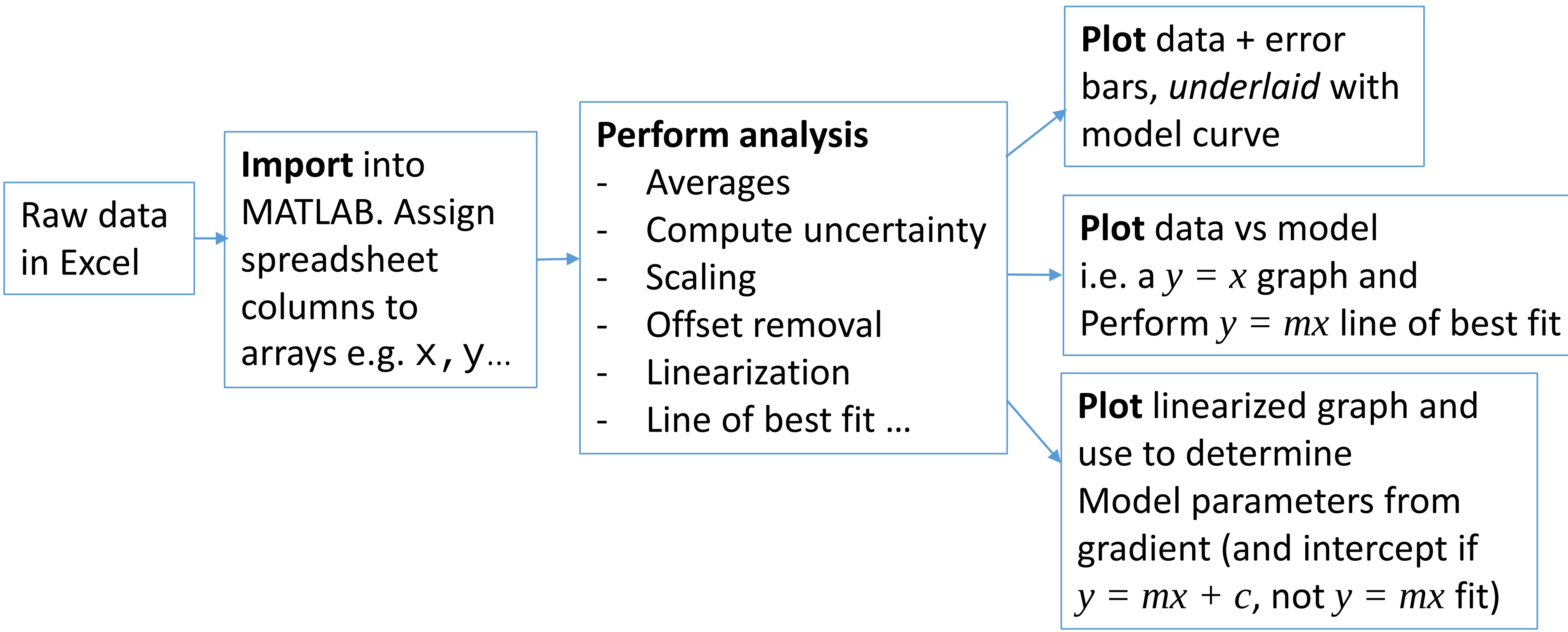
# Experimental data processing pipeline using MATLAB

Dr Andrew French. July 2020.



Art | Books | Comedy | Films | Fitness | Gastronomy | Maths

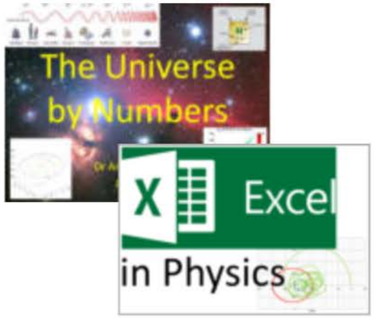
First created July 2012
Last updated May 2020

A-Level Chemistry organic synthesis

# Physics Revision Notes

Constants of Physics
constants of physics A3 classroom posters

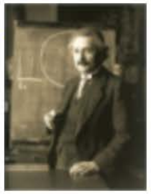
## Mechanics
## Materials
## Waves & Optics
## Thermodynamics
## Relativity




## Electromagnetism & electronics
## Nuclear, Particles & Quantum Physics
## Cosmology

What a lab report should contain

Faces of Physics



## Datalogging
## Questions

UNIVERSITY OF CAMBRIDGE

**My revision notes for the Cambridge University Experimental & Theoretical Physics course 1997-2001**

01 Part 1A Physics Revision 02 Experimental methods 03 Dynamics 04 Waves 05 Electromagnetism 06 Optics 07 Thermodynamics
08 Mathematical methods I 09 Mathematical methods II 10 Quantum Mechanics 11 Fluid dynamics 12 Relativity & Electrodynamics
13 Systems 14 Thermal & Statistical Physics 15 Solid State Physics 16 Nuclear physics 17 Particle physics
18 Dispersion relation for interfacial waves 19 Elastic waves in isotropic media
20 Gravitational astrophysics & cosmology 21 Structure & evolution of stars 22 Keplers Laws 23 General Relativity
24 Information theory 25 Natural and forced flows involving buoyancy

