

# Lecture 2

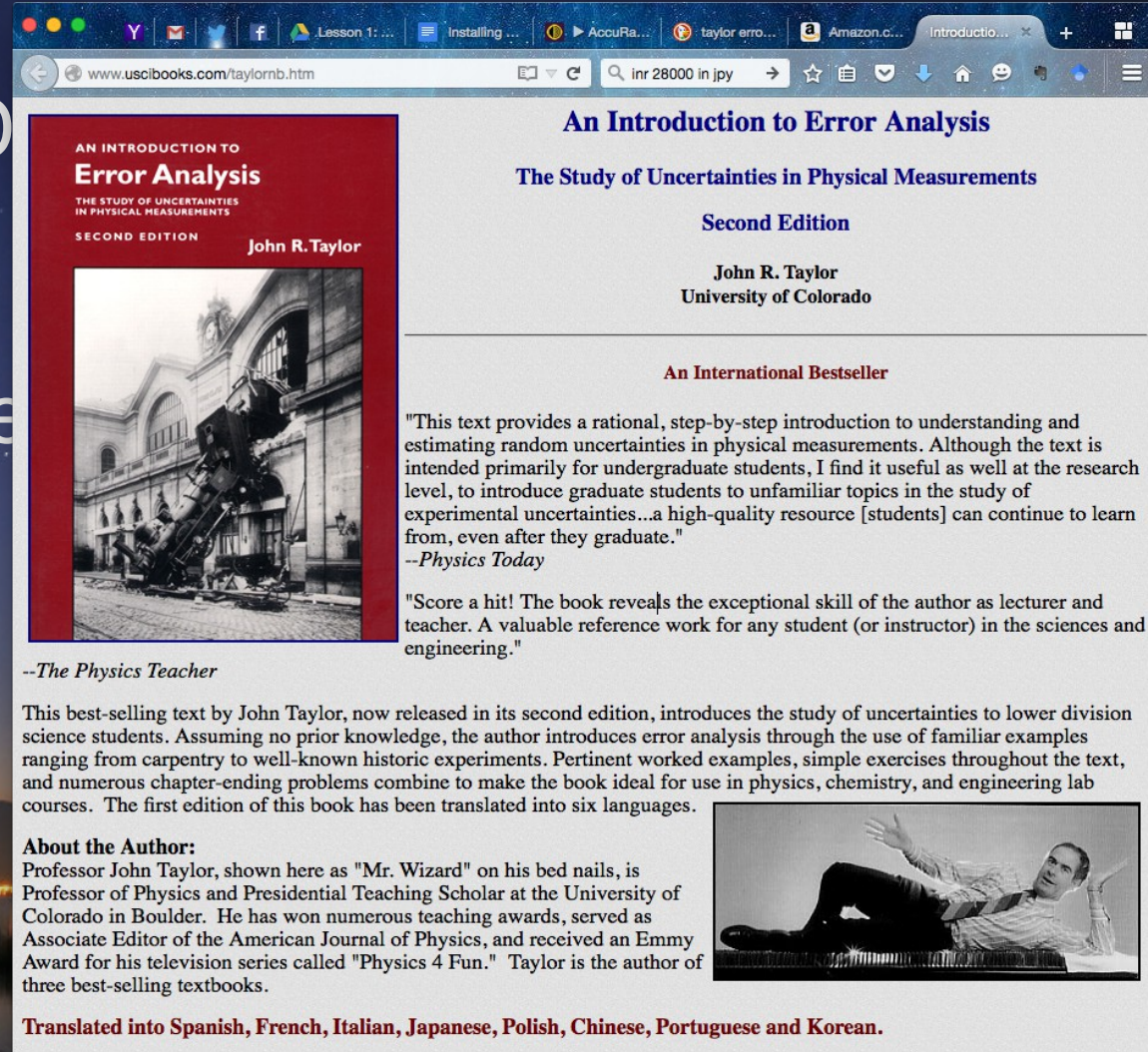
## Basic Statistics

Jayant Murthy  
The Indian Institute of Astrophysics  
[www.iiap.res.in](http://www.iiap.res.in)  
[jmurthy@yahoo.com](mailto:jmurthy@yahoo.com)



# Recommended Reading

- Philip Bevington & D Keith Robinson: Data Reduction and Error Analysis for the Physical Sciences
- John R. Taylor: An Introduction to Error Analysis



The screenshot shows a web browser displaying the website for the book "An Introduction to Error Analysis" by John R. Taylor. The browser's address bar shows the URL "www.uscibooks.com/taylor/b.htm". The website has a red header with the book's title and author. Below the header, there is a section titled "An Introduction to Error Analysis" with the subtitle "The Study of Uncertainties in Physical Measurements" and "Second Edition". The author's name, "John R. Taylor", and affiliation, "University of Colorado", are listed. A quote from "Physics Today" describes the book as a rational, step-by-step introduction to understanding and estimating random uncertainties. Another quote from "The Physics Teacher" praises the book as a hit. A paragraph describes the book as a best-selling text by John Taylor, now in its second edition, introducing the study of uncertainties to lower division science students. A section titled "About the Author:" provides biographical information about Professor John Taylor, including his position at the University of Colorado, his teaching awards, and his television series "Physics 4 Fun." A small black and white photo of Professor Taylor is included. At the bottom, it states that the book has been translated into Spanish, French, Italian, Japanese, Polish, Chinese, Portuguese, and Korean.

