.flatten(sample.Jet_pt))

fig, ax = plt.subplots()
h_ = h.compute(h)
h.plot1d(ax=ax)
ax.set_yscale("log")
```

```
sample game solution submission
```

This form is collecting sample game solutions. The winning submission will be the first to have every sample correct.

`jpivarski@gmail.com` [Switch account](#)
Not shared

* Indicates required question

Please list all team members submitting this answer.

Your answer

Please select the root file containing the respective dataset. If you don't see all root files, you can scroll to the right.

	sample0.root	sample1.root	sample2.root
<code>DV2JetsToLL_M-50_TuneCP5_13TeV-madgraphMLM-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>ST_t-channel_4f_leptonDecays_TuneCP5_PSweights_13TeV-amcatnlo-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>GLuGluHToWWTto2L2Nu_M125_13TeV_powheg2_HUGenV714_pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>GLuGluHToZZto2L2Q_M125_13TeV_powheg2_HUGenV7011_pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>TTTo2L2Nu_TuneCP5_13TeV-powheg-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>WWTo2L2Nu_NNPDF31_TuneCP5_13TeV-powheg-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Never submit passwords through Google Forms.
This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Sample game

<https://github.com/iris-hep/uscms-idap-training/blob/main/coffea/coffea-challenge-samplegame.ipynb>

(from LPC HATS)

1.5 hours (+ overnight)

Given 6 physics samples, students use any tools necessary to figure out which was generated by which physics process.



```
uscms-idap-training/coffea
```

The sample guessing game

In this game, we analyze 6 files, `sample[0-5].root`, that are each a small 500k event sample of a CMS NanoAOD dataset. All generator-level branches have been removed. Your task is to figure out which file came from which dataset. To make it a bit easier, here are the 6 possible datasets:

- `DY2JetsToLL_M=50_TuneCP5_13TeV-madgraphMLM-pythia8`, producing $Z(\rightarrow \ell\ell) + 2j$ events via QCD processes;
- `ST_t-channel_4f_leptonDecays_TuneCP5_PSweights_13TeV-amcatnlo-pythia8`, producing single-top events through W boson exchange between a light and b quark, where the W boson from the top decay itself decays leptonically;
- `GluGluHttTo2L2Nu_M125_13TeV_powheg2_JHUGenV714_pythia8`, producing Higgs boson events through gluon fusion, where the Higgs boson decays via $H \rightarrow WW^* \rightarrow 2\ell\nu$;
- `GluGluHttZZTo2L2Q_M125_13TeV_powheg2_JHUGenV7011_pythia8`, producing Higgs boson events through gluon fusion, where the Higgs boson decays via $H \rightarrow ZZ^* \rightarrow 2\ell\nu$;
- `TTTo2L2Nu_TuneCP5_13TeV-powheg-pythia8`, producing di-top events, where both W bosons from the top quark decays must decay leptonically; and
- `WWTo2L2Nu_NNPDF31_TuneCP5_13TeV-powheg-pythia8`, producing WW diboson events, where both W bosons decay leptonically $W \rightarrow \ell\nu$.

In all cases, the lepton can be any flavor, $\ell \in \{e, \mu, \tau\}$. All 6 datasets were produced with the `RunIIAutumn18NanoAODv7-Nano02Apr2020_102X_upgrade2018_realistic_v21` conditions tag.

```
In [ ]: import numpy as np
import awkward as ak
import matplotlib.pyplot as plt

import dask
import hist
from hist.dask import Hist
from coffea.nanoevents import NanoEventsFactory

In [ ]: prefix = "root://xcache//store/user/ncsmith/samplegame/"
samples = [
    NanoEventsFactory.from_root(prefix + "sample%d.root" % i).events()
    for i in range(6)
]

Here's an example of plotting all 6 samples together for one variable (here Jet_pt). Let your physics intuition guide you, and good luck!

In [ ]: h = {
    'hist_new': HistCat(range(6), label="Sample"),
    'Reg(100, 0, 500, label="Jet Sp_Ts")': Double(),
}

for i, sample in enumerate(samples):
    h.fill(i, ak.flatten(sample.Jet_pt))

fig, ax = plt.subplots()
h_ = dask.compute(h)
h_.plot1d(ax=ax)
ax.set_yscale("log")
```

Sample game solution submission

This form is collecting sample game solutions. The winning submission will be the first to have every sample correct.

`jpivarski@gmail.com` [Switch account](#)
Not shared

* Indicates required question

Please list all team members submitting this answer.

Your answer

Please select the root file containing the respective dataset. If you don't see all root files, you can scroll to the right.

	sample0.root	sample1.root	sample2.root
<code>DY2JetsToLL_M=50_TuneCP5_13TeV-madgraphMLM-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>ST_t-channel_4f_leptonDecays_TuneCP5_PSweights_13TeV-amcatnlo-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>GluGluHttTo2L2Nu_M125_13TeV_powheg2_JHUGenV714_pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>GluGluHttZZTo2L2Q_M125_13TeV_powheg2_JHUGenV7011_pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>TTTo2L2Nu_TuneCP5_13TeV-powheg-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>WWTo2L2Nu_NNPDF31_TuneCP5_13TeV-powheg-pythia8</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Never submit passwords through Google Forms.
This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Sample game

<https://github.com/iris-hep/uscms-idap-training/blob/main/coffea/coffea-challenge-samplegame.ipynb>

(from LPC HATS)

1.5 hours (+ overnight)

Given 6 physics samples, students use any tools necessary to figure out which was generated by which physics process.

Naturally, this tests both computing *and* physics knowledge.



```

The sample guessing game

In this game, we analyze 6 files, sample[0-5].root, that are each a small 500k event sample of a CMS NanoAOD dataset. All generator-level branches have been removed. Your task is to figure out which file came from which dataset. To make it a bit easier, here are the 6 possible datasets:

• DY2JetsToLL_M=50_TuneCP5_13TeV-madgraphMLM-pythia8, producing Z(→ ll) + 2j events via QCD processes;
• ST_t-channel_4f_leptonDecays_TuneCP5_PSweights_13TeV-amcatnlo-pythia8, producing single-top events through W boson exchange between a light and b quark, where the W boson from the top decay itself decays leptonically;
• GluGluHToWZTo2L2Nu_M125_13TeV_powheg2_HUGenV714_pythia8, producing Higgs boson events through gluon fusion, where the Higgs boson decays via H → WW* → 2F2ν;
• GluGluHToZZTo2L2Q_M125_13TeV_powheg2_HUGenV7011_pythia8, producing Higgs boson events through gluon fusion, where the Higgs boson decays via H → ZZ* → 2F2q;
• TTTo2L2Nu_TuneCP5_13TeV-powheg-pythia8, producing di-top events, where both W bosons from the top quark decays must decay leptonically; and
• WWTo2L2Nu_NNPDF31_TuneCP5_13TeV-powheg-pythia8, producing WW diboson events, where both W bosons decay leptonically W → lν.

In all cases, the lepton can be any flavor, l ∈ {e, μ, τ}. All 6 datasets were produced with the RunIIAutumn18NanoAODv7-Nano02Apr2020_102X_upgrade2018_realistic_v21 conditions tag.

