

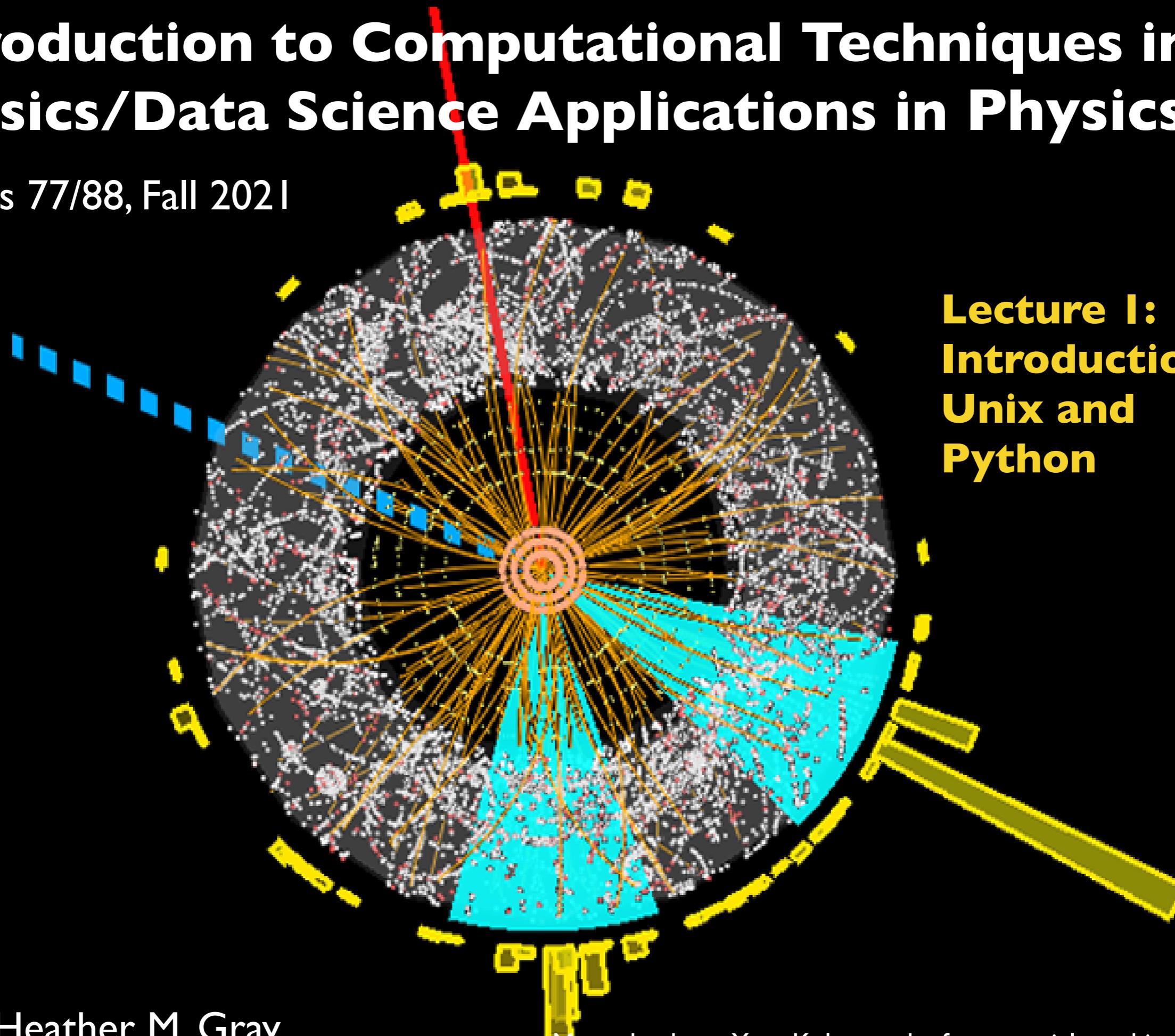
Please ensure that you're wearing your mask and that it is covering your nose and mouth at all times during the lecture



More details here:  
<https://coronavirus.berkeley.edu/return-to-campus/face-coverings/>

# Introduction to Computational Techniques in Physics/Data Science Applications in Physics

Physics 77/88, Fall 2021



**Lecture I:**  
**Introduction to**  
**Unix and**  
**Python**

Prof. Heather M. Gray

Many thanks to Yury Kolomensky for material used in this course

# **The What and Why of Computing**

# Why Computing?

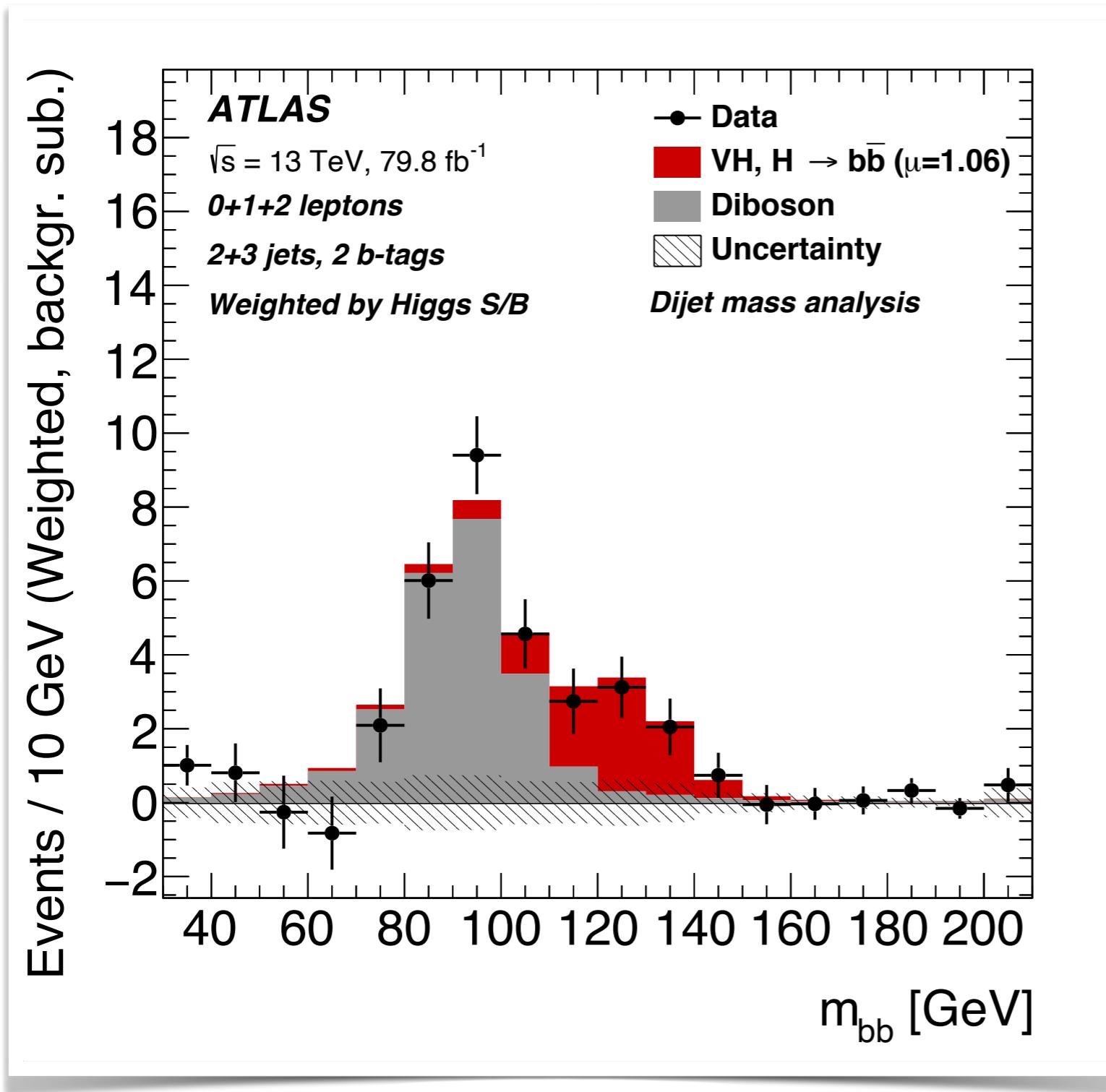
- Necessary tool
  - Physics =
  - Observations →
    - Make a set of
    - Summarize the
  - Most are drawn with some degree of
    - e.g.
    - In reality, we know to some precision
      - e.g.  $G_N =$
  - Many measurements are (e.g. )
    - Have to be interpreted in