PhET

Notes

CLEAR LAYOUT ✓ CORRECT MATHS ✓ WORKINGS ✓ EXPLANATION ✓ ★★★★★ FANTASTIC WORK! ☺ ★★★★★

Feedback stamps for teachers

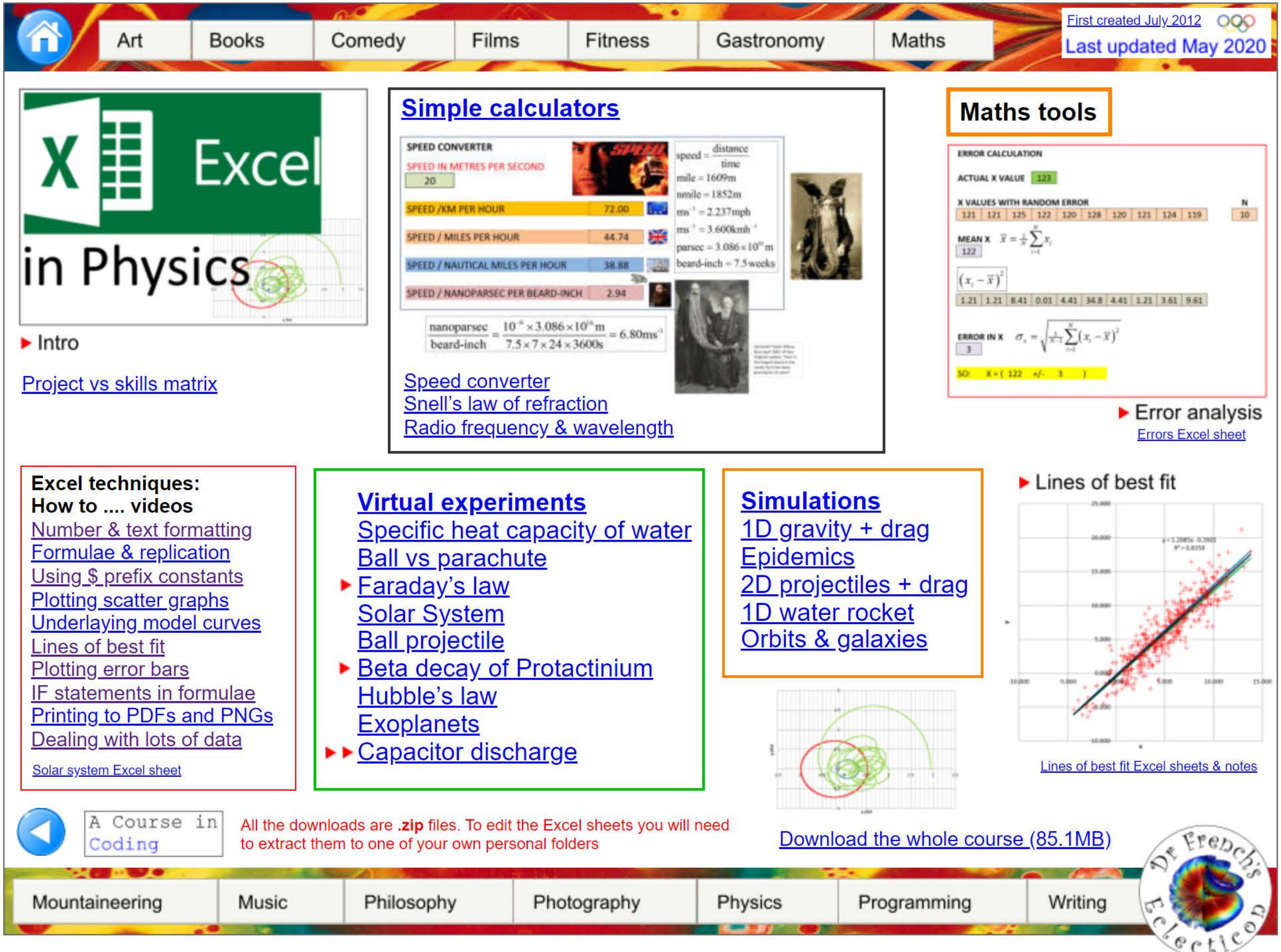
openstax Free Physics textbook

Mountaineering | Music | Philosophy | Photography | Physics | Programming | Writing

Dr French's Eclecticon

www.eclecticon.info/physics_notes.htm

re | eclecticon.info/physics_excel.htm




www.eclecticon.info/physics_excel.htm

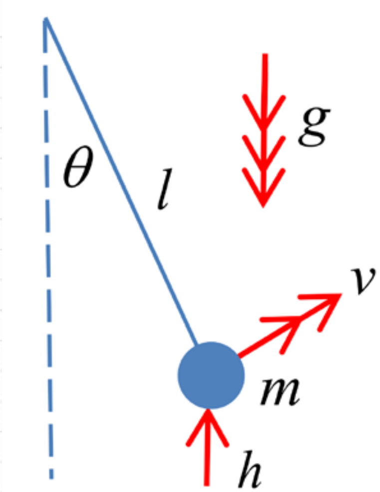
# Example using pendulum data

Raw data in Excel

| length of pendulum L /m | ten periods /s | ten periods /s | ten periods /s |
|---|---|---|---|
| 0.1 | 6.25 | 6.25 | 6.45 |
| 0.15 | 7.86 | 7.84 | 7.85 |
| 0.2 | 8.87 | 8.94 | 8.93 |
| 0.25 | 10.07 | 10.14 | 10.11 |
| 0.3 | 11.17 | 10.97 | 10.93 |
| 0.35 | 11.7 | 11.82 | 11.79 |
| 0.4 | 12.67 | 12.69 | 12.69 |
| 0.45 | 13.55 | 13.55 | 13.49 |
| 0.5 | 14.38 | 14.33 | 14.35 |
| 0.55 | 14.94 | 14.86 | 14.91 |
| 0.6 | 15.66 | 15.81 | 15.76 |
| 0.65 | 16 | 15.96 | 16 |
| 0.7 | 16.9 | 16.92 | 16.92 |
| 0.75 | 17.26 | 17.47 | 17.35 |
| 0.8 | 17.95 | 17.97 | 18.11 |
| 0.85 | 18.67 | 18.72 | 18.65 |
| 0.9 | 19.19 | 19.23 | 19.17 |
| 0.95 | 19.89 | 19.85 | 19.82 |
| 1 | 20.24 | 20.23 | 20.24 |
| 0.72 | 17.11 | 17.14 | 17.17 |

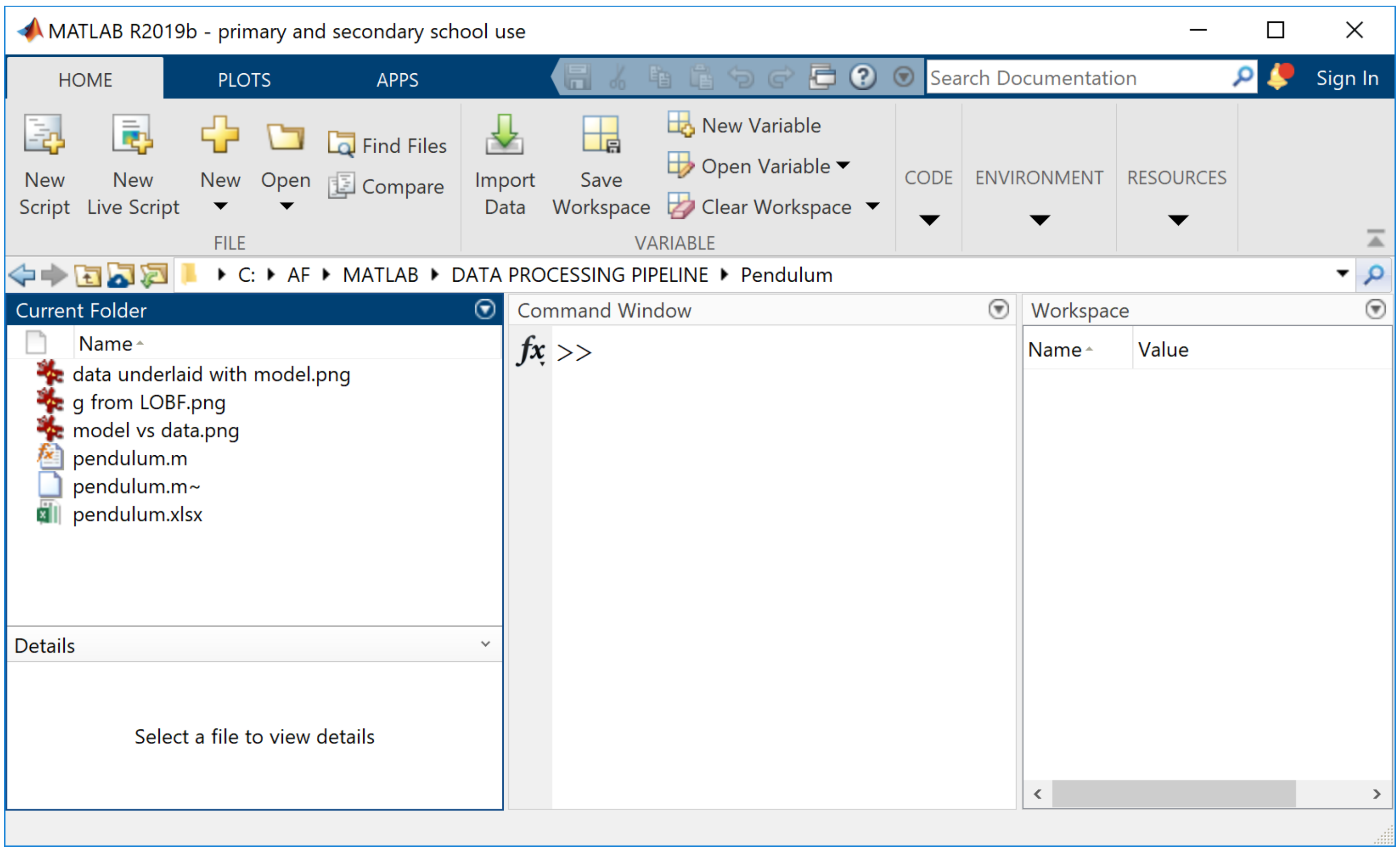
Data recorded manually or via a datalogger in an Excel spreadsheet.

If done manually, I would also recommend a basic graph **plot-as-you-go as a sanity check**, and to guide students to *take more data points* around changes of gradient such as peaks etc.



Run **pendulum.m** (right click, **run**) to execute a series of commands which constitute the rest of the data processing pipeline. The code can be modified for different experiments.