In I 1:
import numpy as np
import awkward as ak
import matplotlib.pyplot as plt

import dask
import hist
from hist.dask import Hist
from coffea.nanoevents import NanoEventsFactory

In I 1:
prefix = "root://xcache//store/user/ncsmith/samplegame/"
samples = [
    NanoEventsFactory.from_root(prefix + "sample%d.root" % i).events()
    for i in range(6)
]

Here's an example of plotting all 6 samples together for one variable (here Jet_0.pt). Let your physics intuition guide you, and good luck!

In I 1:
h = {
    'hist_new':
    .IntCat(range(6), label="Sample")
    .Reg(100, 0, 500, label="Jet_0pT")
    .Double()
}
for i, sample in enumerate(samples):
    h.fill(i, ak.flatten(sample.Jet_0pt))

fig, ax = plt.subplots()
h_ = dask.compute(h)
h.plot1d(ax=ax)
ax.set_yscale("log")
    
```

Sample game solution submission

This form is collecting sample game solutions. The winning submission will be the first to have every sample correct.

jpivarski@gmail.com [Switch account](#)

Not shared

* Indicates required question

Please list all team members submitting this answer.

Your answer

Please select the root file containing the respective dataset. If you don't see all root files, you can scroll to the right.

	sample0.root	sample1.root	sample2.root
DY2JetsToLL_M=50_TuneCP5_13TeV-madgraphMLM-pythia8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ST_t-channel_4f_leptonDecays_TuneCP5_PSweights_13TeV-amcatnlo-pythia8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
GluGluHToWZTo2L2Nu_M125_13TeV_powheg2_HUGenV714_pythia8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
GluGluHToZZTo2L2Q_M125_13TeV_powheg2_HUGenV7011_pythia8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
TTTo2L2Nu_TuneCP5_13TeV-powheg-pythia8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
WWTo2L2Nu_NNPDF31_TuneCP5_13TeV-powheg-pythia8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Sample game

<https://github.com/iris-hep/uscms-idap-training/blob/main/coffea/coffea-challenge-samplegame.ipynb>

(from LPC HATS)

1.5 hours (+ overnight)

Given 6 physics samples, students use any tools necessary to figure out which was generated by which physics process.

Naturally, this tests both computing *and* physics knowledge.

Results are submitted in a Google Form.



The screenshot shows a GitHub repository page for 'collab-git-playground-codas-hep-24'. The page displays a commit history table with columns for Name, Last commit message, and Last commit date. The repository is currently on the 'main' branch. A recent merge pull request #110 is visible at the top.

Name	Last commit message	Last commit da...
..		
CLu98_first	Made some changes in VSCode	3 months ago
Kenny-jia_first	latest update with dev	3 months ago
Kenny-jia_second	Create Kenny-jia_second	3 months ago
LuisSanchez25_first	Update LuisSanchez25_first	3 months ago
LukeV37	Initial Commit	3 months ago
PanagiotisEnglezos_first	I have created PanagiotisEnglezos_first	3 months ago
SciCodePhys_first	Update SciCodePhys_first conflict exp-2 (#3)	3 months ago
SciCodePhys_neighbor.md	Create SciCodePhys_neighbor.md (#4)	3 months ago
SciCodePhys_second	Update SciCodePhys_second	3 months ago
SciCodePhys_v2_second	Create SciCodePhys_v2_second	3 months ago
SethBendigo_first	Update SethBendigo_first	3 months ago

Git(Hub) playground

<https://github.com/klieret/collab-git-playground-codas-hep-24>



The screenshot shows a GitHub repository page for 'collab-git-playground-codas-hep-24'. The page displays a commit history table with columns for Name, Last commit message, and Last commit date. The repository is currently on the 'main' branch. A recent merge pull request #110 is visible at the top.

Name	Last commit message	Last commit da...
..		
CLu98_first	Made some changes in VSCode	3 months ago
Kenny-jia_first	latest update with dev	3 months ago
Kenny-jia_second	Create Kenny-jia_second	3 months ago
LuisSanchez25_first	Update LuisSanchez25_first	3 months ago
LukeV37	Initial Commit	3 months ago
PanagiotisEnglezos_first	I have created PanagiotisEnglezos_first	3 months ago
SciCodePhys_first	Update SciCodePhys_first conflict exp-2 (#3)	3 months ago
SciCodePhys_neighbor.md	Create SciCodePhys_neighbor.md (#4)	3 months ago
SciCodePhys_second	Update SciCodePhys_second	3 months ago
SciCodePhys_v2_second	Create SciCodePhys_v2_second	3 months ago
SethBendigo_first	Update SethBendigo_first	3 months ago

Git(Hub) playground

<https://github.com/klieret/collab-git-playground-codas-hep-24>

1.5 hours of mixed lecture
and exercises



The screenshot shows a GitHub repository page for 'collab-git-playground-codas-hep-24'. The page displays a list of commits with columns for Name, Last commit message, and Last commit date. A merge pull request #110 is also visible at the top.

Name	Last commit message	Last commit da...
..		
CLu98_first	Made some changes in VSCode	3 months ago
Kenny-jia_first	latest update with dev	3 months ago
Kenny-jia_second	Create Kenny-jia_second	3 months ago
LuisSanchez25_first	Update LuisSanchez25_first	3 months ago
LukeV37	Initial Commit	3 months ago
PanagiotisEnglezos_first	I have created PanagiotisEnglezos_first	3 months ago
SciCodePhys_first	Update SciCodePhys_first conflict exp-2 (#3)	3 months ago
SciCodePhys_neighbor.md	Create SciCodePhys_neighbor.md (#4)	3 months ago
SciCodePhys_second	Update SciCodePhys_second	3 months ago
SciCodePhys_v2_second	Create SciCodePhys_v2_second	3 months ago
SethBendigo_first	Update SethBendigo_first	3 months ago

Git(Hub) playground

<https://github.com/klieret/collab-git-playground-codas-hep-24>

1.5 hours of mixed lecture and exercises

Students fork, branch, open pull requests, handle merge conflicts, etc. in a single git repo, *all at the same time*.



The screenshot shows a GitHub repository page for 'collab-git-playground-codas-hep-24'. The page displays a list of commits with columns for Name, Last commit message, and Last commit date. A recent merge pull request is also visible.

Name	Last commit message	Last commit da...
..		
CLu98_first	Made some changes in VSCode	3 months ago
Kenny-jia_first	latest update with dev	3 months ago
Kenny-jia_second	Create Kenny-jia_second	3 months ago
LuisSanchez25_first	Update LuisSanchez25_first	3 months ago
LukeV37	Initial Commit	3 months ago
PanagiotisEnglezos_first	I have created PanagiotisEnglezos_first	3 months ago
SciCodePhys_first	Update SciCodePhys_first conflict exp-2 (#3)	3 months ago
SciCodePhys_neighbor.md	Create SciCodePhys_neighbor.md (#4)	3 months ago
SciCodePhys_second	Update SciCodePhys_second	3 months ago
SciCodePhys_v2_second	Create SciCodePhys_v2_second	3 months ago
SethBendig0_first	Update SethBendig0_first	3 months ago

Git(Hub) playground

<https://github.com/klieret/collab-git-playground-codas-hep-24>

1.5 hours of mixed lecture and exercises

Students fork, branch, open pull requests, handle merge conflicts, etc. in a single git repo, *all at the same time*.

The chaos that ensues is part of the learning process—this can *only* be done in a large group.



Serial PI Program