- Many measurements require complicated
  - Computerization
- are computerized
- And use computers too

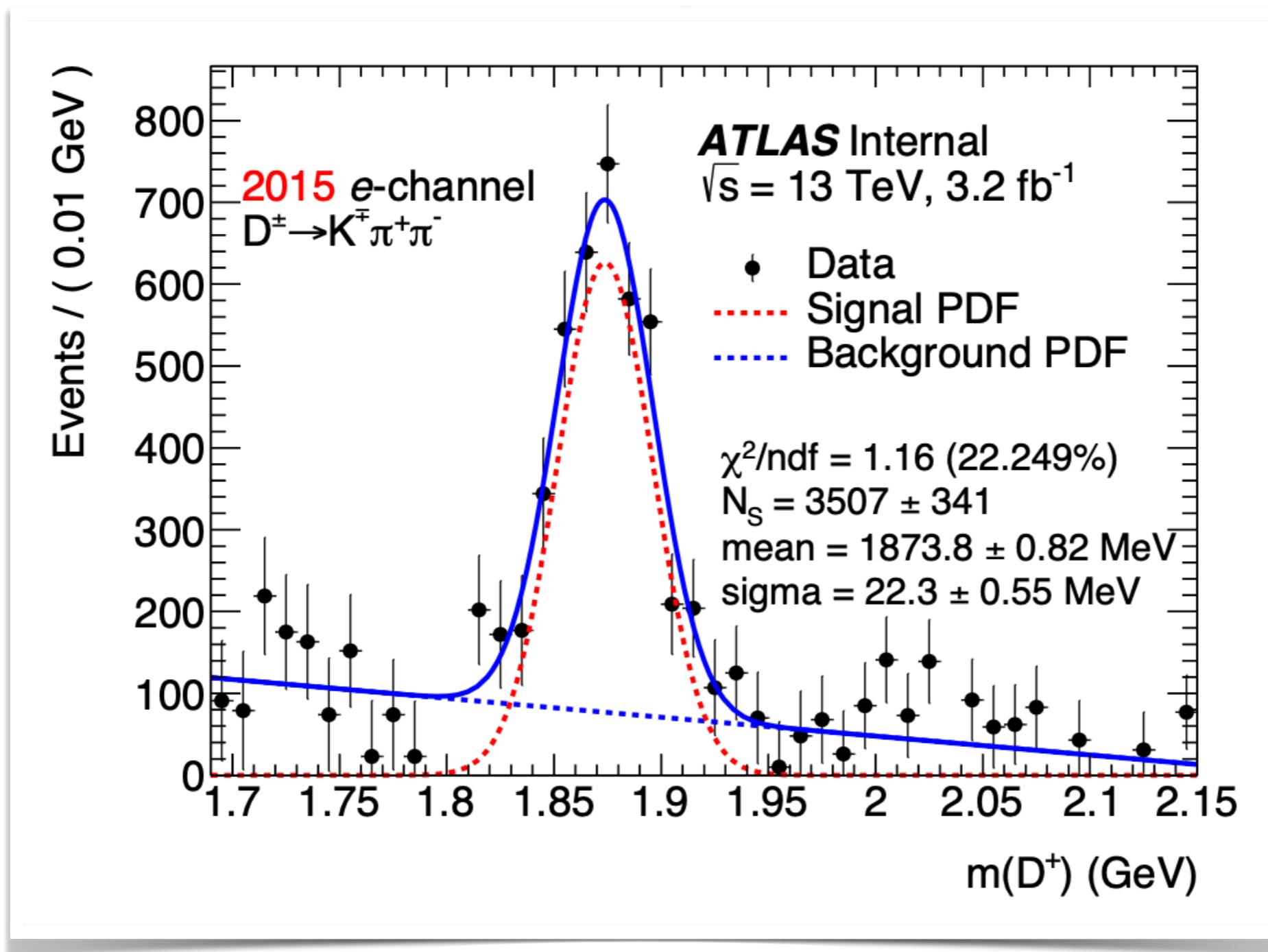
# My First Exposure to Computing



# Examples (My Research)

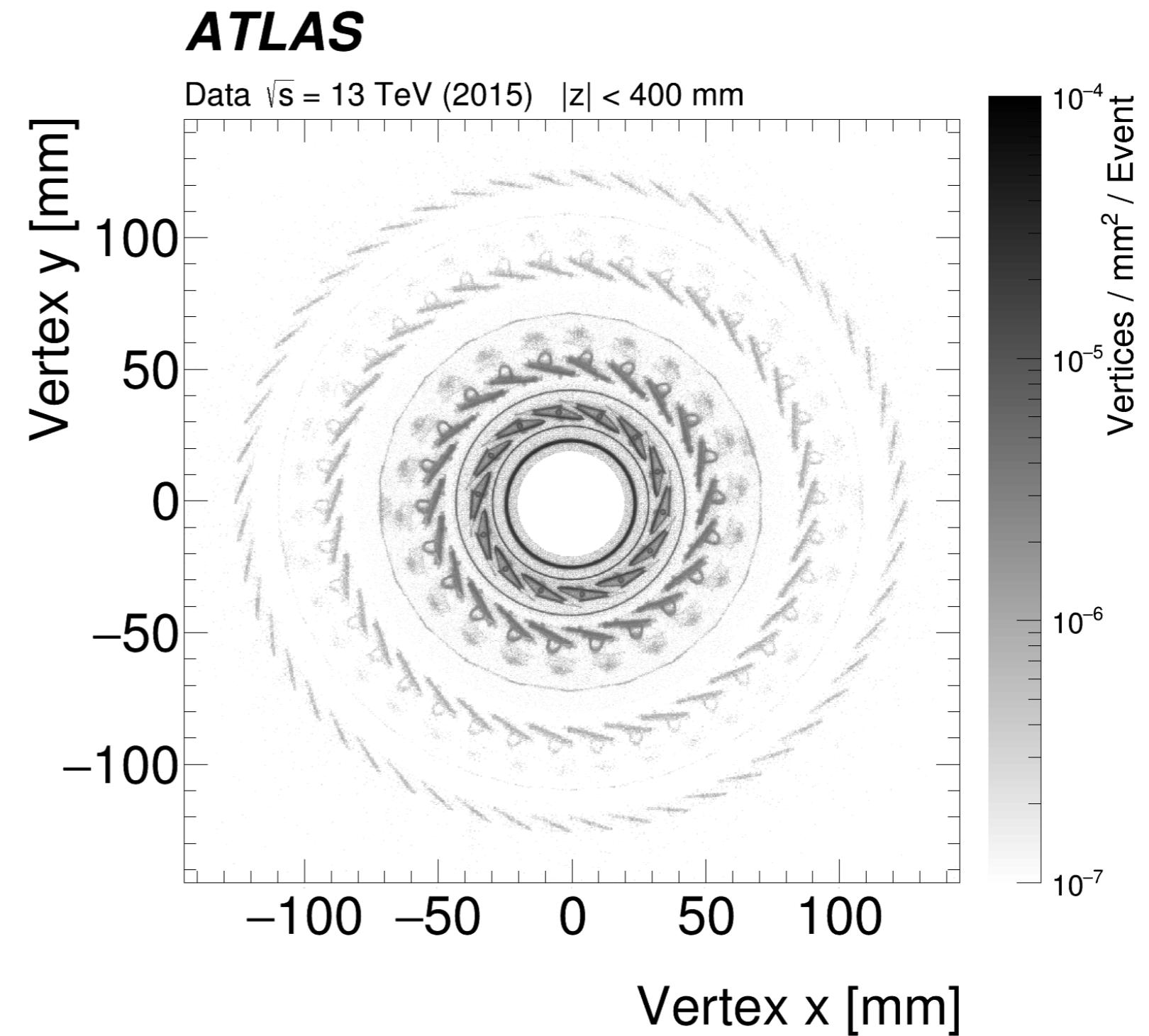


# Examples (My Research)



M. Muskinja (PostDoc)