The key feature is that the code performs the process *automatically*, which can save considerable time when working on new data sets. MATLAB has the ability to perform useful analysis and create bespoke plots to a much higher standard than Excel. Students can focus on the *process,* in modifying the code, rather than the faff of dealing with Excel's defaults! However, I would always start with Excel as a first IT-based analysis.



Editor - C:\AF\MATLAB\DATA PROCESSING PIPELINE\Pendulum\pendulum.m

pendulum.m

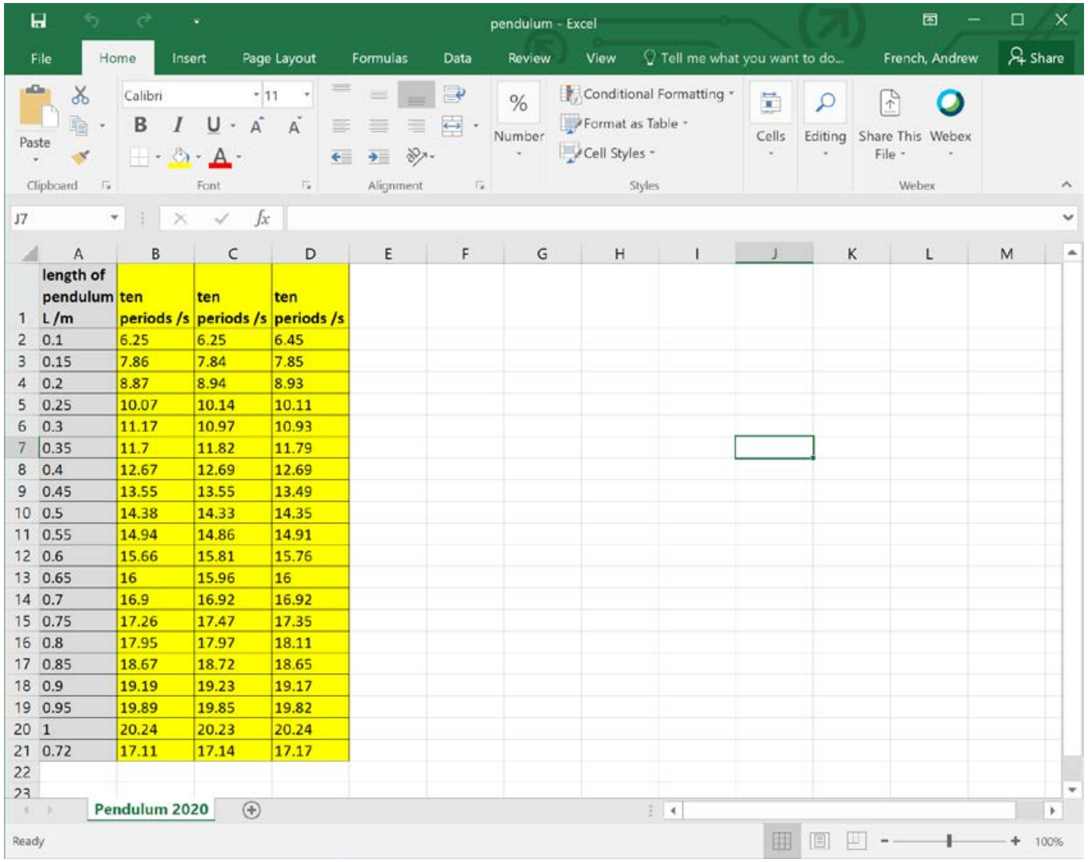
```
% Example physics data processing pipeline: #1 Pendulum
% * Load raw data from an Excel sheet pendulum.xlsx. This has columns of
%   pendulum length L /m, and three repeats of ten periods (10*T) /s.
% * Determine averages and errors
% * Plot y = 4*pi^2 * L vs x =T^2. Determine line of best fit (LOBF) and error,
%   and hence determine g from data. Compare to g = 9.81N/kg.
% * Plot T (data) vs 2*pi*sqrt(L/g) (with actual g). Perform LOBF.
% * Underlay T vs L data and underlay with T = 2*pi*sqrt(L/g) model.
%
% LAST UPDATED by Dr Andrew French. July 2020.

function pendulum

%% INPUTS %%

%Fontsize and marker size for graphs
fsize = 18; msize = 18;

%Set (fixed) error (in m) for pendulum length. Assume no systematic error.
Lerror = 0.01;

%Actual value of g /Nkg^-1
g = 9.81;

%Leave figures or close after printing?
close_after_print = 1;
```

Inside **pendulum.m**

.... It is a text file!

% means commentary

- Vital for humans
- Ignored by machines



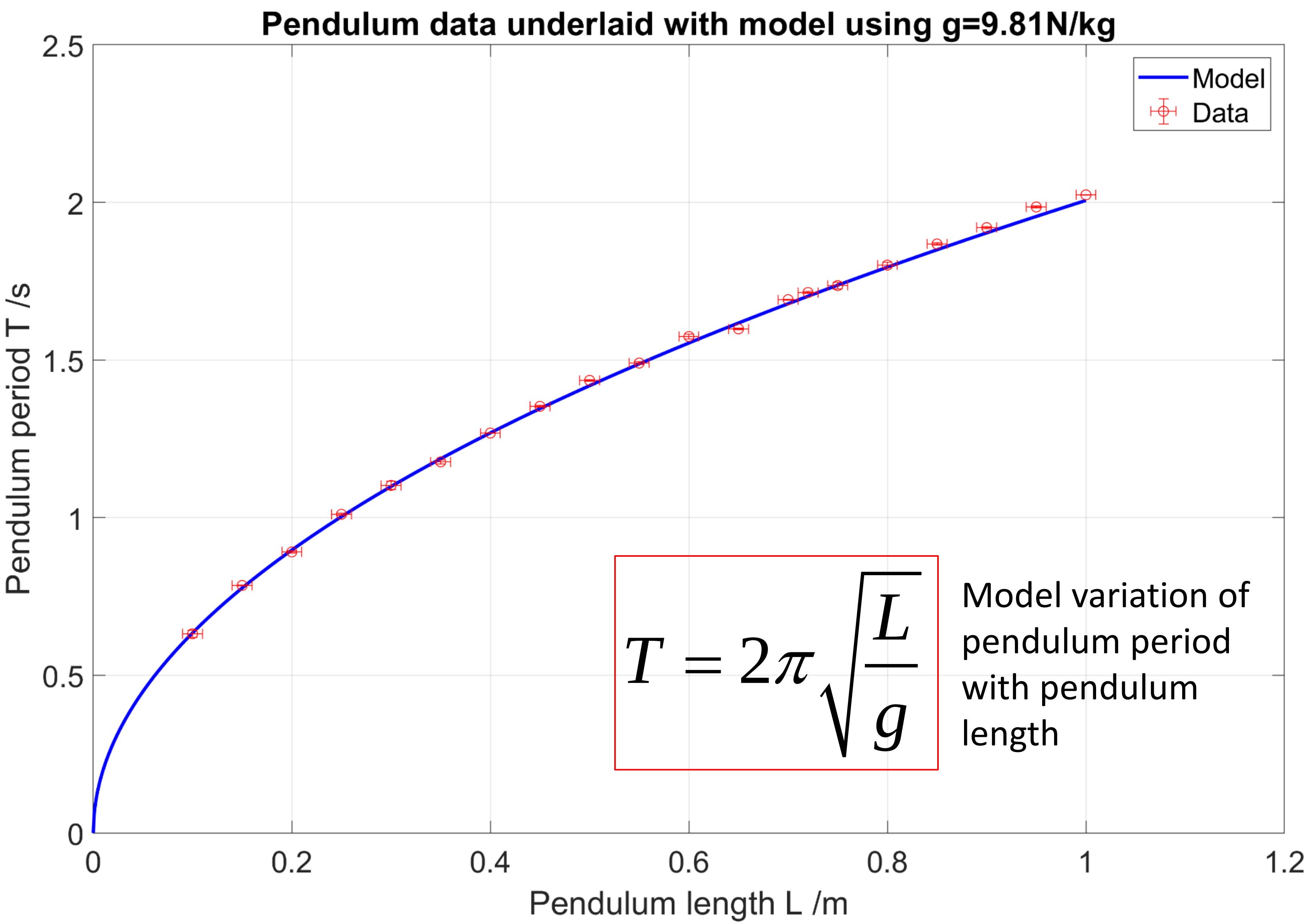
| length of pendulum L /m | ten periods /s | ten periods /s | ten periods /s |
|---|---|---|---|
| 0.1 | 6.25 | 6.25 | 6.45 |
| 0.15 | 7.86 | 7.84 | 7.85 |
| 0.2 | 8.87 | 8.94 | 8.93 |
| 0.25 | 10.07 | 10.14 | 10.11 |
| 0.3 | 11.17 | 10.97 | 10.93 |
| 0.35 | 11.7 | 11.82 | 11.79 |
| 0.4 | 12.67 | 12.69 | 12.69 |
| 0.45 | 13.55 | 13.55 | 13.49 |
| 0.5 | 14.38 | 14.33 | 14.35 |
| 0.55 | 14.94 | 14.86 | 14.91 |
| 0.6 | 15.66 | 15.81 | 15.76 |
| 0.65 | 16 | 15.96 | 16 |
| 0.7 | 16.9 | 16.92 | 16.92 |
| 0.75 | 17.26 | 17.47 | 17.35 |
| 0.8 | 17.95 | 17.97 | 18.11 |
| 0.85 | 18.67 | 18.72 | 18.65 |
| 0.9 | 19.19 | 19.23 | 19.17 |
| 0.95 | 19.89 | 19.85 | 19.82 |
| 1 | 20.24 | 20.23 | 20.24 |
| 0.72 | 17.11 | 17.14 | 17.17 |