**An Introduction to Error Analysis**  
The Study of Uncertainties in Physical Measurements  
Second Edition  
John R. Taylor  
University of Colorado

**An International Bestseller**

"This text provides a rational, step-by-step introduction to understanding and estimating random uncertainties in physical measurements. Although the text is intended primarily for undergraduate students, I find it useful as well at the research level, to introduce graduate students to unfamiliar topics in the study of experimental uncertainties...a high-quality resource [students] can continue to learn from, even after they graduate."  
--*Physics Today*

"Score a hit! The book reveals the exceptional skill of the author as lecturer and teacher. A valuable reference work for any student (or instructor) in the sciences and engineering."  
--*The Physics Teacher*

This best-selling text by John Taylor, now released in its second edition, introduces the study of uncertainties to lower division science students. Assuming no prior knowledge, the author introduces error analysis through the use of familiar examples ranging from carpentry to well-known historic experiments. Pertinent worked examples, simple exercises throughout the text, and numerous chapter-ending problems combine to make the book ideal for use in physics, chemistry, and engineering lab courses. The first edition of this book has been translated into six languages.

**About the Author:**  
Professor John Taylor, shown here as "Mr. Wizard" on his bed nails, is Professor of Physics and Presidential Teaching Scholar at the University of Colorado in Boulder. He has won numerous teaching awards, served as Associate Editor of the American Journal of Physics, and received an Emmy Award for his television series called "Physics 4 Fun." Taylor is the author of three best-selling textbooks.

Translated into Spanish, French, Italian, Japanese, Polish, Chinese, Portuguese and Korean.



# Different Types of Error

- Illegitimate errors:
  - Irreproducible errors.
- Systematic errors:
  - Often due to experimental bias or measurement errors.
- Random errors:
  - Measurement or statistical uncertainty.



# Random Numbers

Note that seed is different in IDL but the concept is the same.

- Write a program to flip coins:
- Random numbers
  - If seed is undefined, get pseudo random numbers.
  - If seed is defined, get the same number.

```
1. gdl
GDL> seed=0
GDL> rnd = randomu(seed) & print,rnd,seed
0.999742      65535
GDL> rnd = randomu(seed) & print,rnd,seed
0.193342      131070
GDL> rnd = randomu(seed) & print,rnd,seed
0.394217      196605
GDL> rnd = randomu(seed) & print,rnd,seed
0.565132      262140
GDL> rnd = randomu(seed) & print,rnd,seed
0.515466      327675
GDL> rnd = randomu(seed) & print,rnd,seed
0.370968      393210
GDL> rnd = randomu(seed) & print,rnd,seed
0.332175      458745
GDL> rnd = randomu(seed) & print,rnd,seed
0.0618024     524280
GDL> rnd = randomu(seed) & print,rnd,seed
0.172894      589815
GDL> rnd = randomu(seed) & print,rnd,seed
0.996569      655350
GDL> rnd = randomu(seed) & print,rnd,seed
0.0416150     720885
GDL>
```

# Use Vectors

```
1. gdl
GDL> tails=0
GDL> if (randomu(seed) ge 0.5) then tails = tails + 1 else heads = heads + 1
GDL> print,heads,tails
      1      0
GDL> if (randomu(seed) ge 0.5) then tails = tails + 1 else heads = heads + 1
GDL> print,heads,tails
      1      1
GDL>
$gdl

GDL - GNU Data Language, Version 0.9.5

- For basic information type HELP,/INFO
- No startup file read (GDL_STARTUP/IDL_STARTUP env. var. not set).
- Please report bugs, feature or help requests and patches at:
  http://sourceforge.net/projects/gnudatalanguage/

GDL> seed=0
GDL> rnd=randomu(seed,10)
GDL> print,rnd
      0.999742      0.162910      0.282618      0.947201      0.231657      0.484974
      0.957477      0.744305      0.540044      0.739953
GDL> print,seed
      655350
GDL> █
```



# Heads or Tails

```
1. vim
;Procedure to flip coins

;Define variables
heads = 0
tails = 0

seed = 0; Define the seed so that the results are always the same
print,"Not using random numbers"

;Now begin the loop
nmax = 10000 ;Number of times to flip the coin
for i = 0,10000 do begin
    rnd = randomu(seed) ;Pick from uniform random variable between 0 and 1
;Pick heads if the random number is less than 0.5 otherwise pick tails
    if (rnd le 0.5)then heads = heads + 1 else $
        tails = tails + 1
endfor ;End loop i

;Print out number of heads and tails
print,"Obtained ",heads," Heads and ",tails," Tails out of ",nmax," Tries"
end
~
~
~
"coin_flip_no_vector.pro" [New] 21L, 610C written

2. gdl
Last login: Thu Jul 9 13:54:04 on ttys001
$cd /Users/jayanth/user/education/course/data_analysis/programs
$$
-bash: $: command not found
$
$gdl

GDL - GNU Data Language, Version 0.9.5

- For basic information type HELP,/INFO
- No startup file read (GDL_STARTUP/IDL_STARTUP env. var. not set).
- Please report bugs, feature or help requests and patches at:
  http://sourceforge.net/projects/gnudatalanguage/

GDL> .run coin_flip_no_vector.pro
% Compiled module: $MAIN$.
Not using random numbers
Obtained      5009 Heads and      4992 Tails out of      10000 Tries
GDL> .run coin_flip_no_vector.pro
% Compiled module: $MAIN$.
Not using random numbers
Obtained      5009 Heads and      4992 Tails out of      10000 Tries
GDL> █
```

What is wrong in the above program?