# Examples (My Research)



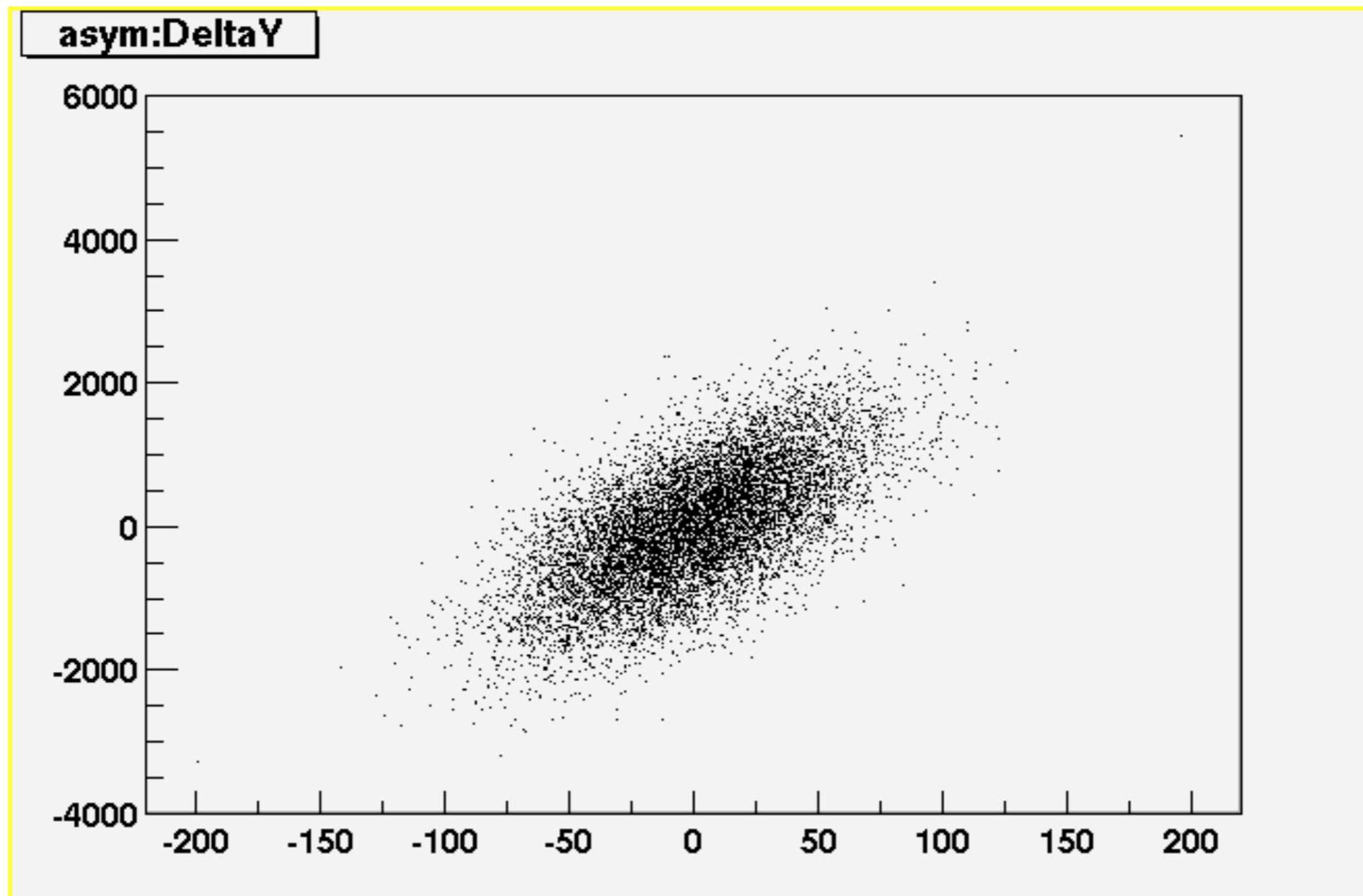
# Describing the Data

- Data: results of
  - In physics we mostly deal with  $\text{kg}$ , i.e.
  - Other fields may deal with
    - An American Robin has gray upper parts and head, and orange under parts, usually brighter in the male
  - Numbers are  $\text{kg}$  to handle
