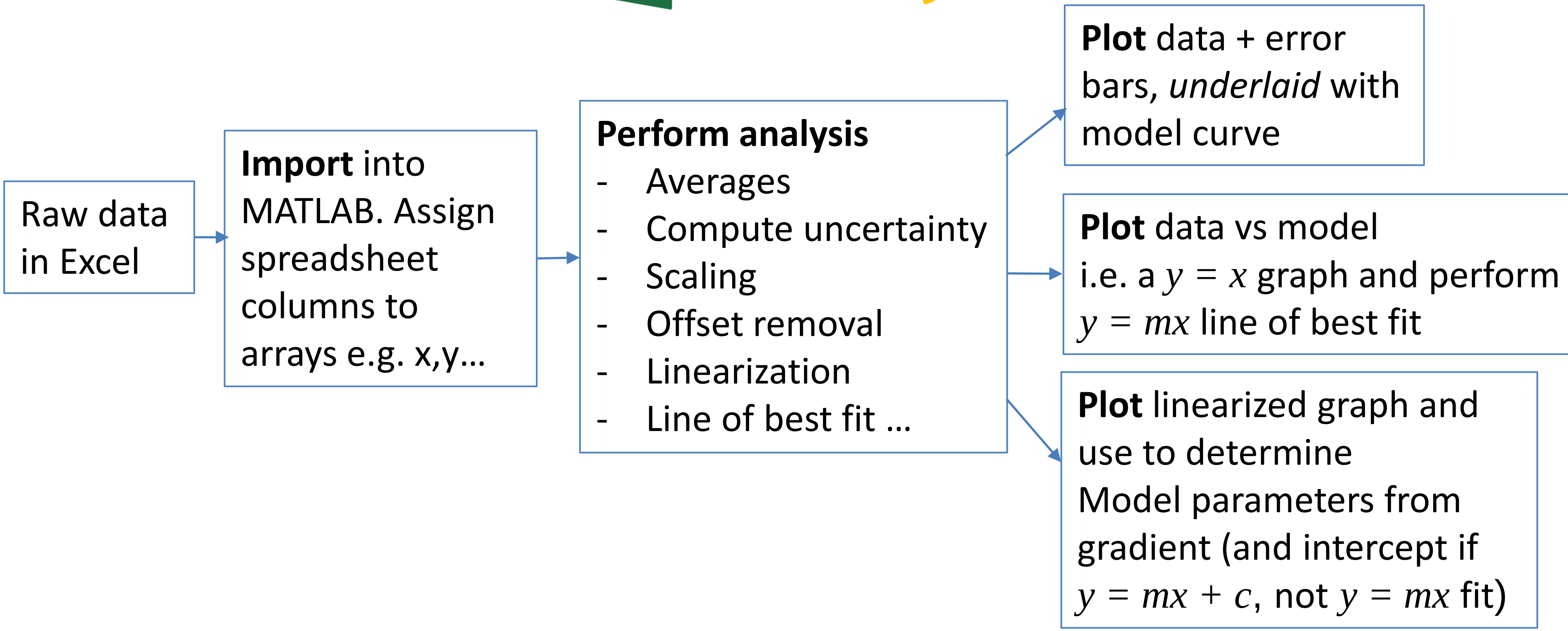
BPhO Computational Challenge

Data processing pipeline

Dr Andrew French.
December 2023.

BPhO British Physics Olympiad

Experimental data processing pipeline using Excel & MATLAB



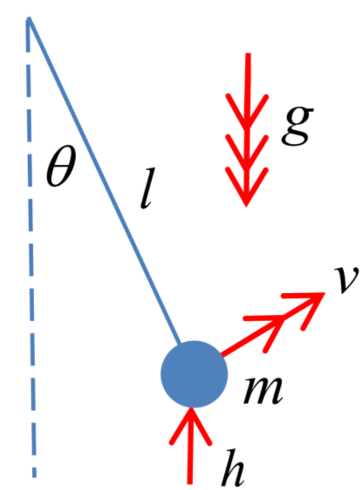
Example using pendulum data

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.

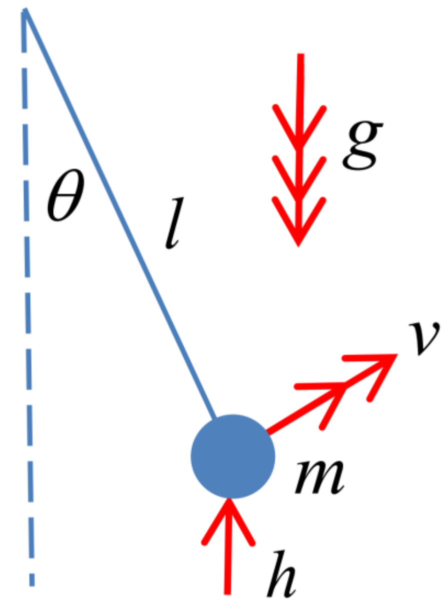


Pendulum 2020

$$ml\ddot{\theta} = -mg\sin\theta \approx -mg\theta \quad \text{NEWTON II}$$

$$\therefore \ddot{\theta} = -\frac{g}{l}\theta$$

T is the pendulum period



$$\text{SHM:} \quad \ddot{\theta} = -\left(\frac{2\pi}{T}\right)^2\theta = -\omega^2\theta$$

$$\theta(t) = \theta_0\cos\left(\frac{2\pi t}{T}\right) = \theta_0\cos\omega t$$

$$\therefore T = 2\pi\sqrt{\frac{l}{g}} \quad T^2 = 4\pi^2\frac{l}{g} \quad \underbrace{g}\,\underbrace{T^2}_{x} = \underbrace{4\pi^2 l}_{y} \quad \text{i.e.} \quad y = gx$$

Simple Harmonic Motion (SHM) of a pendulum

* Ignore air resistance

* Small angle approximation i. $\theta \ll 1\text{radian}$

MEASURING g VIA A PENDULUM 03/06/2020

length of pendulu m L /m	ten periods /s	ten periods /s	ten periods /s	ten periods /s	period T /s	x = T^2	y = 4*pi^2 * L
0.1	6.25	6.25	6.45	6.32	0.63	0.40	3.95
0.15	7.86	7.84	7.85	7.85	0.79	0.62	5.92
0.2	8.87	8.94	8.93	8.91	0.89	0.79	7.90
0.25	10.07	10.14	10.11	10.11	1.01	1.02	9.87
0.3	11.17	10.97	10.93	11.02	1.10	1.22	11.84
0.35	11.7	11.82	11.79	11.77	1.18	1.39	13.82
0.4	12.67	12.69	12.69	12.68	1.27	1.61	15.79
0.45	13.55	13.55	13.49	13.53	1.35	1.83	17.77
0.5	14.38	14.33	14.35	14.35	1.44	2.06	19.74
0.55	14.94	14.86	14.91	14.90	1.49	2.22	21.71
0.6	15.66	15.81	15.76	15.74	1.57	2.48	23.69
0.65	16	15.96	16	15.99	1.60	2.56	25.66
0.7	16.9	16.92	16.92	16.91	1.69	2.86	27.63
0.75	17.26	17.47	17.35	17.36	1.74	3.01	29.61
0.8	17.95	17.97	18.11	18.01	1.80	3.24	31.58
0.85	18.67	18.72	18.65	18.68	1.87	3.49	33.56
0.9	19.19	19.23	19.17	19.20	1.92	3.69	35.53
0.95	19.89	19.85	19.82	19.85	1.99	3.94	37.50
1	20.24	20.23	20.24	20.24	2.02	4.10	39.48
0.72	17.11	17.14	17.17	17.14	1.71	2.94	28.42

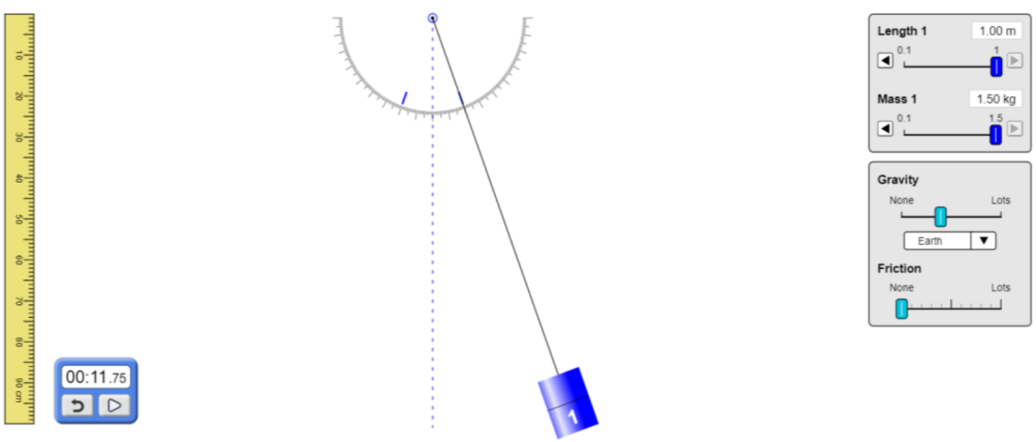
$$ml\ddot{\theta} = -mg\sin\theta \approx -mg\theta \quad \text{NEWTON II}$$

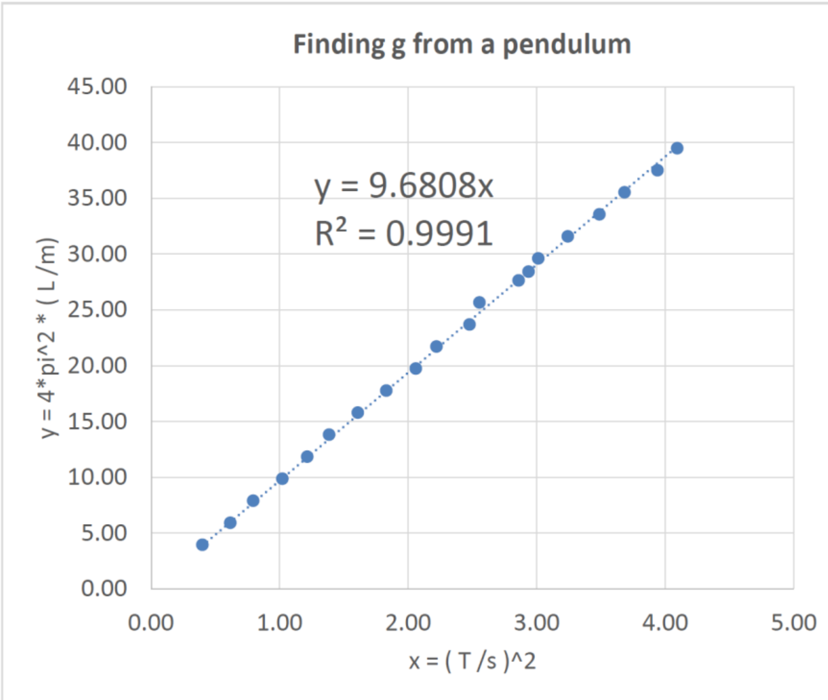
$$\therefore \ddot{\theta} = -\frac{g}{l}\theta$$

$$\text{SHM:} \quad \ddot{\theta} = -\left(\frac{2\pi}{T}\right)^2 \theta = -\omega^2\theta$$

$$\theta(t) = \theta_0 \cos\left(\frac{2\pi t}{T}\right) = \theta_0 \cos\omega t$$

$$\therefore T = 2\pi\sqrt{\frac{l}{g}} \quad T^2 = 4\pi^2\frac{l}{g} \quad g\underbrace{T^2}_{x} = \underbrace{4\pi^2 l}_{y} \quad \text{i.e.} \quad y = gx$$





Initial angle = 20 degrees

To complete, underlay (Period vs pendulum length) data with a model curve

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

MEASURING g VIA A PENDULUM 03/06/2020

length of pendulu m L /m	ten periods /s	ten periods /s	ten periods /s	ten periods /s	period T /s	x = T^2	y = 4*pi^2 * L
0.1	6.25	6.25	6.45	6.32	0.63	0.40	3.95
0.15	7.86	7.84	7.85	7.85	0.79	0.62	5.92
0.2	8.87	8.94	8.93	8.91	0.89	0.79	7.90
0.25	10.07	10.14	10.11	10.11	1.01	1.02	9.87
0.3	11.17	10.97	10.93	11.02	1.10	1.22	11.84
0.35	11.7	11.82	11.79	11.77	1.18	1.39	13.82
0.4	12.67	12.69	12.69	12.68	1.27	1.61	15.79
0.45	13.55	13.55	13.49	13.53	1.35	1.83	17.77
0.5	14.38	14.33	14.35	14.35	1.44	2.06	19.74
0.55	14.94	14.86	14.91	14.90	1.49	2.22	21.71
0.6	15.66	15.81	15.76	15.74	1.57	2.48	23.69
0.65	16	15.96	16	15.99	1.60	2.56	25.66
0.7	16.9	16.92	16.92	16.91	1.69	2.86	27.63
0.75	17.26	17.47	17.35	17.36	1.74	3.01	29.61
0.8	17.95	17.97	18.11	18.01	1.80	3.24	31.58
0.85	18.67	18.72	18.65	18.68	1.87	3.49	33.56
0.9	19.19	19.23	19.17	19.20	1.92	3.69	35.53
0.95	19.89	19.85	19.82	19.85	1.99	3.94	37.50
1	20.24	20.23	20.24	20.24	2.02	4.10	39.48
0.72	17.11	17.14	17.17	17.14	1.71	2.94	28.42

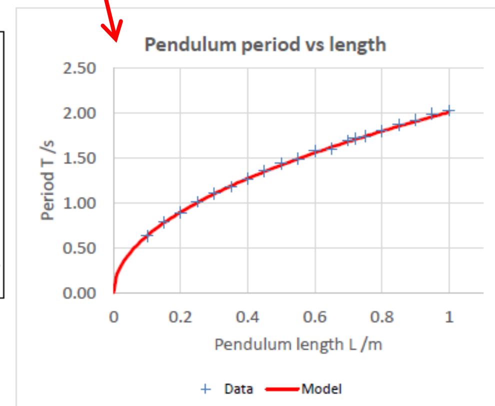
$$ml\ddot{\theta} = -mg\sin\theta \approx -mg\theta \quad \text{NEWTON II}$$

$$\therefore \ddot{\theta} = -\frac{g}{l}\theta$$

$$\text{SHM:} \quad \ddot{\theta} = -\left(\frac{2\pi}{T}\right)^2\theta = -\omega^2\theta$$

$$\theta(t) = \theta_0\cos\left(\frac{2\pi t}{T}\right) = \theta_0\cos\omega t$$

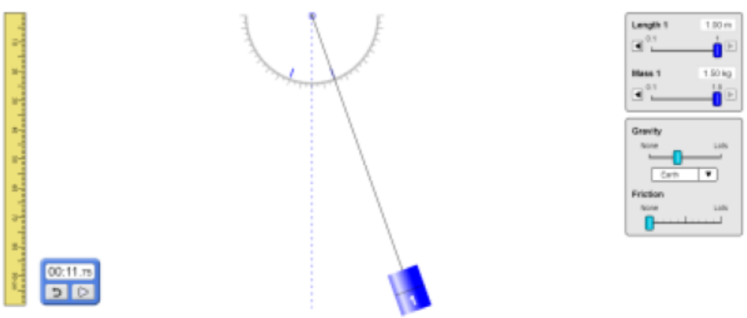
$$\therefore T = 2\pi\sqrt{\frac{l}{g}} \quad T^2 = 4\pi^2\frac{l}{g} \quad g\underbrace{T^2}_{x} = \underbrace{4\pi^2 l}_{y} \quad \text{i.e.} \quad y = gx$$



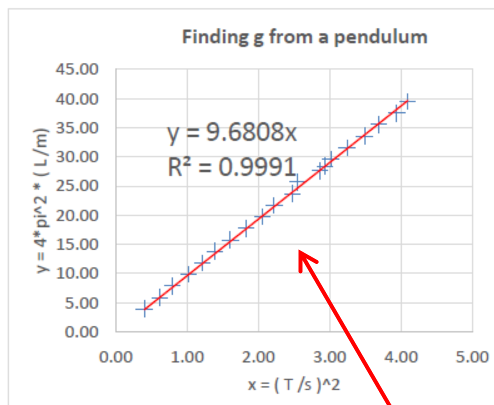
Using g = 9.81N/kg

MODEL curve

L /m	T /s
0.000	0.000
0.010	0.201
0.020	0.284
0.030	0.347
0.040	0.401
0.050	0.449
0.060	0.491
0.070	0.531
0.080	0.567
0.090	0.602
0.100	0.634
0.110	0.665
0.120	0.695
0.130	0.723
0.140	0.751
0.150	0.777
0.160	0.802
0.170	0.827
0.180	0.851
0.190	0.874
0.200	0.897
0.210	0.919
0.220	0.941
0.230	0.962
0.240	0.983
0.250	1.003
0.260	1.023
0.270	1.042
0.280	1.062
0.290	1.080
0.300	1.099
0.310	1.117
0.320	1.135
0.330	1.152
0.340	1.170
0.350	1.187
0.360	1.204
0.370	1.220