# Vector Shortcut

```
1. vim
nmax=10000
seed=0
rnd=randomu(seed,nmax)
heads = n_elements(where(rnd le 0.5))
tails = n_elements(where(rnd gt 0.5))
print,"Obtained ",heads," Heads and ",tails," Tails out of ",nmax," Tries."
end
```

```
2. gdl
$
$gdl

GDL - GNU Data Language, Version 0.9.5

- For basic information type HELP,/INFO
- No startup file read (GDL_STARTUP/IDL_STARTUP env. var. not set).
- Please report bugs, feature or help requests and patches at:
http://sourceforge.net/projects/gnudatalanguage/

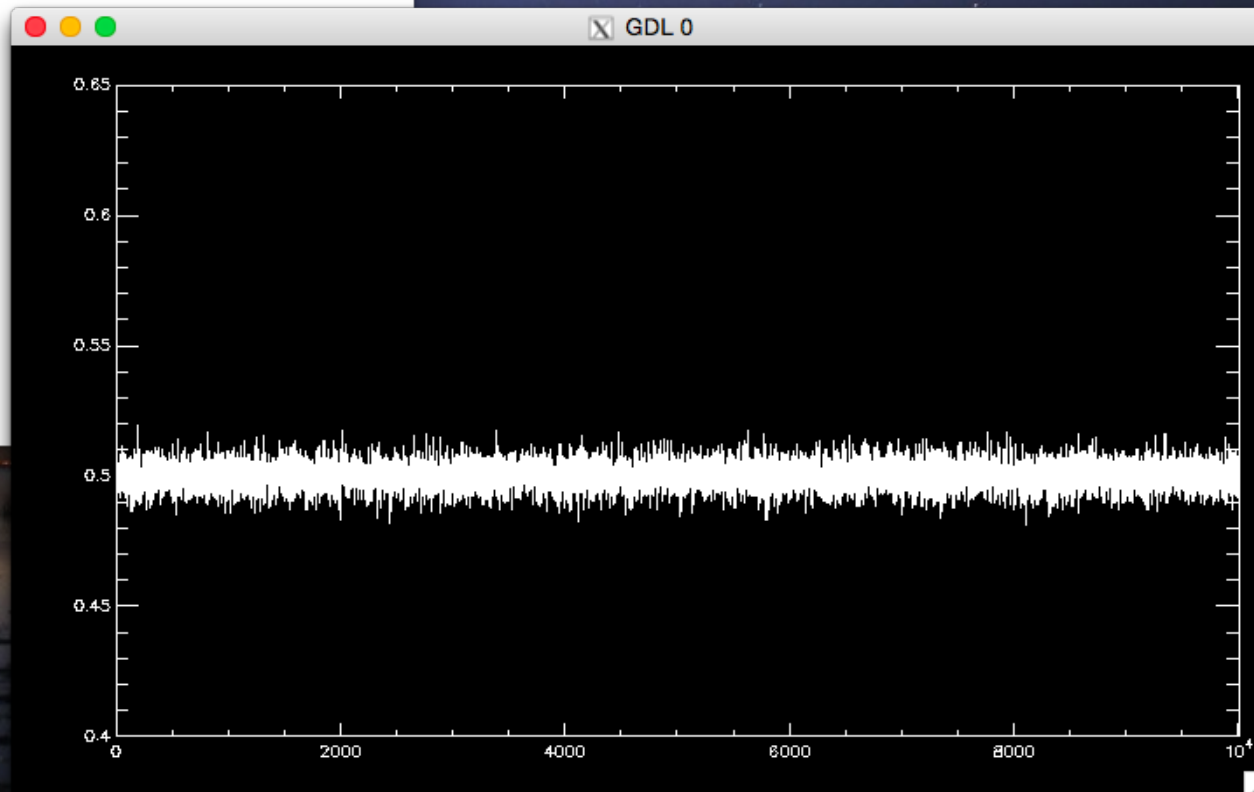
GDL> .run coin_flip_no_vector.pro
% Compiled module: $MAIN$.
Not using random numbers
Obtained      5009 Heads and      4992 Tails out of      10000 Tries
GDL> .run coin_flip_no_vector.pro
% Compiled module: $MAIN$.
Not using random numbers
Obtained      5009 Heads and      4992 Tails out of      10000 Tries
GDL> .run coin_flip_vector.pro
% Compiled module: $MAIN$.
Obtained      4984 Heads and      5016 Tails out of      10000 Tries.
GDL> .run coin_flip_vector.pro
% Compiled module: $MAIN$.
Obtained      4984 Heads and      5016 Tails out of      10000 Tries.
GDL> 
```