**Import** into MATLAB. Assign spreadsheet columns to arrays e.g. x, y…

```
%% IMPORT EXCEL DATA & PREPARE L, T arrays %%

%Import data. Four columns. First is pendulum length, next three are
% ten periods /s.
[num,txt,raw] = xlsread( 'pendulum' );
L = num(:,1); T10_1 = num(:,2); T10_2 = num(:,3); T10_3 = num(:,4);

%Determine period T /s and the (unbiased estimator) of the error in T.
%The second argument of the std function uses the /(N-1) normalization
T = mean( [T10_1 , T10_2 , T10_3 ],2 )/10;
Terror = std( [T10_1 , T10_2 , T10_3 ],0,2 )/10;
```

```
%% ANALYSIS: Compare T vs L data to model T(L) with actual g %%

%Determine model prediction of T using actual value of g
Tmodel = 2*pi*sqrt( L/g );

%Determine model at a much finer grid of L values
LL = linspace( 0,max(L),1000 ); TTmodel = 2*pi*sqrt( LL/g );

%Plot model curve of T vs L
figure('name','model vs data','color',[1 1 1],...
    'units','normalized','position',[0.05, 0.05, 0.9, 0.85]);
plot( LL, TTmodel,'b-','linewidth',2 ); hold on;
set( gca, 'fontsize',fsize ); grid on;

%Plot data error bars
x = L; y = T; yneg = Terror; ypos = Terror;
xneg = Lerror*ones(size(L)); xpos = Lerror*ones(size(L));
errorbar( x,y,yneg,ypos,xneg,xpos,'o','color','r');

%Graph labels etc
xlabel('Pendulum length L /m'); ylabel('Pendulum period T /s');
title('Pendulum data underlaid with model using g=9.81N/kg');
legend({'Model','Data'});

%Print a PNG file
print( gcf, 'data underlaid with model.png','-r300','-dpng' );
if close_after_print==1; close(gcf); end
```

**Plot** data + error bars, *underlaid* with model curve




$$T = 2\pi\sqrt{\frac{L}{g}}$$

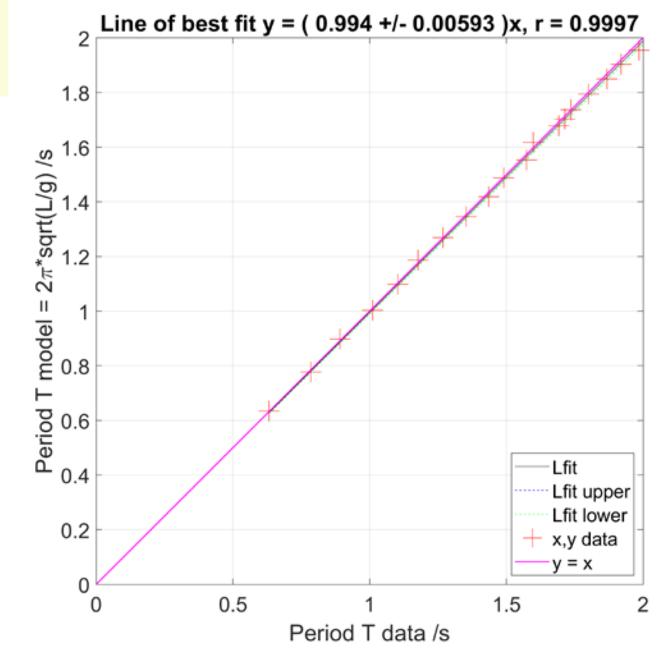
Model variation of pendulum period with pendulum length

```
%% ANALYSIS: Determine line of best fit of the form y = m*x between T data and T model
% For 100% correlation, the gradient m = 1 and product-moment correlation coefficient r = 1.
y = Tmodel; x = T; [yfit,xfit,r,m,dm,yupper,ylower,s] = bestfit( x,y );

%Plot line of best fit
xlabel_str = 'Period T data /s';
ylabel_str = 'Period T model = 2\pi*sqrt(L/g) /s';
plot_LOBF( x,y, yfit,xfit,r,m,dm,yupper,ylower,...
    fsize, msize, xlabel_str, ylabel_str );

%Plot y = x for visual check
plot( [0;x], [0;x], 'm-', 'linewidth', 1 );
legend({'Lfit', 'Lfit upper','Lfit lower','x,y data','y = x'},...
    'location','southeast'); axis equal; axis tight;

%Set sensible x,y limits to include origin
xlimits = get(gca,'xlim'); set( gca, 'xlim',[0,round( xlimits(2) )] );
ylimits = get(gca,'ylim'); set( gca, 'ylim',[0,round( ylimits(2) )] );
print( gcf, 'model vs data.png','-r300','-dpng' );
if close_after_print==1; close(gcf); end
```

