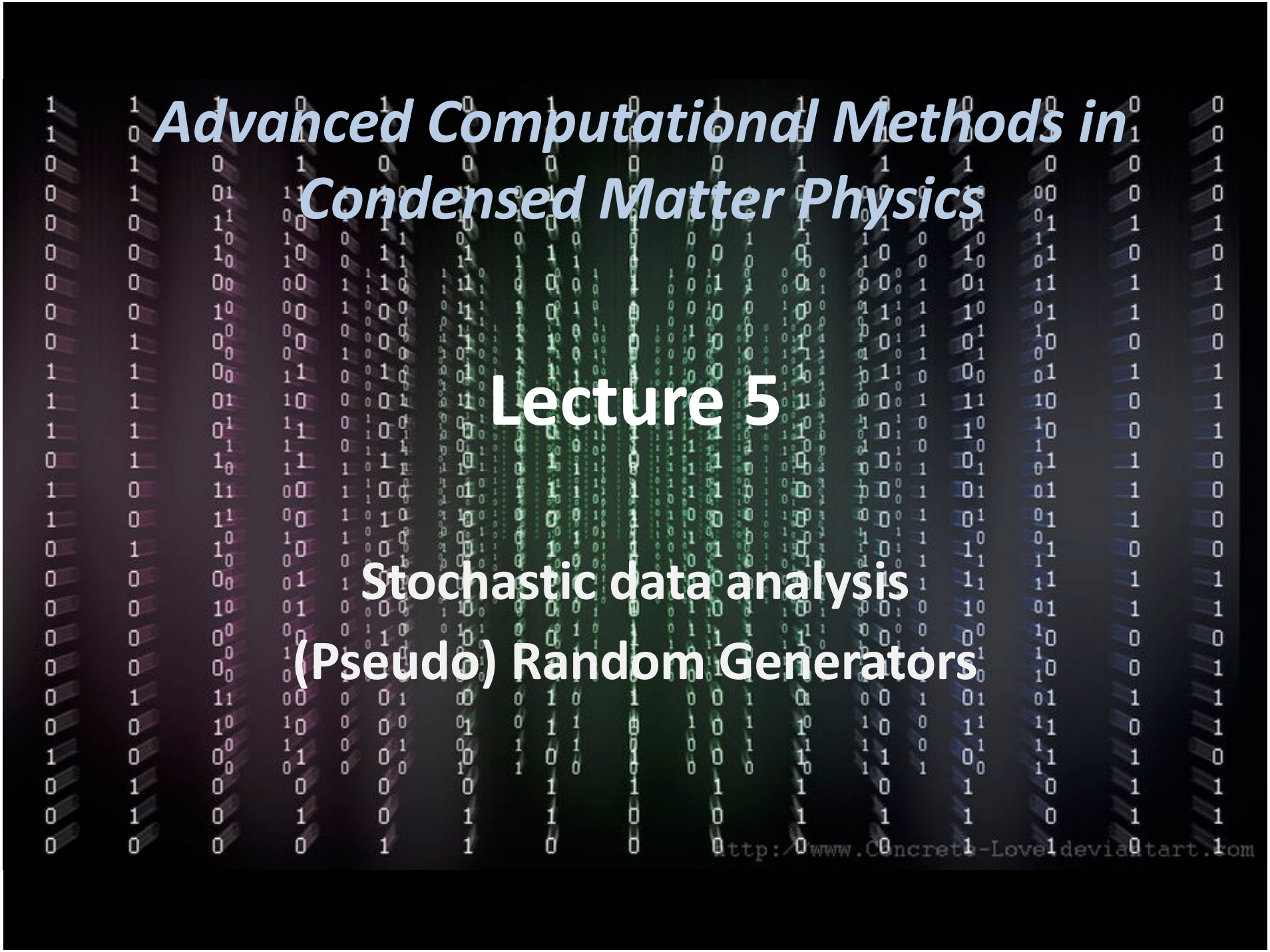

Advanced Computational Methods in Condensed Matter Physics
Lecture 5
Stochastic data analysis
(Pseudo) Random Generators

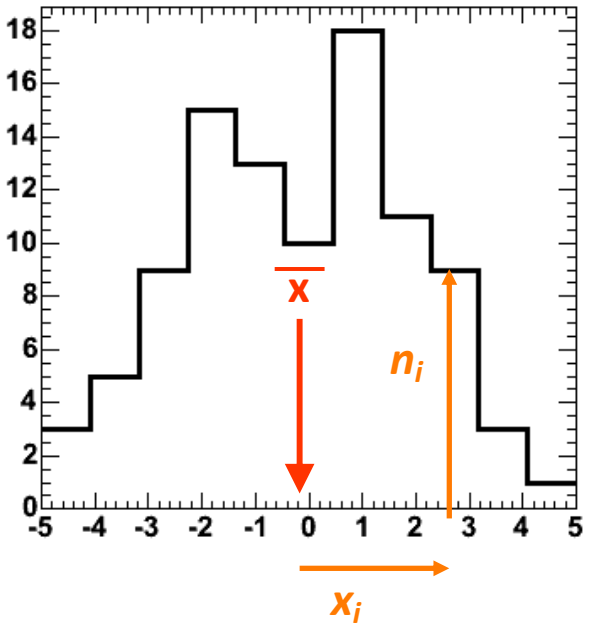
Data Analysis

- Given a set of *unbinned* data points (measurements)
  { x_1, x_2, …, x_N}
  then the mean value or expected value E[x] of quantity x is

$$\overline{x} = \langle x \rangle = \mu = E[x] = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\mathrm{E}[X] = \sum_{i=1}^{N} x_i \, p_i,$$

- For *binned* data $\langle x \rangle = \frac{1}{N_B}\sum_{i=1}^{N_B} n_i x_i$
  - where n_i is bin count and x_i is bin center (N_B=# of bins)
  - Unbinned average more accurate due to rounding



"Spread" of data

- *Variance var(x)* is the expected value of the squared deviation from the mean μ = E[x], or var(x)=E[(x-μ)²]

$$\begin{aligned}
\mathrm{var}(x) &= \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \langle x\rangle\right)^2 \\
&= \frac{1}{N}\sum_{i=1}^{N}\left(x_i^2 - 2x_i\langle x\rangle + \langle x\rangle^2\right) \\
&= \frac{1}{N}\sum_i x_i^2 - \frac{1}{N}2\langle x\rangle\sum_i x_i + \frac{1}{N}\langle x\rangle^2\sum_i 1 \\
&= \langle x^2\rangle - 2\langle x\rangle^2 + \langle x\rangle^2 \\
&= \langle x^2\rangle - \langle x\rangle^2
\end{aligned}$$

- ***Standard deviation***

$$\sigma \equiv \sqrt{V(x)} = \sqrt{\frac{1}{N}\sum_i (x_i - \langle x\rangle)^2} = \sqrt{\langle x^2\rangle - \langle x\rangle^2}$$

Corrected standard deviation

An unbiased estimator for the variance is given by applying Bessel's correction, using N − 1 instead of N to yield the unbiased sample variance, denoted s^2:

$$s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2.$$

- i.e., we multiplied var(x)=σ^2 by the factor N/(N-1).
- This corrected variance should be used when the mean, μ, is unknown.
- The number N-1 corresponds to the number of degrees of freedom
- *Remark:* when calculation the corrected standard deviation, s, one introduces another bias due to the concave nature of the square root. (there is no universal correction formula for that)

Covariance

- Given *2 variables x,y* and a dataset consisting of pairs of numbers

  $\{ (x_1,y_1),\ (x_2,y_2), \ldots (x_N,y_N) \}$

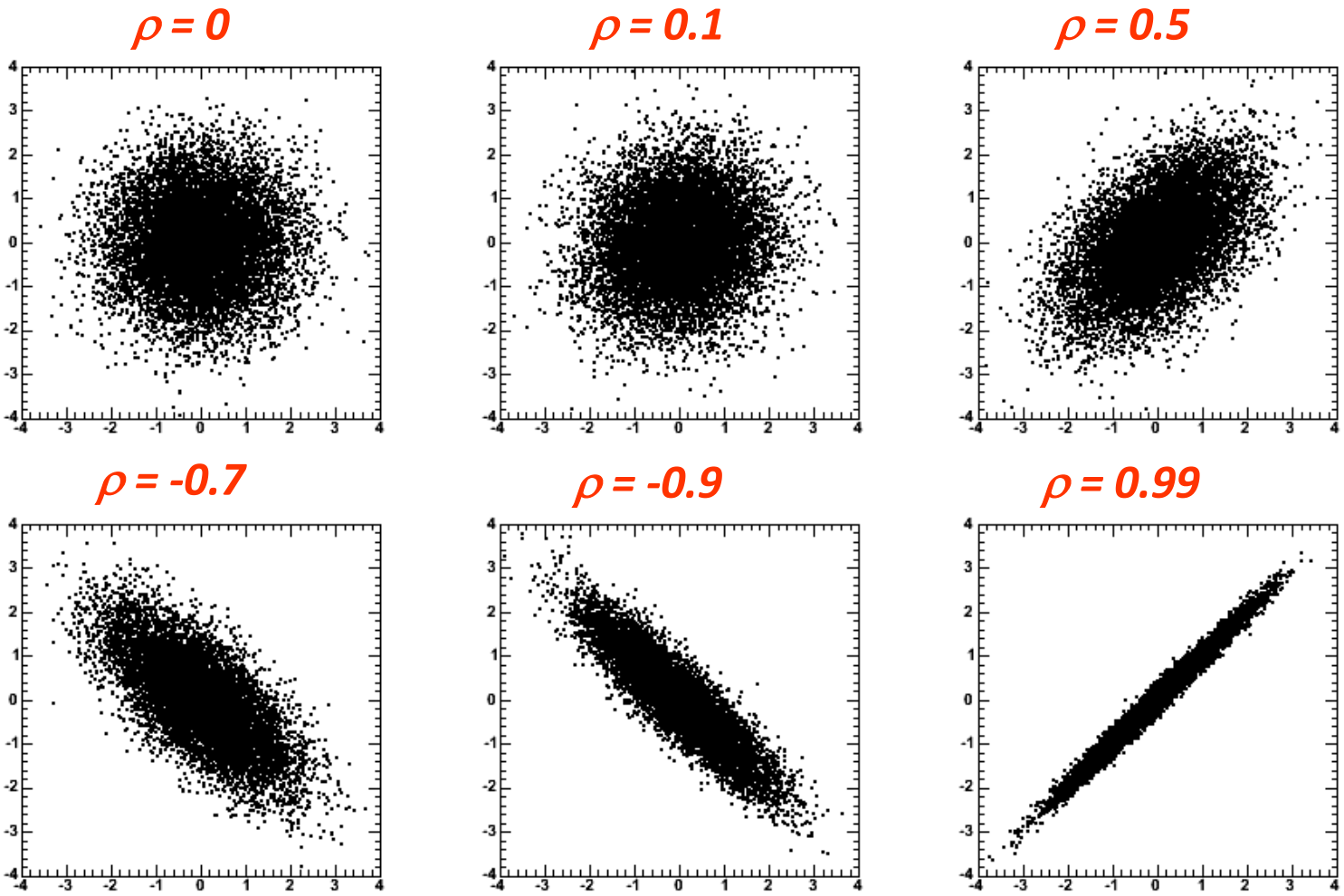
- Definition of <x>, <y>, σ_x, σ_y as usual
- In addition, any *dependence between x,y* described by the ***co***variance