# Monte Carlo Tests

```
1. vim
nmax=10000
nloop = 10000
nheads = lonarr(nloop)
ntails = lonarr(nloop)

for i=0l,nloop - 1 do begin
  rnd=randomu(seed,nmax)
  heads = n_elements(where(rnd le 0.5))
  tails = n_elements(where(rnd gt 0.5))
  nheads(i) = heads
  ntails(i) = tails
endfor

;!psym=10 plots in a box (histogram) format
plot,nheads/float(nmax),psym=10,yrange=[.4,.6]
end
~
~
~
~
~
~
~
~
"coin_flip_mc.pro" 16L, 353C written
```





# More Monte Carlo

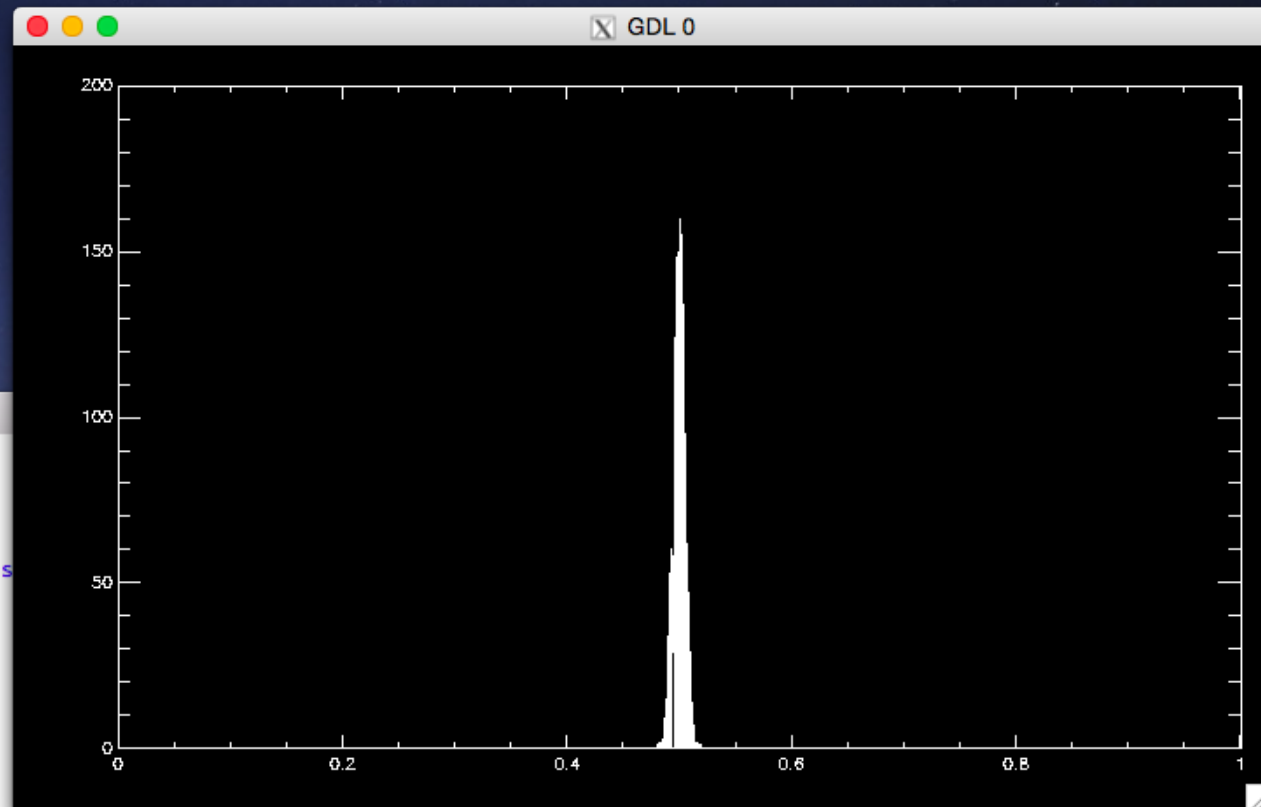
```
2. gdl

GDL - GNU Data Language, Version 0.9.5

- For basic information type HELP,/INFO
- No startup file read (GDL_STARTUP/IDL_STARTUP env. var. not s
- Please report bugs, feature or help requests and patches at:
  http://sourceforge.net/projects/gnudatalanguage/

GDL> .run coin_flip_mc.pro
% Compiled module: $MAIN$.

*** PLPLOT WARNING ***
You said you want pthreads, but they are not available.
GDL> rat=nheads/float(nmax)
GDL> print,mean(rat),median(rat),stdev(rat)
% Compiled module: MEAN.
% Compiled module: STDEV.
   0.499925    0.499900    0.00504734
GDL> h=histogram(rat,min=0,max=1,bin=.0001)
GDL> plot,indgen(10001)*.0001,h,psym=10
GDL> plot,indgen(10001)*.0001,h,psym=10,xrange=[.45,.55]
GDL> plot,indgen(10001)*.0001,h,psym=10,xrange=[.45,.55],xstyle=1
GDL> h=histogram(rat,min=0,max=1,bin=.001)
GDL> plot,indgen(1001)*.001,h,psym=10,xrange=[.45,.55]
GDL>
```