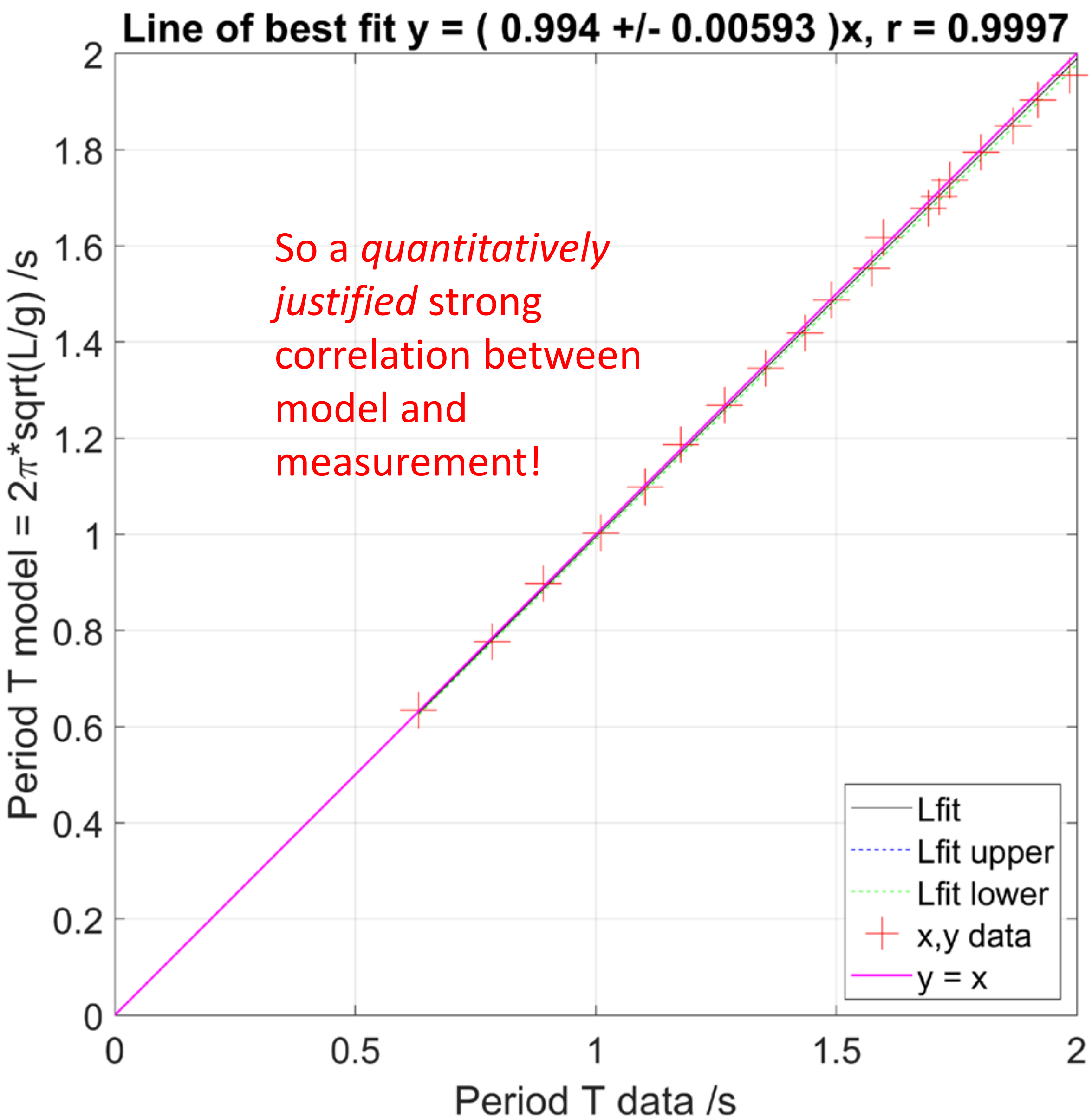
These are *sub-functions* which perform the line of best fit and associated plots. They should be generic, regardless of the dataset.

$$T = 2\pi\sqrt{\frac{L}{g}}$$

Model variation of pendulum period $T$ with pendulum length $L$



**Plot** data vs model
i.e. a $y = x$ graph and
Perform $y = mx$ line of best fit




$$T = 2\pi\sqrt{\frac{L}{g}}$$

Model variation of pendulum period with pendulum length

If you don't need to find parameters from data, **simply comparing model vs measurement** is a very clear first quantitative analysis

```
%% ANALYSIS: Determine g from data %%

%Determine y = 4*pi^2*L and x = T^2
x = T.^2; y = 4*pi^2 * L;

%Determine upper and lower values for error bar calculation
x_upper = ( T + Terror ).^2;  x_lower = ( T - Terror ).^2;
y_upper = 4*pi^2 *( L + Lerror );  y_lower = 4*pi^2 *( L - Lerror );

% Determine line of best fit of the form y = m*x.
% Gradient m is g in this case
[yfit,xfit,r,m,dm,yupper,ylower,s] = bestfit(x,y);

%Plot line of best fit
xlabel_str = '(T/s)^2'; ylabel_str = '4\pi^2*(L/m)';
plot_LOBF( x,y, yfit,xfit,r,m,dm,yupper,ylower,...
    fsize, 0.001, xlabel_str, ylabel_str );

%Plot what the line should be, given the actual value of g
plot( x, g*x, 'm-','linewidth',1 );

%Plot data error bars
yneg = y - y_lower; ypos = y_upper - y; xneg = x - x_lower; xpos = x_upper - x;
errorbar( x,y,yneg,ypos,xneg,xpos,'o','color','r');

%Add a legend
legend({'Lfit', 'Lfit upper','Lfit lower','',...
    'Using g=9.81N/kg','x,y data'},'location','southeast' )
```

But if you *do* need to find parameters, **linearize**, and then perform a line of best fit