$$\begin{aligned}\mathrm{cov}(x,y) &= \frac{1}{N}\sum_i (x_i - \langle x\rangle)(y_i - \langle y\rangle) \\ &= \overline{(x - \langle x\rangle)(y - \langle y\rangle)} \\ &= \langle xy\rangle - \langle x\rangle\langle y\rangle\end{aligned}$$

- The dimensionless ***correlation*** coefficient is defined as

$$\rho = \frac{\mathrm{cov}(x,y)}{\sigma_x \sigma_y} \quad \in [-1,+1]$$

Correlation example



- Concept of covariance, correlation is easily extended to arbitrary number of variables

$$\mathrm{cov}(x_{(i)}, x_{(j)}) = \left\langle x_{(i)} x_{(j)} \right\rangle - \left\langle x_{(i)} \right\rangle \left\langle x_{(j)} \right\rangle$$

- so that $V_{ij} = \mathrm{cov}(x_{(i)}, x_{(j)})$ takes the form of a *n x n symmetric matrix*

- This is called the ***covariance matrix***, or ***error matrix***
- Similarly the correlation matrix becomes

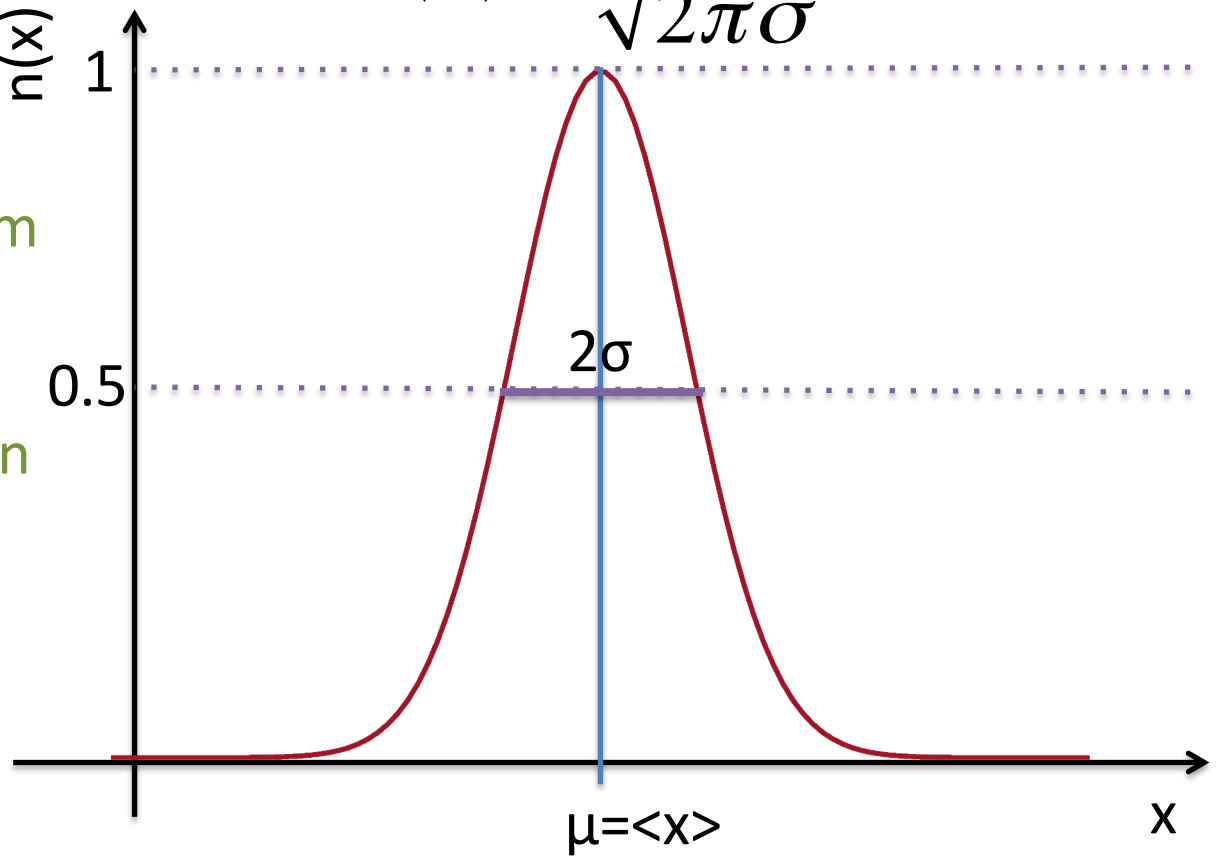
$$\rho_{ij} = \frac{\mathrm{cov}(x_{(i)}, x_{(j)})}{\sigma_{(i)} \sigma_{(j)}} \longrightarrow V_{ij} = \rho_{ij} \sigma_i \sigma_j$$

Distributions

The Gaussian or Normal Distribution: its mean and standard deviation

$$n(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

- 1σ is roughly the half-width at half-maximum of the distribution, probability of a measurement falling in $\pm\sigma$ is 68.3%
- In $\pm 2\sigma$: 95.4%
- In $\pm 3\sigma$: 99.3%



Central Limit Theorem

- Why are errors usually Gaussian?

- The ***Central Limit Theorem*** says
  - If you take the sum X of N independent measurements x_i, each taken from a distribution of mean m_i, a variance $var_i=\sigma_i^2$, the distribution for x

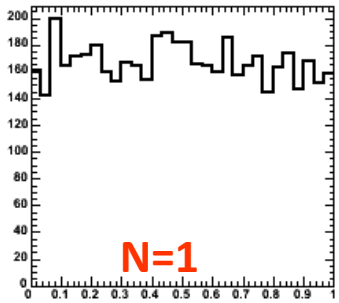
    (a) has expectation value $$\langle X \rangle = \sum_i \mu_i$$

    (b) has variance $$\mathrm{var}(X) = \sum_i \mathrm{var}_i = \sum_i \sigma_i^2$$

    (c) becomes Gaussian as N → ∞

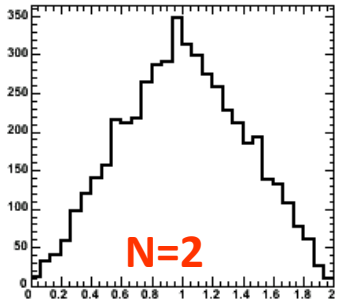
  - *Small print: tails converge very slowly in CLT, be careful in assuming Gaussian shape beyond 2σ*

CLT



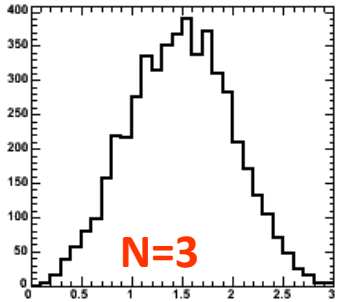
← 5000 numbers taken at random from a uniform distribution between [0,1].

- Mean = $^1/_2$, Variance = $^1/_{12}$

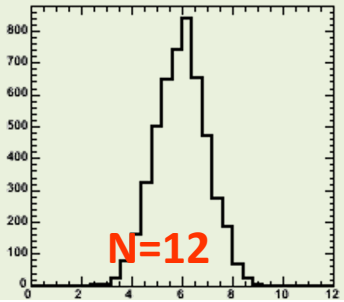


← 5000 numbers, each the sum of 2 random numbers, i.e. $X = x_1+x_2$.