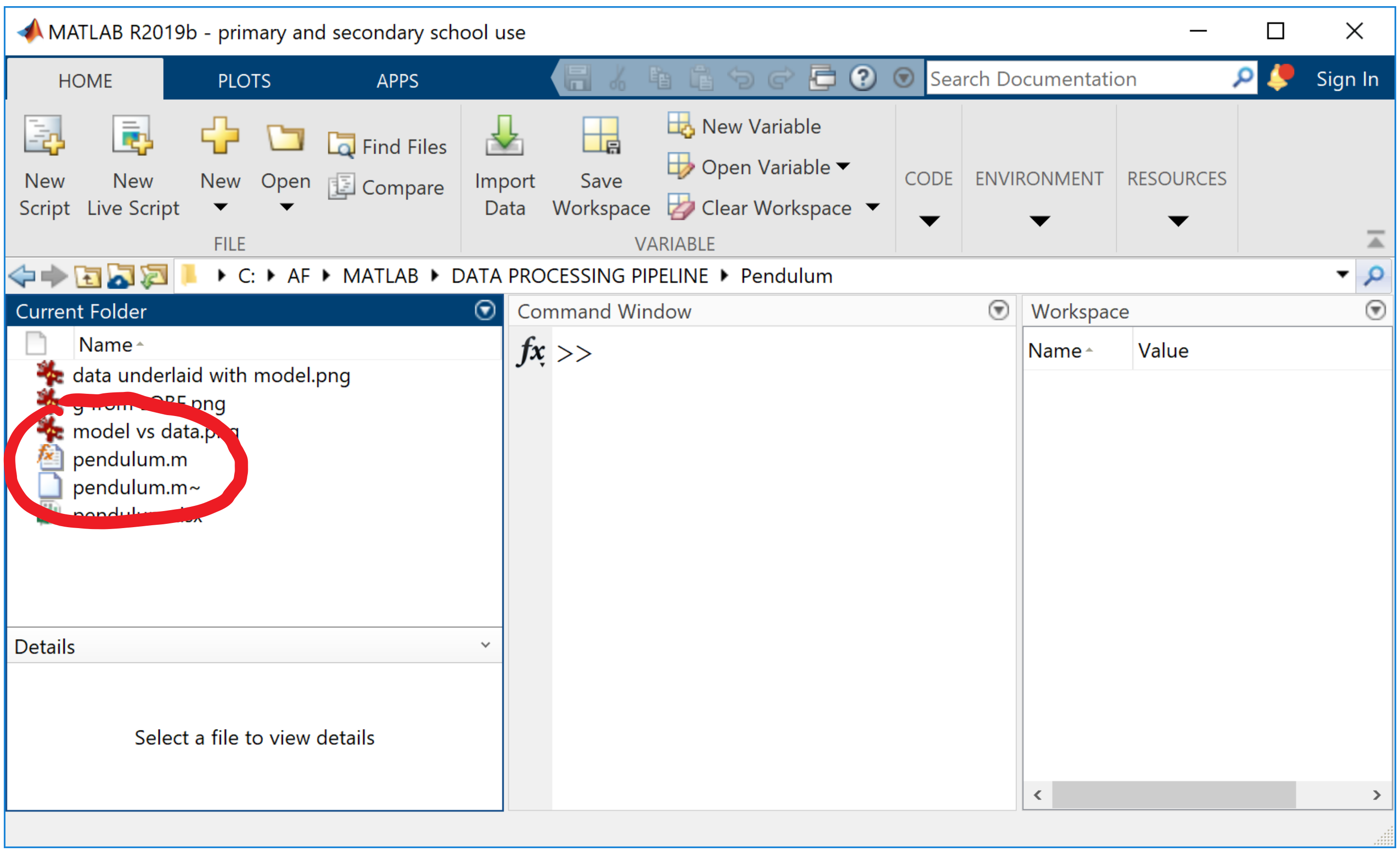
Initial angle = 20 degrees



Linearization, line of best fit to assess model correlation, and determine g

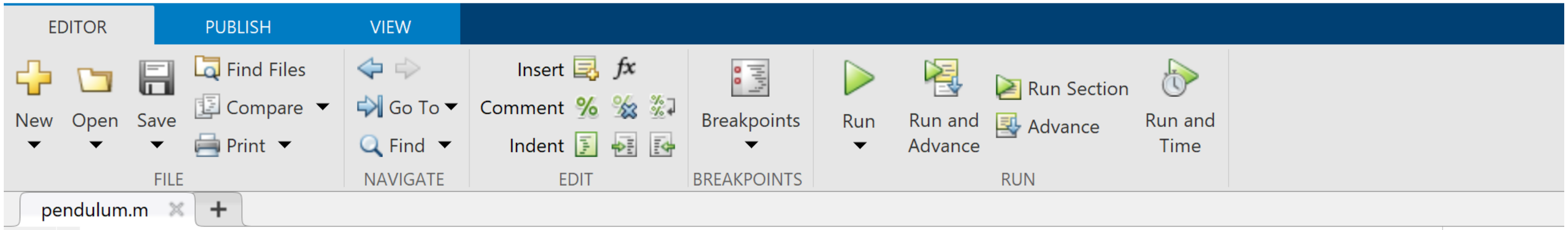
Run **`pendulum.m`** (right click, **run**) to execute a series of commands which constitute the rest of the data processing pipeline. The code can be modified for different experiments.

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.



MATLAB data processing pipeline

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



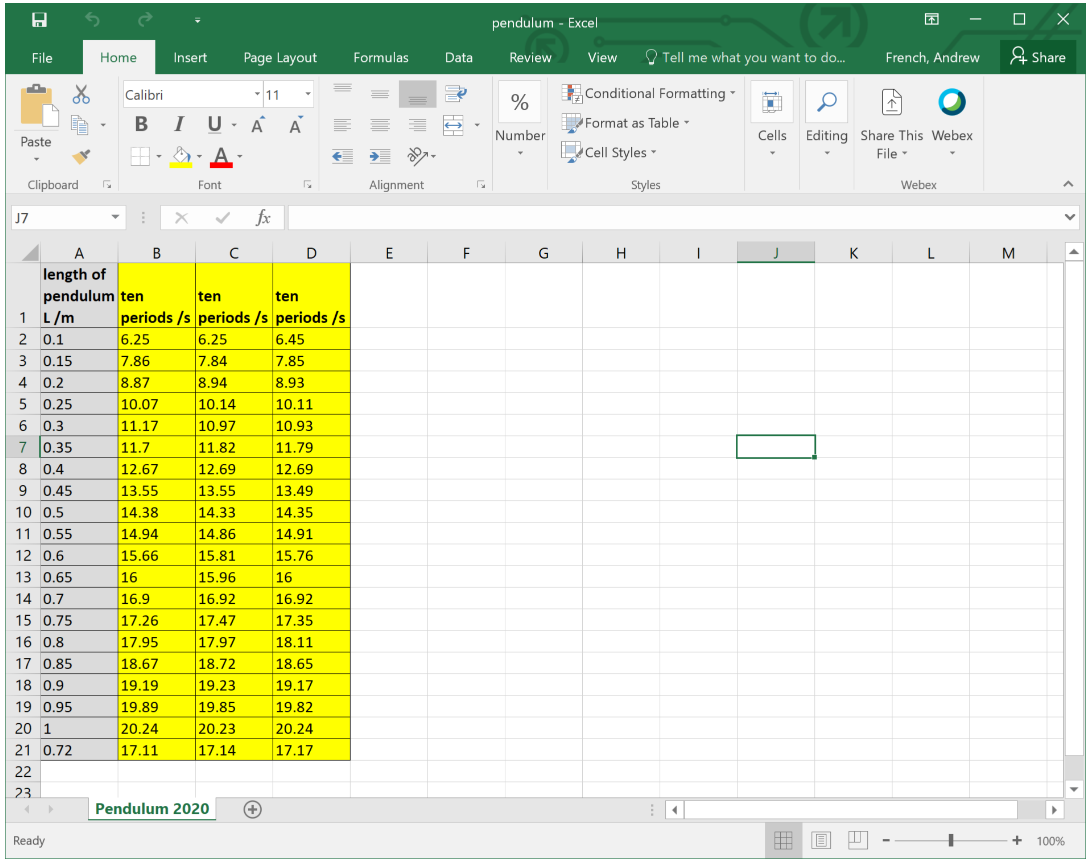

```
% Example physics data processing pipeline: #1 Pendulum
% * Load raw data from an Excel sheet pendulum.xlsx. This has columns of
%   pendulum length L /m, and three repeats of ten periods (10*T) /s.
% * Determine averages and errors
% * Plot y = 4*pi^2 * L vs x =T^2. Determine line of best fit (LOBF) 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



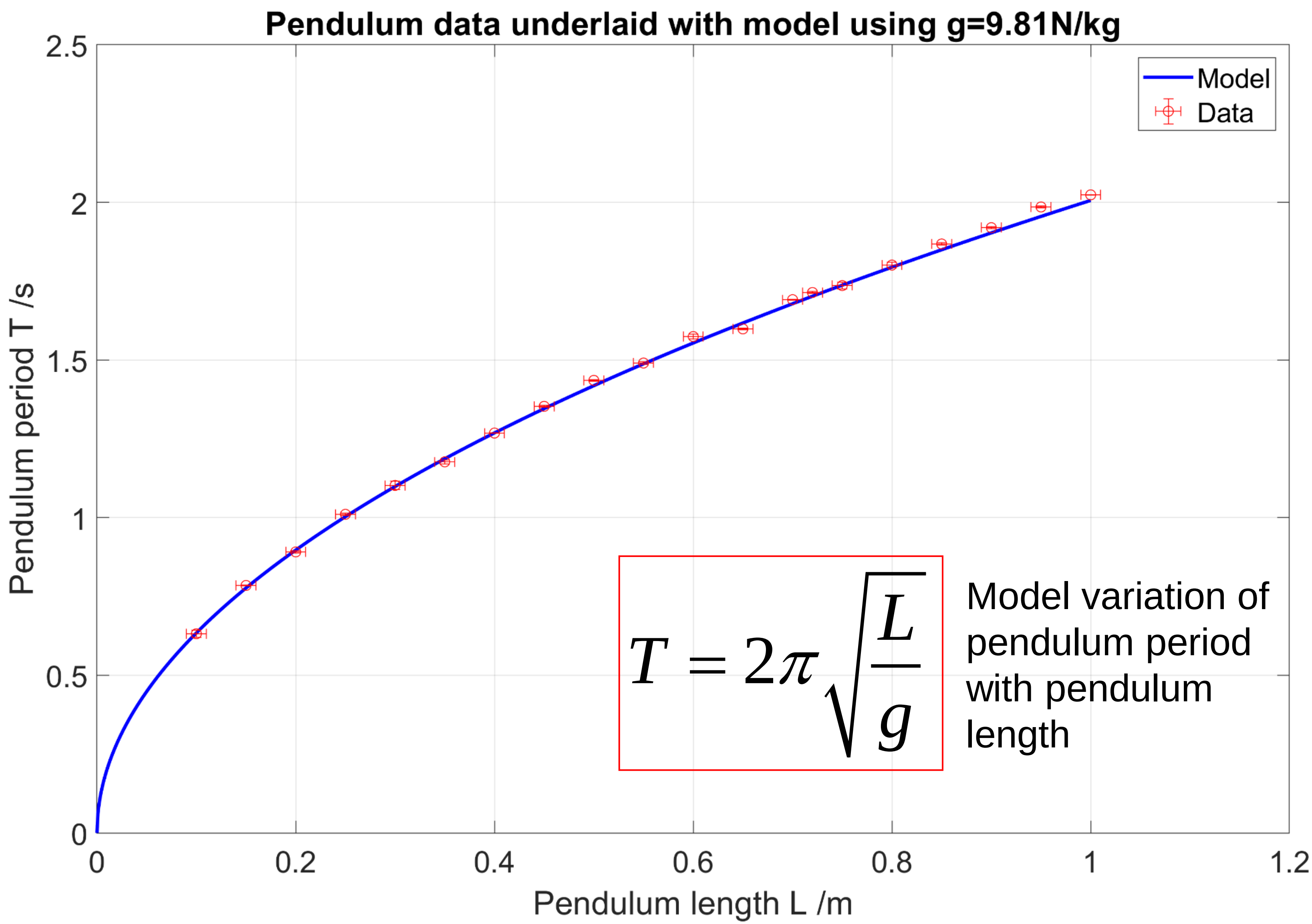
	A	B	C	D
1	length of pendulum L /m	ten periods /s	ten periods /s	ten periods /s
2	0.1	6.25	6.25	6.45
3	0.15	7.86	7.84	7.85
4	0.2	8.87	8.94	8.93
5	0.25	10.07	10.14	10.11
6	0.3	11.17	10.97	10.93
7	0.35	11.7	11.82	11.79
8	0.4	12.67	12.69	12.69
9	0.45	13.55	13.55	13.49
10	0.5	14.38	14.33	14.35
11	0.55	14.94	14.86	14.91
12	0.6	15.66	15.81	15.76
13	0.65	16	15.96	16
14	0.7	16.9	16.92	16.92
15	0.75	17.26	17.47	17.35
16	0.8	17.95	17.97	18.11
17	0.85	18.67	18.72	18.65
18	0.9	19.19	19.23	19.17
19	0.95	19.89	19.85	19.82
20	1	20.24	20.23	20.24
21	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

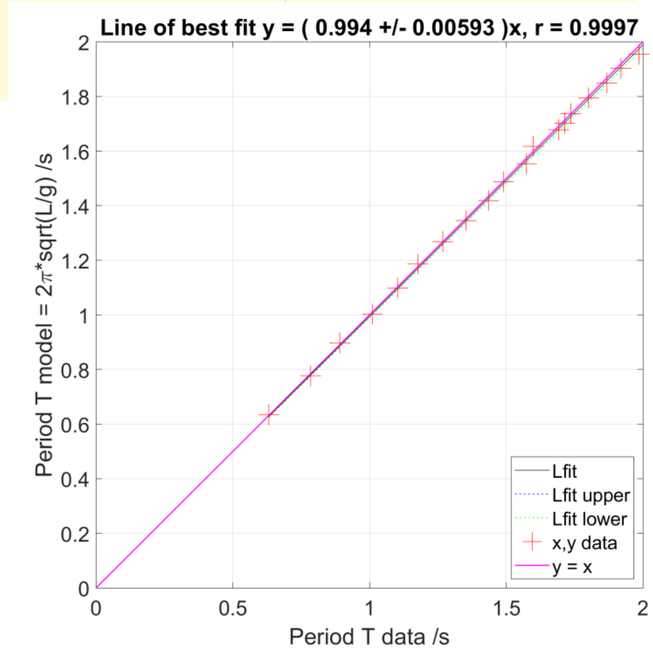


$$T = 2\pi\sqrt{\frac{L}{g}}$$

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
```

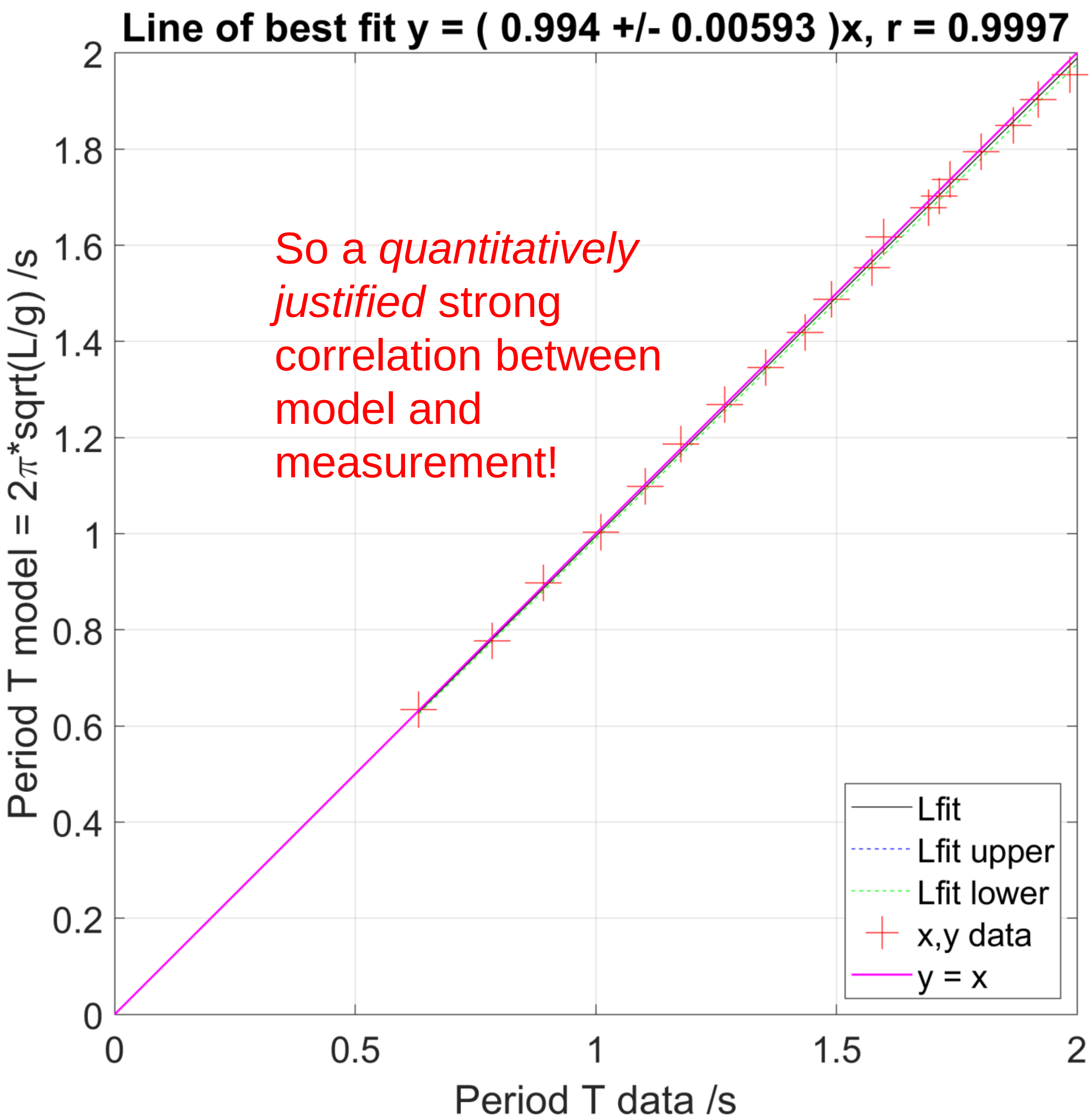
These are *sub-functions* which perform the line of best fit and associated plots. They should be generic, regardless of the dataset.