$$T = 2\pi\sqrt{\frac{L}{g}}$$

$$\therefore \underbrace{4\pi^2 L}_{y} = g\,\underbrace{T^2}_{x}$$

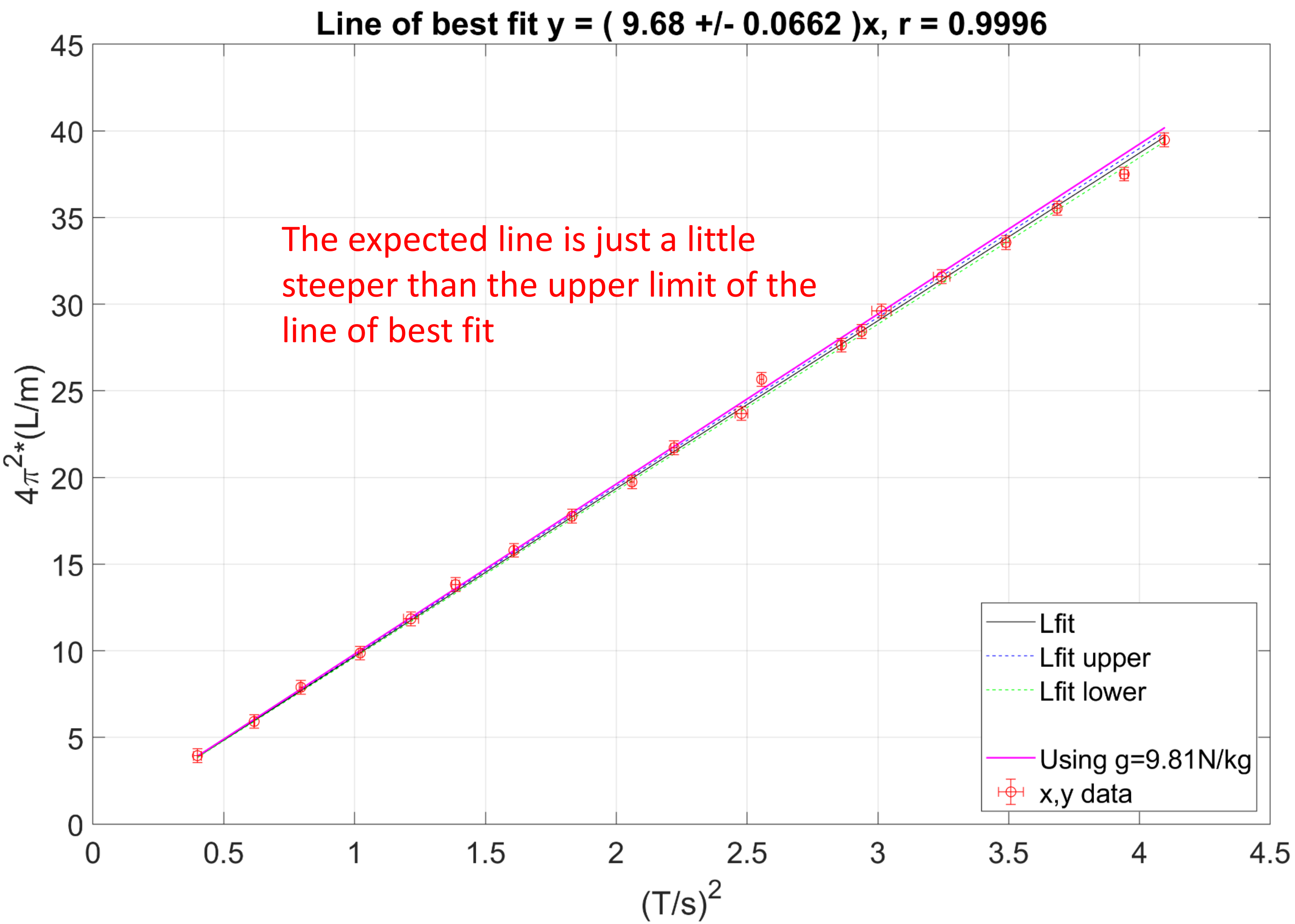
$$\Rightarrow y = gx$$

So $g$ is the gradient of the $x,y$ graph in our case

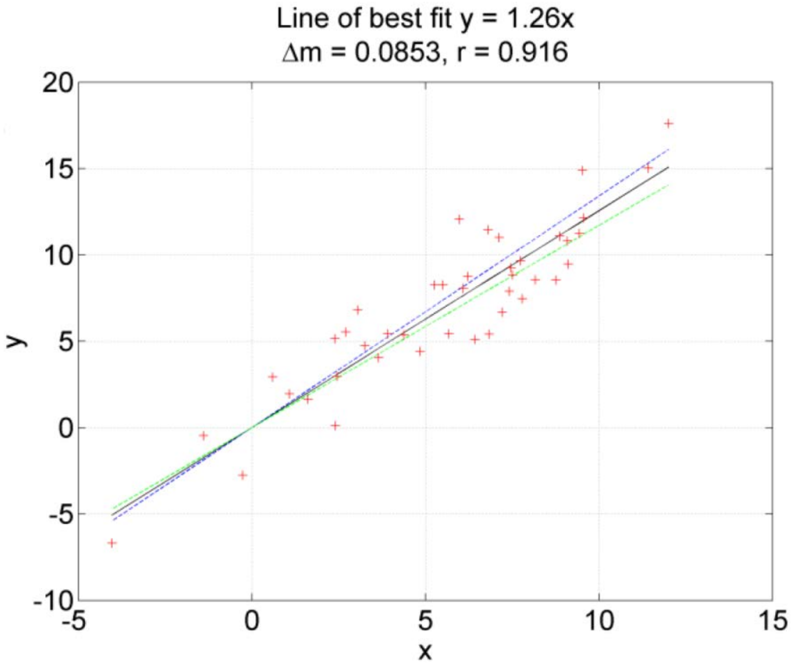
**Plot** linearized graph and use to determine model parameters from gradient (and intercept if $y = mx + c$, not a $y = mx$ fit)

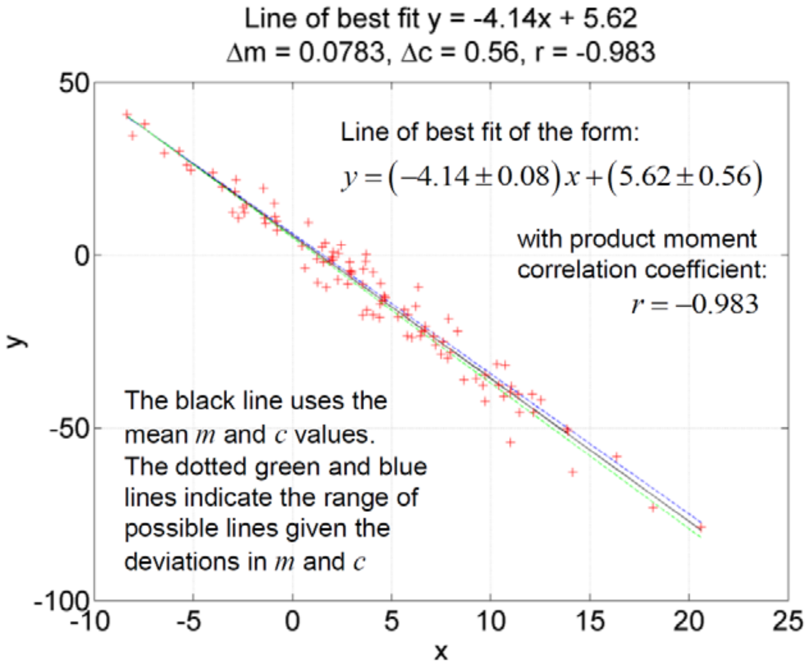
$$T = 2\pi\sqrt{\frac{L}{g}} \quad \therefore \underbrace{4\pi^2 L}_{y} = g \underbrace{T^2}_{x} \quad \Rightarrow y = gx$$

In our case, our gradient (and hence calculated $g$) is systematically lower than what it should be.



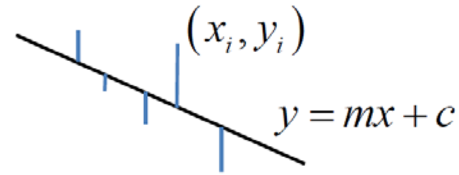
# Lines of best fit using MATLAB



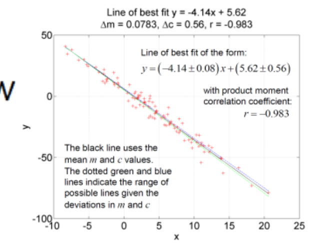



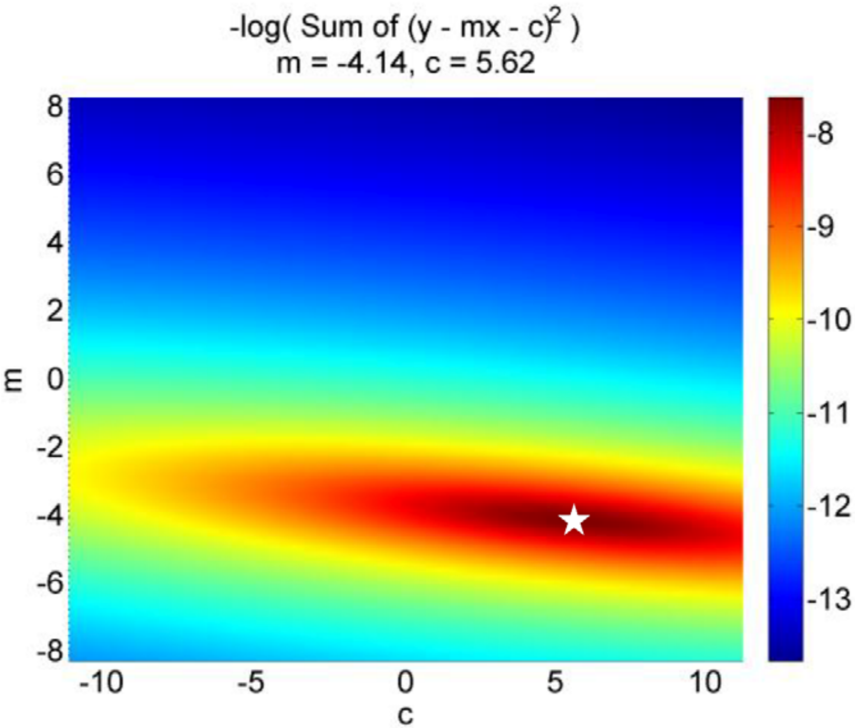
To determine the line of best fit*, let us sum the *squared* deviations of $(x,y)$ from the line of best fit.

