If we represent the vector of modelling functions as $\underset{\sim}{a}(t)$, we can collapse this:

$$\underset{\sim}{a}(t) \equiv \left(1, t, -\tfrac{1}{2}t^2\right)$$

Then

$$\sigma_{h_m}^2 = \left\| \underset{\sim}{a} \underset{\approx}{K} \right\|_2^2 \sigma_h^2$$

Let's generalize the regression analysis we developed above.

We have n observations b_i which we wish to fit with m parameters and modelling functions:

$$b_i \approx \sum_{j=1}^{m} x_j \phi_j(t_i)$$

or:

$$\underset{\sim}{b} \approx \underset{\approx}{A}\, \underset{\sim}{x}$$

where $A_{ij} = \phi_j(t_i)$

We define the residual:

$$\underset{\sim}{r} \equiv \underset{\sim}{b} - \underset{\approx}{A}\,\underset{\sim}{x}$$

We define the best fit modelling parameters $\underset{\sim}{x}$ by:

$$\min_{\underset{\sim}{x}} \| \underset{\sim}{r} \|_2^2 \equiv \min_{\underset{\sim}{x}} \left(\underset{\sim}{r}^T \underset{\sim}{r} \right)$$

$\underset{\sim}{x}$ has the solution:

$$\underset{\approx}{A}^T \underset{\approx}{A}\,\underset{\sim}{x} = \underset{\approx}{A}^T \underset{\sim}{b}$$

or

$$\underset{\sim}{x} = \left[\underset{\approx}{A}^T \underset{\approx}{A} \right]^{-1} \underset{\approx}{A}^T \underset{\sim}{b} \equiv \underset{\approx}{K}\,\underset{\sim}{b}$$

Note that $\underset{\approx}{A}$ is an $n \times m$ matrix, $\underset{\sim}{b}$ is a $n \times 1$ array, $\underset{\sim}{x}$ is a $m \times 1$ array and $\underset{\approx}{K}$ is a $m \times n$ matrix

If we recall that:

$$\sigma_z^2 = E\left\{ (z - \mu_z)^2 \right\}$$

where $\mu_z = E(z)$

Then the matrix of covariance of the regression parameters is:

$$\underset{\approx}{\Sigma}_x^2 \equiv E\left\{ (\underset{\sim}{x} - \underset{\sim}{\mu}_x)(\underset{\sim}{x} - \underset{\sim}{\mu}_x)^T \right\}$$

substituting $\underset{\sim}{x} = \underset{\approx}{K}\,\underset{\sim}{b}$ and $\underset{\sim}{\mu}_x = \underset{\approx}{K}\,\underset{\sim}{\mu}_b$ we get:

$$\underset{\approx}{\Sigma}_x^2 \equiv \underset{\approx}{K}\, E\left\{ (\underset{\sim}{b} - \underset{\sim}{\mu}_b)(\underset{\sim}{b} - \underset{\sim}{\mu}_b)^T \right\} \underset{\approx}{K}^T$$

$$= \underset{\approx}{K}\, \underset{\approx}{\Sigma}_b^2\, \underset{\approx}{K}^T$$

where $\underset{\approx}{\Sigma}_b^2$ is the matrix of covariance of the observations $\underset{\sim}{b}$.

If the observations are <u>independent</u> then $\underset{\approx}{\Sigma}_b^2$ will be a diagonal matrix.

If all of the observations have the same variance, then:

$$\underset{\approx}{\Sigma}_b^2 = \underset{\approx}{I}\, \sigma_b^2$$

and we get

$$\underset{\approx}{\Sigma}_x^2 = \sigma_b^2 \underset{\approx}{K}\underset{\approx}{K}^T$$

We may estimate σ_b^2 from:

$$\sigma_b^2 \approx s_b^2 \equiv \frac{1}{n-m} \| \underset{\approx}{A}\underset{\sim}{x} - \underset{\sim}{b} \|_2^2$$

$$= \frac{1}{n-m} \underset{\sim}{r}^T \underset{\sim}{r}$$

We may calculate the error in model predictions similarly. Suppose we wish to determine the model value at K times $\underset{\sim}{t}^{(m)}$. Thus:

$$\underset{\sim}{b}^{(m)} = \underset{\approx}{A}^{(m)} \underset{\sim}{x}$$

where $A_{ij}^{(m)} = \phi_j(t_i^{(m)})$

Note that $\underset{\approx}{A}^{(m)}$ is a $K \times m$ matrix.