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

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

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

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

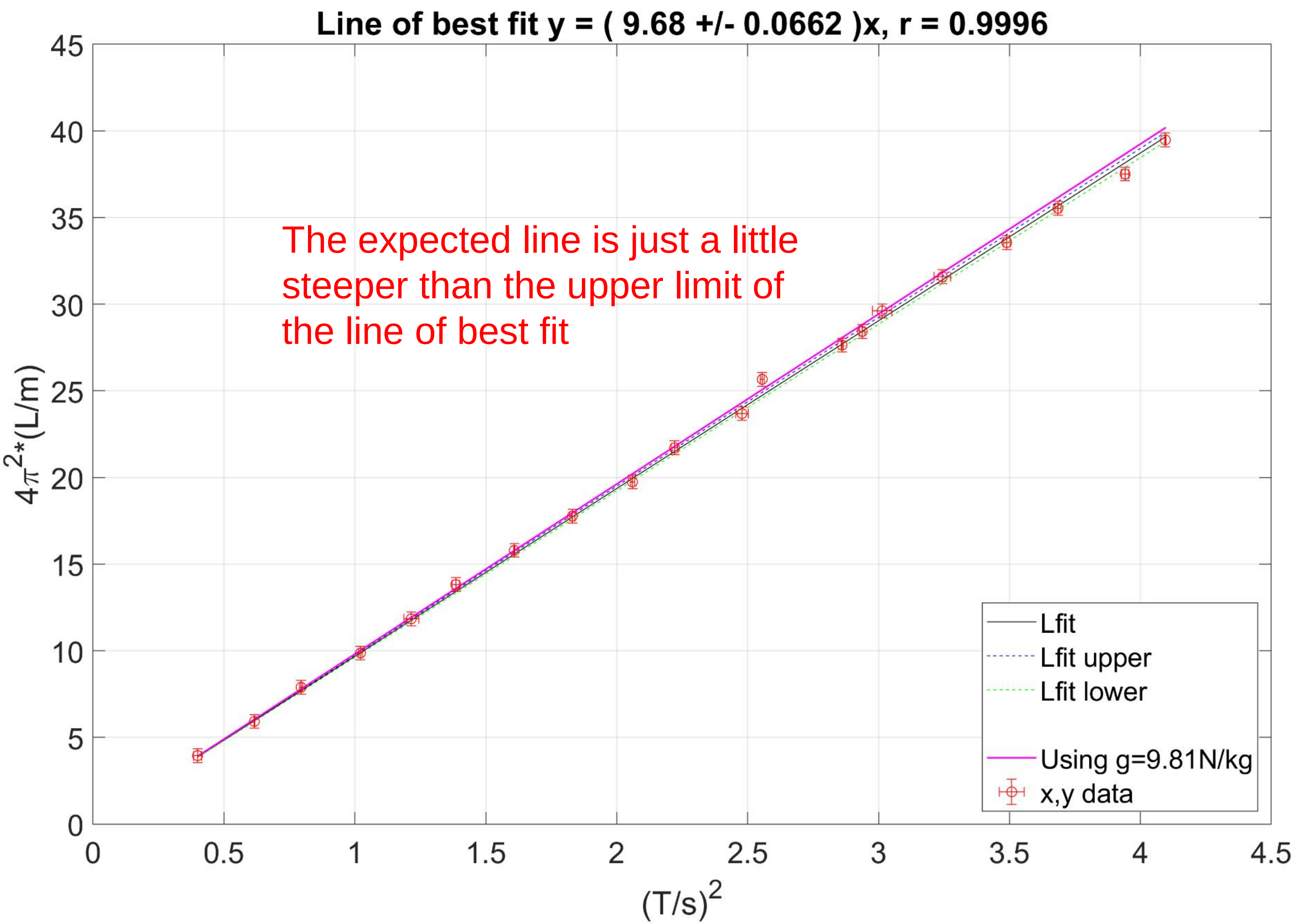
$$\Rightarrow y = gx$$

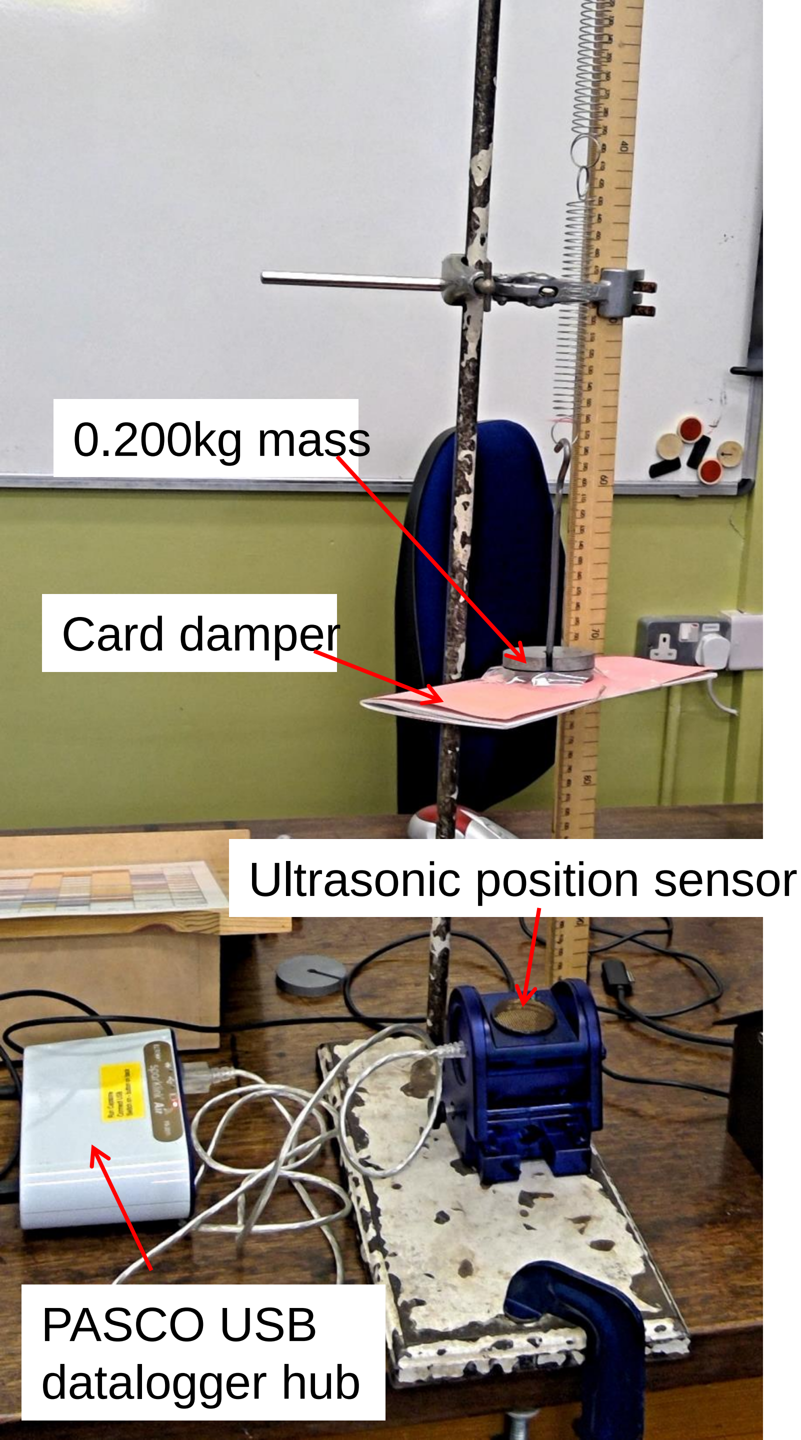
So g is the gradient of the x,y graph in our case

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

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

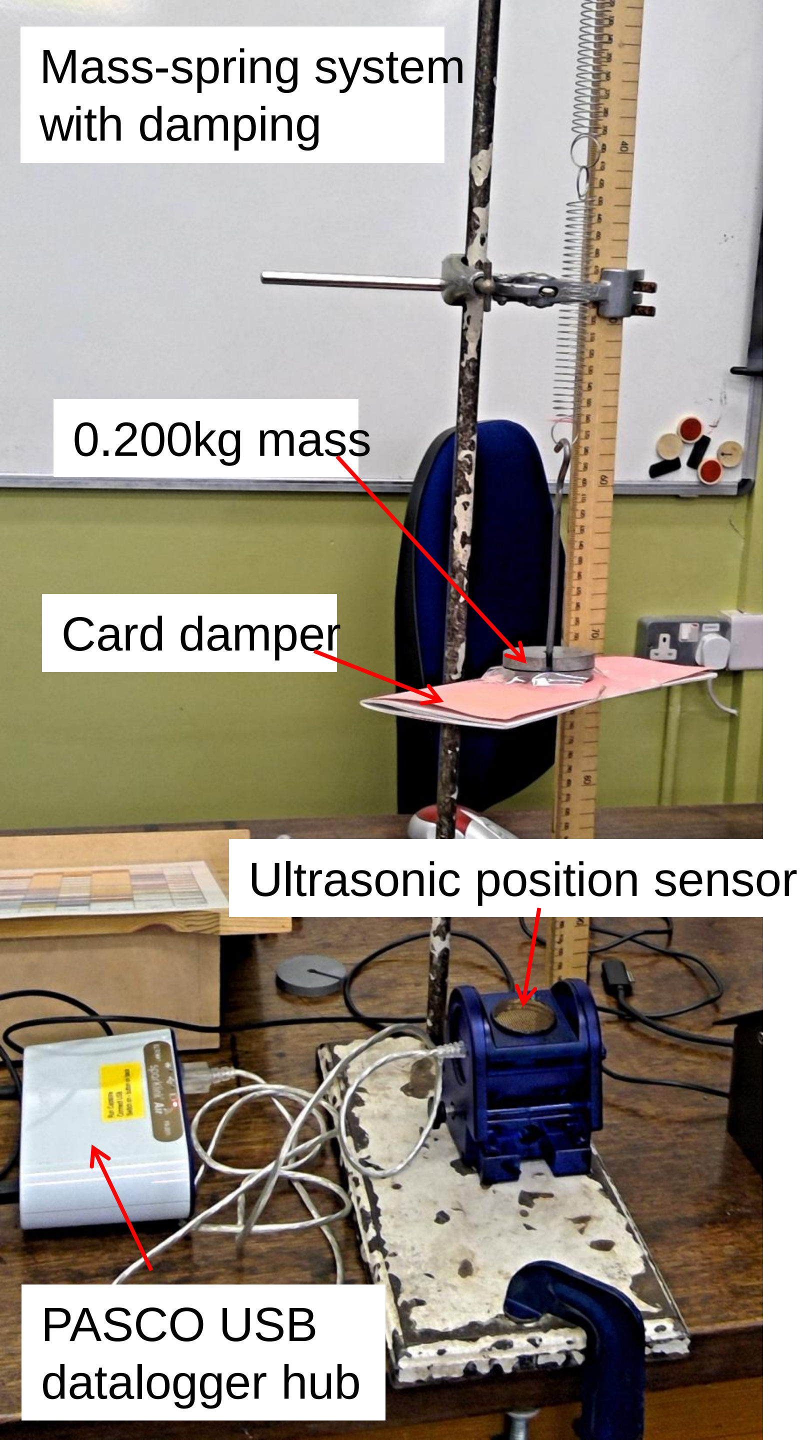
In our case, our gradient (and hence calculated g) is systematically lower than what it should be.

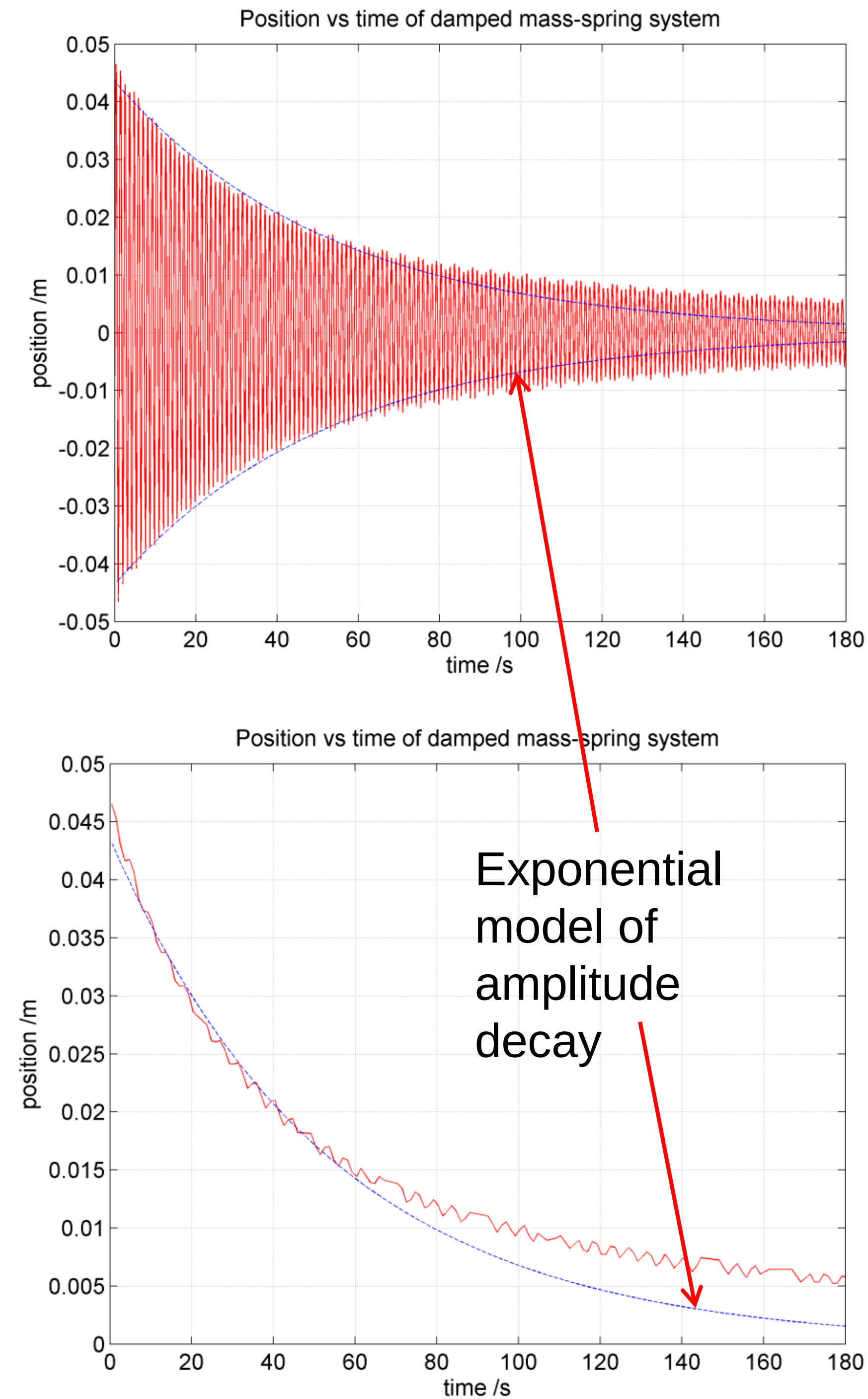


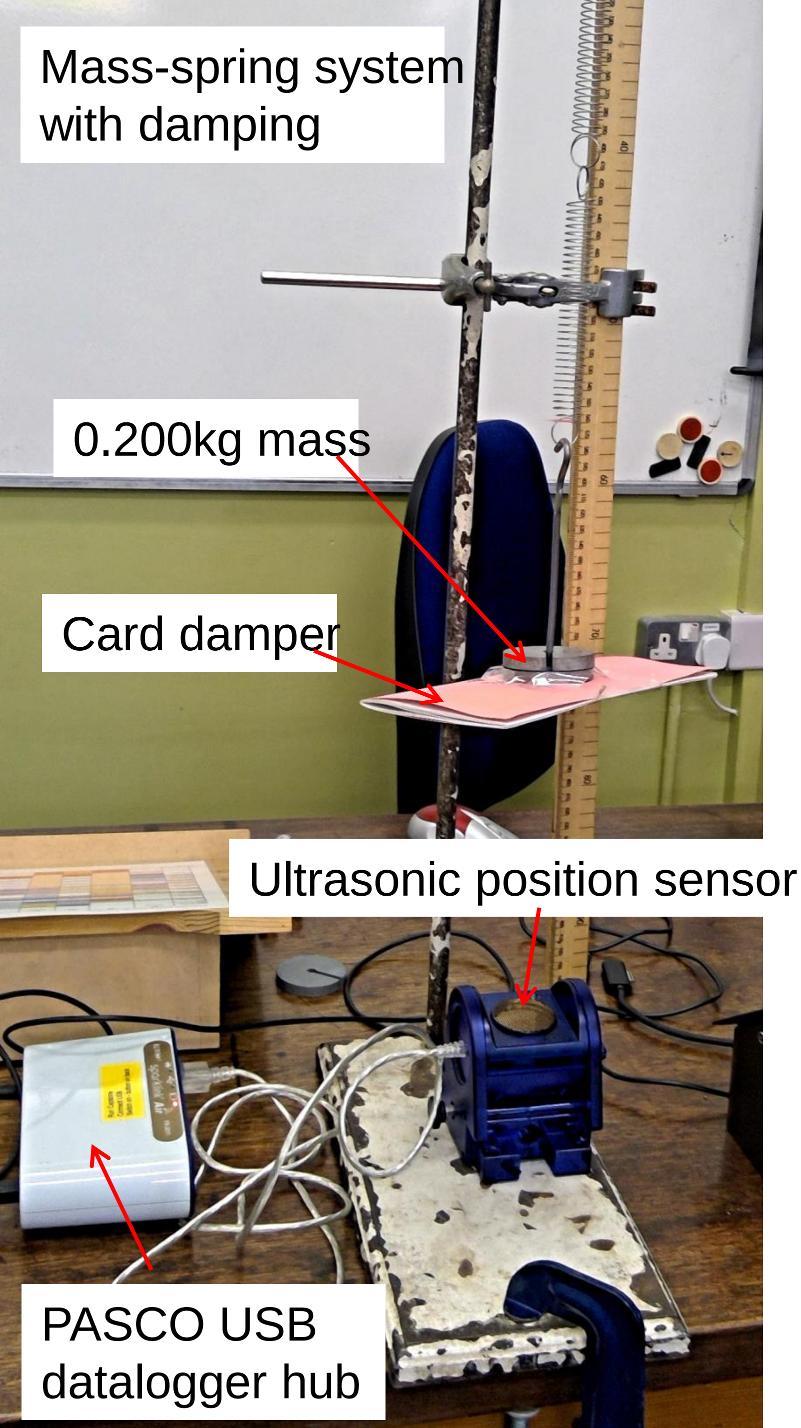


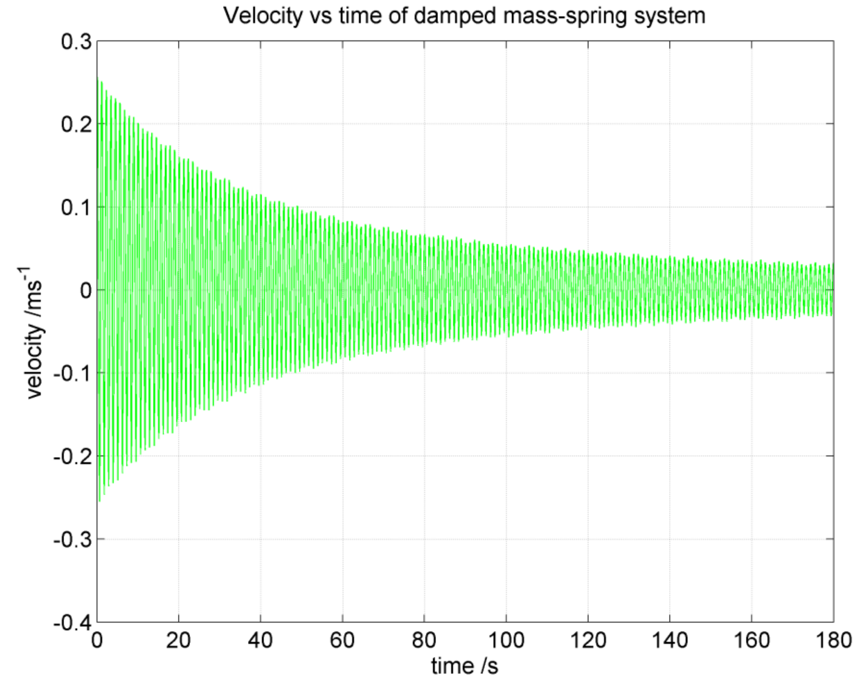
Excel to MATLAB data processing pipeline example:

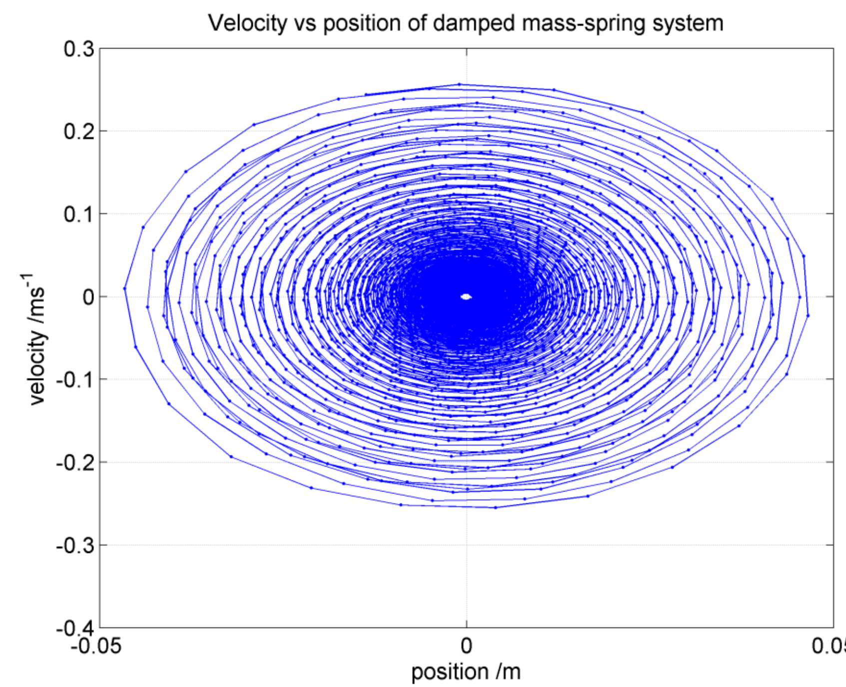
A **mass-spring system** with **damping**, with position recorded via an ultrasonic sensor and a datalogger.