- Triangular shape



← Same for 3 numbers,
$X = x_1 + x_2 + x_3$



← Same for 12 numbers, overlaid curve is exact Gaussian distribution

Binomial distribution

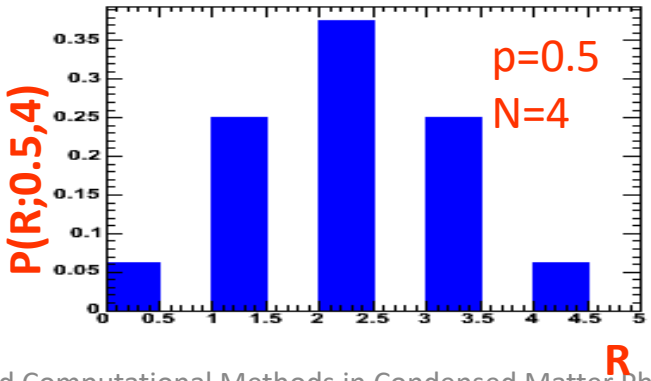
- Simple experiment – Drawing marbles from a bowl
  - Bowl with marbles, fraction **p** are black, others are white
  - Draw **N** marbles from bowl, *put marble back after each drawing*
  - Distribution of **R** black marbles in drawn sample:

Probability of a specific outcome e.g. 'BBBWBWW'

Number of equivalent permutations for that outcome

$$P(R;p,N) = p^R (1-p)^{N-R} \frac{N!}{R!(N-R)!}$$

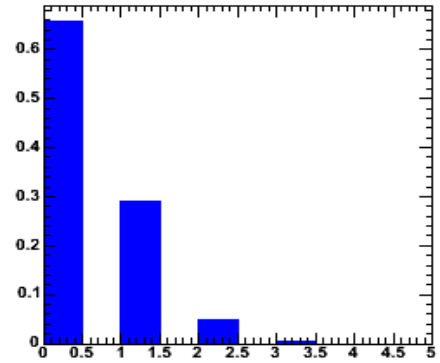
Binomial distribution



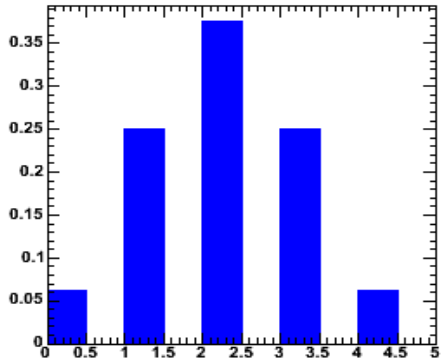
Properties of the binomial dist.

- Mean: $\langle r \rangle = n \cdot p$
- Variance: $\mathrm{var}(r) = np(1-p) \quad \Rightarrow \quad \sigma = \sqrt{np(1-p)}$

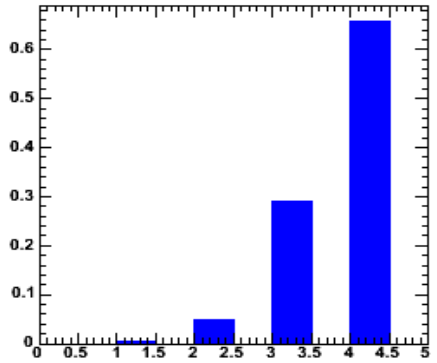
p=0.1, N=4



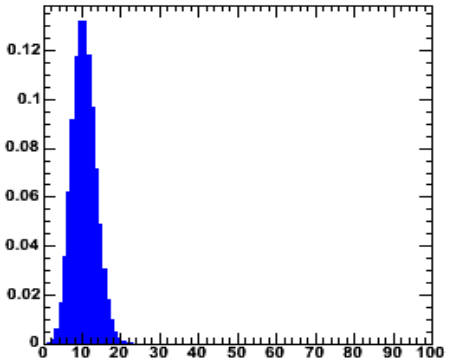
p=0.5, N=4



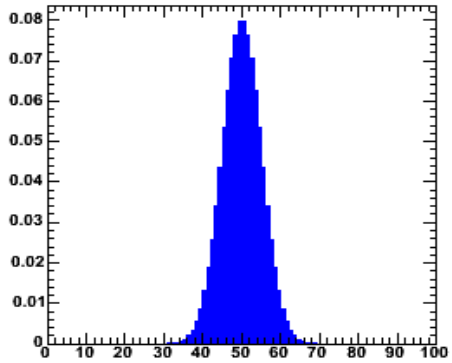
p=0.9, N=4



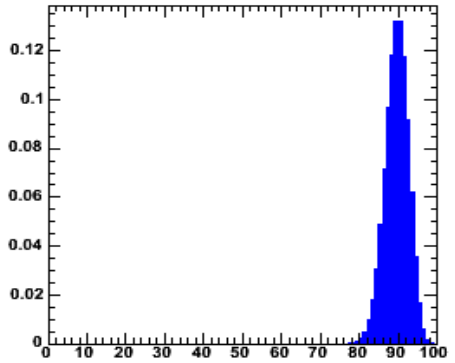
p=0.1, N=1000



p=0.5, N=1000



p=0.9, N=1000



Poisson distribution

- Sometimes we don't know the equivalent of the number of drawings
  - **Example: Geiger counter**
  - Sharp events occurring in a (time) continuum
- What distribution to we expect in measurement over fixed amount of time?
  - Divide time interval λ in n finite chunks,
  - Take binomial formula with p=λ/n and let n→∞

$$P(r;\lambda/n,n)=\frac{\lambda^r}{n^r}(1-\frac{\lambda}{n})^{n-r}\frac{n!}{r!(n-r)!}$$

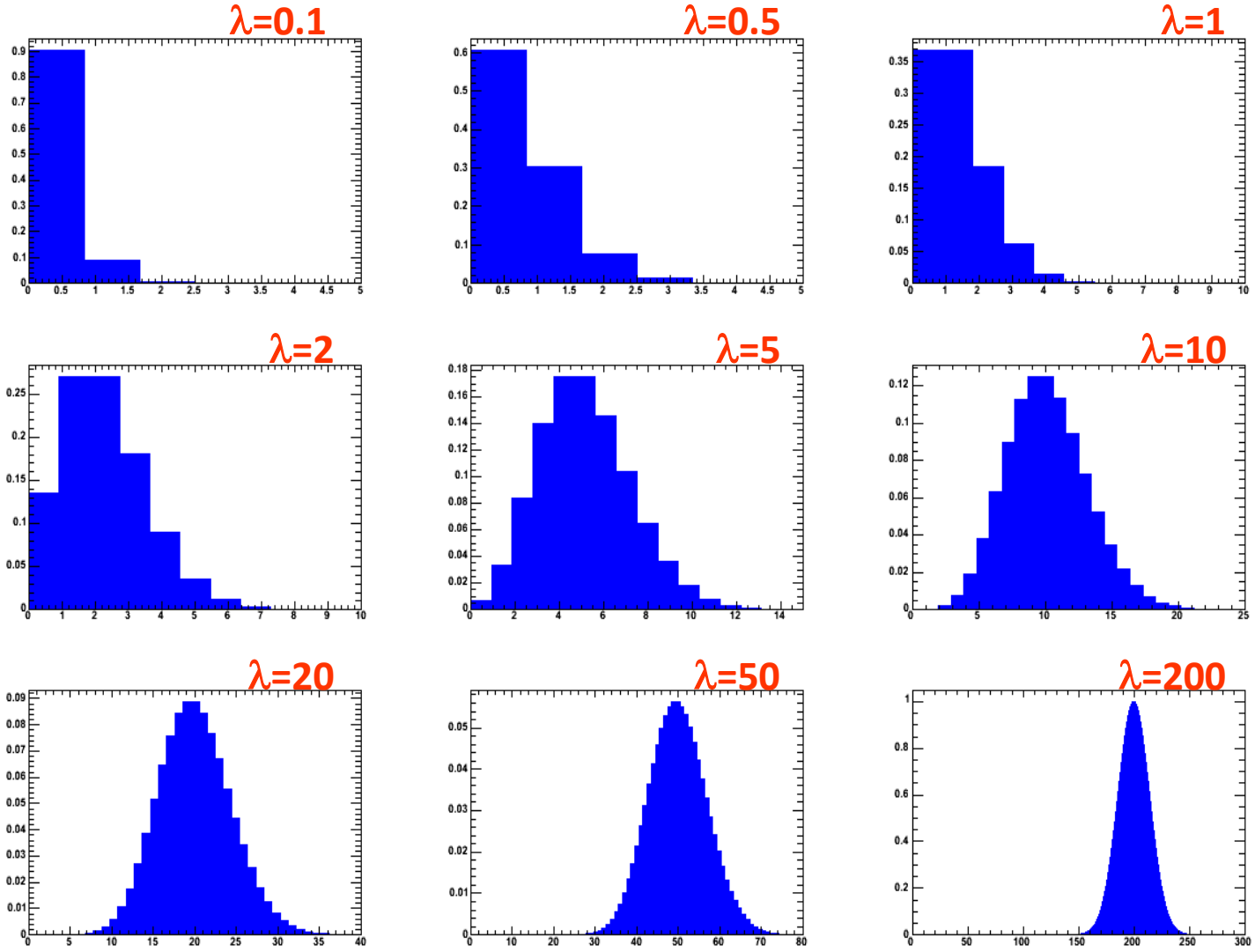
$$\lim_{n\to\infty}\frac{n!}{r!(n-r)!}=n^r,$$

$$\lim_{n\to\infty}(1-\frac{\lambda}{n})^{n-r}=e^{-\lambda}$$

$$P(r;\lambda)=\frac{e^{-\lambda}\lambda^r}{r!}$$

←**Poisson distribution**

Poisson dist.



$$P(r;\lambda) = \frac{e^{-\lambda}\lambda^r}{r!}$$

- Mean, variance: $\langle r \rangle = \lambda$

$$\mathrm{var}(r) = \lambda \quad \Rightarrow \quad \sigma = \sqrt{\lambda}$$

- Convolution of 2 Poisson distributions is also a Poisson distribution with $\lambda_{ab}=\lambda_a+\lambda_b$

$$P(r) = \sum_{r_A=0}^{r} P(r_A;\lambda_A)P(r-r_A;\lambda_B)$$

$$= e^{-\lambda_A}e^{-\lambda_B}\sum \frac{\lambda_A^{r_A}\lambda_B^{r-r_A}}{r_A!(r-r_A)!}$$

$$= e^{-(\lambda_A+\lambda_B)}\frac{(\lambda_A+\lambda_B)^r}{r!}\sum_{r_{A=0}}^{r}\frac{r!}{(r-r_A)!}\left(\frac{\lambda_A}{\lambda_A+\lambda_B}\right)^{r_A}\left(\frac{\lambda_B}{\lambda_A+\lambda_B}\right)^{r-r_A}$$

$$= e^{-(\lambda_A+\lambda_B)}\frac{(\lambda_A+\lambda_B)^r}{r!}\left(\frac{\lambda_A}{\lambda_A+\lambda_B}+\frac{\lambda_B}{\lambda_A+\lambda_B}\right)^r$$

$$= e^{-(\lambda_A+\lambda_B)}\frac{(\lambda_A+\lambda_B)^r}{r!}$$

...