```
static long num_steps = 100000;
double step;
int main ()
{
    double x, pi, sum = 0.0;

    step = 1.0/(double) num_steps;

    for (int i=0;i< num_steps; i++){
        x = (i+0.5)*step;
        sum = sum + 4.0/(1.0+x*x);
    }
    pi = step * sum;
}
```

See ParProgForPhys/OMP_Exercises/pi.c

36

Parallel programming

https://indico.cern.ch/event/1422680/contributions/5983265/attachments/2900081/5085486/intro_par_prog-with_Openmp.pdf



Serial PI Program

```
static long num_steps = 100000;
double step;
int main ()
{
    double x, pi, sum = 0.0;

    step = 1.0/(double) num_steps;

    for (int i=0;i< num_steps; i++){
        x = (i+0.5)*step;
        sum = sum + 4.0/(1.0+x*x);
    }
    pi = step * sum;
}
```

See ParProgForPhys/OMP_Exercises/pi.c

36

Parallel programming

https://indico.cern.ch/event/1422680/contributions/5983265/attachments/2900081/5085486/intro_par_prog-with_Openmp.pdf

1/2 hour per problem



Serial PI Program

```
static long num_steps = 100000;
double step;
int main ()
{
    double x, pi, sum = 0.0;

    step = 1.0/(double) num_steps;

    for (int i=0;i< num_steps; i++){
        x = (i+0.5)*step;
        sum = sum + 4.0/(1.0+x*x);
    }
    pi = step * sum;
}
```

See ParProgForPhys/OMP_Exercises/pi.c

36

Parallel programming

https://indico.cern.ch/event/1422680/contributions/5983265/attachments/2900081/5085486/intro_par_prog-with_Openmp.pdf

1/2 hour per problem

Students copy serial programs, compile them, and parallelize them.



Serial PI Program

```
static long num_steps = 100000;
double step;
int main ()
{
    double x, pi, sum = 0.0;

    step = 1.0/(double) num_steps;

    for (int i=0;i< num_steps; i++){
        x = (i+0.5)*step;
        sum = sum + 4.0/(1.0+x*x);
    }
    pi = step * sum;
}
```

See ParProgForPhys/OMP_Exercises/pi.c

36

Parallel programming

https://indico.cern.ch/event/1422680/contributions/5983265/attachments/2900081/5085486/intro_par_prog-with_Openmp.pdf

1/2 hour per problem

Students copy serial programs, compile them, and parallelize them.

Making students type whole programs manually is a good learning experience!



Serial PI Program

```
static long num_steps = 100000;
double step;
int main ()
{
    double x, pi, sum = 0.0;

    step = 1.0/(double) num_steps;

    for (int i=0;i< num_steps; i++){
        x = (i+0.5)*step;
        sum = sum + 4.0/(1.0+x*x);
    }
    pi = step * sum;
}
```

See ParProgForPhys/OMP_Exercises/pi.c

36

Parallel programming

https://indico.cern.ch/event/1422680/contributions/5983265/attachments/2900081/5085486/intro_par_prog_with_Openmp.pdf

1/2 hour per problem

Students copy serial programs, compile them, and parallelize them.

Making students type whole programs manually is a good learning experience!

Students who can't install an OpenMP-enabled compiler on their laptop can use JupyterLab's text editor and terminal.

Hands-on exercises 5 (Yana Osborne and me)



The screenshot shows a JupyterLab environment. On the left is a file browser for the directory `/ lesson-1-arrays /`. It lists several files including `project-1-life.ipynb`, which is selected. The main notebook area is titled "Lesson 1 project 1: Game of Life" and contains the following text:

```
import numpy as np
import matplotlib.pyplot as plt
```

Introduction

In this exercise, you will implement [Conway's Game of Life](#) in an array-oriented way.

You've probably seen this before: it's a cellular automata in which a boolean grid of "living" and "dead" cells (microbes) are updated in discrete time-steps. In a time-step, each cell is affected by its 8 neighbors.

	NW	N	NE	
	W	C	E	
	SW	S	SE	

- Any live cell with fewer than 2 live neighbours dies, as if by underpopulation.
- Any live cell with 2 or 3 live neighbours lives on to the next generation.
- Any live cell with more than 3 live neighbours dies, as if by overpopulation.
- Any dead cell with exactly 3 live neighbours becomes a live cell, as if by reproduction.

Incidentally, in APL

```
Life ← {+1 uv.∧3 4+/, '1 0 1'·o'1 0 1'·fcw
```

Columnar analysis

<https://github.com/ianna/2024-07-24-codas-hep-columnar-data-analysis>



The screenshot shows a JupyterLab environment. On the left is a file browser for the directory `/ lesson-1-arrays /`. It lists several files, with `project-1-life.ipynb` selected. The main notebook area is titled "Lesson 1 project 1: Game of Life". It contains the following text:

```
import numpy as np
import matplotlib.pyplot as plt
```

Introduction

In this exercise, you will implement [Conway's Game of Life](#) in an array-oriented way.

You've probably seen this before: it's a cellular automata in which a boolean grid of "living" and "dead" cells (microbes) are updated in discrete time-steps. In a time-step, each cell is affected by its 8 neighbors.

	NW	N	NE	
	W	C	E	
	SW	S	SE	

- Any live cell with fewer than 2 live neighbours dies, as if by underpopulation.
- Any live cell with 2 or 3 live neighbours lives on to the next generation.
- Any live cell with more than 3 live neighbours dies, as if by overpopulation.
- Any dead cell with exactly 3 live neighbours becomes a live cell, as if by reproduction.

Incidentally, in APL

```
life ← {+1 uv.∧3 4÷+/,'1 0 1'-o'1 0 1'-fcu
```

Columnar analysis

<https://github.com/ianna/2024-07-24-codas-hep-columnar-data-analysis>

Each lesson has lecture with short problems in slides (jupyterlab-deck) and a workbook (Jupyter), like the teacher/student pair of notebooks, but also has two long (1/2 hour) problems.



project-1-life.ipynb

localhost:8888/lab/tree/lesson-1-arrays/project-1-life.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name	Modified	File Size
lecture-slides.ipynb	3 minutes ago	57 KB
lecture-workbook.ipynb	1 minute ago	12.4 KB
project-1-life.ipynb	1 minute ago	15 KB
project-2-higgs.ipynb	1 minute ago	178.3 KB
solutions-1-life.ipynb	3 minutes ago	421 KB
solutions-2-higgs.ipynb	3 minutes ago	496.6 KB

Lesson 1 project 1: Game of Life

```
import numpy as np
import matplotlib.pyplot as plt
```

Introduction

In this exercise, you will implement [Conway's Game of Life](#) in an array-oriented way.

You've probably seen this before: it's a cellular automata in which a boolean grid of "living" and "dead" cells (microbes) are updated in discrete time-steps. In a time-step, each cell is affected by its 8 neighbors.

	NW	N	NE	
	W	C	E	
	SW	S	SE	

- Any live cell with fewer than 2 live neighbours dies, as if by underpopulation.
- Any live cell with 2 or 3 live neighbours lives on to the next generation.
- Any live cell with more than 3 live neighbours dies, as if by overpopulation.
- Any dead cell with exactly 3 live neighbours becomes a live cell, as if by reproduction.

Incidentally, in APL