    - We will mostly deal with
- Types of quantitative data
  - $\text{kg}$  data, e.g.
  - $\text{kg}$  data, e.g.
    - Some  $\text{kg}$ , e.g. from
  - $\text{kg}$  data, e.g.
  - Sets of  $\text{kg}$ :

# Visualizing the Data

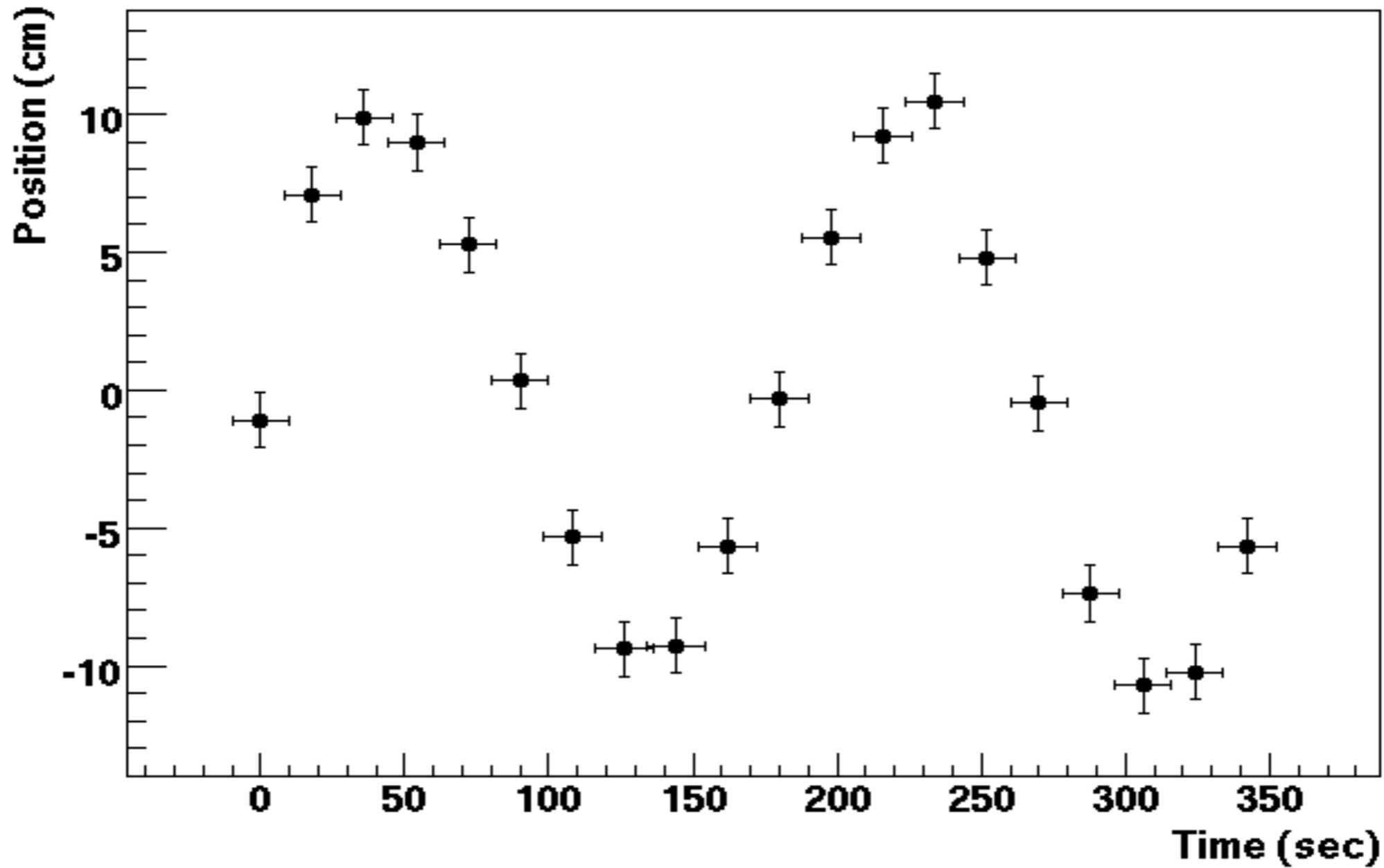
- Tables
  - datasets can be difficult to
- Graphs
  -
- Charts
  -