# More Monte Carlo

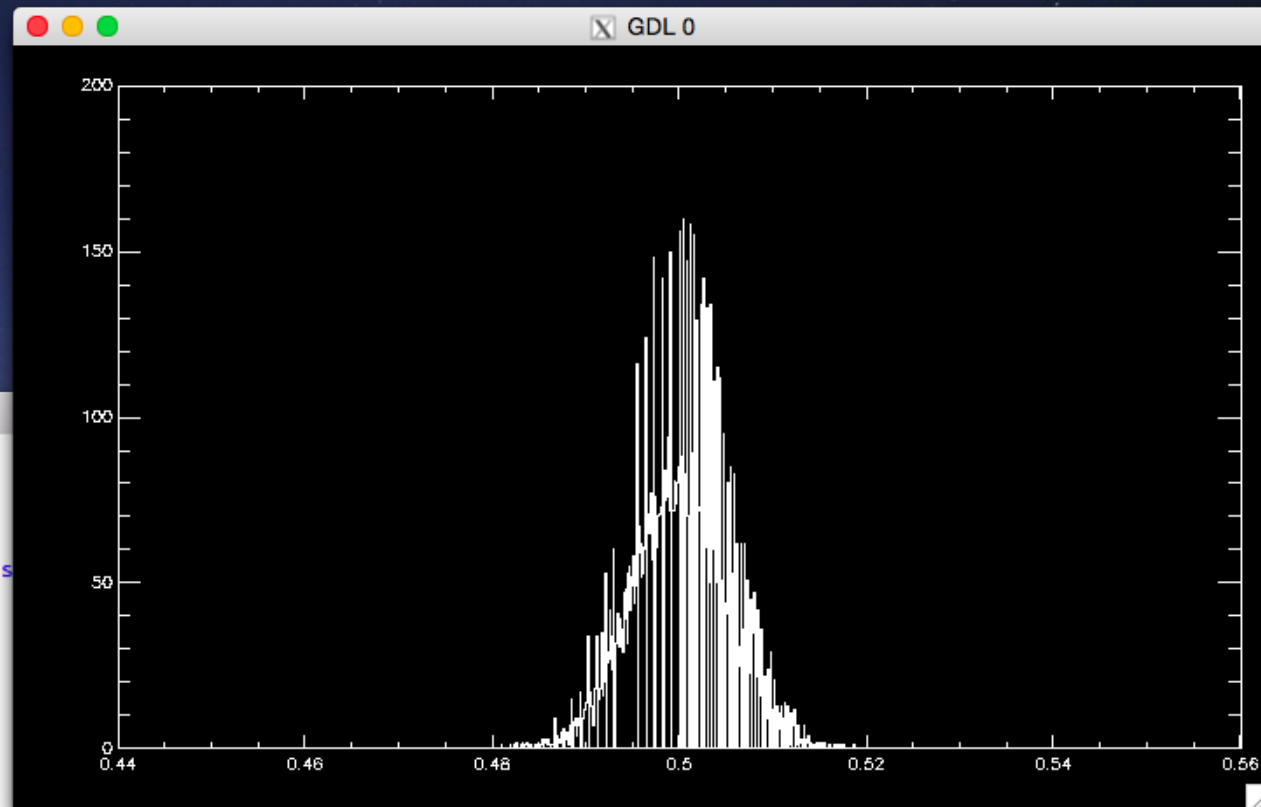
```
2. gdl

GDL - GNU Data Language, Version 0.9.5

- For basic information type HELP,/INFO
- No startup file read (GDL_STARTUP/IDL_STARTUP env. var. not s
- Please report bugs, feature or help requests and patches at:
  http://sourceforge.net/projects/gnudatalanguage/

GDL> .run coin_flip_mc.pro
% Compiled module: $MAIN$.

*** PLPLOT WARNING ***
You said you want pthreads, but they are not available.
GDL> rat=nheads/float(nmax)
GDL> print,mean(rat),median(rat),stdev(rat)
% Compiled module: MEAN.
% Compiled module: STDEV.
    0.499925    0.499900    0.00504734
GDL> h=histogram(rat,min=0,max=1,bin=.0001)
GDL> plot,indgen(10001)*.0001,h,psym=10
GDL> plot,indgen(10001)*.0001,h,psym=10,xrange=[.45,.55]
GDL> plot,indgen(10001)*.0001,h,psym=10,xrange=[.45,.55],xstyle=1
GDL> h=histogram(rat,min=0,max=1,bin=.001)
GDL> plot,indgen(1001)*.001,h,psym=10,xrange=[.45,.55]
GDL>
```