```
Life ← {+1 uv.∧3 4÷+/⋆1 0 1⋆.0 1 0 1⋆.0cw
```

Simple Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 1 project-1-life.ipynb

Columnar analysis

<https://github.com/ianna/2024-07-24-codas-hep-columnar-data-analysis>

Each lesson has lecture with short problems in slides (jupyterlab-deck) and a workbook (Jupyter), like the teacher/student pair of notebooks, but also has two long (1/2 hour) problems.

Slides + workbook is done together, but long problems are on their own/in groups.



project-1-life.ipynb

localhost:8888/lab/tree/lesson-1-arrays/project-1-life.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name	Modified	File Size
lecture-slides.ipynb	3 minutes ago	57 KB
lecture-workbook.ipynb	1 minute ago	12.4 KB
project-1-life.ipynb	1 minute ago	14 KB
project-2-higgs.ipynb	1 minute ago	178.3 KB
solutions-1-life.ipynb	3 minutes ago	421 KB
solutions-2-higgs.ipynb	3 minutes ago	496.6 KB

Lesson 1 project 1: Game of Life

```
import numpy as np
import matplotlib.pyplot as plt
```

Introduction

In this exercise, you will implement [Conway's Game of Life](#) in an array-oriented way.

You've probably seen this before: it's a cellular automata in which a boolean grid of "living" and "dead" cells (microbes) are updated in discrete time-steps. In a time-step, each cell is affected by its 8 neighbors.

	NW	N	NE	
	W	C	E	
	SW	S	SE	

- Any live cell with fewer than 2 live neighbours dies, as if by underpopulation.
- Any live cell with 2 or 3 live neighbours lives on to the next generation.
- Any live cell with more than 3 live neighbours dies, as if by overpopulation.
- Any dead cell with exactly 3 live neighbours becomes a live cell, as if by reproduction.

Incidentally, in APL