# Examples

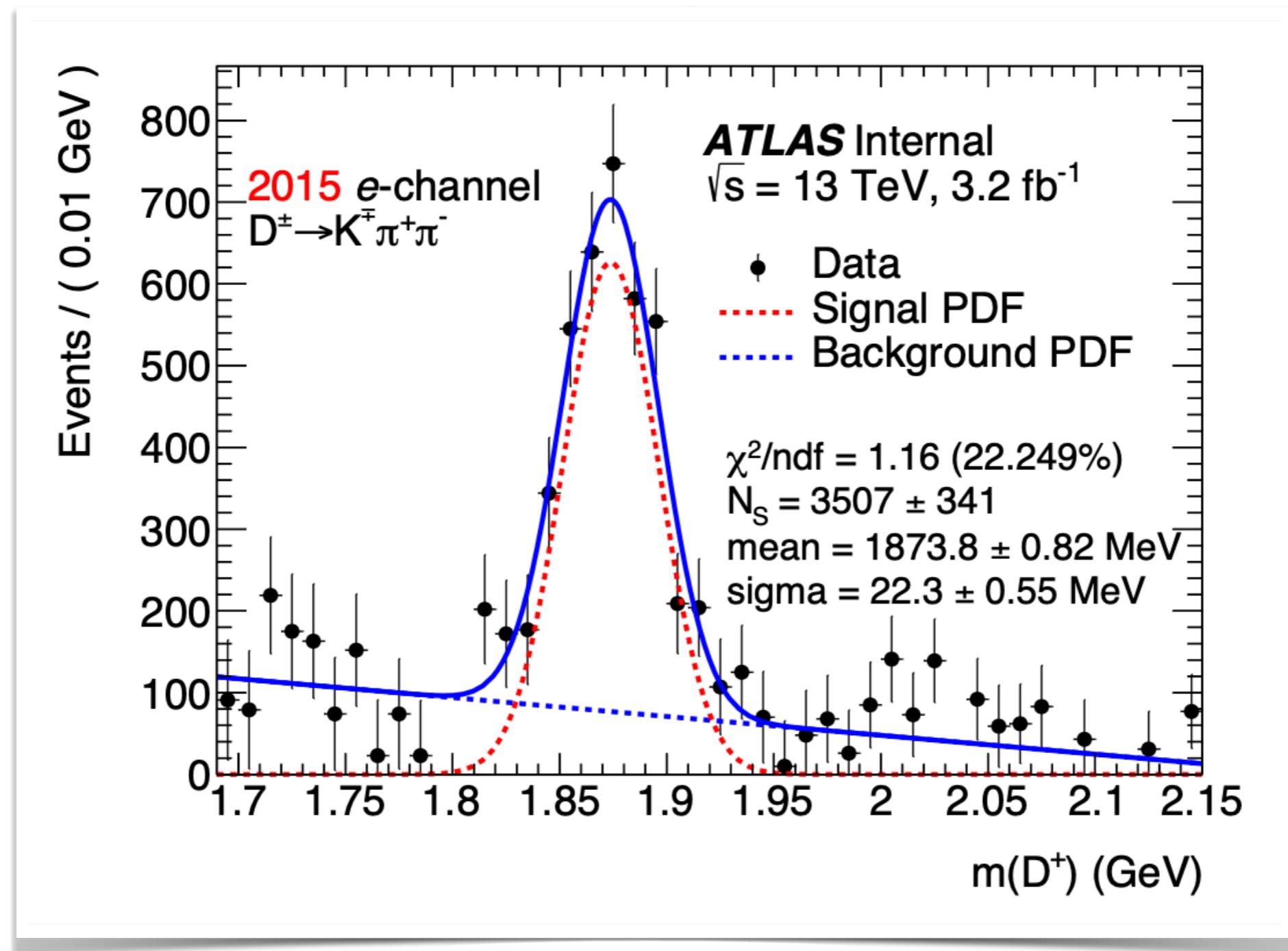


# Examples

**Graph**



# Examples



# **Physics 77/88 Overview and Syllabus**

# Course Organization

- **Lectures:** 2 hours per week (Wheeler 212 at 2 - 4 pm on Mondays)
- **Workshop:** 2 hours per week (Wheeler 212 either 2 - 4 or 4 - 6 pm on Fridays)
  - Extremely important: only learn by doing
  - Interactive/supervised activities in small groups
  - Use [datahub.berkeley.edu](https://datahub.berkeley.edu)
- **Homework:** Most weeks; online submission with bcourses
- Final Project (no midterm, no final exam)
- **Office Hours:**
  - Instructor: On [zoom](#) — please check my [calendar](#) to see the times (sign in with calnet ID)
  - GSI (Elise LePage): TBD, Physics South 420L
  - DS Connector Assistants: TBD, Aneesh Diswania, Sabrina Ma
- Primary means of communication is slack — did everyone get an invite?

# Joining Slack

- Step 1:
  - Join the slack for the physics department: [https://join.slack.com/t/calphysicsworkspace/shared\\_invite/zt-e666a6j5-GgC6gD~oOlgqEWsTw73ztQ](https://join.slack.com/t/calphysicsworkspace/shared_invite/zt-e666a6j5-GgC6gD~oOlgqEWsTw73ztQ)
- Step 2:
  - Join the channel for the course: <slack://channel?id=phys77-fall2021&team=CalPhysics>
- Step 3:
  - Post a message in the class channel
    - Your name, pronouns that you'd like us to use in class, the last thing that you ate



# Physics 77 vs Physics 88

- Room-share two courses
  - Physics 77: **Introduction to Computational Techniques in Physics**
    - Standalone course
    - Assumes no programming knowledge
  - Physics 88: **Data Science Applications in Physics**
    - Connector course to Data 8: The Foundations of Data Science
    - Assumes prior knowledge of Python at the level of Data8 or CS61a
      - Can pass with challenge test (see me for details)
    - Starts in Week 6, room/assignments shared with Physics 77 (including the final project)

# Grading

- Final grades consist of the following components
  - HW: 40%
  - Lecture and workshop attendance and participation: 20%
  - Final Project: 40%
- **Lowest homework grade** will not be included in the final grade
  - This is to allow for those weeks when you have midterms or some other issue that makes it challenging to complete the homework
- **Late Assignment Policy**
  - Late assignments (homework, workshop and project) will be accepted however you will lose 10% of your grade every 24 hours that they are late
  - No late final presentations (unless prior arrangements for exceptional circumstances)

- **Regrading policy**

- If you think that a problem needs regrading, please write a paragraph explaining the issue and why it should be regraded, please send a **private message to me, Elise and Stephanie on slack**. Please note that if your problem is regraded, I will regrade the full homework and your grade may go up or down. Please note that this will be the **only method** by which to request regrading.

- **Attendance policy**

- Attendance is required for the workshop sections and highly recommended for the lecture. Three unexcused absences in Workshop section will drop your grade by a point, i.e. your A becomes a B, your B+ becomes a C+, etc.
- In the event that you aren't able to attend lecture due to sickness, suspected covid-19 exposure or any other reason, please send me a message in slack before the lecture or workshop starts. Notes and recordings for the lectures will be made available on bcourses.

# COVID-19 Policy

- You are required to adhere to all university and UHS policies: <https://coronavirus.berkeley.edu/>
- If you feel sick or have suspected contact with someone with COVID-19, do not come to class/workshops/office hours.
- Recordings of the lectures will be posted on the bcourses page
- A zoom connection will be provided for office hours
- For the workshops, please notify Elise that you can't attend in person. The material will be posted online and please join office hours that week on zoom to ask any questions that you might have.
- The situation is fluid and that we will adapt the best we can with education and public safety in mind.
- I will be clear and communicative about any potential changes to policies -- changes will be discussed in class and will be kept up to date on the syllabus page for the course.