What is the error in $\underset{\sim}{b}^{(m)}$? We have the matrix of covariance:

$$\underset{\approx}{\Sigma}^2_{b^{(m)}} \equiv E\left\{ \left(\underset{\sim}{b}^{(m)} - \underset{\sim}{\mu}_{b^{(m)}}\right)\left(\underset{\sim}{b}^{(m)} - \underset{\sim}{\mu}_{b^{(m)}}\right)^T \right\}$$

$$= \underset{\approx}{A}^{(m)} E\left\{ \left(\underset{\sim}{x} - \underset{\sim}{\mu}_x\right)\left(\underset{\sim}{x} - \underset{\sim}{\mu}_x\right)^T \right\} \underset{\approx}{A}^{(m)T}$$

$$= \underset{\approx}{A}^{(m)} \underset{\approx}{\Sigma}^2_x \underset{\approx}{A}^{(m)T}$$

where $\underset{\approx}{\Sigma}^2_x$ is the matrix of covariance determined before! In general, only the diagonal elements of $\underset{\approx}{\Sigma}^2_{b^{(m)}}$ are of interest:

$$\underset{\sim}{\sigma}^2_{b^{(m)}} = \text{diag}\left\{ \underset{\approx}{\Sigma}^2_{b^{(m)}} \right\}$$

which yields the error in the model predictions.

As a final note on linear regression, let us examine the case where the variance of $\underset{\sim}{\sigma}^2_b$ is <u>not</u> constant!

In this case we wish to weight the points in our regression analysis differently.

This is <u>weighted</u> linear regression!

Suppose we have the <u>diagonal</u> matrix $\underset{\approx}{\Sigma}_b^2$. The weighted problem is just:

$$\min_{\underset{\sim}{x}} \; (\underset{\approx}{A}\underset{\sim}{x} - \underset{\sim}{b})^T \left(\underset{\approx}{\Sigma}_b^2\right)^{-1} (\underset{\approx}{A}\underset{\sim}{x} - \underset{\sim}{b})$$

$$= \min_{\underset{\sim}{x}} \; \underset{\sim}{r}^T \left(\underset{\approx}{\Sigma}_b^2\right)^{-1} \underset{\sim}{r}$$

where $\left(\underset{\approx}{\Sigma}_b^2\right)^{-1}$ is a diagonal matrix whose elements are just $\frac{1}{\sigma_{b_i}^2}$.

Thus we weight each point by the inverse of the variance of that point. This makes sense. Suppose the reason why σ_{b_i} varied was that m_i points were <u>averaged</u> together to get b_i.

Then we have $\sigma_{b_i}^2 = \frac{\sigma_b^2}{m_i}$ and

thus the above regression yields:

$$\frac{1}{\sigma_b^2}\min_{\underset{\sim}{x}} \underset{\sim}{r}^T \begin{pmatrix} m_1 & & 0 \\ & m_2 & \\ 0 & & \ddots \\ & & & m_n \end{pmatrix} \underset{\sim}{r},$$

or the $\bar{i}^{th}$ point is counted m_i times. The result for $\underset{\sim}{x}$ would be the same (on average) as if you had used all of the points individually!

OK, what is the formula for $\underset{\sim}{x}$?