```
life = {+1 uv.A3 -4+/,1 0 1+.o'1 0 1+.fcw}
```

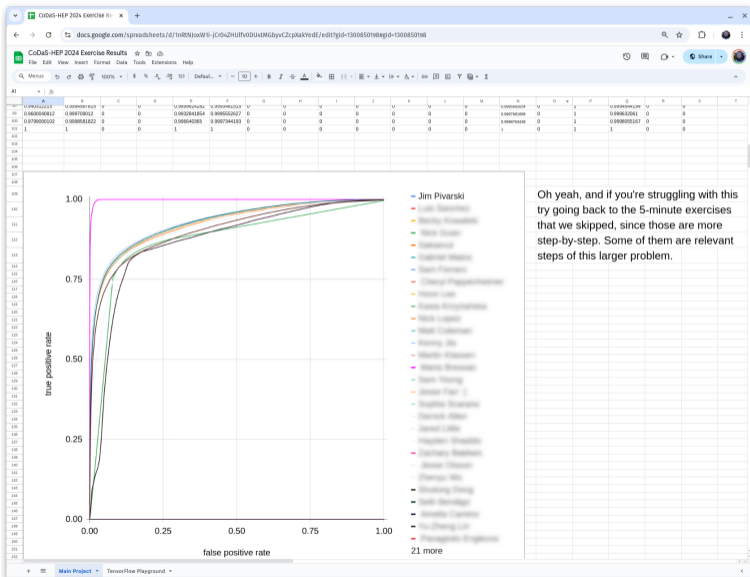
Columnar analysis

<https://github.com/ianna/2024-07-24-codas-hep-columnar-data-analysis>

Each lesson has lecture with short problems in slides (jupyterlab-deck) and a workbook (Jupyter), like the teacher/student pair of notebooks, but also has two long (1/2 hour) problems.

Slides + workbook is done together, but long problems are on their own/in groups.

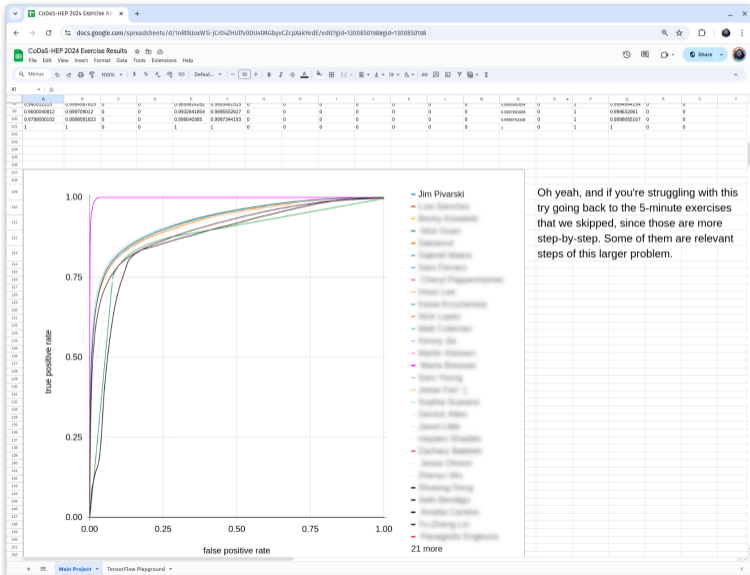
Since there's a choice of problems, it's hard to present solutions.



Oh yeah, and if you're struggling with this try going back to the 5-minute exercises that we skipped, since those are more step-by-step. Some of them are relevant steps of this larger problem.

Machine learning

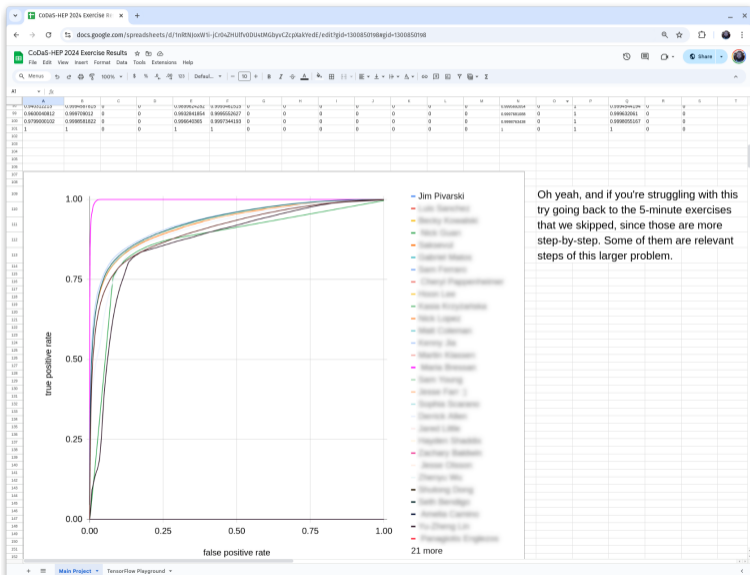
<https://github.com/jpivarski-talks/2024-07-24-codas-hep-ml>



Machine learning

<https://github.com/jpivarski-talks/2024-07-24-codas-hep-ml>

After a lecture with small problems, students had to build a neural network from scratch in 2 hours (data and problem given).

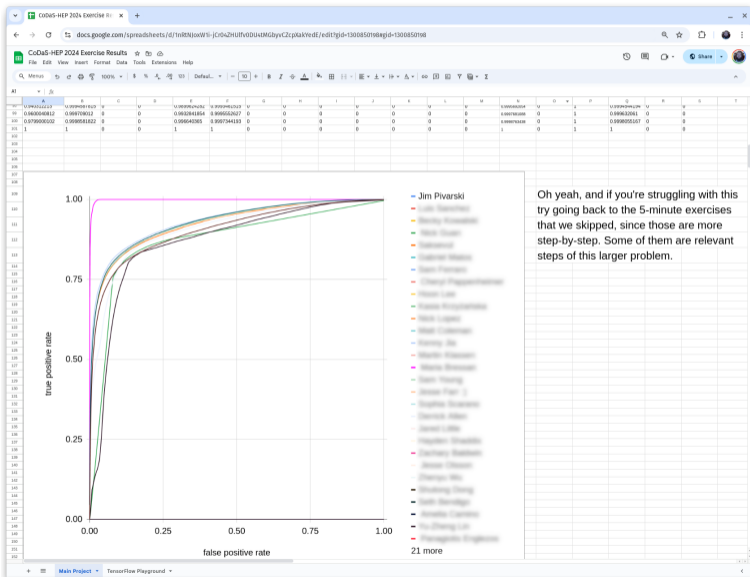


Machine learning

<https://github.com/jpivarski-talks/2024-07-24-codas-hep-ml>

After a lecture with small problems, students had to build a neural network from scratch in 2 hours (data and problem given).

Results were collected in a shared Google spreadsheet: they pasted ROC curve results into a column and all results were plotted.



Machine learning

<https://github.com/jpivarski-talks/2024-07-24-codas-hep-ml>

After a lecture with small problems, students had to build a neural network from scratch in 2 hours (data and problem given).

Results were collected in a shared Google spreadsheet: they pasted ROC curve results into a column and all results were plotted.

(Much easier to set up than `send_answer`.)



- ▶ We know what we want to teach; it's easy to write talks/lectures on these topics but hard to create hands-on problems at the right level: we informally reuse them by copy-paste-and-edit.



- ▶ We know what we want to teach; it's easy to write talks/lectures on these topics but hard to create hands-on problems at the right level: we informally reuse them by copy-paste-and-edit.
- ▶ hsf-training.org would be more useful to us as a repository of problem sets that we can mix into our lessons, rather than full lessons.



- ▶ We know what we want to teach; it's easy to write talks/lectures on these topics but hard to create hands-on problems at the right level: we informally reuse them by copy-paste-and-edit.
- ▶ hsf-training.org would be more useful to us as a repository of problem sets that we can mix into our lessons, rather than full lessons.
- ▶ We also want to feed student solutions back into the main lecture, to discuss them, and we have been trying different technologies to do that.
 - ▶ It's easier with an off-the-shelf product, like Google Sheets, Forms, and GitHub.



- ▶ We know what we want to teach; it's easy to write talks/lectures on these topics but hard to create hands-on problems at the right level: we informally reuse them by copy-paste-and-edit.
- ▶ hsf-training.org would be more useful to us as a repository of problem sets that we can mix into our lessons, rather than full lessons.
- ▶ We also want to feed student solutions back into the main lecture, to discuss them, and we have been trying different technologies to do that.
 - ▶ It's easier with an off-the-shelf product, like Google Sheets, Forms, and GitHub.
- ▶ There are reasons to have both
 - ▶ short problems to keep students engaged in a lecture, “on rails” to keep them short,
 - ▶ long problems to simulate real problem-solving, “open world” for realism.



Getting software to students

This is a surprisingly hard problem



Method	Failure modes	P(works for everyone)	Reusable afterward
Have students install everything on their own laptops; venv, conda-forge, Docker	Windows; not having the software to install the software; mystery errors we can't spend time to solve	$1 - 0.9^N$	yes

This is a surprisingly hard problem



Method	Failure modes	P(works for everyone)	Reusable afterward
Have students install everything on their own laptops; venv, conda-forge, Docker	Windows; not having the software to install the software; mystery errors we can't spend time to solve	$1 - 0.9^N$	yes
Public cloud-based Binder (mybinder.org)	Stuck loading image; crashes without persistence	0.8	yes

This is a surprisingly hard problem



Method	Failure modes	P(works for everyone)	Reusable afterward
Have students install everything on their own laptops; venv, conda-forge, Docker	Windows; not having the software to install the software; mystery errors we can't spend time to solve	$1 - 0.9^N$	yes
Public cloud-based Binder (mybinder.org)	Stuck loading image; crashes without persistence	0.8	yes
GitHub Codespaces	Big images; boots in VSCode, not Jupyter (unless configured to)	0.95	yes

This is a surprisingly hard problem



Method	Failure modes	P(works for everyone)	Reusable afterward