- Look at *Poisson distribution* in limit of *large N*

$$P(r;\lambda) = e^{-\lambda}\frac{\lambda^r}{r!}$$

Take log, substitute, r = λ + x, and use $\ln(r!) \approx r\ln r - r + \ln\sqrt{2\pi r}$

$$\ln(P(r;\lambda)) = -\lambda + r\ln\lambda - (r\ln r - r) - \ln\sqrt{2\pi r}$$

$$= -\lambda + r\left[\ln\lambda - \ln(\lambda(1+\frac{x}{\lambda}))\right] + (\lambda + x) - \ln\sqrt{2\pi\lambda}$$

$$\approx x - (\lambda - x)\left(\frac{x}{\lambda} + \frac{x^2}{2\lambda^2}\right) - \ln(2\pi\lambda)$$

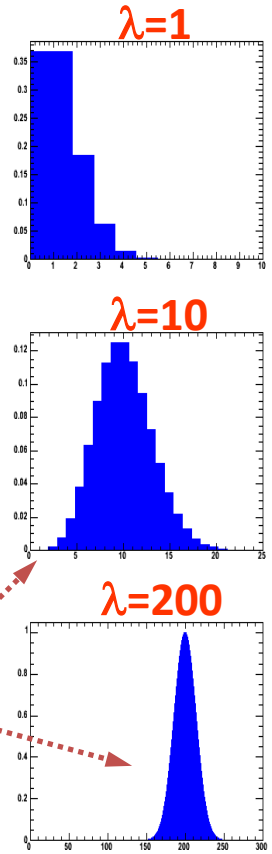
$\ln(1+z) \approx z - z^2/2$

$$\approx \frac{-x^2}{2\lambda} - \ln(2\pi\lambda)$$

Take exp

$$P(x) = \frac{e^{-x^2/2\lambda}}{\sqrt{2\pi\lambda}}$$

Familiar Gaussian distribution,
(approximation reasonable for N>10)



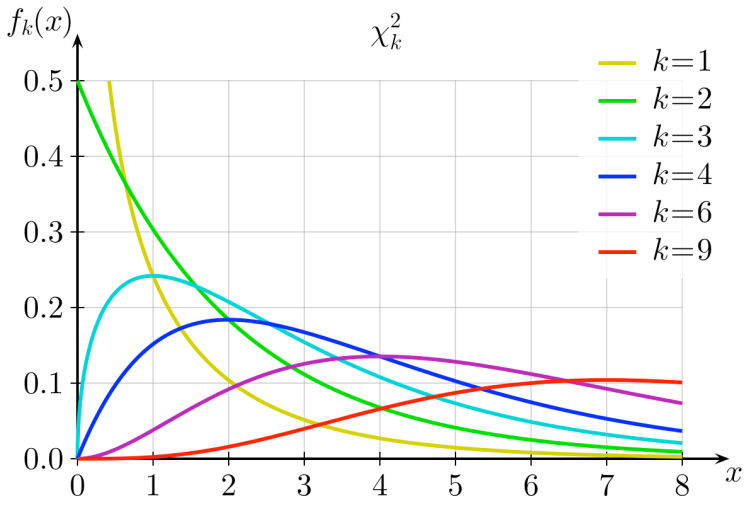
χ^2 test

- A Chi-Square test (χ^2) is a statistical test used to determine whether your experimentally observed results are consistent with your hypothesis (*goodness of fit*).
- Usually refers to *Pearson's chi-squared test*
- The value of the test-statistic is

$$\chi^2 = \sum_{i=1}^{N} \frac{(x_i - E_i)^2}{E_i}$$

- Where x_i is an measured data point, E_i an expected (theoretical) value (asserted by the null hypothesis), and χ^2 Pearson's cumulative test statistic, which asymptotically approaches a χ^2 distribution (a distribution of the sum of k random normal numbers squared.).

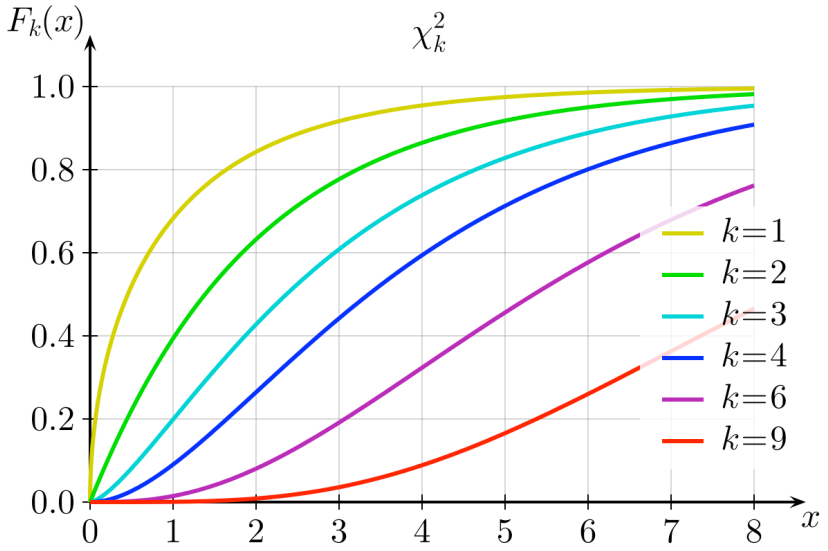
χ^2 distribution



Probability density function (pdf)

$$f(x;\,k) = \begin{cases} \dfrac{x^{(k/2)-1}e^{-x/2}}{2^{k/2}\Gamma\left(\frac{k}{2}\right)}, & x \geq 0; \\ 0, & \text{otherwise.} \end{cases}$$

Mean: k
Variance: 2k



Cumulative distribution function (CDF)

$$F(x;\,k) = \frac{\gamma\left(\frac{k}{2},\,\frac{x}{2}\right)}{\Gamma\left(\frac{k}{2}\right)} = P\left(\frac{k}{2},\,\frac{x}{2}\right)$$

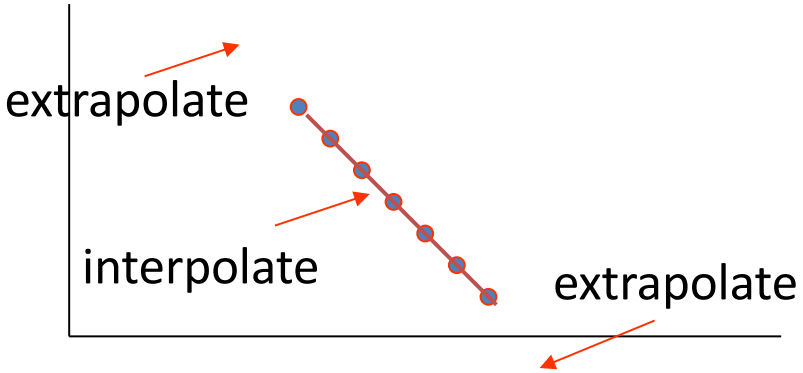
where γ(s,t) is the lower incomplete Gamma function and P(s,t) is the regularized Gamma function

Data/Curve fitting

- Often, we have data sets from experimental/observational measurements
  - Typically, find that the data/dependent variable/output varies…
  - As the control parameter/independent variable/input varies. Examples:
    - Classic gravity drop: location changes with time
    - Pressure varies with depth
    - Wind speed varies with time
    - Temperature varies with location
- Scientific method: Given data identify underlying relationship
- Process known as curve fitting:

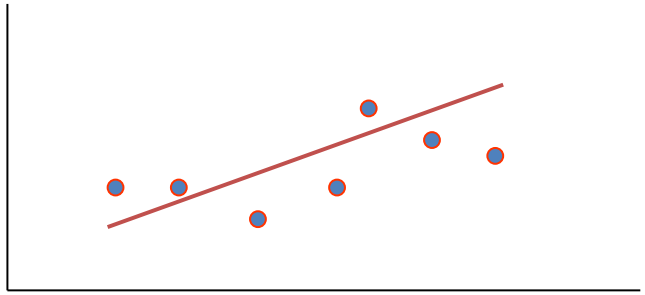
Interpolation vs. Regression

- Distinctly different approaches depending on the quality of the data
- Consider the pictures below:



Pretty confident:
there is a polynomial relationship
Little/no scatter
Want to find an expression
that passes exactly through all the points

Not discussed here!



Unsure what the relationship is
Clear scatter
Want to find an expression
that captures the trend:
minimize some measure of the error
Of all the points…

Linear Regression

- Fitting a **straight line** to a set of paired observations:

$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

y_i : **measured value**

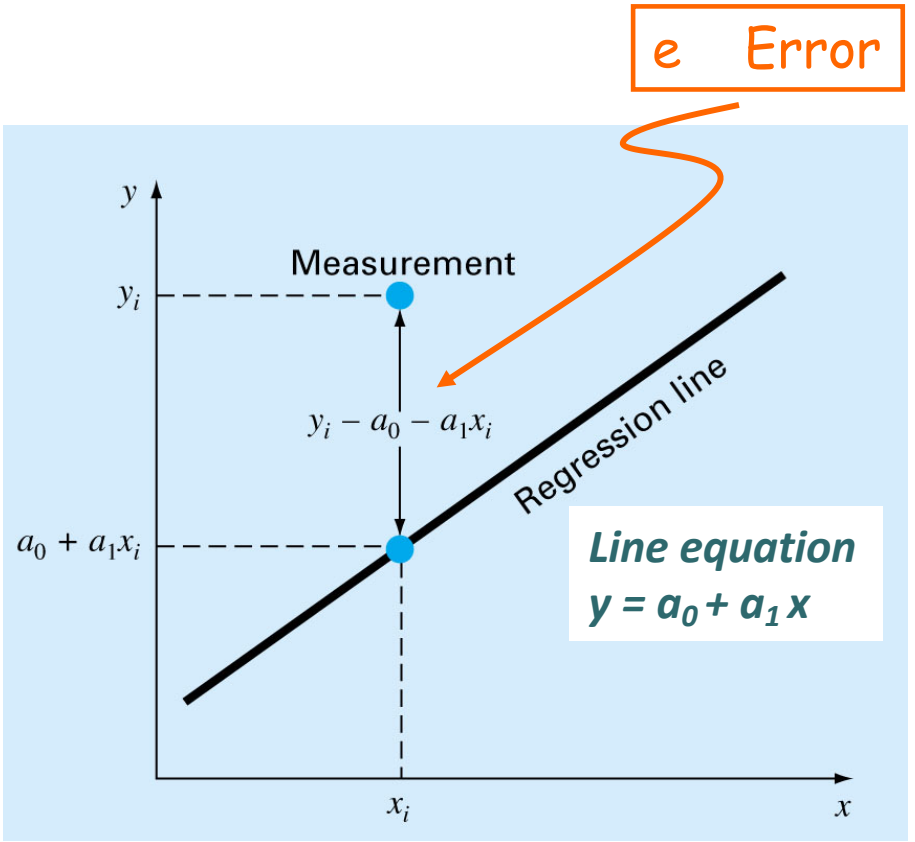
e : **error**

$$y_i = a_0 + a_1 x_i + e$$

$$e = y_i - a_0 - a_1 x_i$$

a_1 : *slope*

a_0 : *intercept*



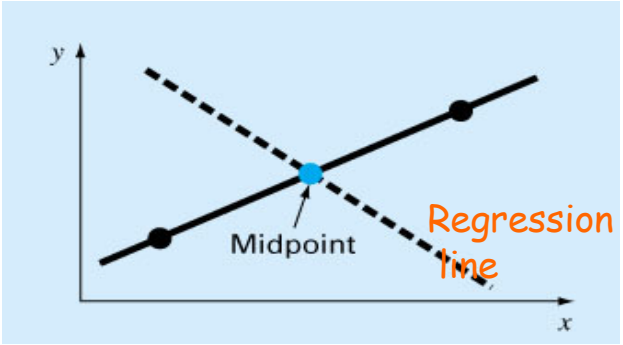
Choosing Criteria For a "Best Fit"

- **Minimize** the sum of the residual errors for all available data?

$$\sum_{i=1}^{n} e_i = \sum_{i=1}^{n} (y_i - a_o - a_1 x_i)$$

Inadequate!

(see ➔➔➔)

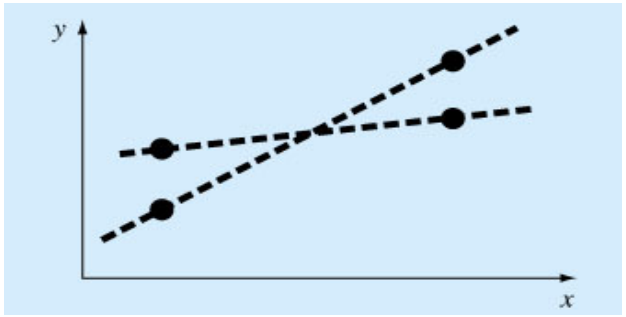


- Sum of the absolute values?

$$\sum_{i=1}^{n} |e_i| = \sum_{i=1}^{n} |y_i - a_0 - a_1 x_i|$$

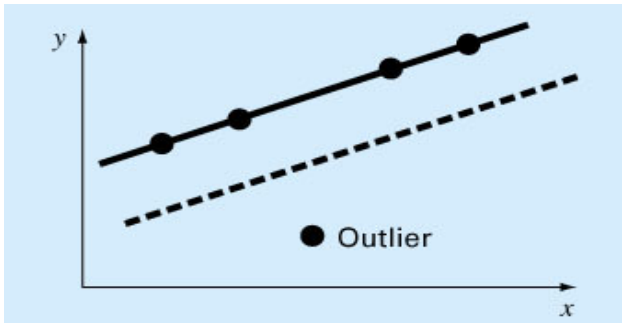
Inadequate!

(see ➔➔➔)



- How about minimizing the distance that an individual point falls from the line?

This does not work either! see ➔➔➔

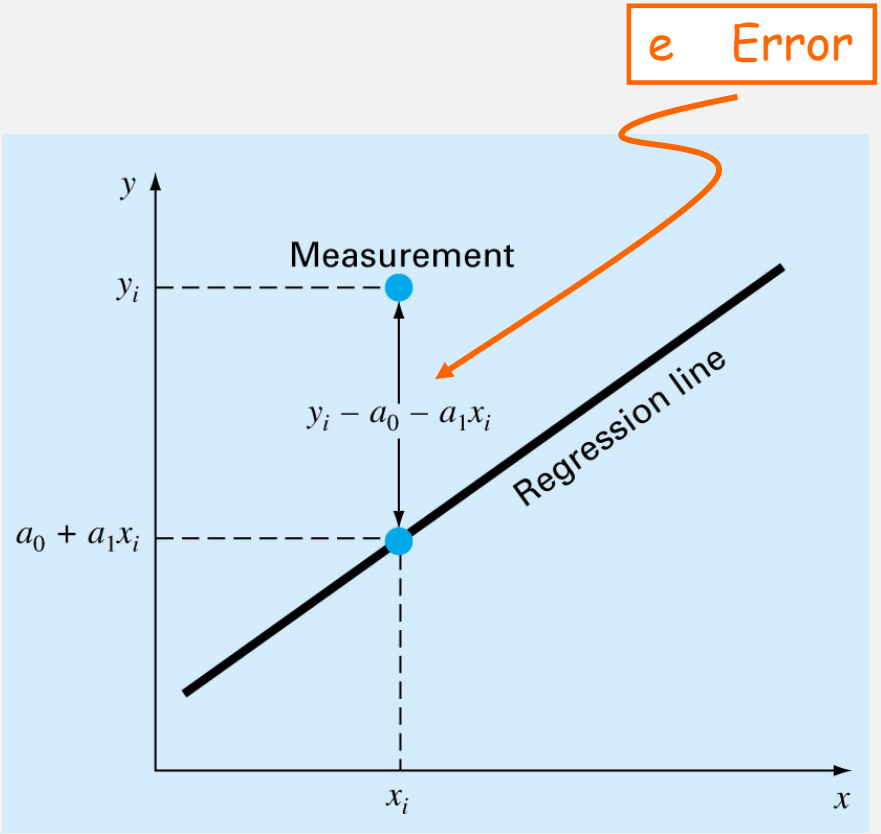


- Best strategy is to *minimize* the sum of the squares of the residuals between the *measured*-y and the y *calculated* with the linear model:

$$S_r = \sum_{i=1}^{n} e_i^2$$

$$= \sum_{i=1}^{n} (y_{i,measured} - y_{i,model})^2$$

$$S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$



- Yields a unique line for a given set of data
- Need to compute a_0 and a_1 such that S_r is minimized!

Least-Squares Fit of a Straight Line

$$\text{Minimize error:} \quad S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$

$$\frac{\partial S_r}{\partial a_o} = -2\sum (y_i - a_o - a_1 x_i) = 0 \quad \Rightarrow \quad \sum y_i - \sum a_0 - \sum a_1 x_i = 0$$

$$\frac{\partial S_r}{\partial a_1} = -2\sum [(y_i - a_o - a_1 x_i) x_i] = 0 \quad \Rightarrow \quad \sum y_i x_i - \sum a_0 x_i - \sum a_1 x_i^2 = 0$$

Since $\sum a_0 = n a_0$

$$(1) \qquad n a_0 + \left(\sum x_i\right) a_1 = \sum y_i$$

$$(2) \qquad \left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 = \sum y_i x_i$$

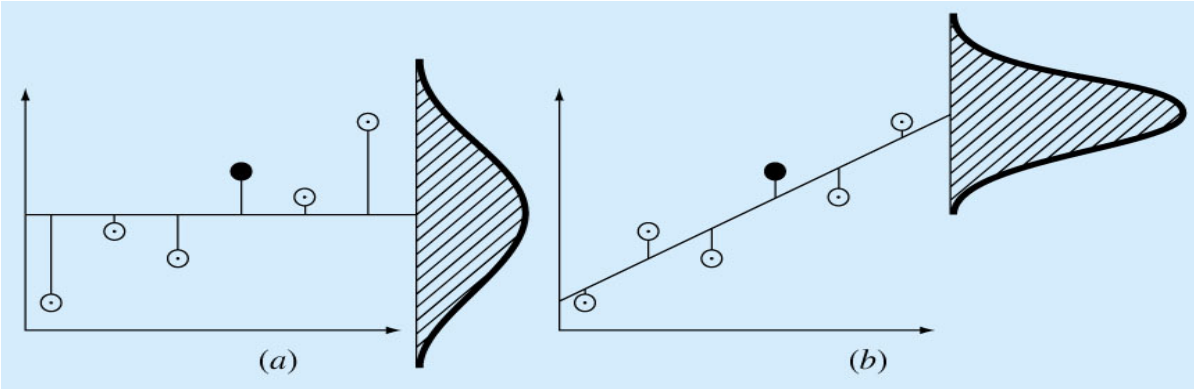
Linear equations for a_0 and a_1

$$a_1 = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2}$$

using (1), a_0 can be expressed as $a_0 = \bar{y} - a_1 \bar{x}$

Mean values

Goodness of the fit



The spread of data

(a) around the mean

(b) around the best-fit line

Notice the improvement in the error due to *linear regression*

- S_r = Sum of the squares of residuals around the regression line
- S_t = total sum of the squares around the mean
- $(S_t - S_r)$ quantifies the improvement or error reduction due to describing data in terms of *a straight line* rather than as *an average value*.