# COVID-19 Policy

- Although it will not be required, it is my preference that everyone wears a mask in the classroom/lab room regardless of the university-wide policy.
- I personally had a difficult experience with covid last year while teaching this class
- Free masks are available here:  
[https://ehs.berkeley.edu/news/  
cloth-face-coverings-  
distribution-schedule](https://ehs.berkeley.edu/news/cloth-face-coverings-distribution-schedule)
- Masks will be available in case you forget or your mask breaks (up to 3 times per semester)



N-95 Respirators



Dust Masks



KN95 Masks



Cloth Masks



Neoprene Masks



Surgical Masks

[https://uhs.berkeley.edu/coronavirus/health-information/protect-  
yourself-and-others-prevention/masks-information](https://uhs.berkeley.edu/coronavirus/health-information/protect-yourself-and-others-prevention/masks-information)

# Lecture Format

- Mostly jupyter notebooks
  - These will be available on bcourses — you can download them and follow along on your laptop during class
- Sometimes (annotated) slides
  - Annotated slides will be posted on bcourses after the lecture
  - (Un)-annotated slides will be posted before the lecture
  - Please use these however you learn best
    - Annotate the slides yourself during the lecture
    - Only listen and use the annotated slides for notes
    - Take your own notes
- Challenge: Technology in the classroom can be challenging for learning but its a key component of this course
  - Strongly suggest that you close email and messaging applications during the class

# Class Participation

- Class participation is key to success in this course
  - That's why its a significant part of the grade
- Each week after class you'll get a brief survey
  - What did you like about the lecture?
  - What did you not understand or would like to have explained again?
  - How did you participate in class this past week?
- Your feedback on questions 1 and 2 will guide the preparation of workshop and the following week's lecture
- Regularly completing the survey (e.g. 4 or more times per semester) will count as credit towards class participation

# Learning Goals

- Guide the presentation of for , and as well as development of , problems for use in discussion section.
- **Representations**
  - move fluidly between representations, descriptions, graphs and tables, and implementations of a physical situation
- **Communication:**
  - Justify and explain step of solving a , in and/or form, at each or writing .
  - Clearly and efficiently and code.
  - Communicate and effectively using information and appropriate presentational technologies
  - e.g.,

- **Tools**

- Identify and apply appropriate techniques to manipulate , implement algorithms for , and solutions to equations, results

- **Problem Solving**

- Identify and apply appropriate techniques to physical problems, and tools and physical systems, and from simulations or experiments

- **Making connections**

- Use and relate new and with those developed in

- **Intellectual maturity and metacognition**

- Take responsibility for learning
  - e.g. identify areas in which you need in the course and take steps to obtain necessary

- **Resourcefulness**

- Use to locate, access, evaluate, and use , and appropriately from digital/electronic media.

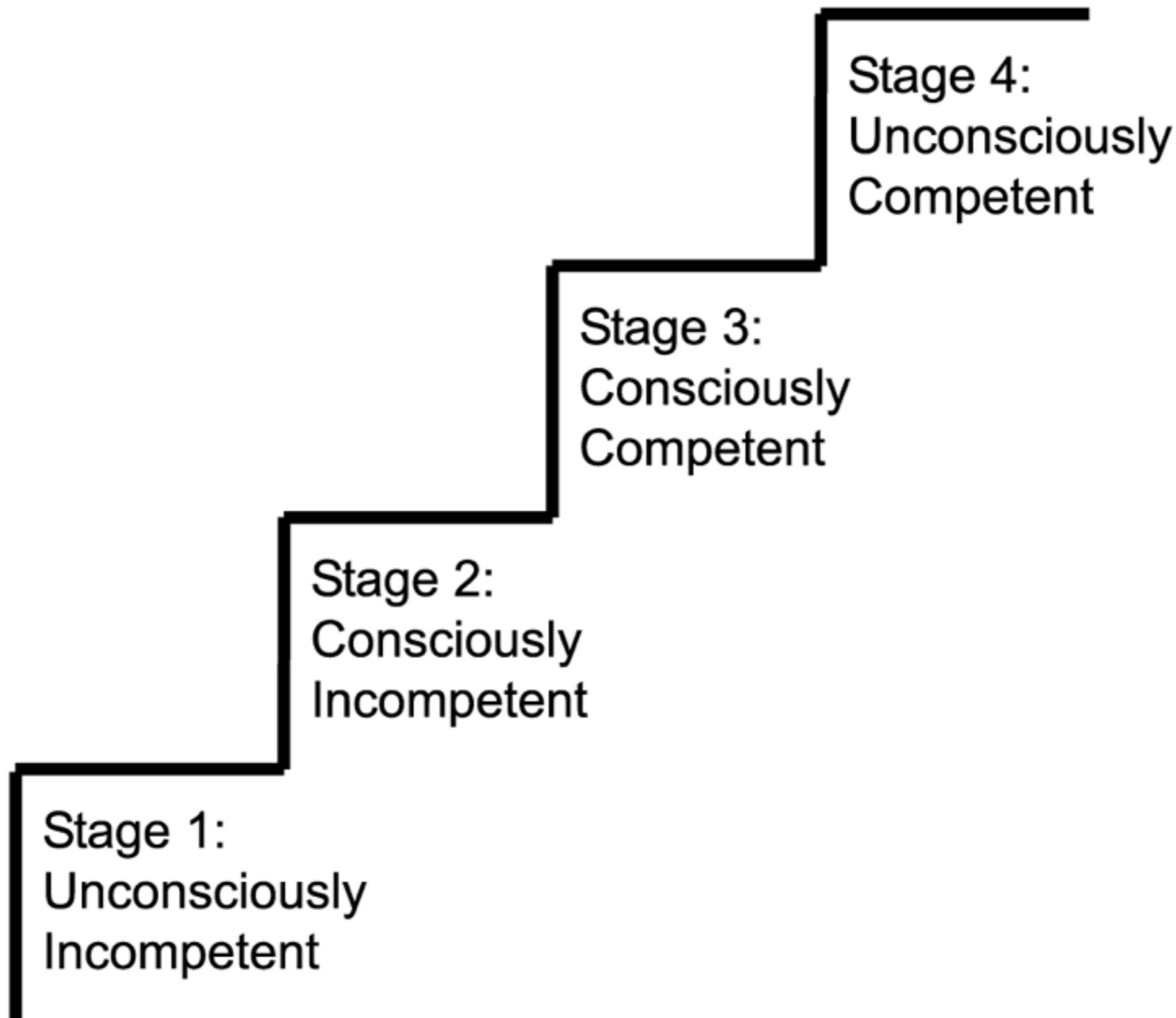
# Tentative Syllabus

Week	Dates	Topics	Reading	Lecture Link	Workshop Link	Homework Link
1	8/30 - 9/3	Introduction to Unix and Python	Newman Ch. 1-2.2	L1	W1: Python basics, and a little plotting	HW1: Python basics and a little plotting
2	9/6 - 9/10	Functions, Loops, Lists, Arrays	Newman Ch. 2.4-2.7	No lecture (Labor Day)	W2: Control Structures 1	
3	9/13 - 9/17	Functions, Loops, Lists, Arrays	Newman Ch. 2.4-2.7	L2	W3: Control Structures 2	HW2: Control Structures
4	9/20 - 9/24	Visualization	Newman Ch. 3	L3	W3: Plotting	
5	9/27 - 10/1	Parsing, Data Processing and File I/O		L4	W4: File I/O	HW3: Arrays, File I/O and Plotting
6	10/4-10/8	Statistics and Probability, Interpreting Measurements	Hughes	L5	W5: PDF Sampling and Statistics.	HW4: File I/O, Statistics
7	10/11 - 10/15	Statistics and Probability, Interpreting Measurement	Hughes	L6	W6: Fitting	HW5: Fitting
8	10/18 - 10/22	Random Numbers, Simulation	Newman Ch. 10	L7	W7: Monte Carlo Techniques	