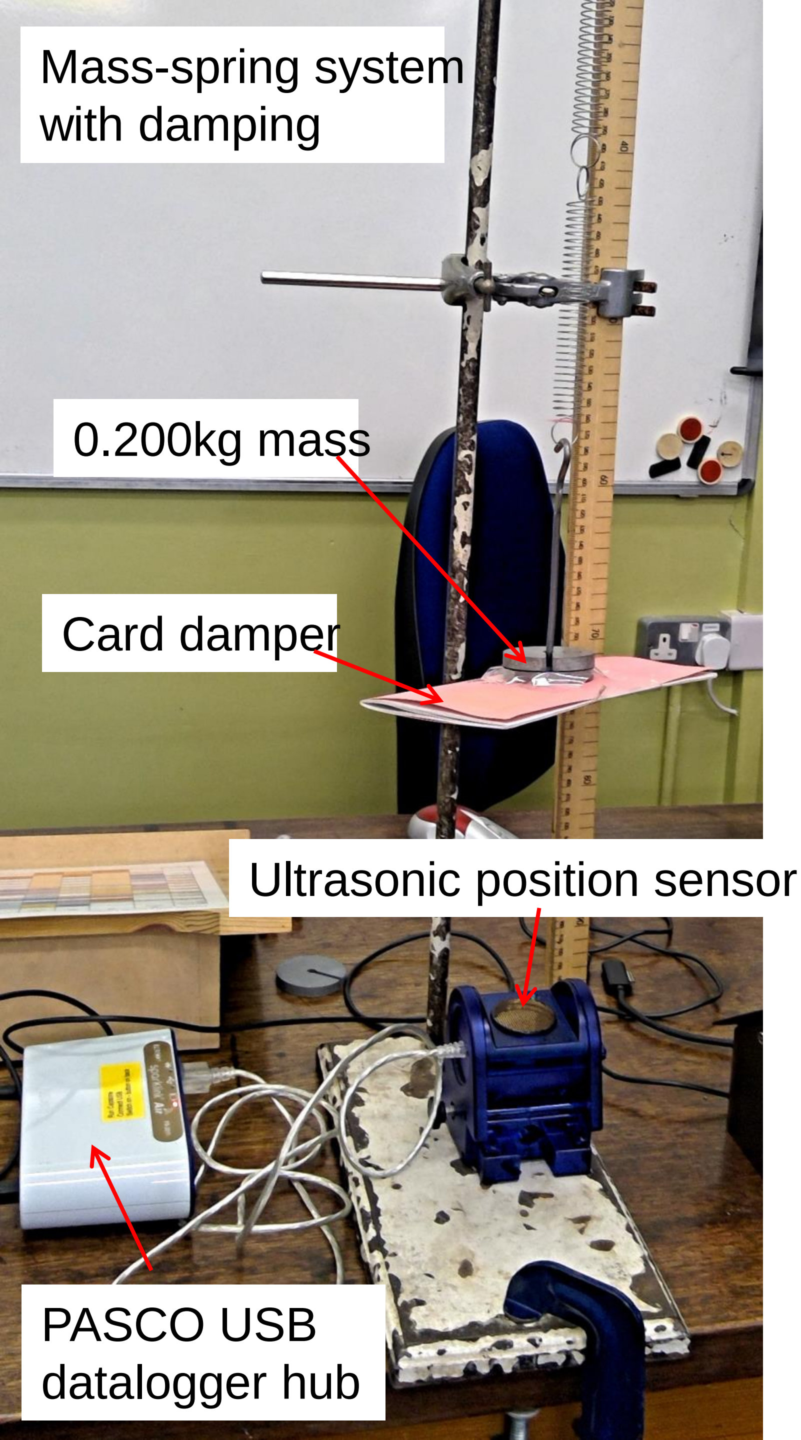

Mass-spring system with damping
0.200kg mass
Card damper
Ultrasonic position sensor
PASCO USB datalogger hub

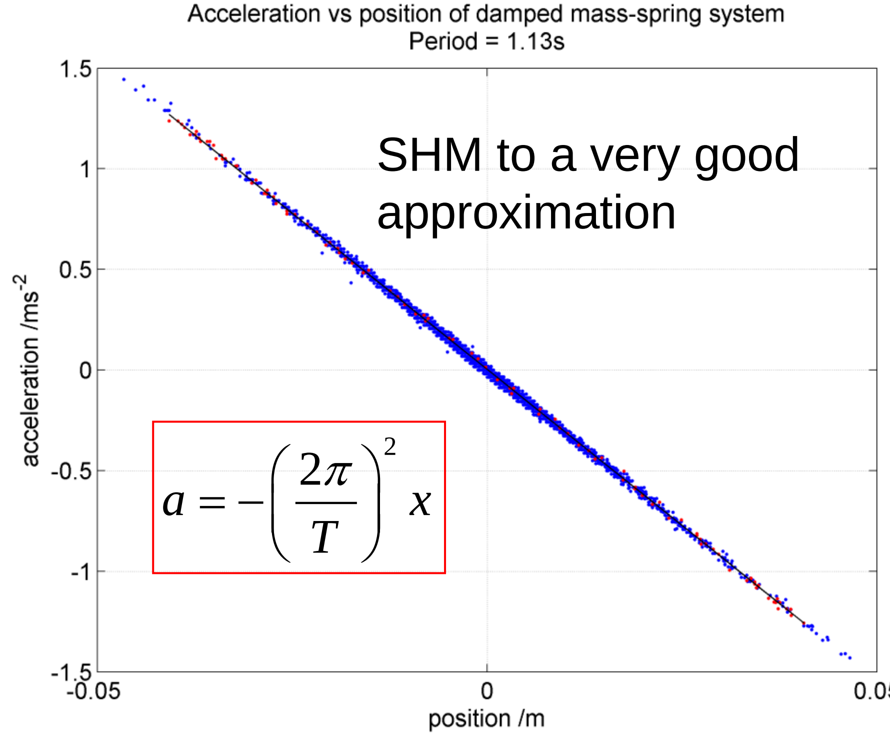

Position vs time of damped mass-spring system
position /m
time /s
Position vs time of damped mass-spring system
position /m
time /s
Exponential model of amplitude decay

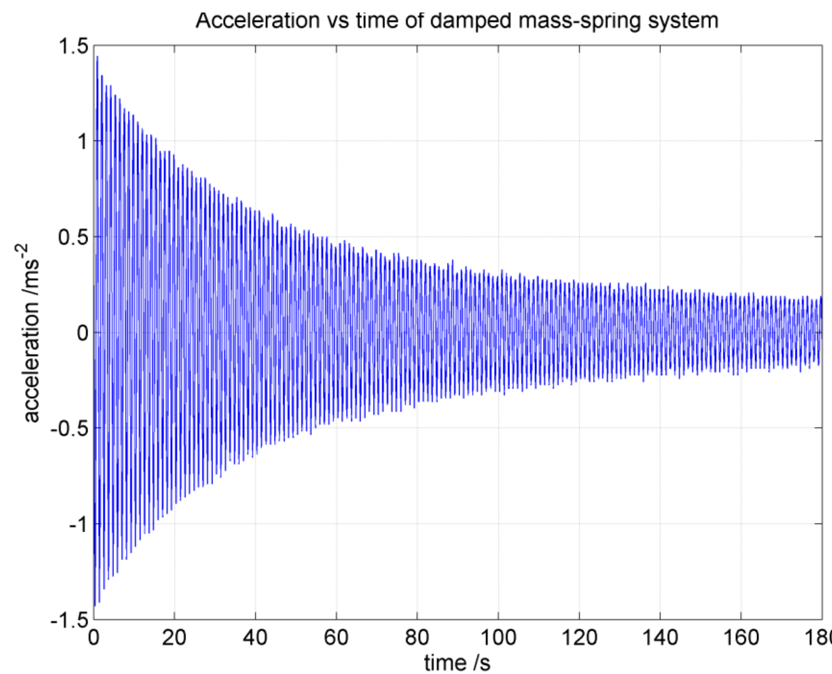

Mass-spring system with damping
0.200kg mass
Card damper
Ultrasonic position sensor
PASCO USB datalogger hub


Velocity vs time of damped mass-spring system
velocity /ms-1
time /s
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
0
20
40
60
80
100
120
140
160
180


Velocity vs position of damped mass-spring system
velocity /ms-1
position /m
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.05
0
0.05


Mass-spring system with damping
0.200kg mass
Card damper
Ultrasonic position sensor
PASCO USB datalogger hub

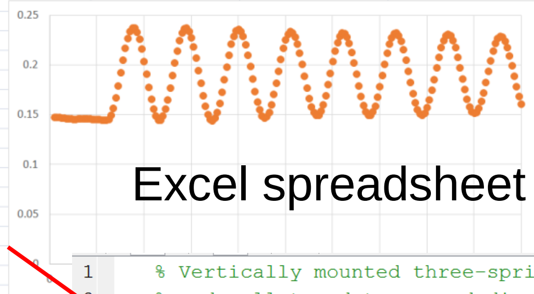

Acceleration vs position of damped mass-spring system
Period = 1.13s
SHM to a very good approximation
$a = -\left(\frac{2\pi}{T}\right)^2 x$
acceleration /ms^{-2}
position /m
1.5
1
0.5
0
-0.5
-1
-1.5
-0.05
0
0.05


Acceleration vs time of damped mass-spring system
acceleration /ms^{-2}
time /s
1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100
120
140
160
180

Data processing pipeline for mass-spring system

Datalogger

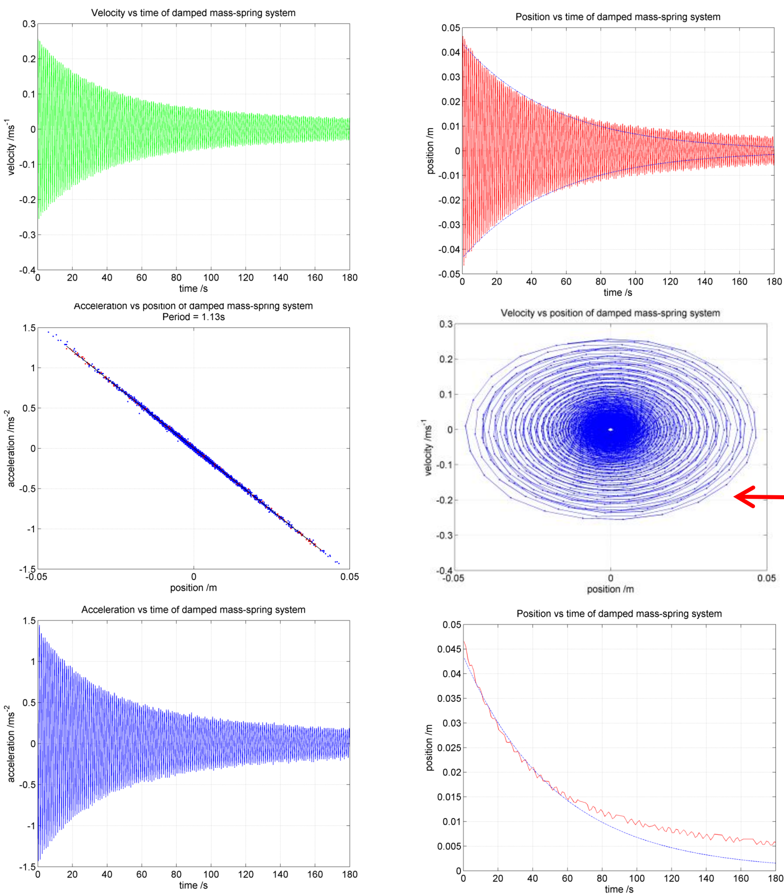
Position (m)	Velocity (m/s)	Acceleration (m/s²)	Time (s)
0.1479	1.48E-05	-0.0169	0.1
0.1479	-0.0017	-0.0516	0.15
0.1478	-0.0051	-0.0519	0.2
0.1474	-0.0069	-0.0176	0.25
0.1471	-0.0069	0.0343	0.3
0.1467	-0.0035	0.0522	0.35
0.1467	-0.0017	-0.0172	0.4
0.1465	-0.0052	-0.0181	0.45
0.1462	-0.0035	0.0861	0.5
0.1462	0.0034	0.0695	0.55
0.1465	0.0035	-0.0172	0.6
0.1465	0.0017	-0.0173	0.65
0.1467	0.0017	4.41E-04	0.7
0.1467	0.0018	-0.0512	0.75
0.1469	-0.0034	-0.1038	0.8
0.1464	-0.0086	-0.0351	0.85
0.146	-0.0069	0.0515	0.9
0.1457	-0.0035	0.0692	0.95



Excel spreadsheet

MATLAB

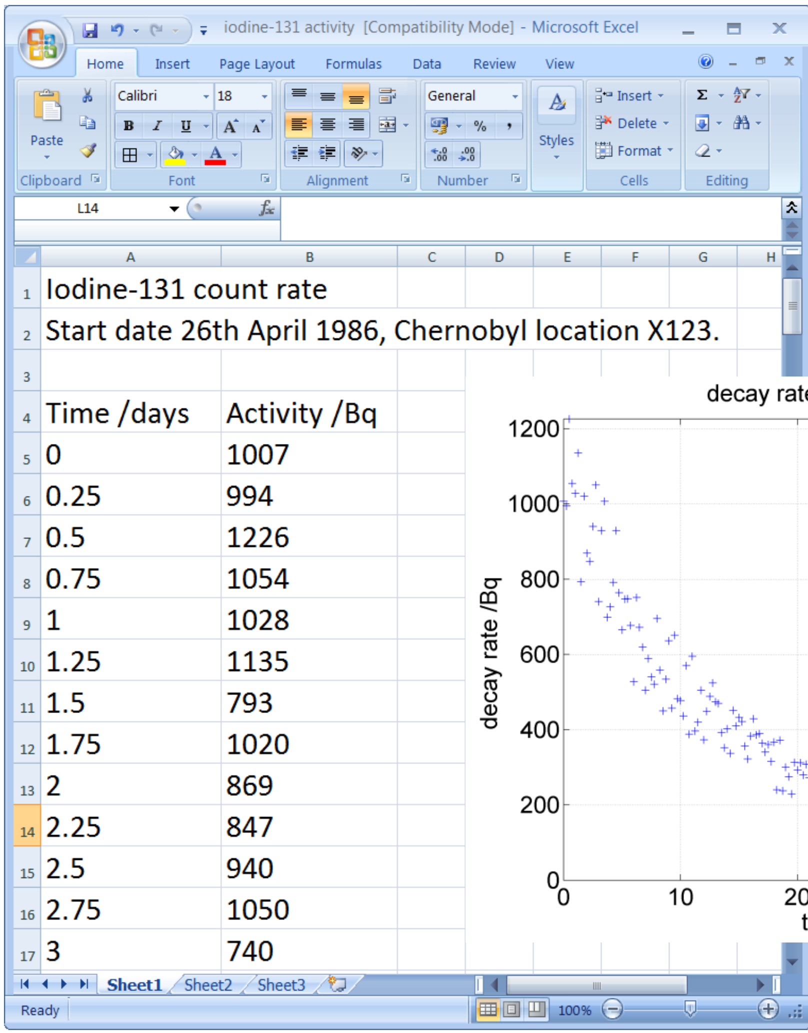
```
% Vertically mounted three-springs-in-series are attched to a 0.2kg mass
% and sellotaped to a card diary to increase air resistance.
% Position of the mass is determined via a PASCO ultrasonic transducer,
% and data is logged at 20Hz using the Capstone system. Capstone
% automatically determines velocity and acceleration via some form of
% numerical differentiation.
function damped_mass_on_spring_analysis
fsize = 14;  %Fontsize for graphs
tstart = 0;  tend = 60*3;  %Time interval

%Load data
[num,txt,raw] = xlsread( 'Damped mass on spring.xlsx');

%Ignore first few data points (oscillation hasn't started yet)
num = num(50:end,:);

%Determine position, velocity, acceleration, time in SI units
t = num(:,4); t = t - t(1);
x = num(:,1); v = num(:,2); a = num(:,3);

%Determine equilibrium height from the average position, and shift x
%such that equilibrium displacement is zero.
x = x - mean(x);

%Restricted data set (i.e. where effect of damping can be ignored)
aa = a(100:200); tt = t(100:200); xx = x(100:200);

%Determine line of best fit between a and x to determine period
[m,c,yfit,r] = lfit(xx,aa); T = 2*pi/sqrt(-m);

%Find peaks and plot these to determine exponential envelope
[tp,xp] = peakfindergeneral(t,x); N = length(xp);

%Linearize exponential envelope. Choose which set of data you want to use
%for this computation. NOTE THE ENVELOPE IS NOT EXPONENTIAL IN THE SAME WAY
%DURING THE WHOLE DECAY, SINCE THE DRAG FORCE WON'T BE LINEAR WITH SPEED.
i = find(xp>0); i = i(1:50);
Y = log( xp(i) ); X = tp(i);

%Determine exponential envelope in the form x = (+/-) A*exp(-k*t)
[m,c,YFIT,r] = lfit( X,Y);
```



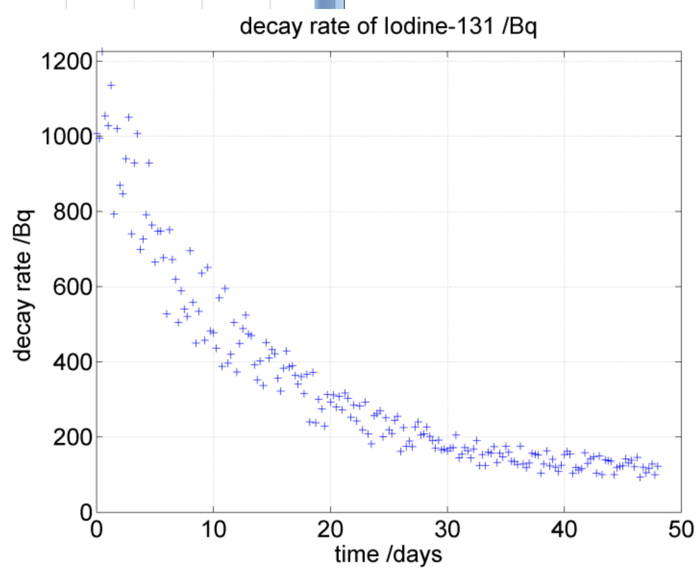
Auto-generated graphs and analysis

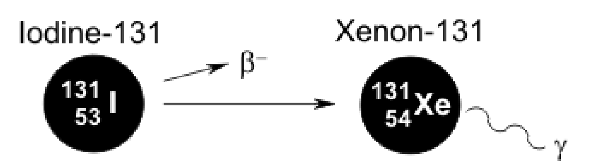
Chernobyl

You are a soviet nuclear physicist sent to help with the Chernobyl disaster in 1986. You need to determine the presence of an isotope from its half life, but background levels are huge.... All you have is a text file of count rates. Your military commander demands results as soon as possible.