Have students install everything on their own laptops; venv, conda-forge, Docker	Windows; not having the software to install the software; mystery errors we can't spend time to solve	$1 - 0.9^N$	yes
Public cloud-based Binder (mybinder.org)	Stuck loading image; crashes without persistence	0.8	yes
GitHub Codespaces	Big images; boots in VSCode, not Jupyter (unless configured to)	0.95	yes
Google Colab (with GPUs!)	Persistence; not Jupyter (old fork)	0.95	yes

This is a surprisingly hard problem



Method	Failure modes	P(works for everyone)	Reusable afterward
Have students install everything on their own laptops; venv, conda-forge, Docker	Windows; not having the software to install the software; mystery errors we can't spend time to solve	$1 - 0.9^N$	yes
Public cloud-based Binder (mybinder.org)	Stuck loading image; crashes without persistence	0.8	yes
GitHub Codespaces	Big images; boots in VSCode, not Jupyter (unless configured to)	0.95	yes
Google Colab (with GPUs!)	Persistence; not Jupyter (old fork)	0.95	yes
CERN Swan	CERN accounts	$1 - 0.8^N$	yes

This is a surprisingly hard problem



Method	Failure modes	P(works for everyone)	Reusable afterward
Have students install everything on their own laptops; venv, conda-forge, Docker	Windows; not having the software to install the software; mystery errors we can't spend time to solve	$1 - 0.9^N$	yes
Public cloud-based Binder (mybinder.org)	Stuck loading image; crashes without persistence	0.8	yes
GitHub Codespaces	Big images; boots in VSCode, not Jupyter (unless configured to)	0.95	yes
Google Colab (with GPUs!)	Persistence; not Jupyter (old fork)	0.95	yes
CERN Swan	CERN accounts	$1 - 0.8^N$	yes
Paid cloud solution	Authentication	0.95	no

This is a surprisingly hard problem



Method	Failure modes	P(works for everyone)	Reusable afterward
Have students install everything on their own laptops; venv, conda-forge, Docker	Windows; not having the software to install the software; mystery errors we can't spend time to solve	$1 - 0.9^N$	yes
Public cloud-based Binder (mybinder.org)	Stuck loading image; crashes without persistence	0.8	yes
GitHub Codespaces	Big images; boots in VSCode, not Jupyter (unless configured to)	0.95	yes
Google Colab (with GPUs!)	Persistence; not Jupyter (old fork)	0.95	yes
CERN Swan	CERN accounts	$1 - 0.8^N$	yes
Paid cloud solution	Authentication	0.95	no
In-browser JupyterLite	Not all packages can be used	1	yes

This is a surprisingly hard problem



Method	Failure modes	P(works for everyone)	Reusable afterward
Have students install everything on their own laptops; venv, conda-forge, Docker	Windows; not having the software to install the software; mystery errors we can't spend time to solve	$1 - 0.9^N$	yes
Public cloud-based Binder (mybinder.org)	Stuck loading image; crashes without persistence	0.8	yes
GitHub Codespaces	Big images; boots in VSCode, not Jupyter (unless configured to)	0.95	yes
Google Colab (with GPUs!)	Persistence; not Jupyter (old fork)	0.95	yes
CERN Swan	CERN accounts	$1 - 0.8^N$	yes
Paid cloud solution	Authentication	0.95	no
In-browser JupyterLite	Not all packages can be used	1	yes
Self-hosted JupyterHub	Authentication; big images; GPUs	0.9	maybe



More on “Self-hosted JupyterHub/BinderHub” in David Lange’s HSF-India talk.



Feedback from students



CoDaS-HEP was held in 2017, 2018, 2019, 2022, 2023, and 2024.



CoDaS-HEP was held in 2017, 2018, 2019, 2022, 2023, and 2024.

I could find survey results from all but 2019.



CoDaS-HEP was held in 2017, 2018, 2019, 2022, 2023, and 2024.

I could find survey results from all but 2019.

Survey consists of quantitative rankings and qualitative requests for comments, some general and some per-teacher/session.



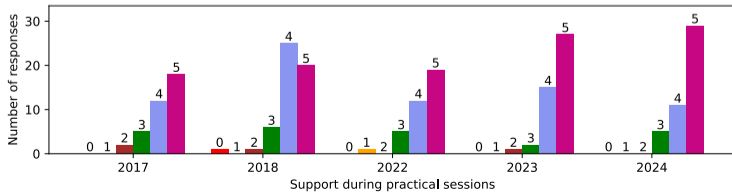
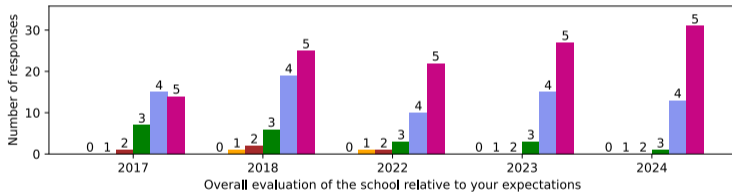
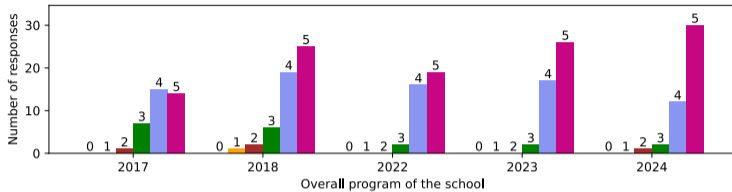
CoDaS-HEP was held in 2017, 2018, 2019, 2022, 2023, and 2024.

I could find survey results from all but 2019.

Survey consists of quantitative rankings and qualitative requests for comments, some general and some per-teacher/session.

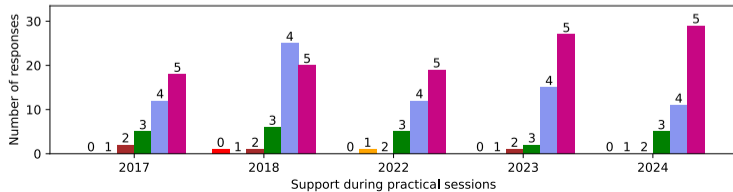
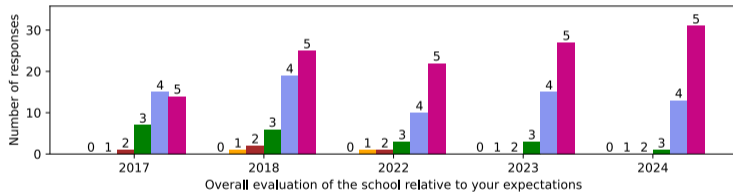
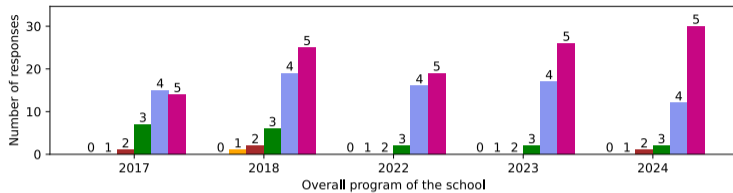
Per-teacher questions are useful for improving the program, but we'll only look at general questions here.

General quantitative questions



Mostly unchanging,
mostly positive.

General quantitative questions



Mostly unchanging,
mostly positive.

They reflect well on us,
but aren't useful for
making decisions or
changes.



Analyzing qualitative data: the free-form comment boxes

The students made a lot of good suggestions, but

- ▶ what were the most common suggestions?
- ▶ how can I show them without explicit quotes? (We didn't say they'd be public.)



The students made a lot of good suggestions, but

- ▶ what were the most common suggestions?
- ▶ how can I show them without explicit quotes? (We didn't say they'd be public.)

I can summarize them for you, but

- ▶ my diligence might not be constant: I might weigh early comments more than later ones (there's a lot of text to read),
- ▶ some of the comments are about me: how can I convince you/myself that I've summarized them fairly?

Trying a new thing: CharGPT for semantic clustering