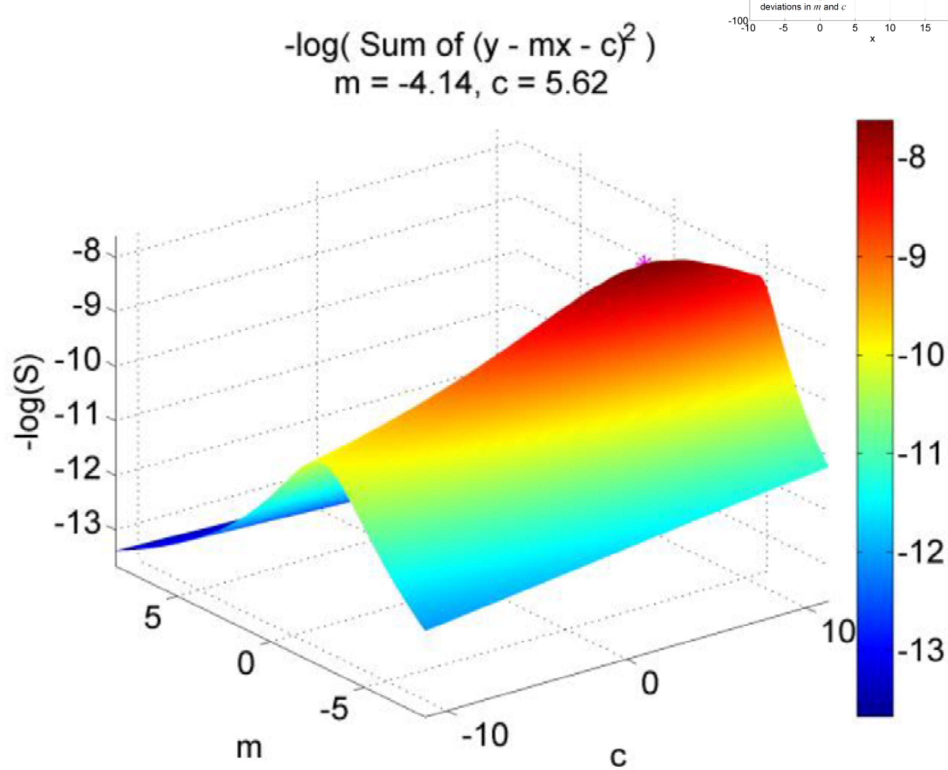
$$S = \sum_{i=1}^{N} (y_i - mx_i - c)^2$$




Using the (*negatively correlated*) data on the right, we can plot a surface of $S$ vs $m$ and $c$ values. We can see this has a **minimum** at a particular $(m,c)$ coordinate. (Note for clarity the plots below are of $-\log S$, so the $(m,c)$ coordinate corresponds to the peak, i.e. maximum, instead).




The idea is to find optimum $m$ and $c$ values (or just $m$) given $x,y$ data

## Correlation & Linear Regression

Perhaps the most important analytical tool in the physical sciences is the ability to quantify the validity of a model relating a set of measurable parameters. The idea is as follows:

(1) Rearrange the model in such a way that it becomes a *linear equation* of the form $y = mx + c$

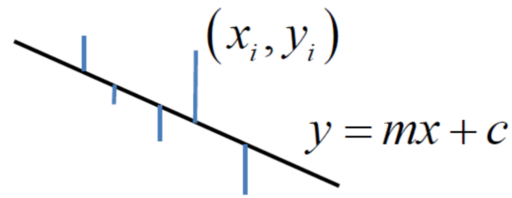
(2) Plot experimental $(x,y)$ data on a graph and determine the **line of best fit** through the data.

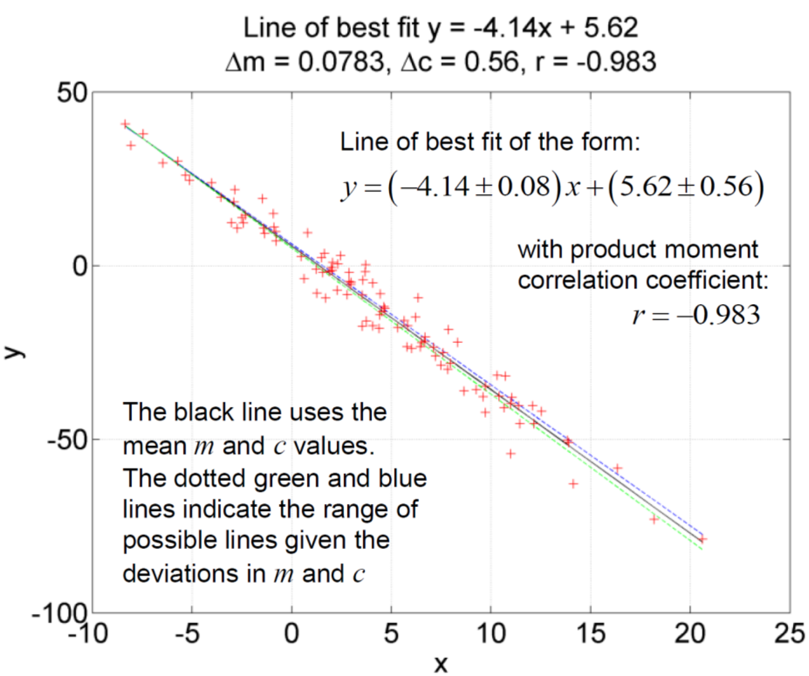
(3) Determine *gradient* $m$ and *vertical intercept* $c$ from the line of best fit.

(4) Determine the standard deviation of both gradient $m$ and intercept $c$, and a quantitative measure of how good the fit is (this is called the **product moment correlation coefficient**).

To determine the line of best fit*, let us sum the *squared* deviations of $(x,y)$ from the line of best fit.

$$S = \sum_{i=1}^{N} \left( y_i - mx_i - c \right)^2$$




We can find $S$ given a range of $m$ and $c$ values. Which pairing results in the *minimum* value of $S$?

**Summary: Line of Best Fit for:** $y = mx + c$

$N$ data point pairs $(x,y)$

$$\bar{x} = \tfrac{1}{N}\sum_{i=1}^{N} x_i, \quad y = \tfrac{1}{N}\sum_{i=1}^{N} y_i, \quad \overline{x^2} = \tfrac{1}{N}\sum_{i=1}^{N} x_i^2, \quad \overline{y^2} = \tfrac{1}{N}\sum_{i=1}^{N} y_i^2, \quad \overline{xy} = \tfrac{1}{N}\sum_{i=1}^{N} x_i y_i$$

$$V[x] = \overline{x^2} - \bar{x}^2, \quad V[y] = \overline{y^2} - \bar{y}, \quad \mathrm{cov}[x,y] = \overline{xy} - \bar{x}\bar{y}$$

$$m = \frac{\overline{xy} - \bar{y}\bar{x}}{\overline{x^2} - \bar{x}^2} = \frac{\mathrm{cov}[x,y]}{V[x]}, \qquad c = \bar{y} - m\bar{x}$$

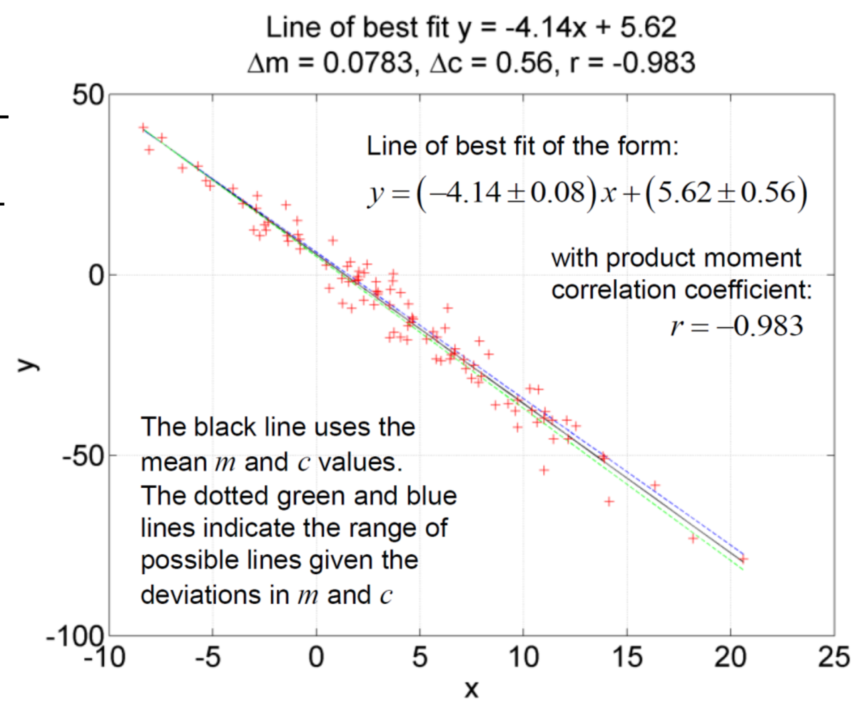
$$r = \frac{\mathrm{cov}[x,y]}{\sqrt{V[x]V[y]}}$$