Time /days	Activity /Bq
0	1007
0.25	994
0.5	1226
0.75	1054
1	1028
1.25	1135
1.5	793
1.75	1020
2	869
2.25	847
2.5	940
2.75	1050
3	740





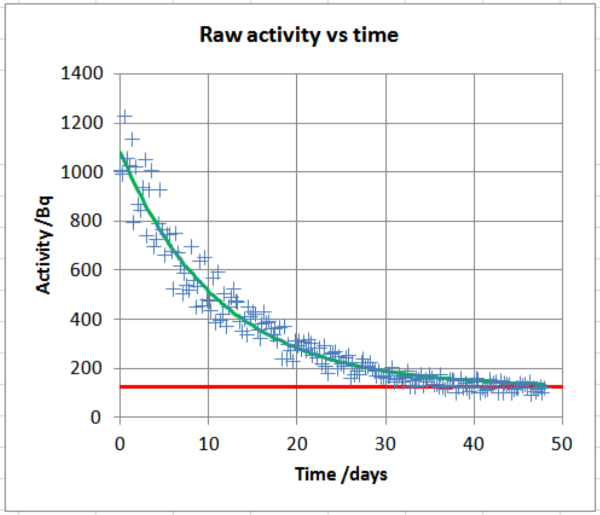
PERFORM ANALYSIS IN EXCEL

Iodine-131 count rate

Start date 26th April 1986, Chernobyl location X123.

Time /days	Activity /Bq	A = Activity - Background	log10(A)	Model Activity /Bq
0	1007	884	2.946	1080.2
0.25	994	871	2.940	1059.1
0.5	1226	1103	3.043	1038.4
0.75	1054	931	2.969	1018.2
1	1028	905	2.957	998.39
1.25	1135	1012	3.005	979.05
1.5	793	670	2.826	960.14
1.75	1020	897	2.953	941.65
2	869	746	2.873	923.57
2.25	847	724	2.860	905.89
2.5	940	817	2.912	888.6
2.75	1050	927	2.967	871.69
3	740	617	2.790	855.15
3.25	929	806	2.906	838.98
3.5	1007	884	2.946	823.16
3.75	699	576	2.760	807.7
4	726	603	2.780	792.58
4.25	791	668	2.825	777.79
4.5	929	806	2.906	763.32
4.75	764	641	2.807	749.18
5	665	542	2.734	735.35
5.25	747	624	2.795	721.83
5.5	748	625	2.796	708.6
5.75	677	554	2.744	695.66
6	527	404	2.606	683.02
6.25	751	628	2.798	670.65
6.5	672	549	2.740	658.55
6.75	619	496	2.695	646.72
7	505	382	2.582	635.15
7.25	589	466	2.668	623.84
7.5	540	417	2.620	612.78
7.75	521	398		
8	695	572		
8.25	558	435		
8.5	450	327		
8.75	534	411		

NOTE IGNORE DATA AFTER 28.75 DAYS FOR BEST FIT



Note this estimate is slightly different to the 100Bq used in the subsequent MATLAB analysis

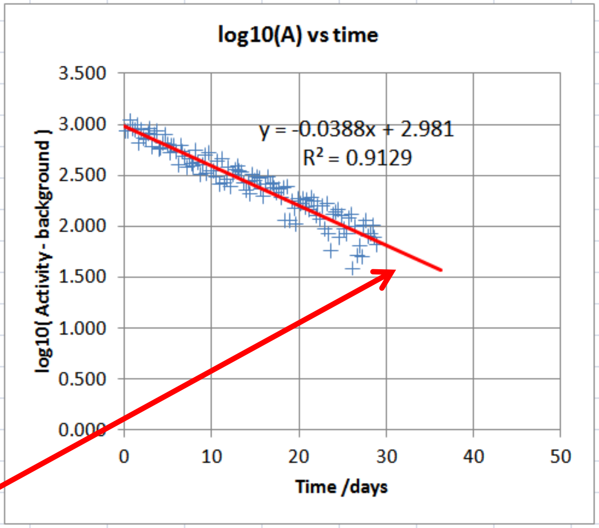
Estimate background level /Bq

123

Time /days	Activity /Bq
0	123
50	123

m	-0.039
c	2.981

Half life /days	7.759
A0 /Bq	957.2



$$A = \frac{A_0}{2^{t/t_{1/2}}}$$

$$\log_{10} A = \log_{10} A_0 - \log_{10}\left(2^{t/t_{1/2}}\right)$$

$$\log_{10} A = \log_{10} A_0 - \frac{t}{t_{1/2}}\log_{10} 2$$

$$y = \log_{10} A$$

$$x = t$$

$$y = mx + c$$

$$m = -\frac{\log_{10} 2}{t_{1/2}} \qquad c = \log_{10} A_0$$

$$t_{1/2} = -\frac{\log_{10} 2}{m} \qquad A_0 = 10^c$$

Data flow
Data processing and
Information
Presentation

is often best achieved by ***writing code*.**

In other words
a **text file** which
is interpreted
by a *programming language* like
MATLAB or Python

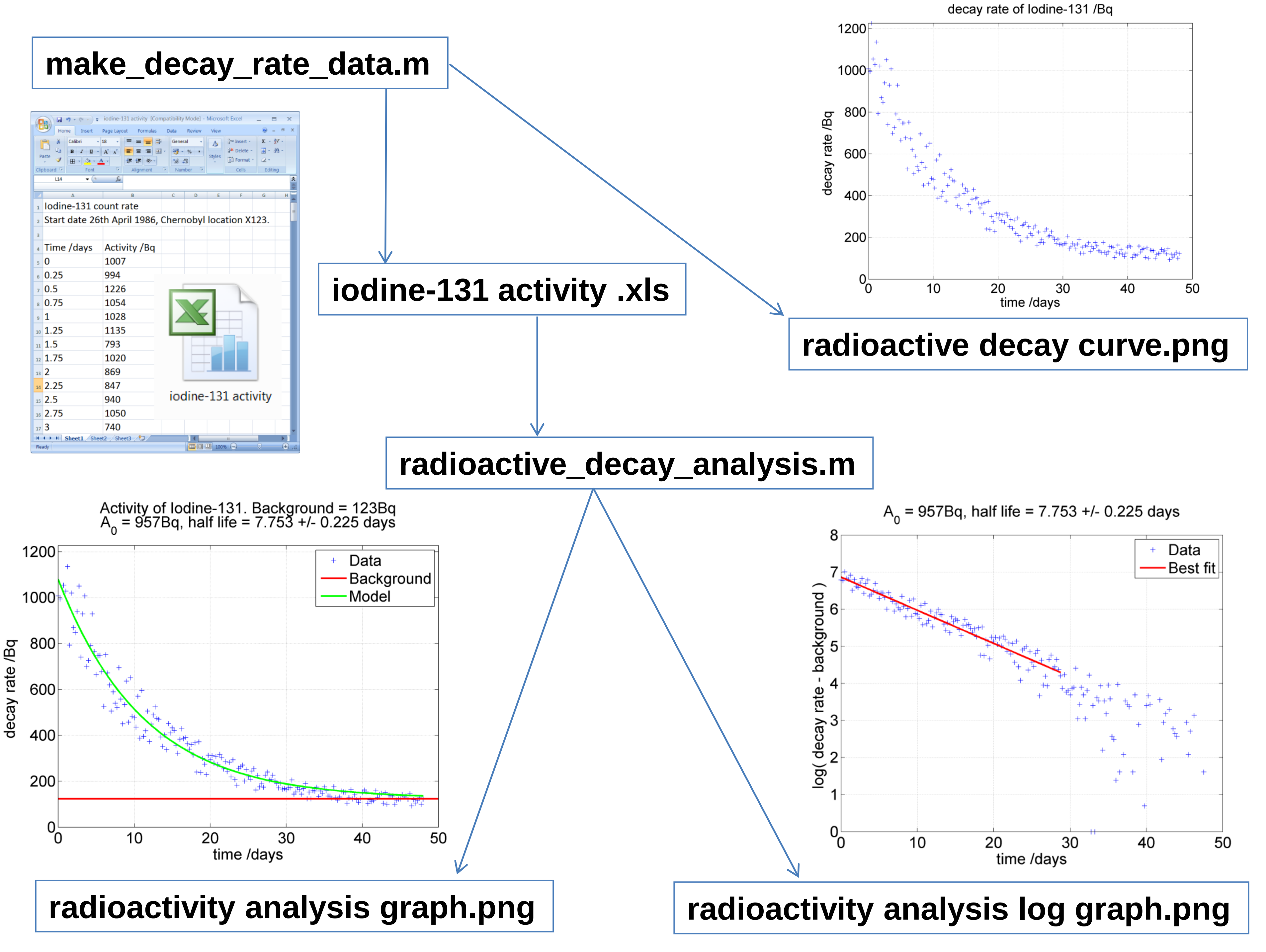
E:\Programming\A Course in Coding\2. MATLAB\Short Scientific Computing Course\1. The Signal and the Noise\radioactive_decay_analysis.m

File Edit Text Go Cell Tools Debug Desktop Window Help

Stack: Base

radioactive_decay_analysis.m

```
%radioactive_decay_analysis
% Analysis of Iodine-131 decay rate vs time data.
%
% LAST UPDATED by Andy French June 2019

function radioactive_decay_analysis

%Estimated background rate /Bq
B = 100;

%Fontsize for graphs
fsize = 18;

%

%Ingest Excel file of activity vs time
[num,txt,raw] = xlsread( 'iodine-131 activity.xls' );

%Extract vectors for time /days and activity /Bq
t = num(:,1); A = num(:,2);

%Plot activity vs time
fig1 = figure('color',[1 1 1],'name','radioactive decay curve');
plot(t,A,'+');
xlabel('time /days','fontsize',fsize);
ylabel('decay rate /Bq ','fontsize',fsize);
set(gca,'fontsize',fsize);
grid on; ylim([0,max(A)]);

%Overlay background level
xlimits = get( gca, 'xlim' ); hold on; plot( xlimits,[B,B],'r-' );

```


make_decay_rate_data.m
iodine-131 activity .xls
radioactive decay curve.png
radioactive_decay_analysis.m
radioactivity analysis graph.png
radioactivity analysis log graph.png
Iodine-131 count rate
Start date 26th April 1986, Chernobyl location X123.
Time /days Activity /Bq
0 1007
0.25 994
0.5 1226
0.75 1054
1 1028
1.25 1135
1.5 793
1.75 1020
2 869
2.25 847
2.5 940
2.75 1050
3 740
iodine-131 activity
decay rate of Iodine-131 /Bq
decay rate /Bq
time /days
Activity of Iodine-131. Background = 123Bq
A0 = 957Bq, half life = 7.753 +/- 0.225 days
Data
Background
Model
A0 = 957Bq, half life = 7.753 +/- 0.225 days
Best fit
log(decay rate - background)

Geostationary

We'll focus on data collected from this system

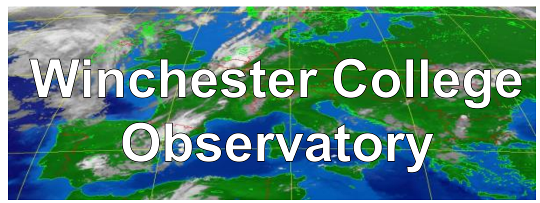
Temperature
Pressure
Humidity
Solar radiation

Wind speed
Wind direction
UV index

EUMETCAST
Earth Observation data

e.g. full hemisphere weather every **15 minutes at 1 pixel per km² resolution!**

Davis Vantage Pro
automated weather station



Dartcom PC based receiver system running software to ingest and process each data stream simultaneously

Processed data (e.g. temperature variation vs time Excel sheet, indexed images for plotting cloud cover over UK vs time etc)

Workstation console in room beneath observatory

USB sticks/hard drives (possibly internal network) to Z drive / Firefly for general Wincoll access

Investigate patterns in local measurements of temperature, humidity, pressure using our meteorological system on the roof of Science School. *Start with an Excel analysis of one month of data (November 2021), then investigate the MATLAB tools.*

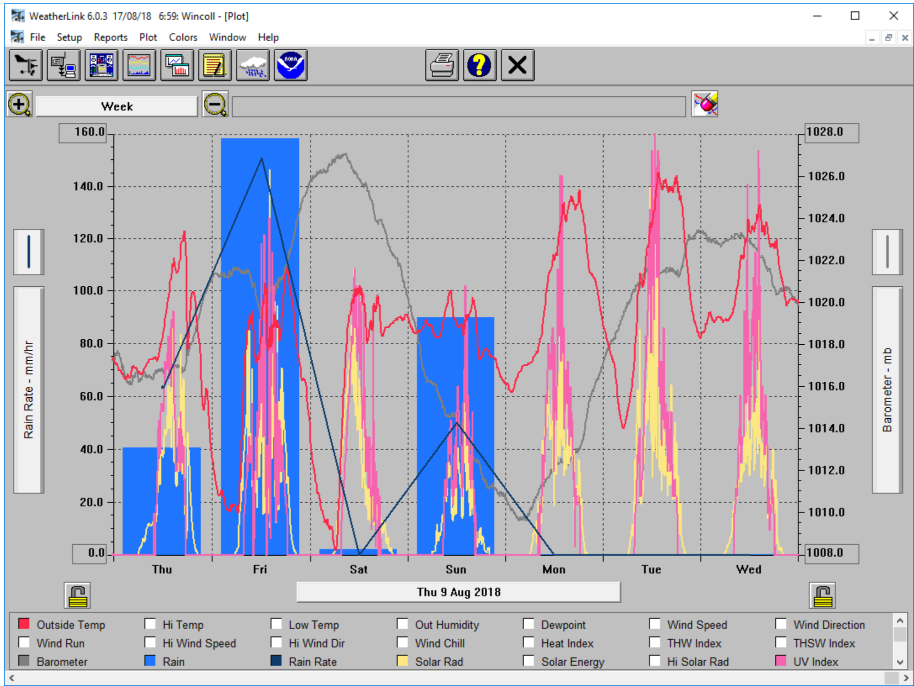
- Run **plot_met_data.m** and generate graphs from the files in the Met data directory
Can you spot any trends?

- Load a met_data file into **MATLAB**. Investigate its structure.

- Adapt code from **plot_met_data.m** to make your own graphs.

- How about temperature vs pressure, or temperature vs humidity. Are there any correlations?

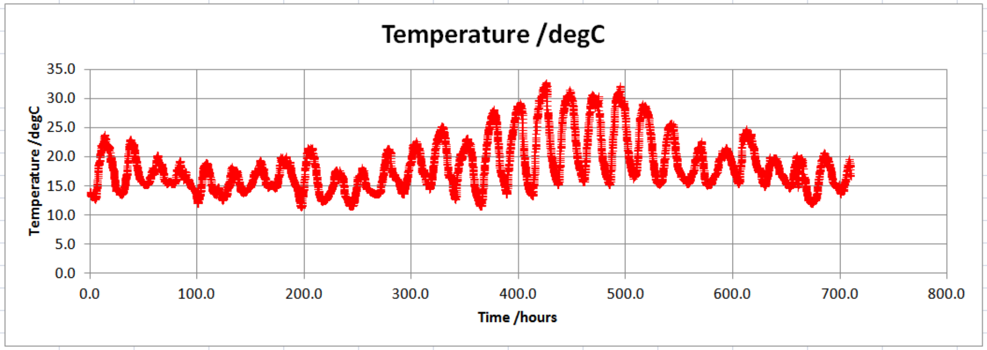
- Could you work out the *rate* of change of temperature, time etc? (And plot this).

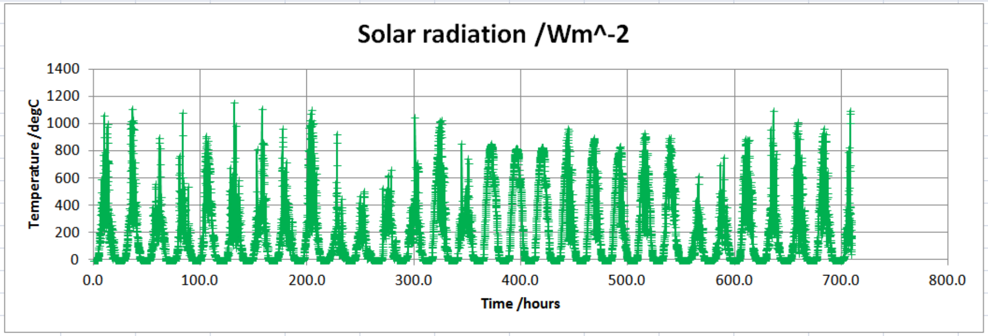


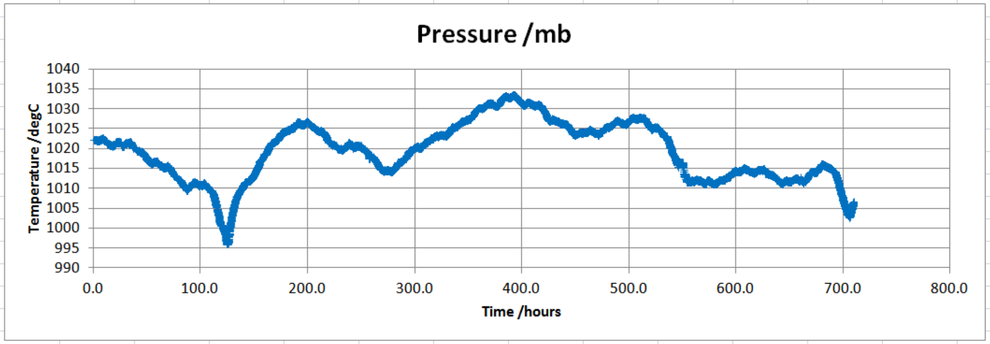
Example analysis of November 2021 Winchester College meteorological data using Microsoft Excel

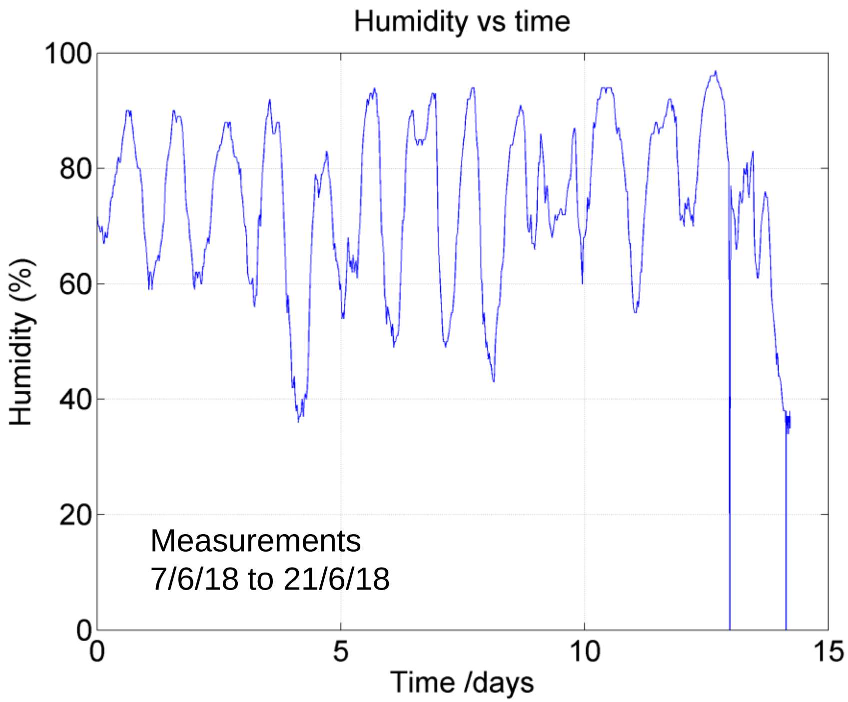
WINCHESTER COLLEGE WEATHER DATA FOR NOVEMBER 2021