r : *correlation coefficient*

r^2 : *coefficient of determination*

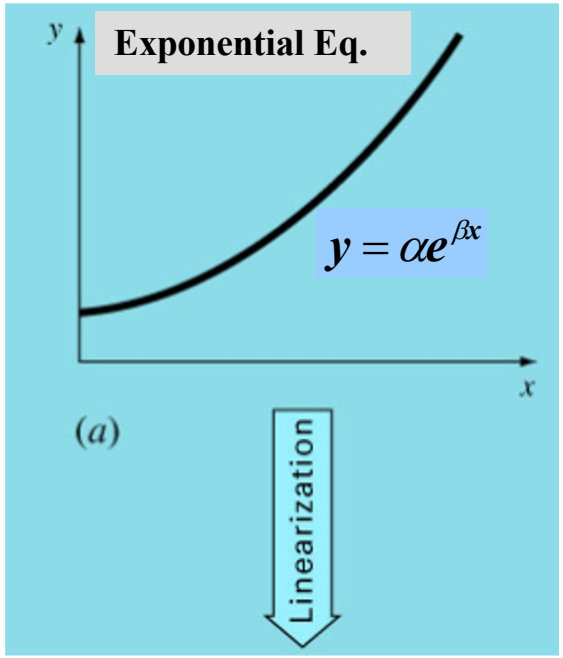
$$r^2 = \frac{S_t - S_r}{S_t}$$

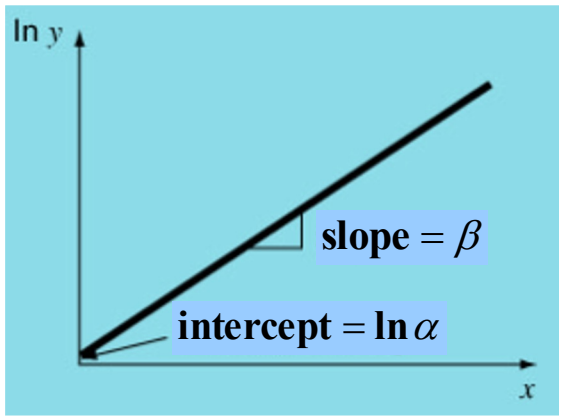
$$S_t = \sum (y_i - \bar{y})^2$$

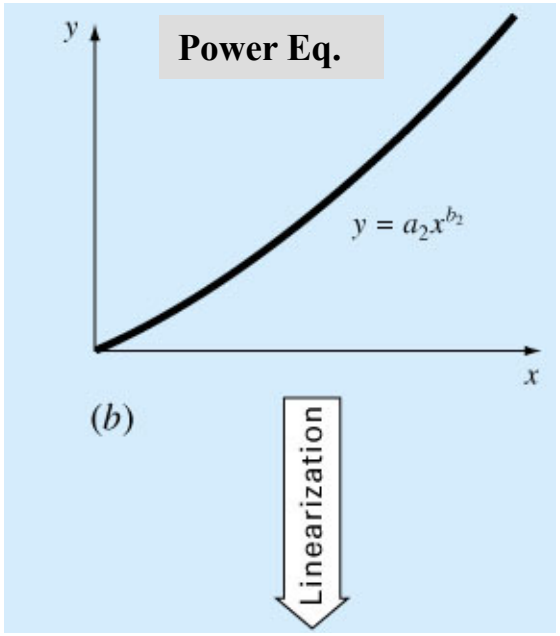
$$S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$

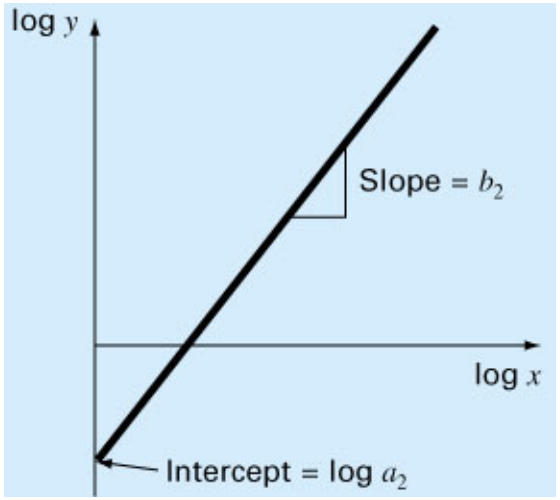
- For a perfect fit S_r=0 and $r = r^2 = 1$ signifies that the line explains 100 percent of the variability of the data.
- For $r = r^2 = 0$ ➔ S_r=S_t ➔ the fit represents no improvement

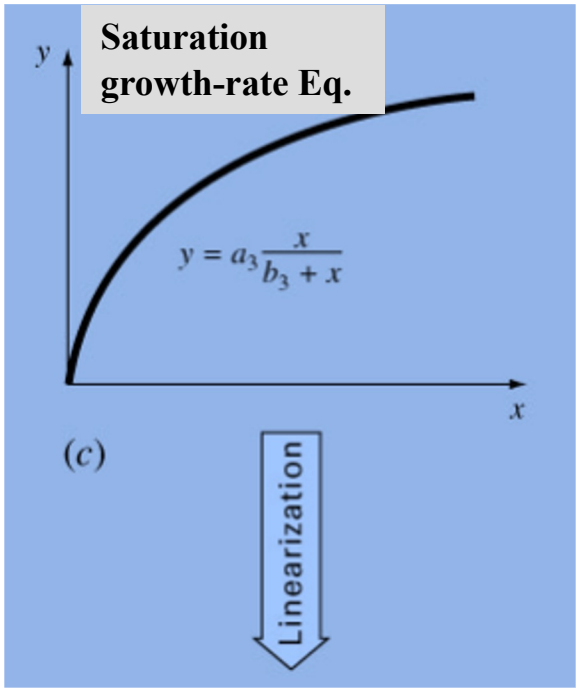
Linearization of Nonlinear Relationships

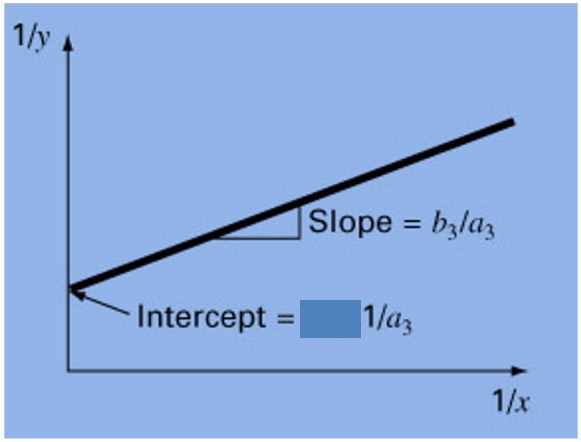












Polynomial Regression

- Some data is poorly represented by a straight line. A curve (polynomial) may be better suited to fit the data. The least squares method can be extended to fit the data to higher order polynomials.
- As an example let us consider a second order polynomial to fit the data points:

$$y = a_0 + a_1 x + a_2 x^2$$

$$\text{Minimize error}: \quad S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2$$

$$\frac{\partial S_r}{\partial a_o} = -2\sum (y_i - a_o - a_1 x_i - a_2 x_i^2) = 0$$

$$\frac{\partial S_r}{\partial a_1} = -2\sum x_i (y_i - a_o - a_1 x_i - a_2 x_i^2) = 0$$

$$\frac{\partial S_r}{\partial a_2} = -2\sum x_i^2 (y_i - a_o - a_1 x_i - a_2 x_i^2) = 0$$

$$n a_0 + \left(\sum x_i\right) a_1 + \left(\sum x_i^2\right) a_2 = \sum y_i$$

$$\left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 + \left(\sum x_i^3\right) a_2 = \sum x_i y_i$$

$$\left(\sum x_i^2\right) a_0 + \left(\sum x_i^3\right) a_1 + \left(\sum x_i^4\right) a_2 = \sum x_i^2 y_i$$

...

- To fit the data to an m^{th} order polynomial, we need to solve the following system of linear equations ((m+1) equations with (m+1) unknowns)

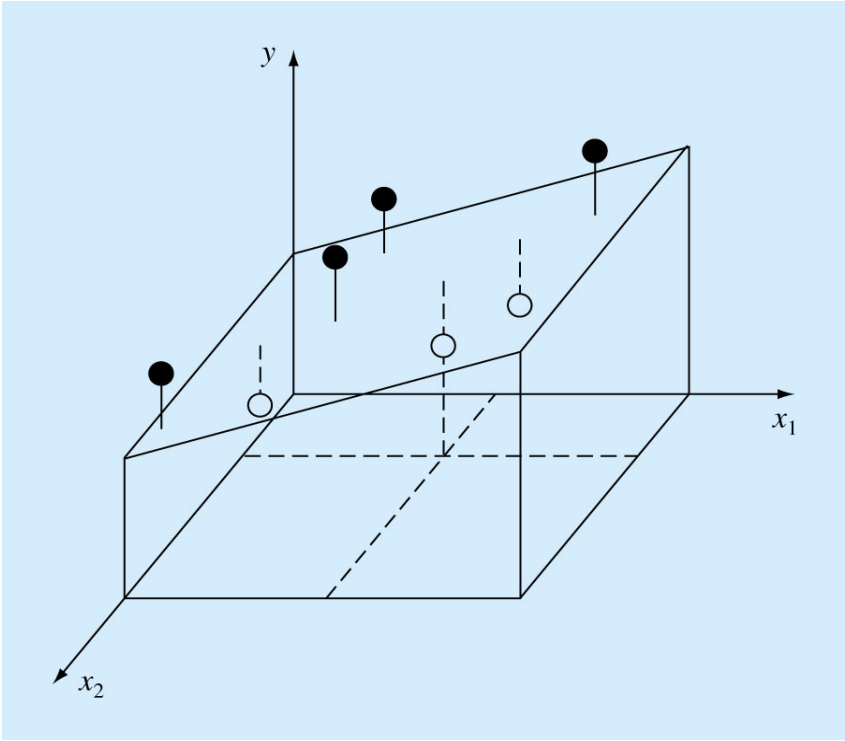
$$\begin{bmatrix} n & \sum x_i & \cdots & \sum x_i^m \\ \sum x_i & \sum x_i^2 & \cdots & \sum x_i^{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum x_i^m & \sum x_i^{m+1} & \cdots & \sum x_i^{m+m} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \vdots \\ \sum x_i^m y_i \end{bmatrix}$$

Multiple Linear Regression

- A useful extension of linear regression is the case where *y* is a linear function of two or more independent variables. For example:

$$y = a_o + a_1 x_1 + a_2 x_2 + e$$

- For this 2-dimensional case, the regression line becomes a plane as shown in the figure below.