Just as before we take the gradient:

$$\nabla_{\underset{\sim}{x}} \left\{ \underset{\sim}{r}^T \left(\underset{\approx}{\Sigma}_b^{2^{-1}} \right) \underset{\sim}{r} \right\} = 0$$

which yields:

$$\underset{\approx}{A}^T \left(\underset{\approx}{\Sigma}_b^2 \right)^{-1} \underset{\approx}{A}\, \underset{\sim}{x} = \underset{\approx}{A}^T \left(\underset{\approx}{\Sigma}_b^2 \right)^{-1} \underset{\sim}{b}$$

or $\underset{\sim}{x} = \underset{\approx}{K}_w \underset{\sim}{b} \equiv \left\{ \underset{\approx}{A}^T \left(\underset{\approx}{\Sigma}_b^2 \right)^{-1} \underset{\approx}{A} \right\}^{-1} \underset{\approx}{A}^T \left(\underset{\approx}{\Sigma}_b^2 \right)^{-1} \underset{\sim}{b}$

with error:

$$\underset{\approx}{\Sigma}_x^2 = \underset{\approx}{K}_w \underset{\approx}{\Sigma}_b^2 \underset{\approx}{K}_w^T$$

ex 13

```
clear
echo on
format compact
% In this problem we examine the behavior of a dilute
% suspension of particles which is being sheared.  As the
% suspension is sheared (the motion which is produced
% when a fluid is confined between two concentric cylinders
% and one of them is rotated) the particles will tumble over
% one another.  This will lead to a random walk of the
% particles which can be characterized by a diffusion
% coefficient, much like a molecular diffusivity.  We can
% measure this quantity - the shear-induced self-diffusivity -
% by examining the random walk of a single tracer particle in
% a suspension of otherwise identical particles.  Because the
% random motion is due to the interaction of the tracer with
% other particles, the diffusivity will be identically zero if the
% concentration of the other particles is zero.
pause
% In our lab we measured the shear-induced self-diffusivity in
% the dilute limit.  If the particles are perfect spheres, theory
% suggests that the leading order term in the diffusivity
% should go as c^2 where c is the concentration.  Other
% models suggest that if the spheres are not perfect, the
% diffusivity should be proportional to c to leading order.  More
% recently, models have suggested that even for smooth spheres the
% presence of bounding walls gives rise to an O(c) diffusivity.
%
% In this example we examine data to determine which model best
% describes the experiments:
%
% c       diffusivity     error
% 0.01    2.37e-4         2.2e-5
% 0.025   5.63e-4         6.5e-5
% 0.05    1.42e-3         1.6e-4
% 0.075   2.27e-3         4.0e-4
% 0.10    4.06e-3         7.6e-4
% 0.15    9.96e-3         1.8e-3
%
% The errors given above are the one standard deviation errors
% in the measured diffusivities calculated from the statistics
% governing the measurement process.
pause
% Using this data, we wish to fit a constitutive equation for the
% diffusivity of the form:
%
%         Diffusivity = c * x(1) + c^2 * x(2) + c^3 * x(3)
%
% using weighted linear regression, and determine the error
% in each of the fitting coefficients.  In particular, we need to
% determine if the difference between the O(c) coefficient x(1) and
% zero is statistically significant.

% First we put in the data:
c=[.01,.025,.05,.075,.1,.15]';
```

```
diff=[2.37e-4,5.63e-4,1.4153e-3,2.27e-3,4.055e-3,9.965e-3]';
sigma=[2.2e-5,6.491228e-5, 1.617643e-4,3.98208e-4,7.605929e-4,1.775587e-3]';
pause
%Let's plot this data up:
figure(1)
errorbar (c,diff,sigma)
set(gca,'FontSize',18)
xlabel('concentration')
ylabel('diffusivity')
grid on
%As you can see, it certainly goes to zero as the
%concentration goes to zero.
pause
%The asymptotic behavior can be visualized better
%by plotting diff/c rather than just diff:
figure(2)
errorbar(c,diff./c,sigma./c,sigma./c,'o')
set(gca,'FontSize',18)
xlabel('concentration')
ylabel('diffusivity / concentration')
grid on
hold on
%Here there is clearly a non-zero asymptotic value
%of diff/c as c goes to zero.  This is what we are
%trying to capture.
pause
%OK, now for the weighted linear regression.  The
%unweighted linear regression problem is of the
%form:
%
%   min || A x - b ||
%
%In the weighted regression problem we weight each
%data point by the inverse of its variance when forming
%the sum of the squares of the deviation.  Let's do this:
%
%First we set up the matrix of modeling functions.  We
%have a linear term, a quadratic term, and a cubic term:
a=[c,c.^2,c.^3]
pause
%This is the weighting function:
vardiff=diag(sigma.^2);
weight=inv(vardiff)
pause
%We now define a matrix kw:
kw=inv(a'*weight*a)*a'*weight
%which can be used the conventional way:
x=kw*diff
%which has a lead coefficient different from zero.
pause
%Now for the error:
varx=kw*vardiff*kw'
xerror=diag(varx).^.5;
%So the coefficients and errors are:
[x,xerror]
%and in particular,
```