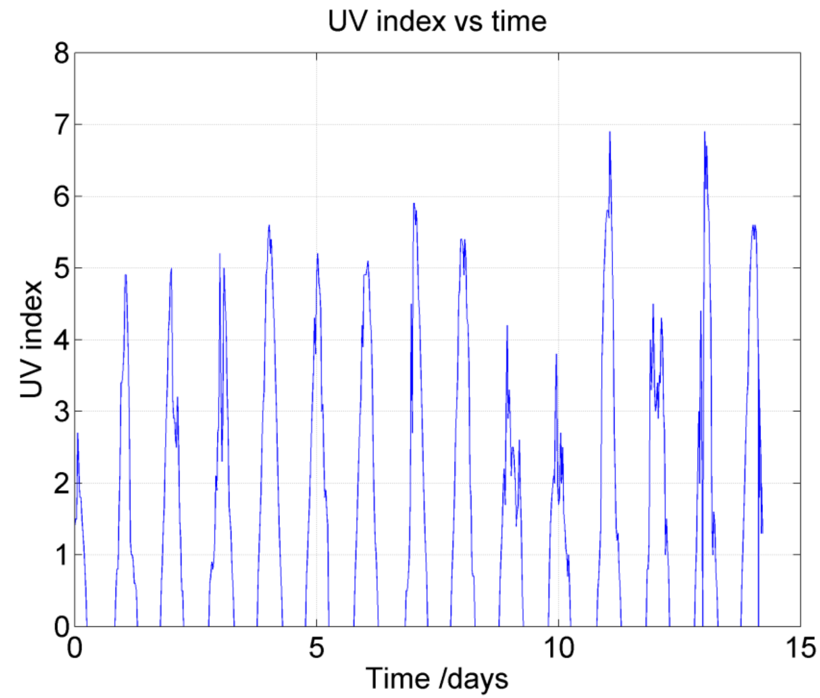
time /hours	temp /degC	solar radiation /Wm^-2	UV index	wind speed /ms^-1	wind direction /deg	Pressure /mb	humidity %	rain rate /mm/hour
0.0	14.1	0	0	0		1022.3	90	0
0.1	14.1	0	0	0		1022.3	90	0
0.2	14.0	0	0	0		1022.3	91	0
0.2	13.9	0	0	0		1022.4	91	0
0.3	13.8	0	0	0		1022.4	91	0
0.4	13.7	0	0	0		1022.3	91	0
0.5	13.7	0	0	0		1022.3	91	0
0.6	13.6	0	0	0		1022.3	91	0
0.7	13.4	0	0	0		1022.3	91	0
0.8	13.4	0	0	0		1022.4	92	0
0.8	13.4	0	0	0		1022.4	92	0
0.9	13.5	0	0	0		1022.4	92	0
1.0	13.5	0	0	0		1022.4	92	0
1.1	13.4	0	0	0		1022.4	92	0
1.2	13.3	0	0	0		1022.4	92	0
1.2	13.3	0	0	0		1022.4	92	0
1.3	13.4	0	0	0		1022.4	93	0
1.4	13.3	0	0	0		1022.4	93	0
1.5	13.3	0	0	0		1022.4	93	0
1.6	13.2	0	0	0		1022.3	93	0
1.7	13.2	0	0	0		1022.4	93	0
1.7	13.2	0	0	0		1022.3	93	0
1.8	13.2	0	0	0		1022.3	93	0
1.9	13.2	0	0	0		1022.3	93	0
2.0	13.2	0	0	0		1022.3	93	0
2.1	13.2	0	0	0		1022.3	93	0
2.2	13.3	0	0	0		1022.2	94	0
2.3	13.4	0	0	0		1022.4	94	0
2.3	13.4	0	0	0		1022.3	94	0
2.4	13.4	0	0	0		1022.4	94	0
2.5	13.4	0	0	0		1022.4	94	0
2.6	13.4	0	0	0		1022.4	94	0
2.7	13.4	0	0	0		1022.3	94	0
2.7	13.4	0	0	0		1022.4	94	0
2.8	13.4	0	0	0		1022.3	94	0
2.9	13.4	0	0	0		1022.2	94	0
3.0	13.3	0	0	0		1022.2	94	0
3.1	13.3	0	0	0		1022.3	94	0
3.2	13.3	0	0	0		1022.1	93	0
3.2	13.2	0	0	0		1022.2	94	0
3.3	13.2	0	0	0		1022.1	94	0
3.4	13.2	0	0	0		1022.1	93	0
3.5	13.1	0	0	0		1022	94	0



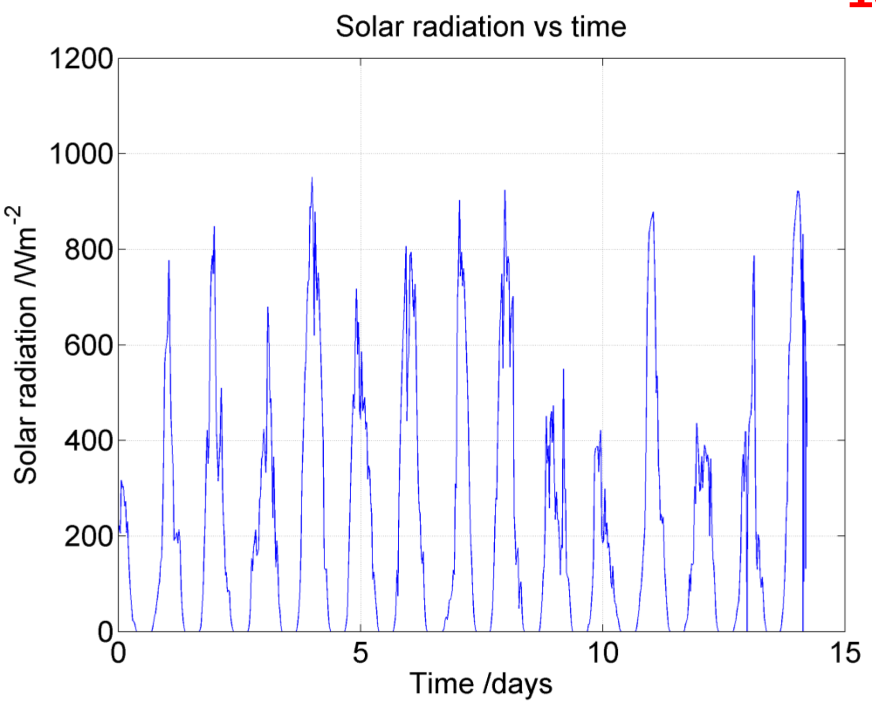


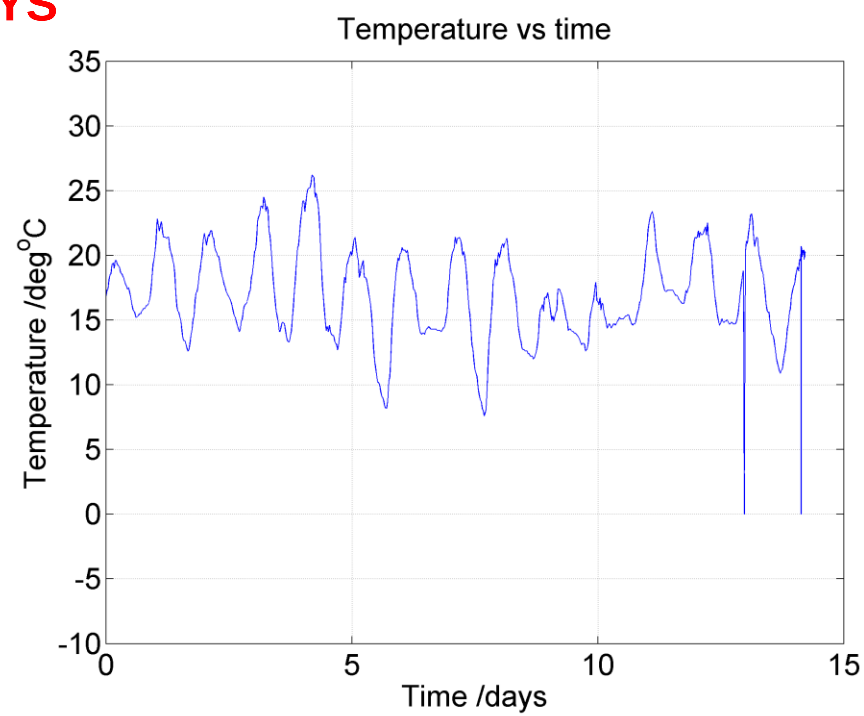


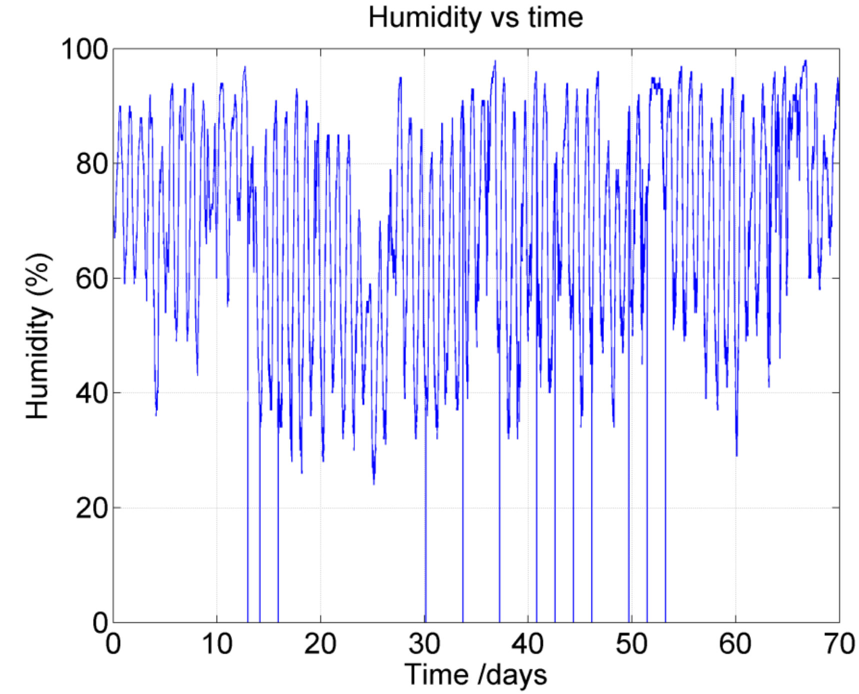

Humidity vs time
Humidity (%)
Time /days
Measurements
7/6/18 to 21/6/18

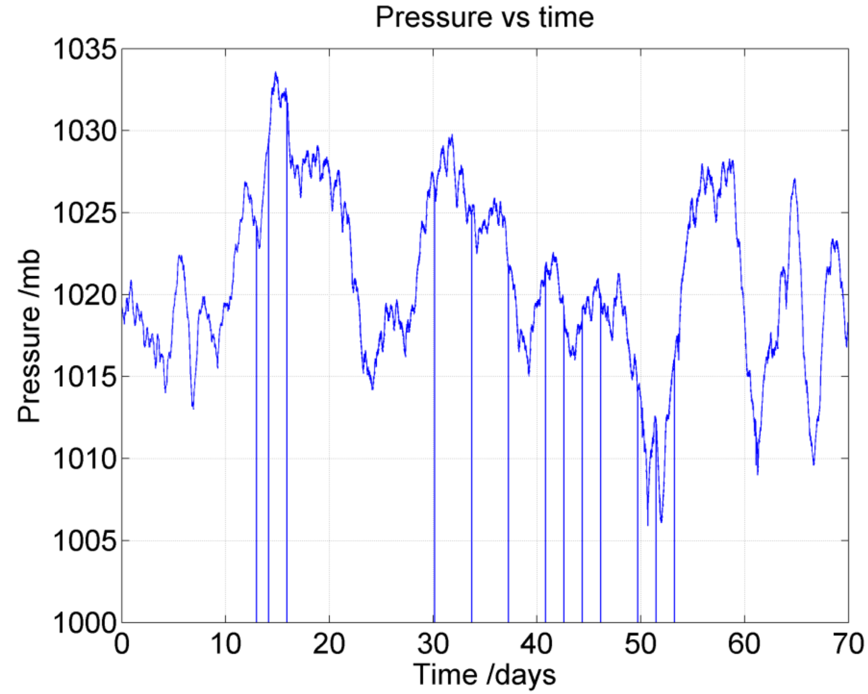

UV index vs time
UV index
Time /days

15 DAYS

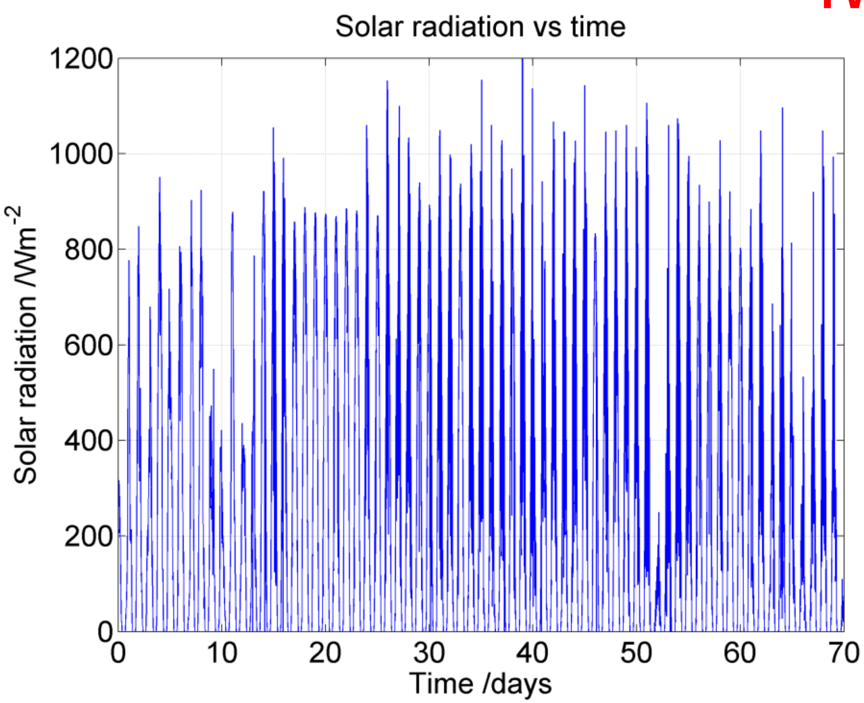

Solar radiation vs time
Solar radiation /Wm-2
Time /days

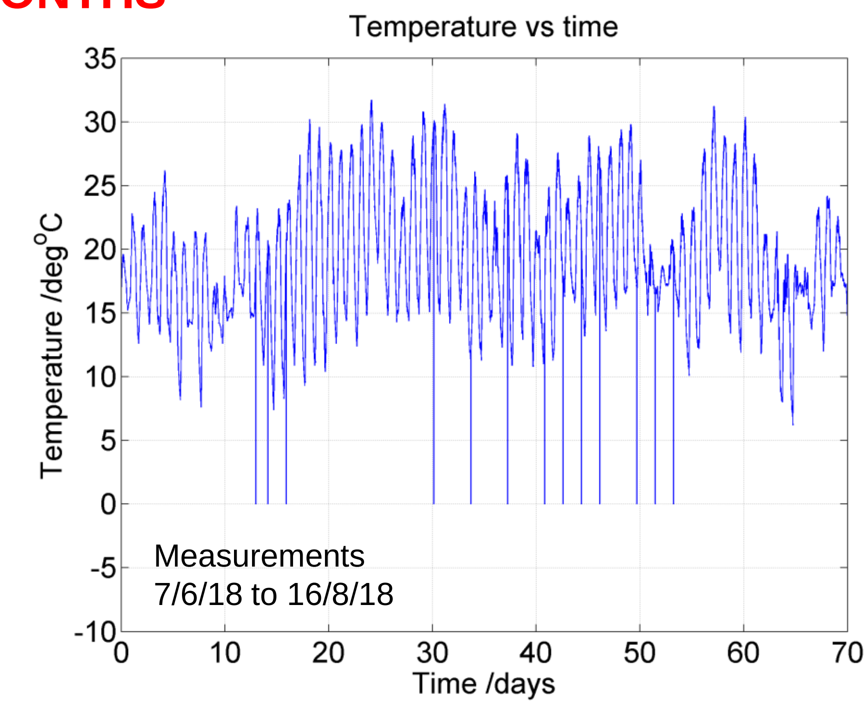

Temperature vs time
Temperature /degoC
Time /days

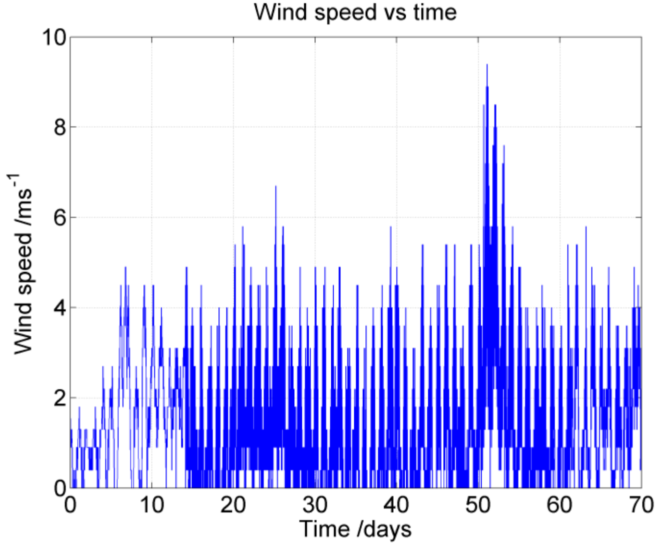

Humidity vs time
Humidity (%)
Time /days
0
20
40
60
80
100
0
10
20
30
40
50
60
70