```
{
  "model": model,          # "gpt-4o"
  "temperature": temperature, # 0.7
  "messages": [
    {"role": "system", "content": textwrap.dedent(f"""
      All of the following statements are students' answers to the question, \"{question}\",
      meaning the CoDaS-HEP Computing in High Energy Physics school, over five years: 2017, 2018, 2022, 2023, and 2024.
      Each statement is numbered by year and a unique identifier. Summarize these statements by grouping them into
      approximately {number} categories. Format the categories as JSON with a title and a several-sentence long paragraph
      description for each. Don't mention any personal names and don't use any exact quotes. Make sure that all
      concerns are addressed in the long descriptions. For each category, list all of the statements that fit that
      category by their year-hyphen-identifier string. Put any uncategorized statements into a list of other_statements
      and summarize them by a few sentences in other_statement_summary.
    """).strip()},
  ] + [{"role": "user", "content": f"{year}-{n + 1}. {stmt}" for year in data for n, stmt in enumerate(data[year])}],
  "response_format": {
    "type": "json_schema",
    "json_schema": {
      "name": "response",
      "schema": {"type": "object", "properties": {
        "categories": {"type": "array", "items": {"type": "object", "properties": {
          "title": {"type": "string"},
          "long_description": {"type": "string"},
          "statements": {"type": "array", "items": {"type": "string"}}, # to verify that assignments are sensible
        }},
        "other_statements": {"type": "array", "items": {"type": "string"}}, # to see which were unassigned
        "other_statement_summary": {"type": "string"},
      }},
    }},
  ],
},
}
```

“What did you like most about the school?”



Diverse and Relevant Topics

Many students appreciated the wide array of topics covered during the school. The broad coverage allowed participants to learn about areas they might not have been exposed to otherwise and gain insights into diverse computational tools and techniques applicable to High Energy Physics (HEP). Students felt that the topics were relevant to both their current research and future endeavors, with subjects like machine learning, parallel programming, and Python often highlighted as particularly beneficial.

Hands-On Exercises and Practical Learning

The hands-on exercises were highly valued by students, as they provided an opportunity to apply what they learned in a practical setting. Many participants found the interactive sessions and exercises helpful for reinforcing theoretical knowledge and developing practical skills. The approach of integrating exercises into the lectures was appreciated, as it allowed students to learn by doing and facilitated better retention of the material.

Parallel Programming and Computational Tools

Parallel programming emerged as a standout topic among the participants, with many students expressing high interest and appreciation for the sessions dedicated to it. The lectures on parallel programming, including OpenMP and other tools, were frequently mentioned as highlights. Students valued learning about these advanced computational techniques, which they found applicable to their research and beneficial for their future work.

Social Interaction and Networking

Participants valued the opportunity to network and interact socially with peers, lecturers, and experts in the field. The school provided a platform for students to connect with others, exchange ideas, and build professional relationships. Social events, coffee breaks, and the overall atmosphere were highlighted as conducive to forming meaningful connections and enhancing the learning experience.

Overall Organization and Supportive Environment

The organization of the school, including logistics, accommodation, and food, was frequently praised by participants. Students appreciated the well-structured program, the attention to detail, and the supportive environment created by the organizers. The school's atmosphere was described as welcoming and conducive to learning, allowing participants to focus on their studies without unnecessary stress.

Other statements

Several statements reflected individual preferences or specific experiences that didn't neatly fit into the main categories. These include mentions of specific lectures or tools that were particularly enjoyed, appreciation for the interactive style of the school, enjoyment of the campus or food, and personal learning outcomes. Some students highlighted the novelty of the topics and the exposure to new concepts, while others focused on the logistics and organization of the event.

“What did you like most about the school?”



Diverse and Relevant Topics

Many students appreciated the wide array of topics covered during the school. The broad coverage allowed participants to learn about areas they might not have been exposed to otherwise and gain insights into diverse computational tools and techniques applicable to High Energy Physics (HEP). Students felt that the topics were relevant to both their current research and future endeavors, with subjects like machine learning, parallel programming, and Python often highlighted as particularly beneficial.

Hands-On Exercises and Practical Learning

The hands-on exercises were highly valued by students, as they provided an opportunity to apply what they learned in a practical setting. Many participants found the interactive sessions and exercises helpful for reinforcing theoretical knowledge and developing practical skills. The approach of integrating exercises into the lectures was appreciated, as it allowed students to learn by doing and facilitated better retention of the material.

Para

Parallel
to it. T
advance

Soci

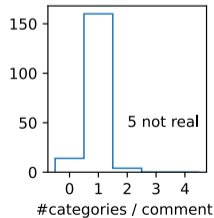
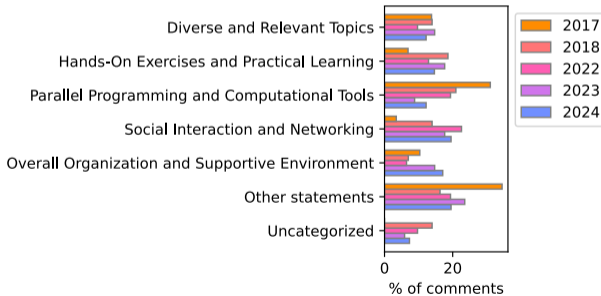
Particip
connect
forming

Over

The org
program
learning

Othe

Several
or tools
students highlighted the novelty of the topics and the exposure to new concepts, while others focused on the logistics and organization of the event.



icated
se

ts to
ive to

d
ive to

ctures
Some

“What did you like least about the school?”



Machine Learning and Lecture Pace

Many students expressed concerns about the pace and structure of the machine learning sessions. The lectures were often described as rushed, chaotic, or difficult to follow, with insufficient time for hands-on exercises or deeper understanding of the material. There was a desire for more structured, interactive sessions and a broader introduction to machine learning concepts. Some students also felt that certain lectures moved too quickly over material, making it hard to grasp the content effectively.

Living Conditions and Accommodations

A significant number of participants were dissatisfied with the dormitory conditions, including the cleanliness and comfort of the rooms and bathrooms. Issues with bedding, humidity, and pests were mentioned, as well as discomfort with shared facilities. Some students felt the living arrangements detracted from their overall experience at the school.

Time Constraints and Schedule

The schedule of the school was a common concern, with many students feeling that the days were too packed, leaving insufficient time for rest, understanding, and application of the material. Early morning starts and long, intensive days contributed to fatigue and limited the ability to fully engage with the content. A longer duration for the school was suggested to allow more time for exercises and in-depth exploration of topics.

Technical and Organizational Issues

Several students experienced technical difficulties during the sessions, which affected their learning experience. Problems with software setup, internet connectivity, and technical support were noted. Some lectures were poorly organized, with inconsistent setups for exercises and technical issues that interrupted the flow of learning. There was a call for better preparation and organization to streamline these processes.

Content Relevance and Diversity

Some students felt that the content of the school was too focused on specific areas, such as the Large Hadron Collider (LHC), and did not cater to a broader range of interests within high energy physics. There were also comments on the lack of diversity among presenters and topics, and a desire for more inclusive and varied subject matter that would appeal to a wider audience.

Other statements

Some students had generally positive experiences and did not express specific complaints about the school. There were comments about the physical environment, such as classroom temperature and weather, as well as minor logistical issues like timing of start times. A few remarks pointed to specific lectures or speakers that did not meet expectations, either due to presentation style or content relevance. Overall, the feedback included a mix of satisfaction and minor grievances not directly related to the main categories identified.