...

Example (2-vars): Minimize error: $S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_{1i} - a_2 x_{2i})^2$

$$\frac{\partial S_r}{\partial a_o} = -2\sum (y_i - a_o - a_1 x_{1i} - a_2 x_{2i}) = 0$$

$$\frac{\partial S_r}{\partial a_1} = -2\sum x_{1i} (y_i - a_o - a_1 x_{1i} - a_2 x_{2i}) = 0$$

$$\frac{\partial S_r}{\partial a_2} = -2\sum x_{2i} (y_i - a_o - a_1 x_{1i} - a_2 x_{2i}) = 0$$

$$n a_0 + \left(\sum x_{1i}\right) a_1 + \left(\sum x_{2i}\right) a_2 = \sum y_i$$

$$\left(\sum x_{1i}\right) a_0 + \left(\sum x_{1i}^2\right) a_1 + \left(\sum x_{1i} x_{2i}\right) a_2 = \sum x_{1i} y_i$$

$$\left(\sum x_{2i}\right) a_0 + \left(\sum x_{1i} x_{2i}\right) a_1 + \left(\sum x_{2i}^2\right) a_2 = \sum x_{2i} y_i$$

$$\begin{bmatrix} n & \sum x_{1i} & \sum x_{2i} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{1i} x_{2i} \\ \sum x_{2i} & \sum x_{1i} x_{2i} & \sum x_{2i}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_{1i} y_i \\ \sum x_{2i} y_i \end{bmatrix}$$

Curve fitting summary

Use existing software for plotting and data analysis like:

- Mathematica, Maple, MathCad
- (Excel)
- MatLab, scilab, octave
- R
- OriginPro, IGOR Pro, Labplot, qtiplot
- Grace
- *gnuplot, MayaVi, ParaView*

... (see also http://en.wikipedia.org/wiki/List_of_graphing_software ,
http://en.wikipedia.org/wiki/List_of_statistical_packages ,
http://en.wikipedia.org/wiki/List_of_numerical_analysis_software ,
http://en.wikipedia.org/wiki/List_of_computer_algebra_systems)

What is random?

- Definition of random from Merriam-Webster:
- Main Entry: **random**
  Function: *adjective*
  Date: 1565
  1 a : lacking a definite plan, purpose, or pattern **b :** made, done, or chosen at random <read *random* passages from the book>
  2 a : relating to, having, or being elements or events with definite probability of occurrence <*random* processes> **b :** being or relating to a set or to an element of a set each of whose elements has equal probability of occurrence <a *random* sample>; *also* **:** characterized by procedures designed to obtain such sets or elements <*random* sampling>

Random numbers

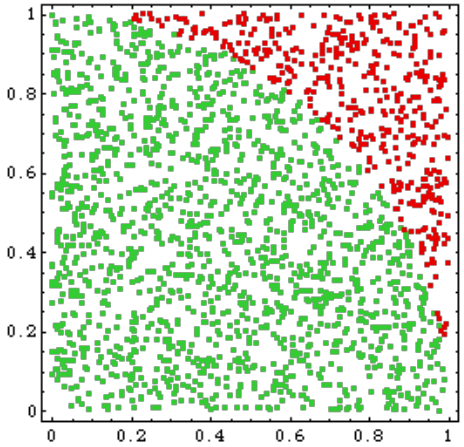
- Utopia
  - True random generators: exhibiting “true” randomness, such as the time between “tics” from a Geiger counter exposed to a radioactive element
    - Hard to find
    - Hard to proof
    - Complex implementation
- Reality
  - Pseudo random number generators
    - Sequences having the appearance of randomness, but nevertheless exhibiting a specific, repeatable pattern.
    - numbers calculated by a computer through a deterministic process, cannot, by definition, be random

“*Any one who consider arithmetical methods of producing random digits is, of course, in a state of sin.*”

John von Neumann [1951]

(Pseudo) Random number generators (RNG)

- **Desirable Attributes:**
  - **Uniformity**
  - **Independence**
  - **Efficiency**
  - **Replicability**
  - **Long Cycle Length**
- Needed for:
  - Numerical Algorithms
  - Simulations
  - "Monte-Carlo" Methods
  - encryption
- Each random number x_i is an independent sample drawn from a continuous uniform distribution between 0 and 1

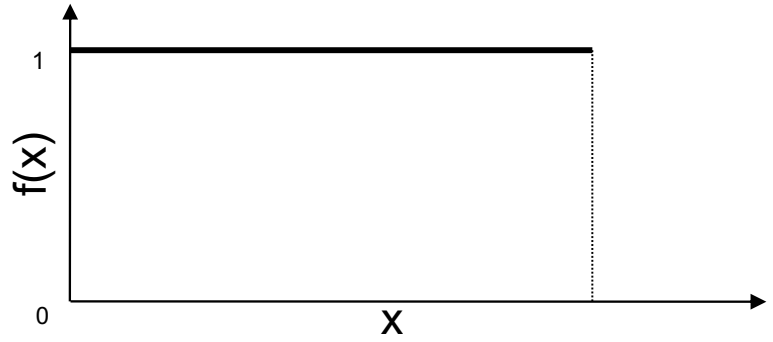


Example: calculation of π using MC

$$\text{pdf:}\quad f(x) = \begin{cases} 1\,,\ 0 \le x \le 1 \\ 0\,,\ \text{else} \end{cases}$$

...

PDF:



$$E[x] = \int_0^1 y dy = [y^2/2]_0^1 = 1/2$$

$$\mathrm{var}(x) = \int_0^1 y^2 dy - [E[x]]^2$$

$$= [y^3/3]_0^1 - (1/2)^2 = 1/3 - 1/4$$

$$= 1/12$$

RNG algorithms

Remember:

- Given knowledge of the algorithm used to create the numbers and its internal state (i.e. seed), you can predict all the numbers returned by subsequent calls to the algorithm, whereas with genuinely random numbers, knowledge of one number or an arbitrarily long sequence of numbers is of no use whatsoever in predicting the next number to be generated.
- Computer-generated "random" numbers are more properly referred to as *pseudorandom numbers*, and *pseudorandom sequences* of such numbers.

Mid-Square "generator"

MidSquare

Example:

X_0 = 7182 (seed)

X_0^2 = 51581124

➔ R_1 = 0.5811

X_1^2 = $(5811)^2$ = 33767721

➔ R_2 = 0.7677

etc.

Problem

Note: Cannot choose a seed that guarantees that the sequence will not degenerate and will have a long period. Also, zeros, once they appear, are carried in subsequent numbers.

Ex1: X_0 = 5197 (seed) X_0^2= 27008809

➔ R_1 = 0.0088 X_1^2= 00007744

➔ R_2 = 0.0077

Ex2: X_0 = 4500 (seed) X_0^2= 20250000

➔ R_1 = 0.2500 X_1^2= 06250000

➔ R_2 = 0.2500

Linear congruential generators

- Linear Congruential Method:
  - Basic generator
    $X_{n+1} = (a\, X_n + c) \pmod m$,
  - With modulus $m \geq 0$, multiplier $m > a > 0$, increment $0 \leq c < m$
  - Most natural choice for m is one that equals to the capacity of a computer integer type used.
  - $m = 2^b$ (binary machine), where b is the number of bits in the integer type.
  - $m = 10^d$ (decimal machine), where d is the number of digits in the integer type.
  - X_0 is called the seed

...

- The appearance of randomness is provided by performing modulo arithmetic or remaindering
- With X_n determined, we generate a corresponding real number as follows:
  $R_n=X_n/\text{float}(m)$ or $R_n=X_n/\text{float}(m+1)$
- When dividing by m, the values, R_n, are then distributed on [0,1).
- We desire uniformity, where any particular R_n is just as likely to appear as any other R_n, and the average of the R_n is very close to 0.5.
- Again: the next result depends upon only the previous integer – This is a characteristic of linear, congruential generators which minimizes storage requirements, but at the same time, imposes restrictions on the period.

LCGs

- Used in -
  - `rand()` function in C / C++ (libc)
  - `Java.util.Random`
  - ..
- The period is at most m
- For c=0 LCGs are also called multiplicative *congruential* random number generator

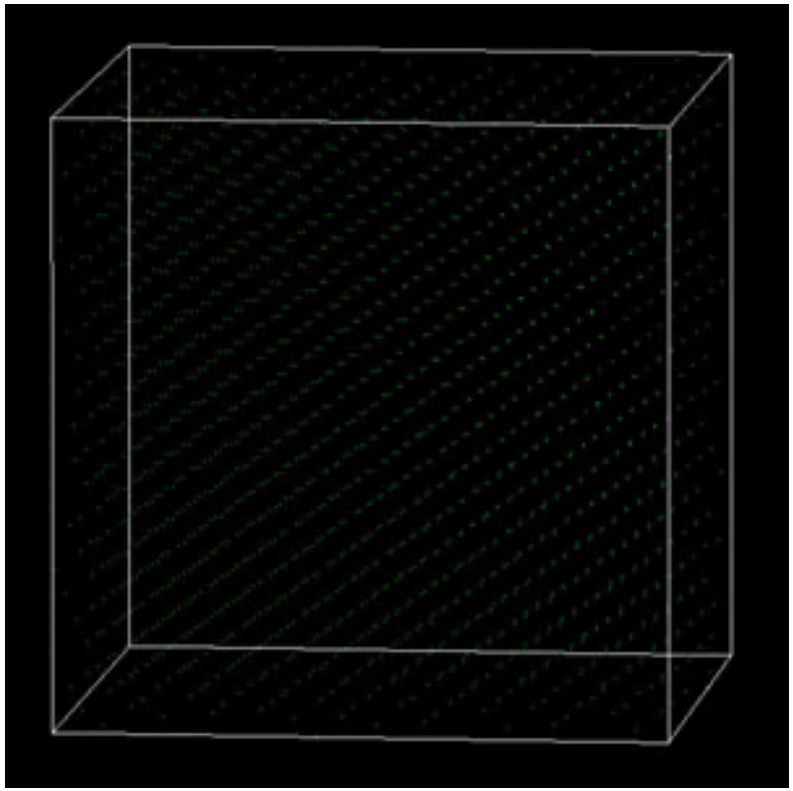
Advantages/Disadvantages

LCGs:

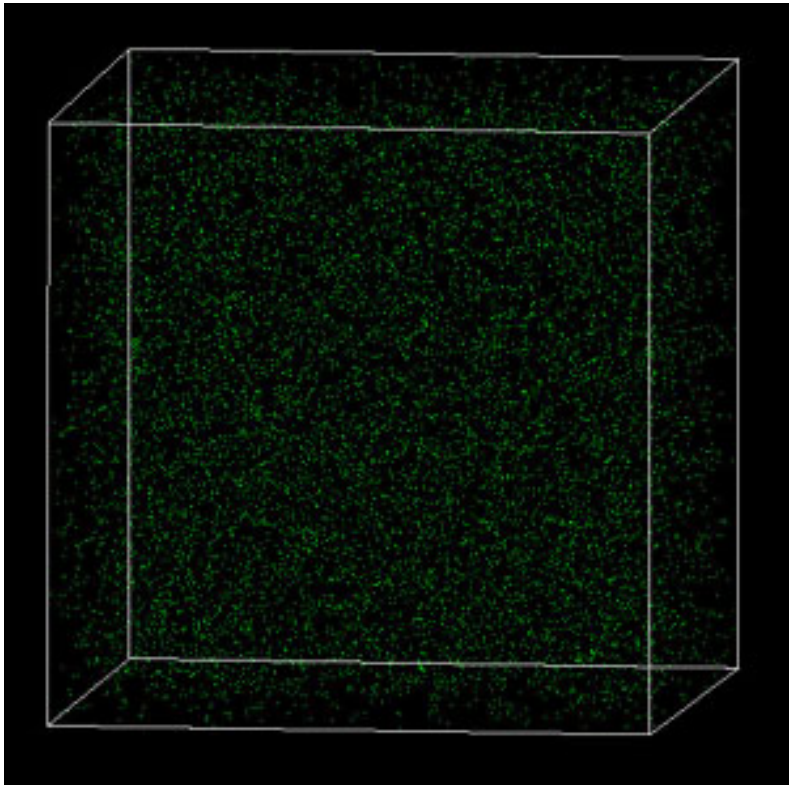
- fast and require minimal memory (typically 32 or 64 bits) to retain state
- valuable for simulating multiple independent streams
- should not be used for applications where high-quality randomness is critical
- not suitable for a Monte Carlo simulation because of the serial correlation (among other things)
- should also not be used for cryptographic applications
- LCGs tend to exhibit some severe defects:
  - For instance, if an LCG is used to choose points in an n-dimensional space, the points will lie on, at most, $m^{1/n}$ hyperplanes (Marsaglia's Theorem, developed by George Marsaglia). This is due to serial correlation between successive values of the sequence x_i. The spectral test, which is a simple test of an LCG's quality, is based on this fact.

LCG

- 3D points generated using a congruential RNG



Points fall on planes



Ideal random points

Other RNGs

- MT – **Mersenne Twister** – fast, negligible serial correlation, good for MC, cycle = 2^{19937}-1
- Blum Blum Shub (slow, not suitable for simulations, but for cryptography)
- ANSI X9.17
  - Based on triple-DES
- Capstone/Fortezza
- DSA (Digital Signature Specification)
- Yarrow-160
- Fortuna

- And many others

Tests for RNGs

1. ***Frequency test***. Uses the Kolmogorov-Smirnov or the chi-square test to compare the distribution of the set of numbers generated to a uniform distribution.
2. ***Runs test***. Tests the runs up and down or the runs above and below the mean by comparing the actual values to expected values. The statistic for comparison is the chi-square.
3. ***Autocorrelation test***. Tests the correlation between numbers and compares the sample correlation to the expected correlation of zero.
4. ***Gap test***. Counts the number of digits that appear between repetitions of a particular digit and then uses the Kolmogorov-Smirnov test to compare with the expected number of gaps.
5. ***Poker test***. Treats numbers grouped together as a poker hand. Then the hands obtained are compared to what is expected using the chi-square test.

- In testing for **uniformity**, the hypotheses are as follows:

  H_0: $x_i \sim U[0,1]$

  H_1: $x_i \neq U[0,1]$

The null hypothesis, H_0, reads that the numbers are distributed uniformly on the interval [0,1].

- In testing for **independence**, the hypotheses are as follows;

  H_0: $x_i \sim$ independently

  H_1: $x_i \neq$ independently

This null hypothesis, H_0, reads that the numbers are independent. Failure to reject the null hypothesis means that no evidence of dependence has been detected on the basis of this test. This does not imply that further testing of the generator for independence is unnecessary.

χ^2 tests

- Measure how well the presumed distribution (usually uniform) is represented.

- Algorithm for the test:
  - Divide the whole interval, within which the random number would be into finite number of bins (class intervals). Assume they have same size.
  - Count the number of random numbers within each interval and calculate the "expected" number of observations [(number of random numbers used) / (number of class intervals) for uniform intervals].
  - Calculate: $\chi^2 = \Sigma(i=1,m)(\text{observed}_i - \text{expected}_i)^2 / (\text{expected}_i)$
  - The value of χ^2 determines if the numbers generated represent a chosen distribution, by looking up in a table, some critical values of χ^2.

Run Tests (Up and Down)

Consider the 40 numbers; both the Kolmogorov-Smirnov and Chi-square would indicate that the numbers are uniformly distributed. But, not so.

0.08	0.09	0.23	0.29	0.42	0.55	0.58	0.72	0.89	0.91
0.11	0.16	0.18	0.31	0.41	0.53	0.71	0.73	0.74	0.84
0.02	0.09	0.30	0.32	0.45	0.47	0.69	0.74	0.91	0.95
0.12	0.13	0.29	0.36	0.38	0.54	0.68	0.86	0.88	0.91

The number and length of runs should approximately follow a normal distribution with appropriate means and variances.

Poker test

- based on the frequency with which certain digits are repeated.

  Example:

  0.255 0.577 0.331 0.414 0.828 0.909

  Note: a pair of like digits appear in each number generated.

  In 3-digit numbers, there are only 3 possibilities:

  - p(3 different digits) =

    p(2^{nd} diff. from 1^{st}) * p(3^{rd} diff. from 1^{st} & 2^{nd})

    = (0.9) (0.8) = 0.72
  - p(3 like digits) =

    p(2nd digit same as 1^{st}) ∗ p(3^{rd} digit same as 1^{st})

    = (0.1) (0.1) = 0.01
  - p(exactly one pair) = 1 - 0.72 - 0.01 = 0.27

Example

A sequence of 1000 three-digit numbers has been generated and an analysis indicates that 680 have three different digits, 289 contain exactly one pair of like digits, and 31 contain three like digits. Based on the poker test, are these numbers independent?

Combination, i	Observed Frequency, O_i	Expected Frequency, E_i	$\frac{(O_i - E_i)^2}{E_i}$
Three different digits	680	720	2.24
Three like digits	31	10	44.10
Exactly one pair	289	270	1.33
	1000	1000	47.65

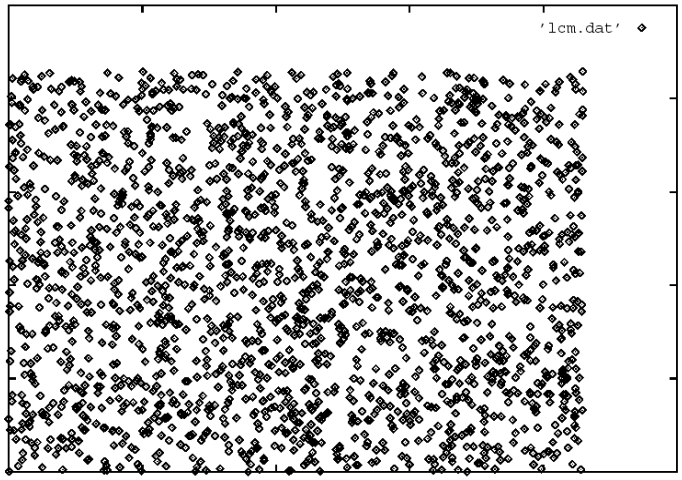
The appropriate degrees of freedom are one less than the number of class intervals. Since $\chi^2_{0.05,\,2} = 5.99 < 47.65$, the independence of the numbers is rejected on the basis of this test. (0.05 is the level of significance)

More on tests…

DIEHARD - http://stat.fsu.edu/pub/diehard/
NIST - http://csrs.nist.gov/rng

Qualitative tests

- Scatter Plots
  - Plot pairs of random numbers.
  - Clumps of numbers, gaps and patterns are easily visible.



- Random Walk
  - Divide the range of the RNG into equal intervals (e.g. 4 intervals for a random walk in two dimensions)
  - Generate a number, if number falls in:
    - First interval, increment X
    - Second interval, increment Y
    - Third interval, decrement X
    - Fourth interval, decrement Y
  - Generate t steps for a random walk for n walks
  - Calculate the means squared distance reached
  - Plot this distance against time
  - A plot for several values of t and distance should roughly be linear- otherwise the random numbers are not correctly distributed.

Physical (True?) RNG

- Radioactive decay
- Air Turbulence in disk drives
- Lava lamp
  e.g., https://en.wikipedia.org/wiki/Lavarand
- http://www.random.org
- Intel 8xx chipset

- Timing of keystrokes when a user enters a password.
- Measurement of timing skew between two systems timers:
  - A hardware timer
  - A software timer

More RNG resources

True Random Numbers

http://www.fourmilab.ch/hotbits/
http://www.robertnz.net/hwrng.htm
https://quantumnumbers.anu.edu.au/

Pseudo-random Number Generator documentation

https://en.cppreference.com/w/cpp/numeric/random.html
https://docs.python.org/3/library/random.html

Online PRNGs

https://leventozturk.com/engineering/random/

Next lecture:

- Introduction to CUDA
- Complex Ginzburg Landau equation