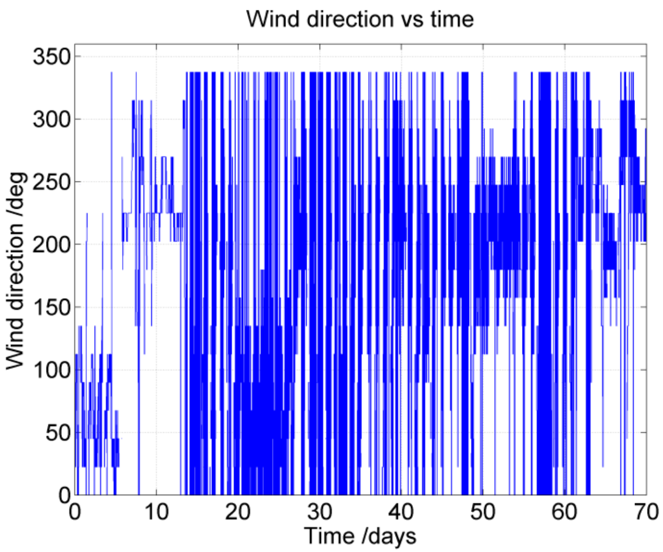

Pressure vs time
Pressure /mb
Time /days
1000
1005
1010
1015
1020
1025
1030
1035
0
10
20
30
40
50
60
70

TWO MONTHS


Solar radiation vs time
Solar radiation /Wm-2
Time /days
0
200
400
600
800
1000
1200
0
10
20
30
40
50
60
70

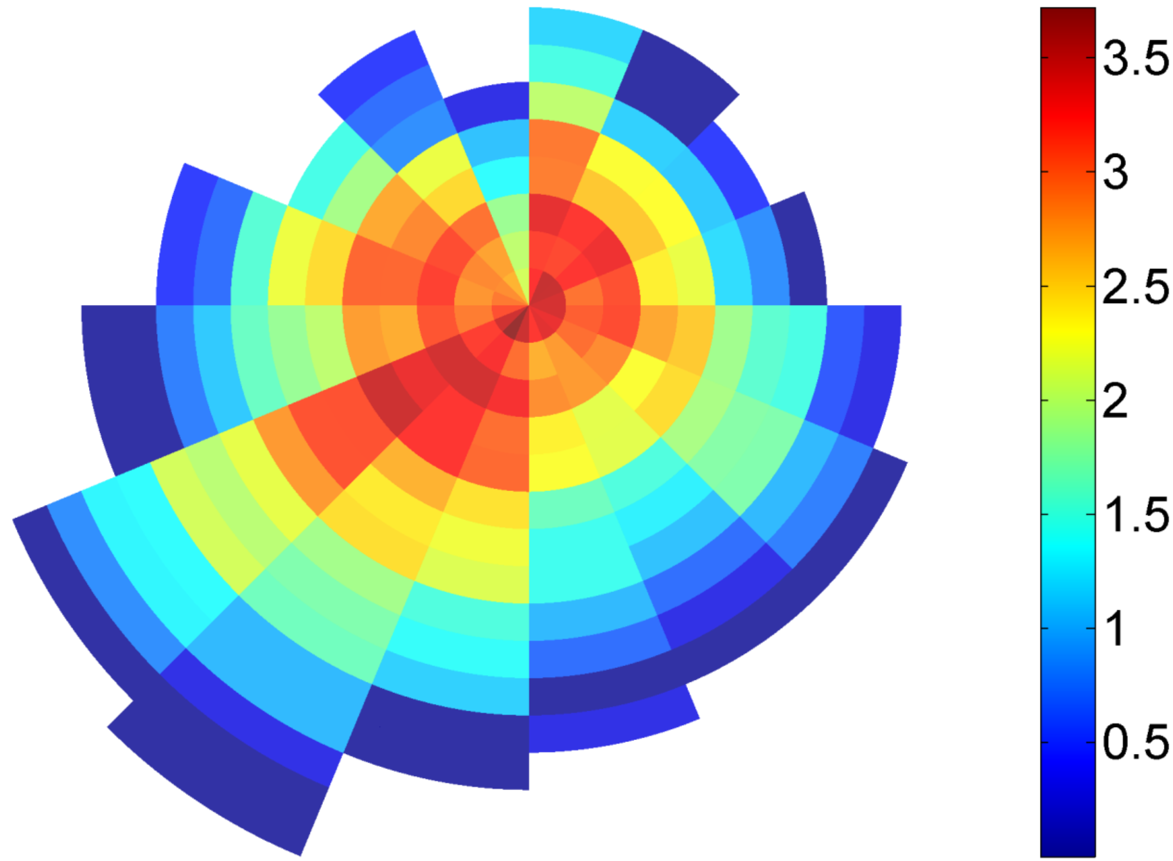

Temperature vs time
Temperature /degoC
Time /days
-10
-5
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60
70
Measurements
7/6/18 to 16/8/18



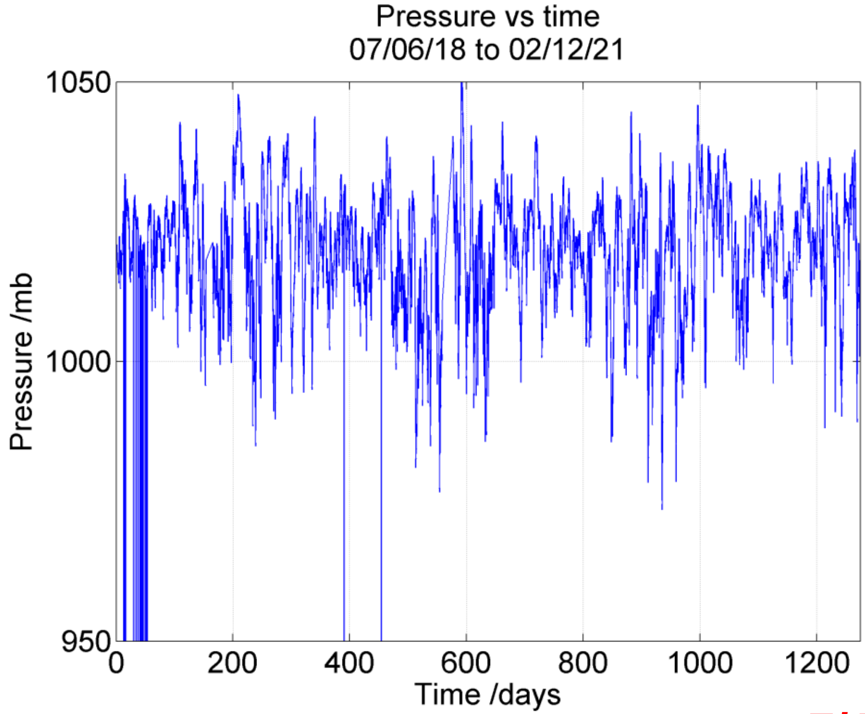


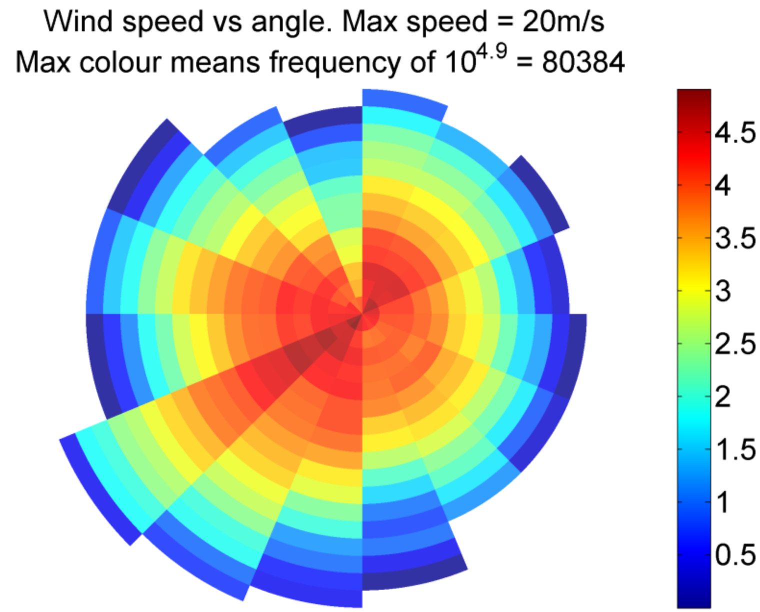
Measurements
7/6/18 to 16/8/18

Wind speed vs angle. Max speed = 20m/s
Max colour means frequency of $10^{3.7}$ = 5247

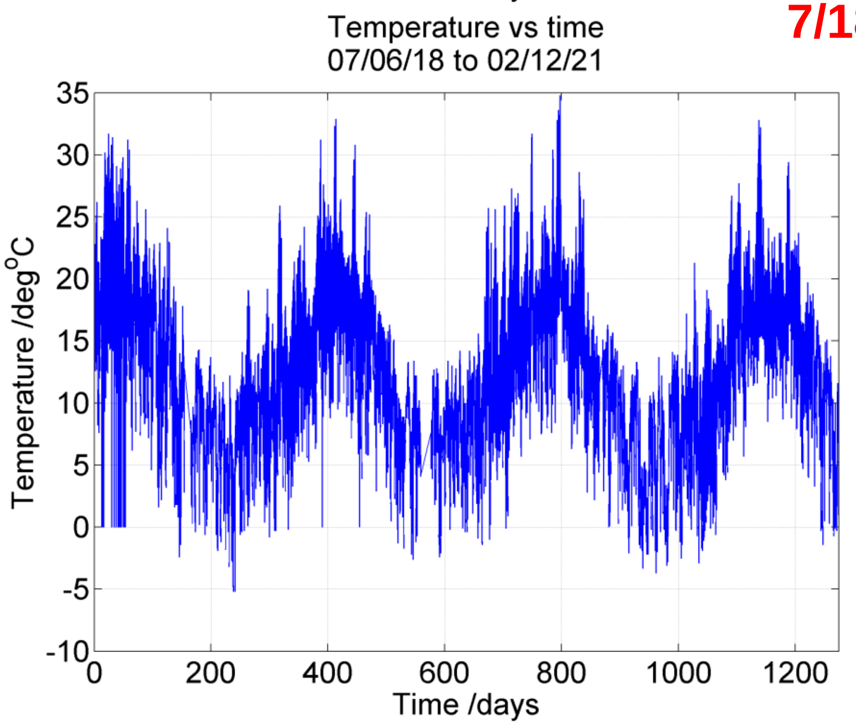


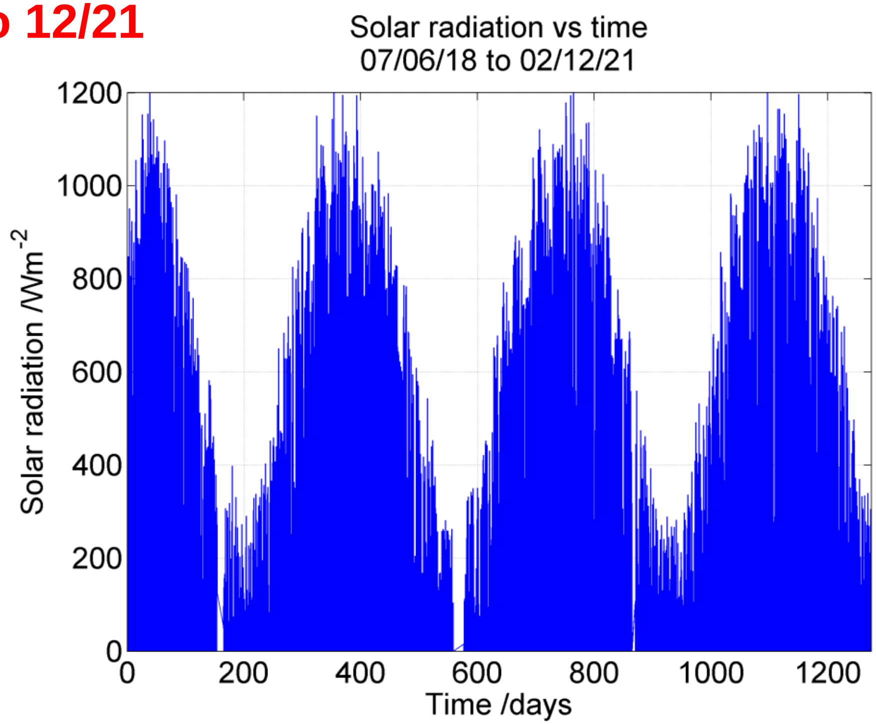
This ‘wind rose’ displays the frequency of wind measurements in circular sectors. Angle corresponds to 16 wind direction sectors (e.g. N, NNE etc) and range corresponds to wind speed. The colour scale is the *logarithm* of frequency.


Pressure vs time
07/06/18 to 02/12/21
Pressure /mb
1050
1000
950
0
200
400
600
800
1000
1200
Time /days


Wind speed vs angle. Max speed = 20m/s
Max colour means frequency of $10^{4.9}$ = 80384
4.5
4
3.5
3
2.5
2
1.5
1
0.5

7/18 to 12/21

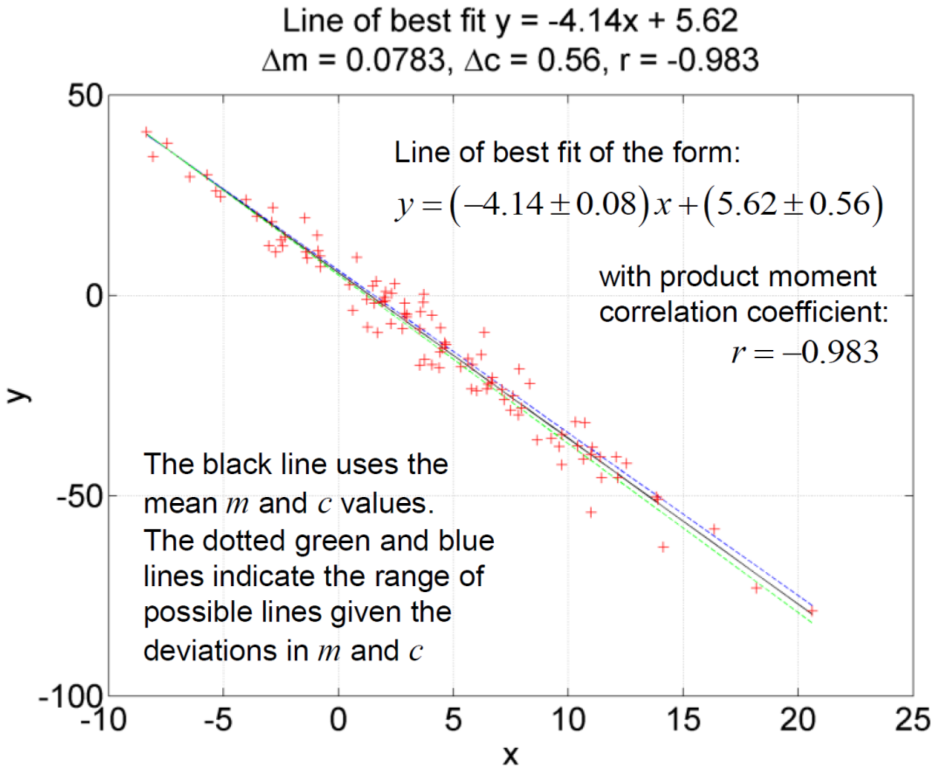

Temperature vs time
07/06/18 to 02/12/21
Temperature /deg°C
35
30
25
20
15
10
5
0
-5
-10
0
200
400
600
800
1000
1200
Time /days

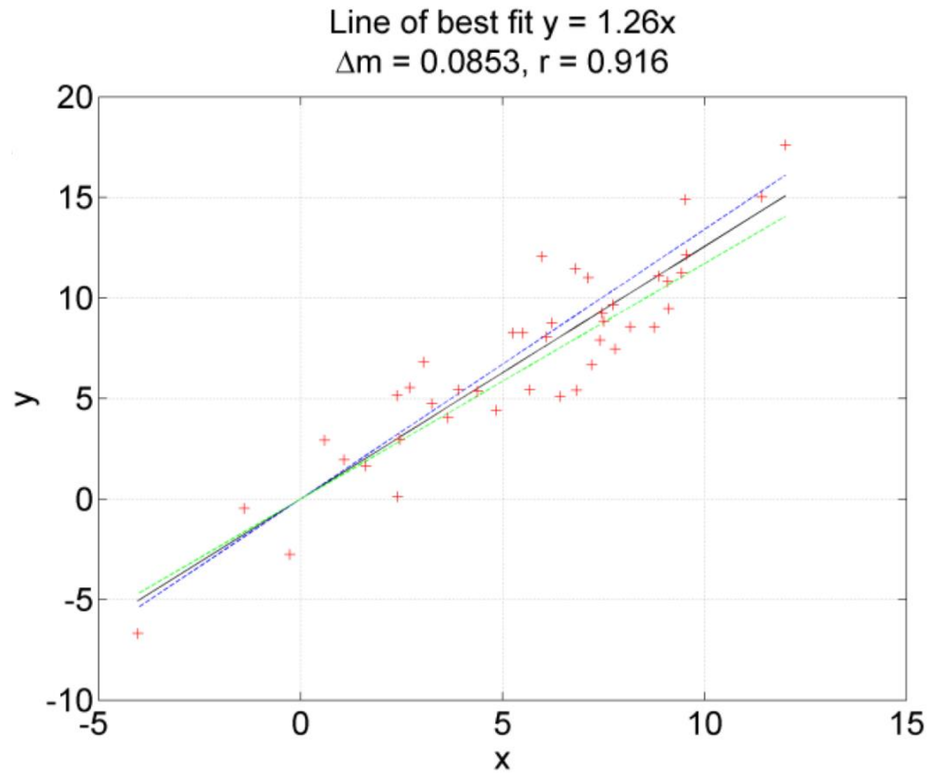

Solar radiation vs time
07/06/18 to 02/12/21
Solar radiation /Wm^{-2}
1200
1000
800
600
400
200
0
0
200
400
600
800
1000
1200
Time /days

A recipe for finding lines of best fit

$$y = mx + c$$

$$y = mx$$





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

N data point pairs (x,y)

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

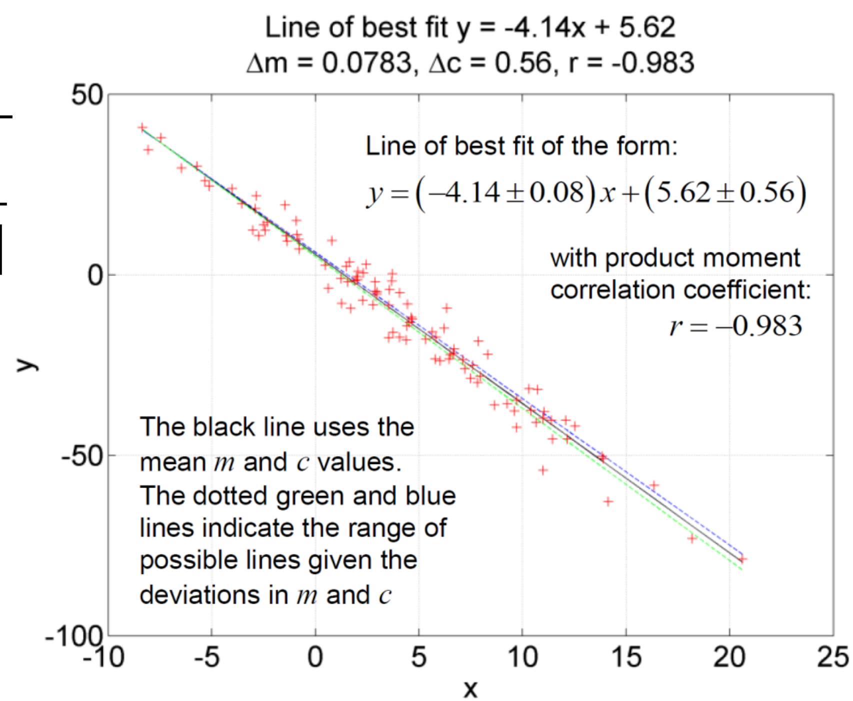
$$V[x] = \overline{x^2} - \bar{x}^2, \quad V[y] = \overline{y^2} - \bar{y}, \quad \mathrm{cov}[x,y] = \overline{xy} - \bar{x}\bar{y}$$

