1) Course organization

2) A bit about Error Analysis

3) Scientific programming

- Show web page

- Goals

- Formal error analysis: Tails through curve, getting binomial + Poisson

- Scientific programming

- Data wrangling

- 5 hrs + 5 labs (every 2 weeks)

- Labs involve using scientific programming for simulation + to ascertain data, reduce it to a usable + preferred form; make tables + graphs + interpret it + fit it with mathematical relationships.
1) continued
   - we'll be looking at data from biophysics, astrophysics, geophysics
     + climate science realms
   - we will explore the practical sides of data sampling + Fourier
     acquisition

2) Intro to Error

In science the term "error" is an estimate of the inevitable uncertainty in
measurement, not a mistake.

No physical quantity can be measured with complete certainty, e.g. who 'era'.

But why should we care about "error" which is, after all, inevitable?

a) are 2 measurements consistent?

i) one person measures length of pendulum as \( l = 348.9 \) cm

ii) another person measures period of same pendulum to be \( T = 1.64 \) s
2) Error - continued

are measurements i) d ii) calculated

to check, turn iii) T moment to equivalent lii

\[ \mathcal{I} \]

cause \& forces

Suggest we use forces:

\[ \tau = \mathcal{I} \phi \]

\[ -mg\sin\phi \cdot l = \mathcal{I} \phi^2 = ml^2 \phi^2 \]

\[ \Rightarrow m \text{ cancels, why?} \]
2) Error - continued

\[ g \sin \gamma \times g \gamma = L_1 \gamma \]

operation bit if we need a function
whose double derivative is \( \pm \) to itself

Suggest \( \gamma = \frac{g}{2} \cos (\omega_0 t + \delta) \)

solve \( w \) with \( \omega_0 = \frac{2\pi}{T} = \sqrt{\frac{K}{m}} \)

[Could plug in solve and check]

(use matlab for this)

\[ l_1 = 30.9 \text{ cm} = 0.309 \text{ m} \]

compared to \( \frac{\omega_0^2}{l_1} = \frac{g}{L_1} \)

\[ l = \frac{g}{\omega_0^2} \]

plug in \( T \) \( \Rightarrow \) 32.26

or \( l = 0.323 \text{ m} \)

\[ 3226 \]

\[ l_1 \]

doesn't work prime face consistent, but we need error to factor?
2) Error continued

i) Error estimation, measured at
near 86.56 30.9 cm mean
measuring to ±0.1 mm hard to
do. Likely could measure to ±0.5 cm

\[ \text{measured value at} \ (30.9 ± 0.5) \text{ cm} \]

"slackers"

ii) Measure 10 complete oscillations
each time, does error pop →

\[ (30.9 ± 2) \text{ cm} \]

⇒ OK measurements continued

b) Why study error part 2

1904-1905, Einstein publishes general
theory of relativity.

predicts light passing our sun
beats by \( d = 1.8'' \) (seconds of arc),
double the \( d = 0.9'' \) predicted by "classical" arguments

1980: Dyon Eddington, Davidson do "small
experiment" & got \( d = (0.10 ± 0.3)'' \)
Eddington → Prince (Whi, 6 of 4's)
Dyson → Sobral, Brazil

why
2 measurements
moon back to
blackboard
next slide
3) Intro to scientific programming

Suppose we want to "simulate"
ball dropped from 14.1 m height
above the floor

First, "simulate" means to write a
computer program that gives details about
the ball's position during the drop, etc.

Second, let's make some approximations. We will
initially ignore air resistance (drag), but
want to add this in later.
step 2) draw experiment to help visualize. For simulation think of recently high-speed near or pull drop. What characteristics of the experiment should we consider?

step 3)

- Mass of ball?
- Shape of ball? (radius?)
- Is it dropping through air or something else?
- Wind?
- g of planet?
- Are we near planet’s surface?
- Color of ball
- Speed, position, etc. of ball
call well characteristics from lot

symbolic

step 2) assign variable to characteristics

m (mass), g (accel of gravity), etc.

step 5) need to consider physics and set governing equations

\[ F_{net} = ma \implies F_{net} / m = a = \frac{V_f - V_i}{\Delta t} \]

\( (\Delta t = \text{time step}) \)
3) sci. proj. continued

\[ v_x = u_x + \frac{F_{\text{tot}}}{m} \, dt \]

-this looks useful.

Also, \[ v_{\text{avg}} = \frac{y_f - y_i}{dt} \implies y_f = y_i + v_{\text{avg}} \, dt \]
gives us new ball position from initial (mit) position + avg velocity after time step \( dt \).

Also, \( t_f = t_i + dt \) (helps us build a time array)

To step 6, code:

- Build "function" to simulate ball drop
  - [call part]
  - [set constants and initial variables]
  - for loop to step through ball drop
    - step = \( dt \)

(Do this in ML + Python?)

* in sci. programming sense
3) Sci. prog. continued

(At time)

Note that we:

a) described the scene (both pictorially + with characteristics) before

b) many to physics to get after

c) built with using standard fan

At meta level we started simple + neglected drag. But we formulate so we can easily add it in later.