# More Monte Carlo

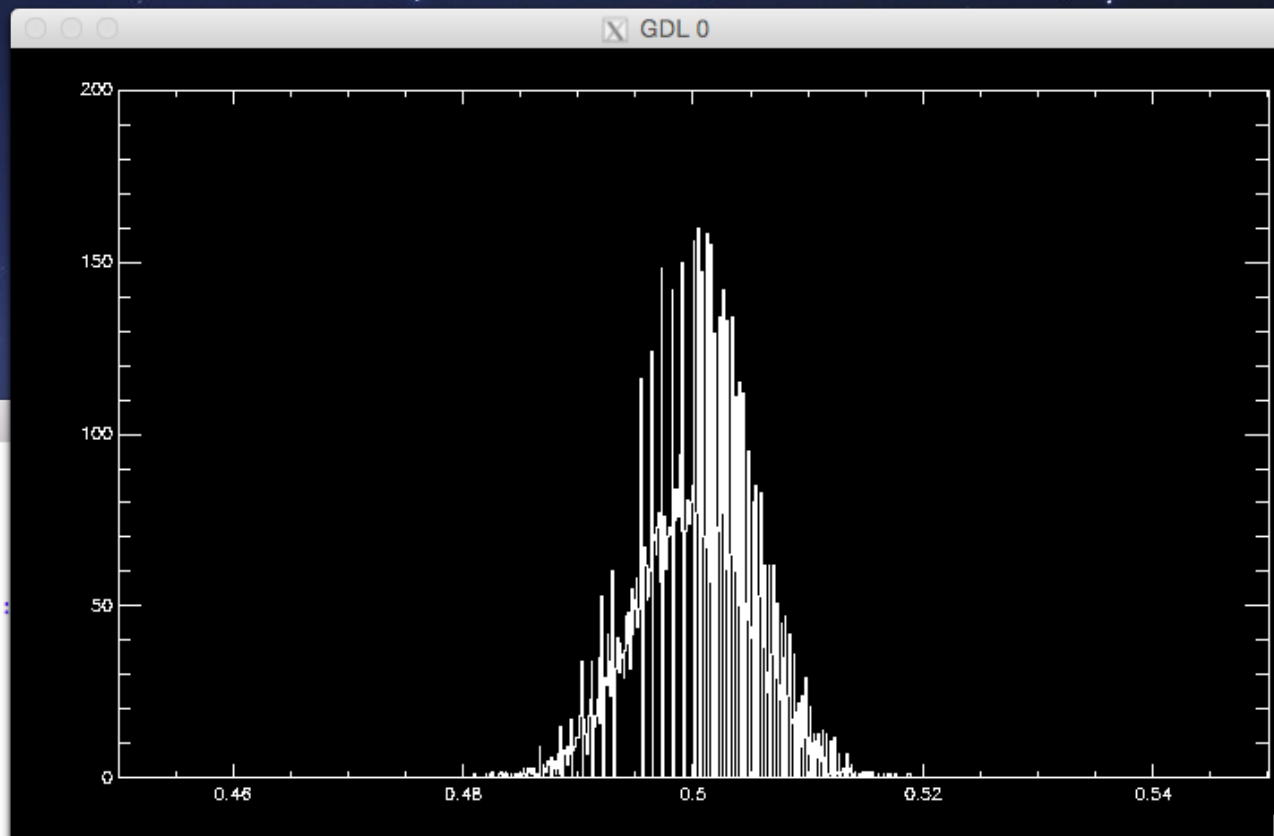
```
2. gdl

GDL - GNU Data Language, Version 0.9.5

- For basic information type HELP,/INFO
- No startup file read (GDL_STARTUP/IDL_STARTUP env. var. not
- Please report bugs, feature or help requests and patches at:
  http://sourceforge.net/projects/gnudatalanguage/

GDL> .run coin_flip_mc.pro
% Compiled module: $MAIN$.

*** PLPLOT WARNING ***
You said you want pthreads, but they are not available.
GDL> rat=nheads/float(nmax)
GDL> print,mean(rat),median(rat),stdev(rat)
% Compiled module: MEAN.
% Compiled module: STDEV.
    0.499925    0.499900    0.00504734
GDL> h=histogram(rat,min=0,max=1,bin=.0001)
GDL> plot,indgen(10001)*.0001,h,psym=10
GDL> plot,indgen(10001)*.0001,h,psym=10,xrange=[.45,.55]
GDL> plot,indgen(10001)*.0001,h,psym=10,xrange=[.45,.55],xstyle=1
GDL> h=histogram(rat,min=0,max=1,bin=.001)
GDL> plot,indgen(1001)*.001,h,psym=10,xrange=[.45,.55]
GDL>
```