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

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

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

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



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

N data point pairs (x,y)

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

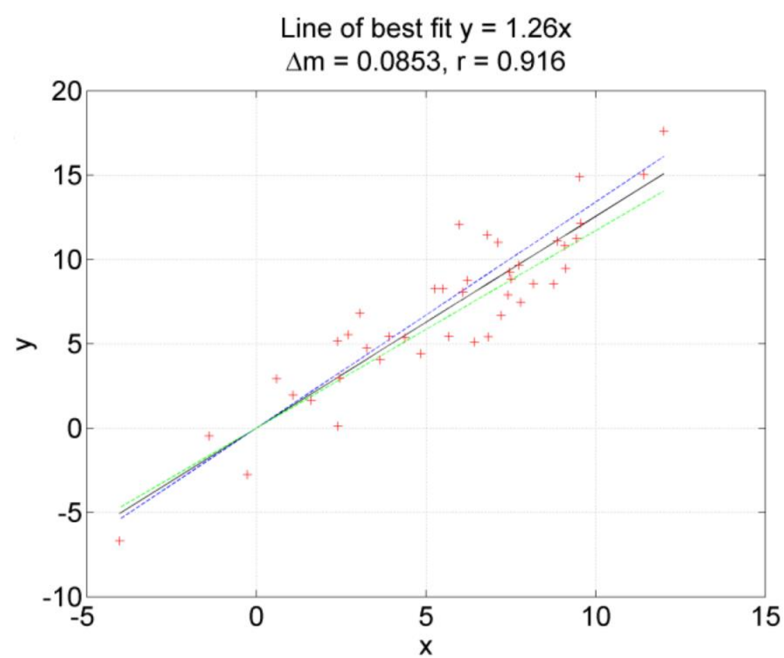
$$V[x] = \overline{x^2} - \overline{x}^2, \quad V[y] = \overline{y^2} - \overline{y}, \quad \mathrm{cov}[x,y] = \overline{xy} - \overline{xy}$$

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

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

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

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



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 $y = mx + c$

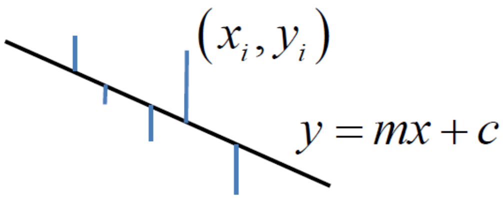
(2) Plot experimental (x,y) data on a graph and determine the **line of best fit** through the data.

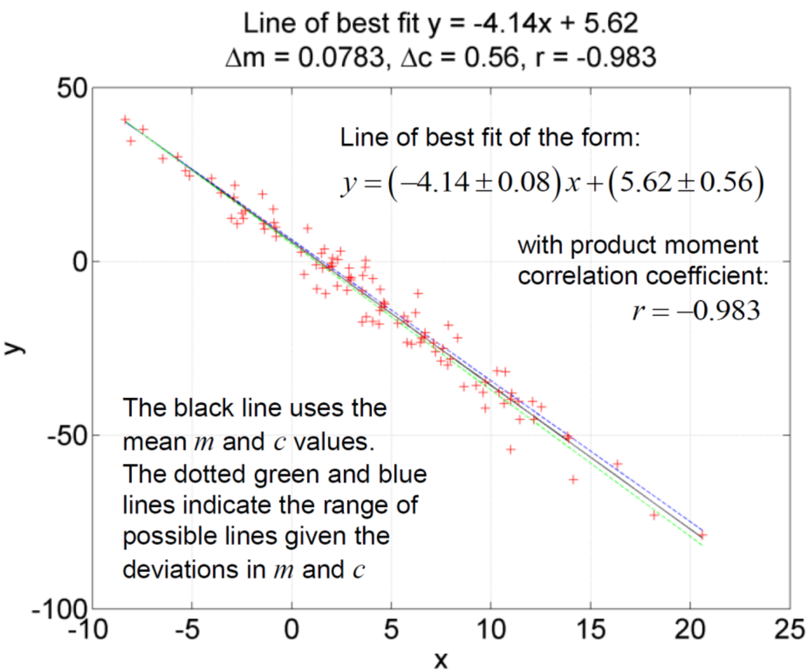
(3) Determine *gradient* m and *vertical intercept* c from the line of best fit.

(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.

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

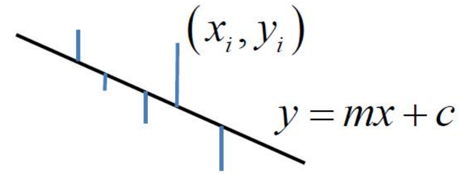




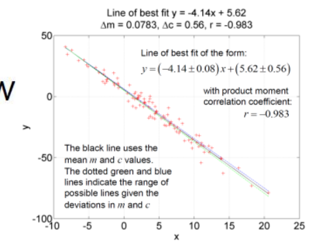
We can find S given a range of m and c values. Which pairing results in the *minimum* value of S?

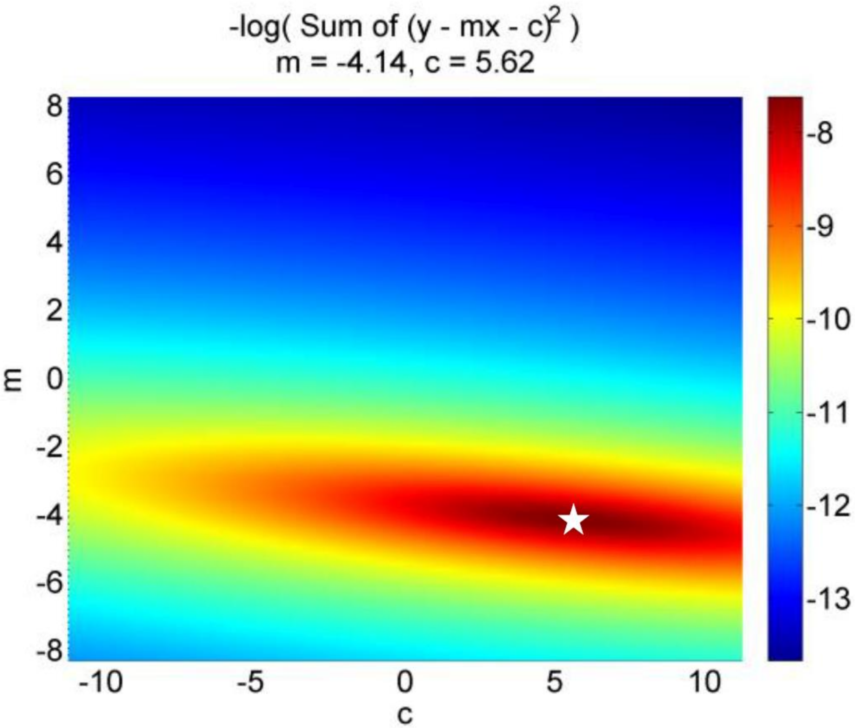
To determine the line of best fit*, let us sum the *squared* deviations of (x,y) from the line of best fit.

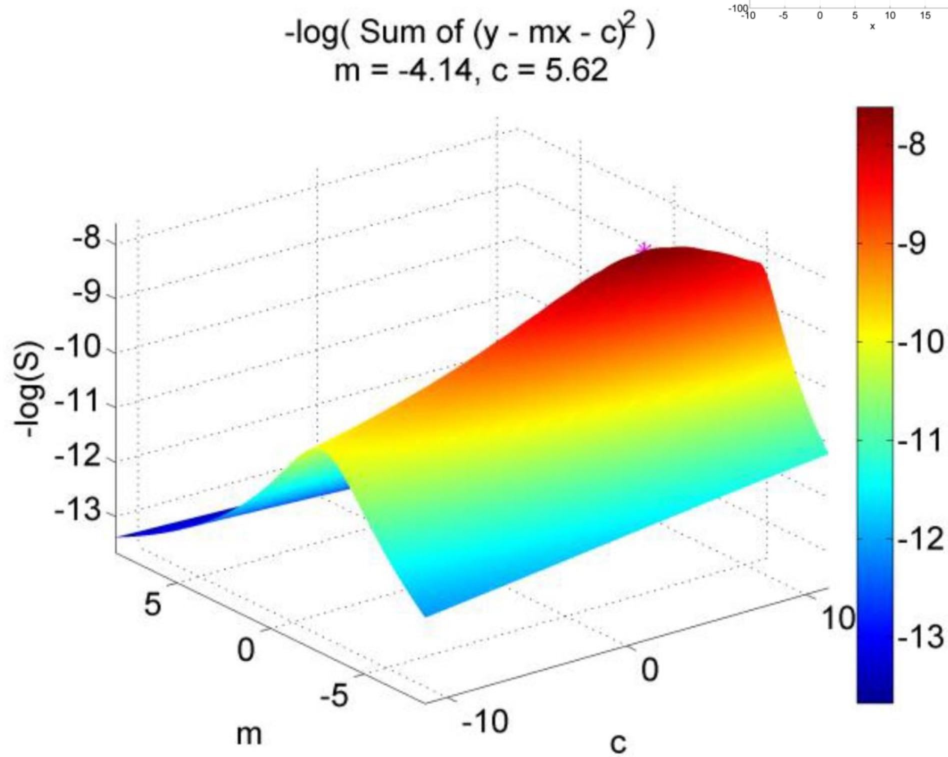
$$S = \sum_{i=1}^{N} \left(y_i - mx_i - c \right)^2$$



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 minimum of S can be found by differentiating S with respect to m and c, and setting these expressions equal to zero. Since S is a function of two variables we must use *partial derivatives*.

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

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

$$\therefore \frac{\partial S}{\partial m} = 0 \Rightarrow \sum_{i=1}^{N} x_i (y_i - mx_i - c) = 0$$

$$\therefore \sum_{i=1}^{N} x_i y_i - m\sum_{i=1}^{N} x_i^2 - c\sum_{i=1}^{N} x_i = 0$$

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

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

$$\therefore \frac{\partial S}{\partial c} = 0 \Rightarrow \sum_{i=1}^{N} (y_i - mx_i - c) = 0$$

$$\therefore \sum_{i=1}^{N} y_i - m\sum_{i=1}^{N} x_i - cN = 0$$

Hence:

$$\sum_{i=1}^{N} x_i y_i - m\sum_{i=1}^{N} x_i^2 - c\sum_{i=1}^{N} x_i = 0 \quad \therefore \overline{xy} - m\overline{x^2} - c\overline{x} = 0$$

$$\sum_{i=1}^{N} y_i - m\sum_{i=1}^{N} x_i - cN = 0 \quad \therefore \overline{y} - m\overline{x} - c = 0$$

Therefore:

$$\boxed{c = \overline{y} - m\overline{x}}$$

$$\therefore \overline{xy} - m\overline{x^2} - (\overline{y} - m\overline{x})\overline{x} = 0$$

$$\therefore m\left(\overline{x}^2 - \overline{x^2}\right) + \overline{xy} - \overline{y}\overline{x} = 0$$

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

If we repeat the analysis for the line: $x = My + d \Rightarrow M = \frac{\text{cov}[x,y]}{V[y]}$

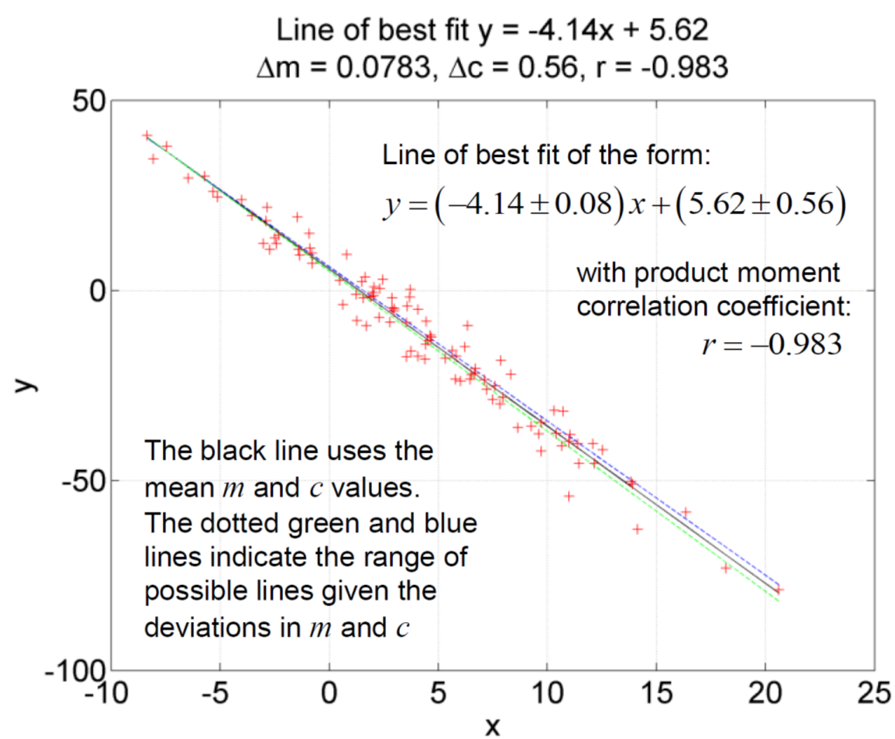
If this was the *same line but rearranged*: ← $y = mx + c$

$$M = \tfrac{1}{m} \quad \therefore mM = 1$$

Hence define a **product moment correlation coefficient:**

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

This will be +1 for a perfect positive correlation and -1 for a perfect negative correlation (i.e. $S = 0$ in both cases).



It is possible to show* that the standard deviations (i.e. 'errors') in m and c are:

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

$$\Delta c = \frac{s}{\sqrt{N}} \sqrt{1 + \frac{\overline{x}^2}{V[x]}}$$

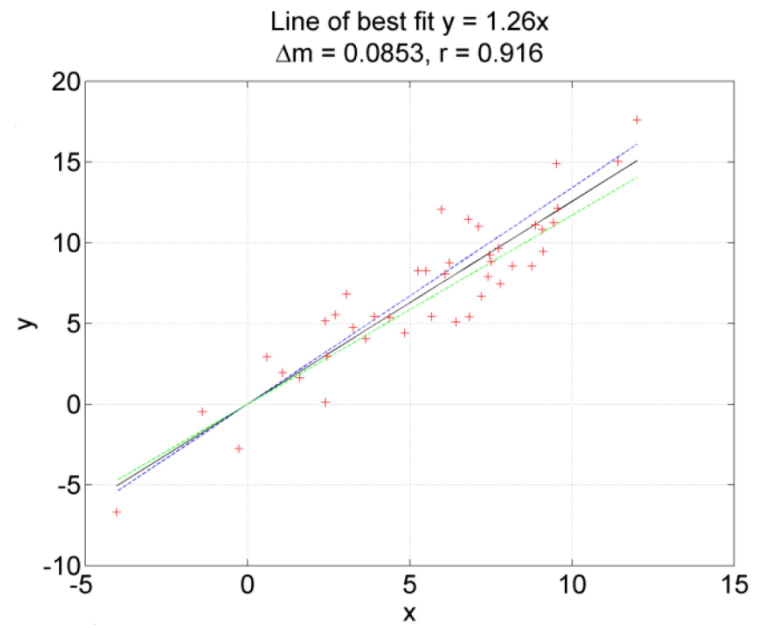
$$s = \sqrt{\frac{1}{N-2} \sum_{i=1}^{N} \left(y_i - m x_i - c \right)^2}$$

This is very useful in the physical sciences, as the errors in m and c will often be the uncertainties in model parameters (e.g. the strength of gravity…)

s is the *unbiased estimator* of the standard deviation in the y values from the line of best fit. The N-2 factor is due to two parameters (m and c) being used in the calculation, which are of course derived from the sample data themselves as shown above.

*http://mathworld.wolfram.com/LeastSquaresFitting.html

In many situations a **direct proportion** is asserted between y and x. The computation of the line of best fit (which passes through $(0,0)$ follows a similar argument to the one above.



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

$$\frac{\partial S}{\partial m} = 2\sum_{i=1}^{N} (y_i - mx_i)(-x_i)$$

$$\therefore \frac{\partial S}{\partial m} = 0 \Rightarrow \sum_{i=1}^{N} x_i (y_i - mx_i) = 0$$

$$\therefore \sum_{i=1}^{N} x_i y_i - m \sum_{i=1}^{N} x_i^2 = 0$$

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

The product moment correlation coefficient is the same as before but the standard deviation in m is slightly different since only *one* parameter is used in the computation of s (i.e. m).

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

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

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

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

N data point pairs (x,y)

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

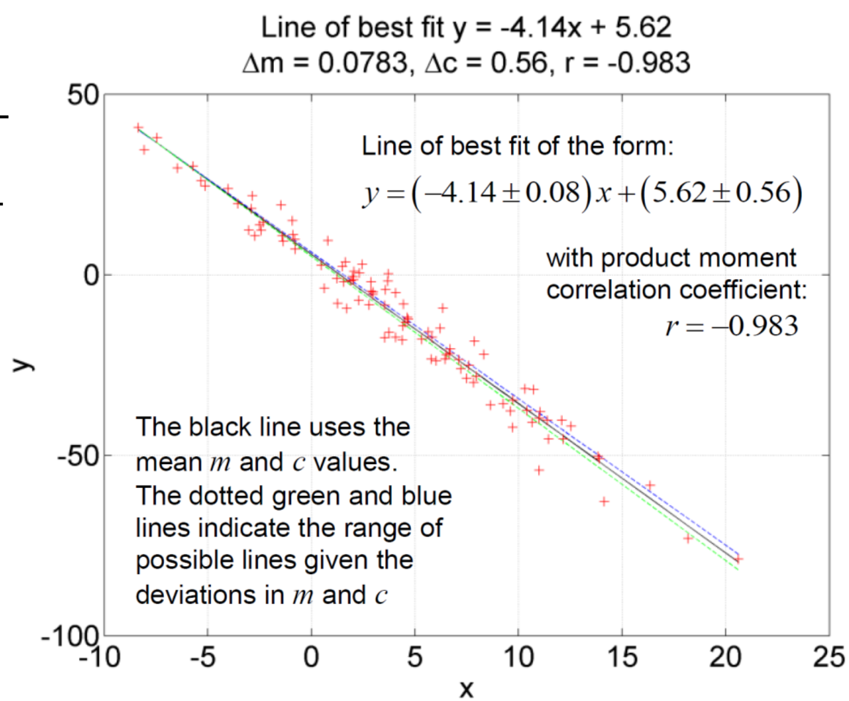
$$V[x] = \overline{x^2} - \bar{x}^2, \quad V[y] = \overline{y^2} - \bar{y}, \quad \mathrm{cov}[x,y] = \overline{xy} - \bar{x}\bar{y}$$

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

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

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

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



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

N data point pairs (x,y)

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

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

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

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

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

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