# Tentative Syllabus (cont.)

9	10/25 - 10/29	Numerical Differentiation	Newman Ch. 5.1-5.9	L8	W8: Numerical Differentiation	
10	11/01 - 11/05	Numerical Integration	Newman Ch. 5.10-5.11	L9	W9: Numerical Integration	HW6: Numerical Integration
11	11/08 - 11/12	Approximation: Root finding, interpolation	Newman Ch. 6	L10	W10: Approximate Root-Finding Methods	Final Project Proposals Due
12	11/15- 11/19	Introduction to Linear Algebra	Newman Ch. 6	L11	W11: Linear Algebra in Python	
13	11/22 - 11/26	Ordinary Differential Equations	Newman Ch. 8	L12	No workshop: Thanksgiving	
14	11/29 - 12/03	Fourier Transforms, Signal Processing	Newman Ch. 7	L13	W12: Introduction to Numerical ODE Solutions	
15	12/06 - 12/10	Final Presentations		Final Presentations	Final Presentations	

# How You Should Learn



# How to Approach This Course

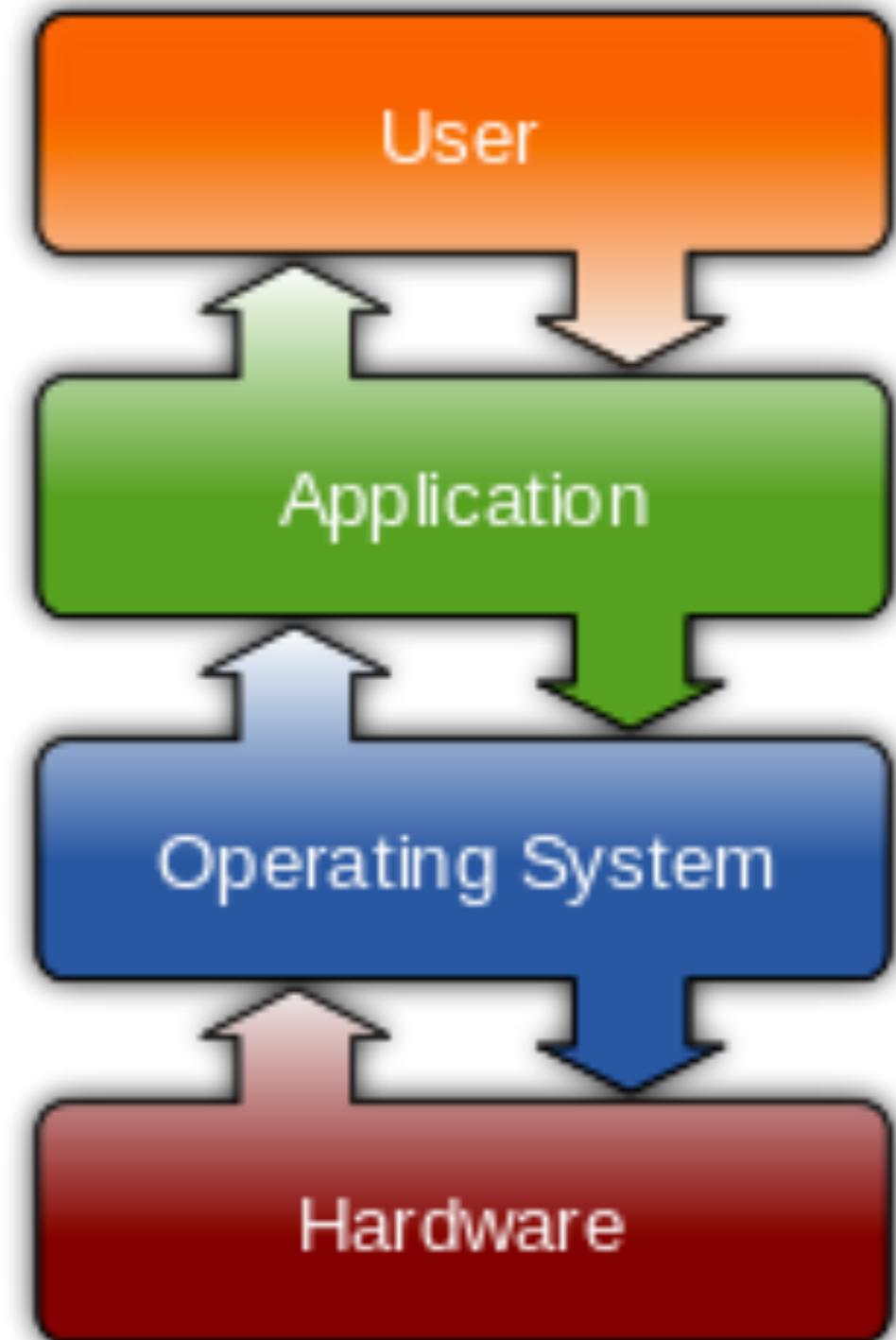
- Active Learning
  - Learn by
- Three main threads to the course
  - 
  - 
  -
- No single book covers all of it
  - Recommend two (Newman, Hughes & Hase)
  - Many online resources for each step
    - Book → Lecture → Workshop → HW → Book cycle applies

**Let's take a look at the  
jupyter notebook**

# **Operating Systems, Programming Environments, Representations**

# Brief Spiel: Operating Systems

- Operating System ( ) is an interface between a and a
  - Translate to electronic signals, report results back
  - Optimized for hardware, efficient
    - Low-level code
  - Interface can be graphical ( ), text-based ( ), or mixed ( )