# More Monte Carlo

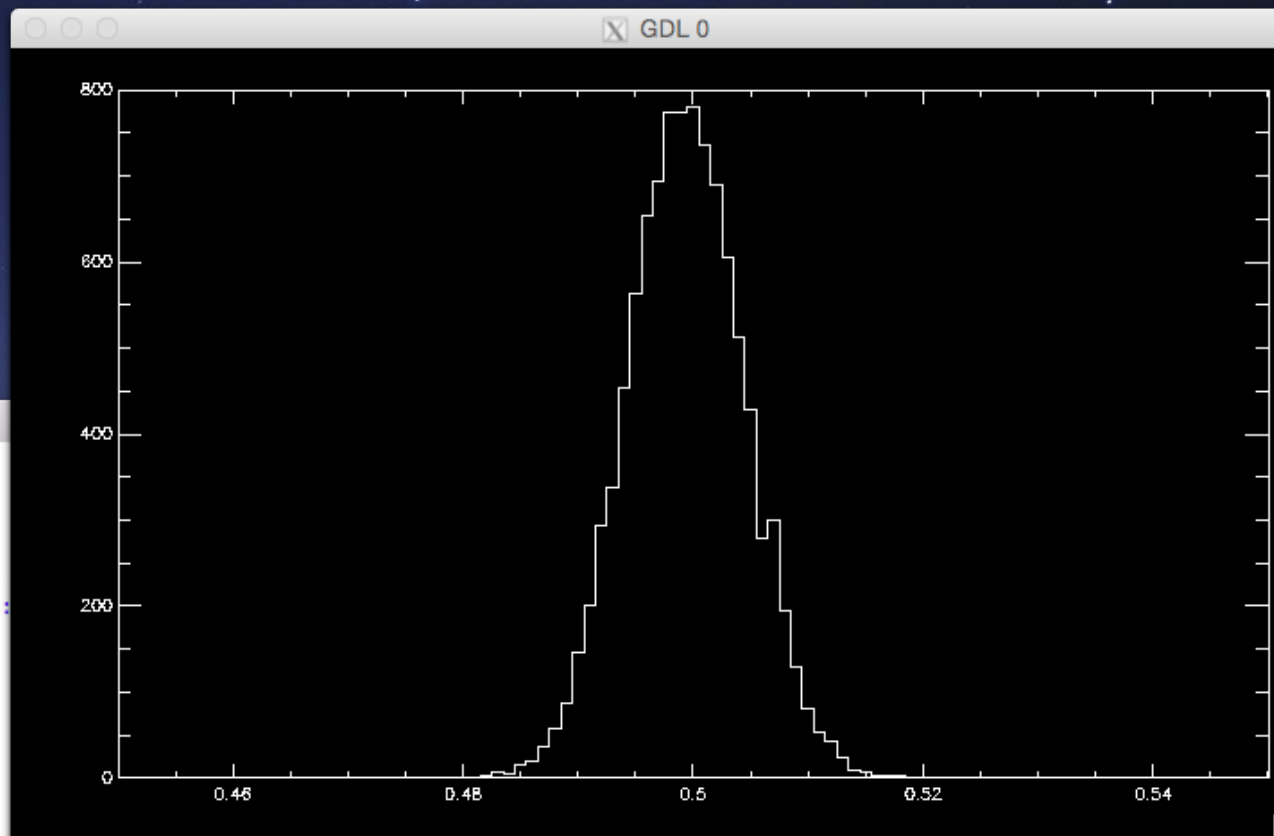
```
2. gdl

GDL - GNU Data Language, Version 0.9.5

- For basic information type HELP,/INFO
- No startup file read (GDL_STARTUP/IDL_STARTUP env. var. not
- Please report bugs, feature or help requests and patches at:
  http://sourceforge.net/projects/gnudatalanguage/

GDL> .run coin_flip_mc.pro
% Compiled module: $MAIN$.

*** PLPLOT WARNING ***
You said you want pthreads, but they are not available.
GDL> rat=nheads/float(nmax)
GDL> print,mean(rat),median(rat),stdev(rat)
% Compiled module: MEAN.
% Compiled module: STDEV.
    0.499925    0.499900    0.00504734
GDL> h=histogram(rat,min=0,max=1,bin=.0001)
GDL> plot,indgen(10001)*.0001,h,psym=10
GDL> plot,indgen(10001)*.0001,h,psym=10,xrange=[.45,.55]
GDL> plot,indgen(10001)*.0001,h,psym=10,xrange=[.45,.55],xstyle=1
GDL> h=histogram(rat,min=0,max=1,bin=.001)
GDL> plot,indgen(1001)*.001,h,psym=10,xrange=[.45,.55]
GDL>
```