Product moment correlation coefficient

$$\Delta m = \frac{s}{\sqrt{N}}\frac{1}{\sqrt{V[x]}}, \qquad \Delta c = \frac{s}{\sqrt{N}}\sqrt{1 + \frac{\bar{x}^2}{V[x]}}$$

$$s = \sqrt{\frac{1}{N-2}\sum_{i=1}^{N}\left(y_i - mx_i - c\right)^2}$$

Errors in gradient and intercept



**Summary: Line of Best Fit for:** $y = mx$

$N$ data point pairs $(x,y)$

$$\overline{x} = \tfrac{1}{N}\sum_{i=1}^{N} x_i, \quad y = \tfrac{1}{N}\sum_{i=1}^{N} y_i, \quad \overline{x^2} = \tfrac{1}{N}\sum_{i=1}^{N} x_i^2, \quad \overline{y^2} = \tfrac{1}{N}\sum_{i=1}^{N} y_i^2, \quad \overline{xy} = \tfrac{1}{N}\sum_{i=1}^{N} x_i y_i$$

$$V[x] = \overline{x^2} - \overline{x}^2, \quad V[y] = \overline{y^2} - \overline{y}, \quad \mathrm{cov}[x,y] = \overline{xy} - \overline{xy}$$

$$m = \frac{\overline{xy}}{\overline{x^2}}$$

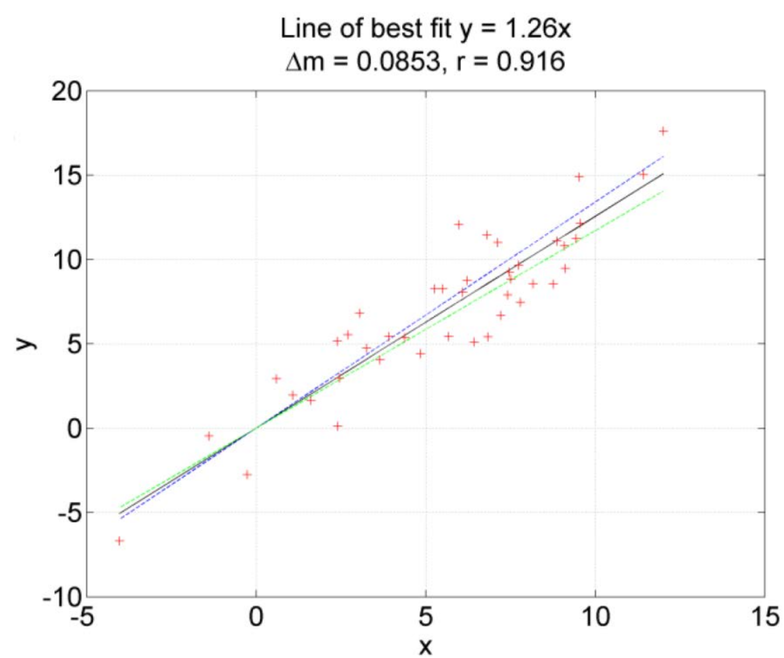
$$r = \frac{\mathrm{cov}[x,y]}{\sqrt{V[x]V[y]}}$$

Product moment correlation coefficient

$$\Delta m = \frac{s}{\sqrt{N}} \frac{1}{\sqrt{V[x]}}$$

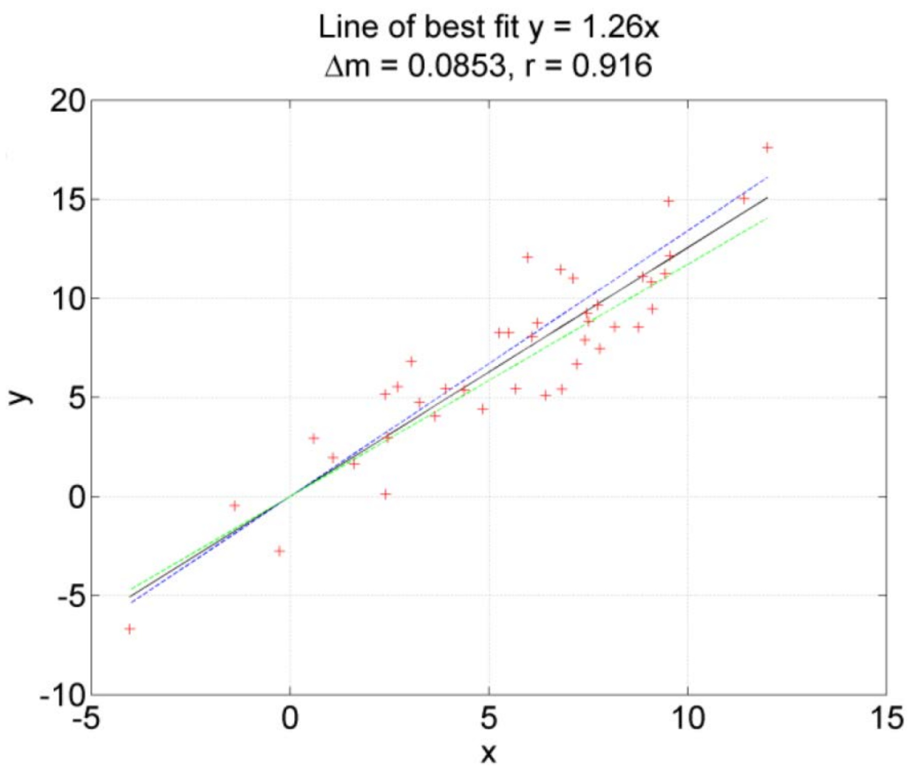
Error in gradient

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(y_i - mx_i)^2}$$



```
%Line of best fit function yfit = m*x, with product moment correlation
%coefficient r
function [yfit,xfit,r,m,dm,yupper,ylower,s] = bestfit(x,y)

%Find any x or y values that are NaN or Inf
ignore = isnan(abs(x)) | isnan(abs(y)) | isinf(abs(x)) | isinf(abs(y));
x(ignore) = []; y(ignore) = [];

%Compute line of best fit
xbar = mean(x); ybar = mean(y); xybar = mean(x.*y);
xxbar = mean(x.^2 ); yybar = mean(y.^2 );
Vx = xxbar - xbar^2; Vy = yybar - ybar^2;
COVxy = xybar - xbar*ybar;
m = xybar/xxbar; r = COVxy/sqrt( Vx*Vy );
[x,i] = sort(x); y = y(i);
yfit = m*x; xfit = x;

%Compute error in gradient m
n = length(x); s = sqrt( (1/(n-1))*sum( (y - yfit).^2 ) );
dm = s/sqrt(n*Vx);

%Determine envelope lines
yupper = (m+dm)*x; ylower = (m-dm)*x;
```

```
%Plot line of best fit
function plot_LOBF( x,y, yfit,xfit,r,m,dm,yupper,ylower,...
    fsize, msize, xlabel_str, ylabel_str  )
figure('name','line of best fit','color',[1 1 1],...
    'units','normalized','position',[0.05, 0.05, 0.9, 0.85]);
plot( xfit, yfit, 'k-', xfit, yupper, 'b--', xfit, ylower, 'g--' );
hold on; plot( x,y,'r+','markersize',msize); set(gca, 'fontsize',fsize );
xlabel(xlabel_str); ylabel(ylabel_str); grid on;
title( {['Line of best fit y = ( ',num2str(m,3),' +/- ',num2str(dm,3),...
    ' )x, r = ',num2str(r,4)] } );
```




Note this generic plot code is for a $y = mx$ line of best fit