Source: Wikipedia

# Unix/Linux

- One of the oldest (surviving) OS ( )
- Initially developed for , ported to (Linux and spinoffs)
- Variants are now running on a variety of hardware
  - PC ( )
  - Mac ( )
  - smartphones ( , )
  - even the occasional microwave
- Robust, efficient
- Designed to be , so basic interface is a

# Programming Environments

- interface
- interface
- Connecting to a server (ssh, terminal)
- Scripts
- Compiled vs interpreted languages

```
[root@localhost ~]# ping -q fa.wikipedia.org
PING text.pmtpt.wiki (208.80.152.2) 56(84) bytes of data.
^C
--- text.pmtpt.wiki ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 540.528/540.528/540.528/0.000 ms
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd /var
[root@localhost var]# ls -la
total 72
drwxr-xr-x. 18 root root 4096 Jul 30 22:43 .
drwxr-xr-x. 23 root root 4096 Sep 14 20:42 ..
drwxr-xr-x. 2 root root 4096 May 14 00:15 account
drwxr-xr-x. 11 root root 4096 Jul 31 22:26 cache
drwxr-xr-x. 3 root root 4096 May 18 16:03 db
drwxr-xr-x. 3 root root 4096 May 18 16:03 empty
drwxr-xr-x. 2 root root 4096 May 18 16:03 games
drwxrwx-T. 2 root gdm 4096 Jun 2 18:39 gdm
drwxr-xr-x. 38 root root 4096 May 18 16:03 lib
drwxr-xr-x. 2 root root 4096 May 18 16:03 local
lrwxrwxrwx. 1 root root 11 May 14 00:12 lock -> ../run/lock
drwxr-xr-x. 14 root root 4096 Sep 14 20:42 log
lrwxrwxrwx. 1 root root 10 Jul 30 22:43 mail -> spool/mail
drwxr-xr-x. 2 root root 4096 May 18 16:03 nis
drwxr-xr-x. 2 root root 4096 May 18 16:03 opt
drwxr-xr-x. 2 root root 4096 May 18 16:03 preserve
drwxr-xr-x. 2 root root 4096 Jul 1 22:11 report
lrwxrwxrwx. 1 root root 6 May 14 00:12 run -> ../run
drwxr-xr-x. 14 root root 4096 May 18 16:03 spool
drwxrwxrwt. 4 root root 4096 Sep 12 23:50 tmp
drwxr-xr-x. 2 root root 4096 May 18 16:03 yp
[root@localhost var]# yum search wiki
Loaded plugins: langpacks, presto, refresh-packagekit, remove-with-leaves
rpmfusion-free-updates | 2.7 kB 00:00
rpmfusion-free-updates/primary_db | 206 kB 00:04
rpmfusion-nonfree-updates | 2.7 kB 00:00
updates/metalink | 5.9 kB 00:00
updates | 4.7 kB 00:00
updates/primary_db | 2.6 MB 00:15 ETA
```

Source: Wikipedia

# Brief Spiel: Programming Languages

- Programming languages allow to translate sets of (an algorithm) to a form understandable by a computer
- Classifications
  - 
  - 
  - 
  -
- Implementations
  - 
  - 
  - 
  -

# Data Representation

- Data on computers represented in format
  - Base representation (as opposed to base )
  - $abcd_2 =$
  - Examples:

Decimal numbers	Binary equivalent	Decimal numbers	Binary equivalent
0	0000	8	1000
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111

- Smallest memory cell:
  - bits = byte (B)
  - Measures of memory
    - 1 kB = ; 1 MB = , etc

# Binary Representation

- Practical consequences
  - All numbers in the digital format are  $\dots$ , i.e. they have precision
    - This is easy to understand with  $\dots$  (by construction)
    - Real numbers: think about representation in
      - $0.125 = \dots$
      - Could you represent  $1/10$  in powers of 2 ? What about  $\pi$  ?
    - Basic data types have  $\dots$  and  $\dots$  value, as well as precision, determined by the data type size
      - i.e. how much  $\dots$  is allocated for each data type
      - Most common:  $\dots$  or  $\dots$  bytes for integer and real values

# Examples: Integer Data

- C/C++

```
root [0] `sizeof(int) // number of bytes for interger  
(const int)4  
root [1] sizeof(long) // number of bytes for long integer  
(const int)8  
root [2] []
```

- This means:

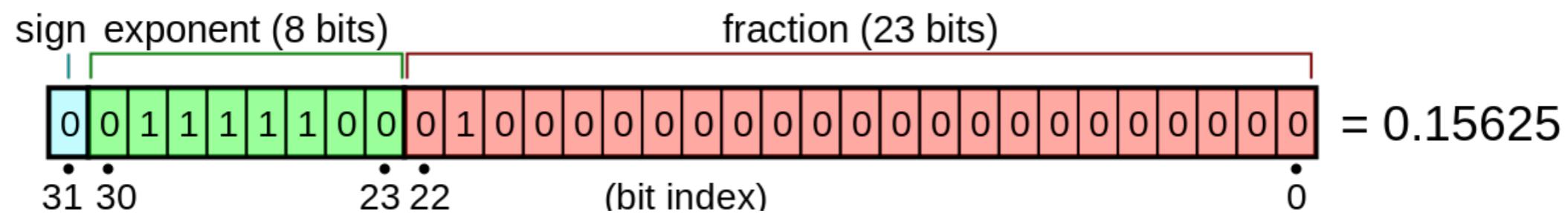
- Range of signed ints in C is  $\pm 2^{14}7483647$
- Range of unsigned ints in C is  $0..4294967295$
- E.g. time in Unix is represented as — in seconds from Jan 1, . Hence the impending end of time in
  - (Google “end of unix time”)



# Examples: Real numbers

- Real numbers are represented by 3 fields

- 
- 
- 



$$\text{value} = (-1)^{\text{sign}} \times \left(1 + \sum_{i=1}^{23} b_{23-i} 2^{-i}\right) \times 2^{(e-127)}$$

- For example, most languages use **float** and **double** floating point numbers
- and **float** in C/C++

# Precision and Range

... both single and double precision IEEE floating point numbers.

<b>Property</b>	<b>Value for float</b>	<b>Value for double</b>
Largest representable number	3.402823466e+38	1.7976931348623157e+308
Smallest number without losing precision	1.175494351e-38	2.2250738585072014e-308
Smallest representable number(*)	1.401298464e-45	5e-324
Mantissa bits	23	52
Exponent bits	8	11
Epsilon(**)	1.1929093e-7	2.220446049250313e-16

[http://www.cprogramming.com/tutorial/floating\\_point/understanding\\_floating\\_point\\_representation.html](http://www.cprogramming.com/tutorial/floating_point/understanding_floating_point_representation.html)

*What about Python ? Let's go back to the Notebook*