# Effects of Sample Size

```
1. vim
nloop=10000
nmax = (10l)^(lindgen(5)+1)

for j = 0,4 do begin
  nheads = lonarr(nloop)
  ntails = lonarr(nloop)
  for i=0l,nloop - 1 do begin
    rnd=randomu(seed,nmax[j])
    heads = n_elements(where(rnd le 0.5))
    tails = n_elements(where(rnd gt 0.5))
    nheads(i) = heads
    ntails(i) = tails
  endfor ; end i
  rat=nheads/float(nmax[j])
  print,nmax[j],mean(rat),stdev(rat)
endfor ; end j
end
~
~
~
~
~
~
"coin_flip_mc3.pro" 17L, 384C written
```

```
2. gdl
$gdl

GDL - GNU Data Language, Version 0.9.5

- For basic information type HELP,/INFO
- No startup file read (GDL_STARTUP/IDL_STARTUP env. var. not set).
- Please report bugs, feature or help requests and patches at:
  http://sourceforge.net/projects/gnudatalanguage/

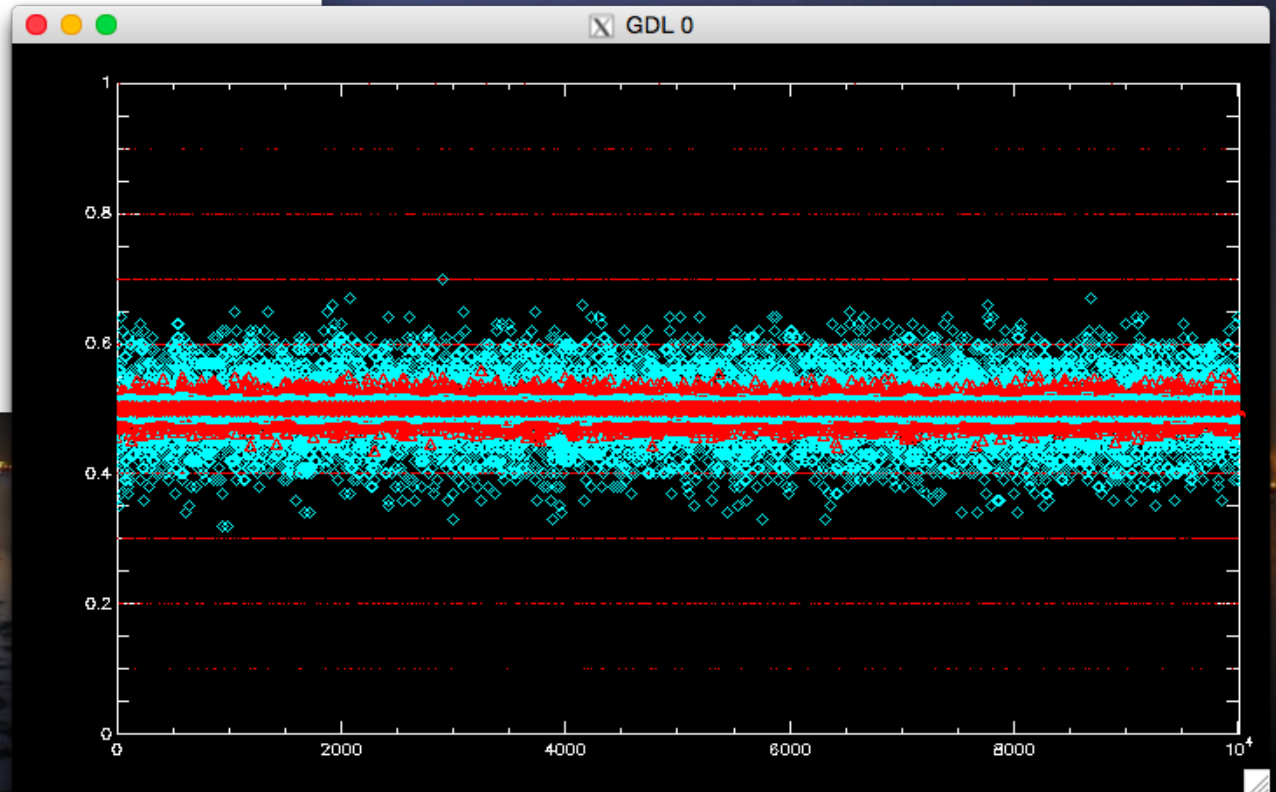
GDL> .run coin_flip_mc3.pro
% Compiled module: $MAIN$.
% Compiled module: MEAN.
% Compiled module: STDEV.
      10      0.497480      0.157527
     100      0.500109      0.0498468
    1000      0.500047      0.0158167
   10000      0.499984      0.00495140
  100000      0.500006      0.00158280

GDL> 
```

# Do it with *Style*

```
1. vim
nloop=10000
nmax = (101)^(lindgen(5)+1)
plot,[0,nloop],[0,1],/nodata

for j = 0,4 do begin
nheads = lonarr(nloop)
ntails = lonarr(nloop)
for i=01,nloop - 1 do begin
    rnd=randomu(seed,nmax[j])
    heads = n_elements(where(rnd le 0.5))
    tails = n_elements(where(rnd gt 0.5))
    nheads(i) = heads
    ntails(i) = tails
endfor ; end i
rat=nheads/float(nmax[j])
oplot,rat,psym=(j+3),col=255^(j+1)
wait,1
print,nmax[j],mean(rat),stdev(rat)
endfor ; end j
end
~
~
~
~
"coin_flip_mc3.pro" 20L, 455C written
```





# Central Limit Theorem

- If we take a large enough sample of independent random numbers, the distribution of their mean ( $\mu$ ) will approach a normal distribution with mean  $\mu$  and standard deviation  $\sigma/\sqrt{N}$ .

```
1. vim
nloop= 10000
nmax = 100

tot_heads = lonarr(nmax)
for i=01,nloop - 1 do begin
    rnd=randomu(seed,nmax)
    heads = where(rnd le 0.5, nheads)
    tails = where(rnd gt 0.5, ntails)
    tot_heads[nheads] = tot_heads[nheads]+1
plot,tot_heads,psym=10
xyouts,20,max(tot_heads)/2,i
if (i lt 10)then wait,1
endfor ; end i
end
~
~
~
~
~
~
~
~
~
~
"coin_flip_cvt.pro" 14L, 316C written
```