“What did you like least about the school?”



Machine Learning and Lecture Pace

Many students expressed concerns about the pace and structure of the machine learning sessions. The lectures were often described as rushed, chaotic, or difficult to follow, with insufficient time for hands-on exercises or deeper understanding of the material. There was a desire for more structured, interactive sessions and a broader introduction to machine learning concepts. Some students also felt that certain lectures moved too quickly over material, making it hard to grasp the content effectively.

Living Conditions and Accommodations

A significant number of participants were dissatisfied with the dormitory conditions, including the cleanliness and comfort of the rooms and bathrooms. Issues with bedding, humidity, and pests were mentioned, as well as discomfort with shared facilities. Some students felt the living arrangements detracted from their overall experience at the school.

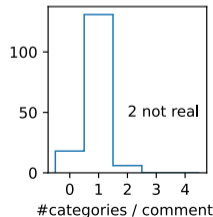
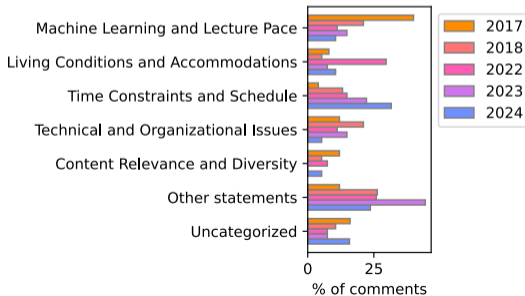
Time Constraints and Schedule

The sch
applicat
duration

Tech
Several
and tec
learning

Cont
Some s
of inter
subject

Othe
Some s
such as
did not
directly related to the main categories identified.



g, and
nger

ctivity,
of

r range
varied

ment,
s that
t

“Any other general comments, feedback, or suggestions?”



General Appreciation and Positive Feedback

Participants consistently expressed gratitude and positive feedback about the school. Many found the experience enlightening, well-organized, and beneficial for their academic and professional growth. The workshops and lectures were generally well-received, with attendees appreciating the opportunity to learn and network with peers and experts in the field.

Suggestions for Program Improvement

Participants suggested that the program could be improved by extending its duration, providing more time for hands-on exercises, and offering parallel sessions to cater to different levels of expertise in topics like machine learning. Some attendees recommended adding competitions, extra challenges, or large projects to foster engagement and practical application of skills learned.

Logistical and Organizational Concerns

Several participants noted logistical issues, such as the need for clearer setup instructions before the event, better coordination with support staff, and improvements to room conditions. Suggestions included pre-event setup instructions, solutions for technical difficulties, and improvements to venue facilities to enhance the learning experience.

Diversity and Inclusivity Concerns

There were concerns raised about diversity and inclusivity, particularly regarding gender and racial representation among participants and presenters. Suggestions included incorporating diversity statements and being mindful of inclusivity during all school activities to ensure a comfortable environment for everyone.

Food and Accommodation Feedback

Feedback on food and accommodation was mixed, with many praising the quality and variety, while others suggested improvements such as offering vegan or halal options and addressing environmental concerns about waste. These aspects significantly influenced the overall satisfaction of participants during the school.

Other statements

Some participants expressed gratitude and shared additional logistical suggestions, such as providing a clearer schedule, enhancing communication, and organizing campus tours. There were also comments on the importance of providing a welcoming and inclusive environment, even in informal settings like meals and social events.

“Any other general comments, feedback, or suggestions?”



General Appreciation and Positive Feedback

Participants consistently expressed gratitude and positive feedback about the school. Many found the experience enlightening, well-organized, and beneficial for their academic and professional growth. The workshops and lectures were generally well-received, with attendees appreciating the opportunity to learn and network with peers and experts in the field.

Suggestions for Program Improvement

Participants suggested that the program could be improved by extending its duration, providing more time for hands-on exercises, and offering parallel sessions to cater to different levels of expertise in topics like machine learning. Some attendees recommended adding competitions, extra challenges, or large projects to foster engagement and practical application of skills learned.

Logistical and Organizational Concerns

Several attendees mentioned the need for more room for learning activities.

Diversity and Inclusivity Concerns

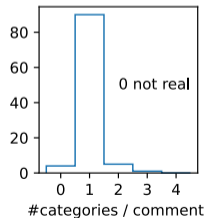
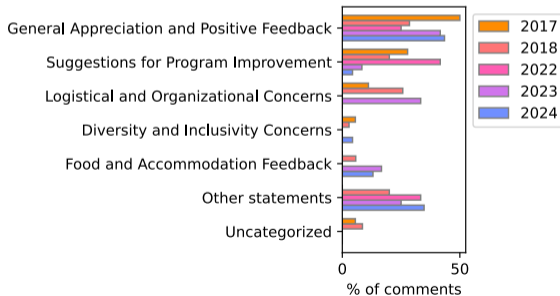
There were some concerns regarding the inclusivity of the program.

Food and Accommodation Feedback

Feedback was provided on the food options and accommodation arrangements.

Other comments

Some participants provided additional comments on campus facilities and event organization.



ements

ions

halal

izing
cial

“Should we extend the school to two weeks?” (2024 only)



Concerns About Length and Format

Some participants express concerns about extending the school to two weeks. They believe that a longer duration might make it difficult for students to attend due to commitments at their home institutions. Suggestions include keeping the school format similar to a course that offers credits, making the program flexible, or maintaining a shorter duration like a week. Moreover, improvements in accommodations and facilities are considered necessary if the school is extended, as well as structured social programming to balance the intensive schedule.

Benefits of Extended Duration

Extending the school to 10 days or two weeks is seen as an opportunity to delve deeper into the topics. Participants suggest more time for exercises, hands-on work, and small group projects. A longer duration could also allow for covering more material or exploring specific topics in greater depth, such as machine learning or neural networks. This could enhance the overall learning experience by providing students with more time to digest and apply the knowledge.

Suggestions for Structure and Content

Participants recommend structuring the school to include shorter days, more free time, and breaks to prevent burnout. Incorporating group projects and social activities can enhance interactions among students. Suggestions include starting later in the day, including local cultural activities, and allowing time for students to explore independently. There is also a proposal to extend the school incrementally and gather feedback each year to refine the experience.

Other statements

Some participants express general support for extending the school without specific suggestions for improvements. Others feel the current duration suffices for the material covered, while one participant expresses enthusiasm but acknowledges potential challenges in attending a longer program.

“Should we extend the school to two weeks?” (2024 only)



Concerns About Length and Format

Some participants express concerns about extending the school to two weeks. They believe that a longer duration might make it difficult for students to attend due to commitments at their home institutions. Suggestions include keeping the school format similar to a course that offers credits, making the program flexible, or maintaining the current structure.

Benefits

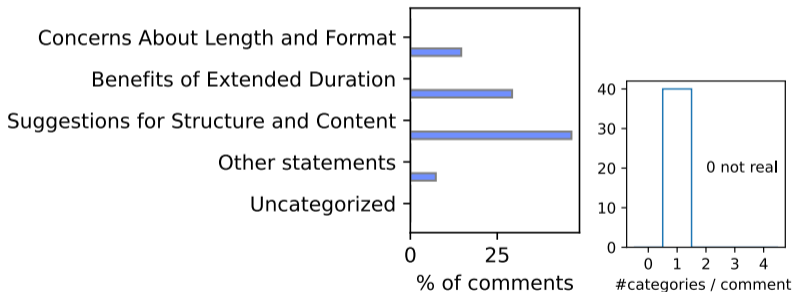
Extending the program and smaller neural networks.

Suggestions

Participants suggest activities to explore.

Other

Some participants express enthusiasm but acknowledge potential challenges in attending a longer program.





- ▶ Hands-on exercises are highlighted as important.



- ▶ Hands-on exercises are highlighted as important.
- ▶ The topics are about right, but the applications are a little too LHC-focused.



- ▶ Hands-on exercises are highlighted as important.
- ▶ The topics are about right, but the applications are a little too LHC-focused.
- ▶ We've had problems (and high turn-over) with the machine learning content, but it's decreasing.



- ▶ Hands-on exercises are highlighted as important.
- ▶ The topics are about right, but the applications are a little too LHC-focused.
- ▶ We've had problems (and high turn-over) with the machine learning content, but it's decreasing.
- ▶ New ideas: pre-event preparatory content and/or parallel sessions.



- ▶ Hands-on exercises are highlighted as important.
- ▶ The topics are about right, but the applications are a little too LHC-focused.
- ▶ We've had problems (and high turn-over) with the machine learning content, but it's decreasing.
- ▶ New ideas: pre-event preparatory content and/or parallel sessions.
- ▶ Students enjoy the in-person aspect of the school, though not the accommodations, and are split on whether it should be extended to two weeks. Those in favor want a longer, lower-intensity event; those opposed think that a longer school would conflict with other obligations.