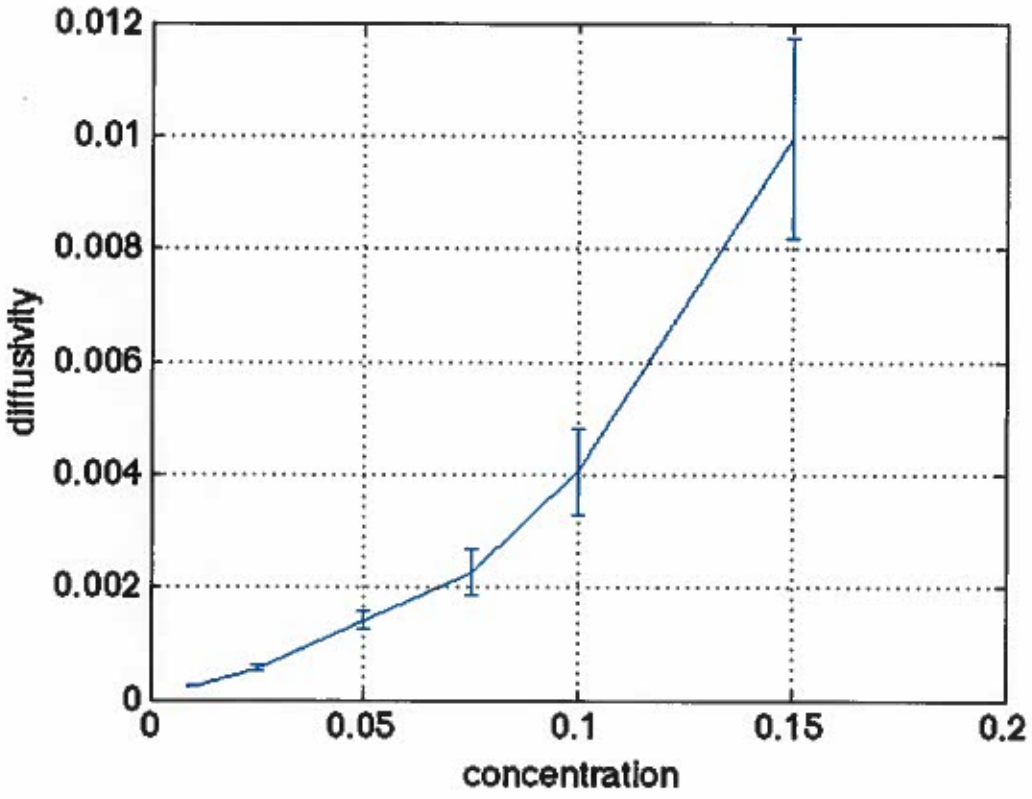
```
x(1)/xerror(1)
%So the order c coefficient is statistically different from
%zero (about 8 standard deviations).
pause
%Let's finish off with a bit of graphics.  Let's plot up
%the model predictions together with its uncertainty.
cp=[.001:.001:.15]';
ap=[cp,cp.^2,cp.^3];
dp=ap*x;
perror=diag(ap*varx*ap').^.5;
figure(2)
plot(cp,dp./cp,'b')
plot(cp,(dp+perror)./cp,'g')
plot(cp,(dp-perror)./cp,'g')
axis([0 .15 0 .1]);
%where we have plotted the 1-sigma confidence interval
%of the model function, based on the given standard deviations.
hold off
echo off
```

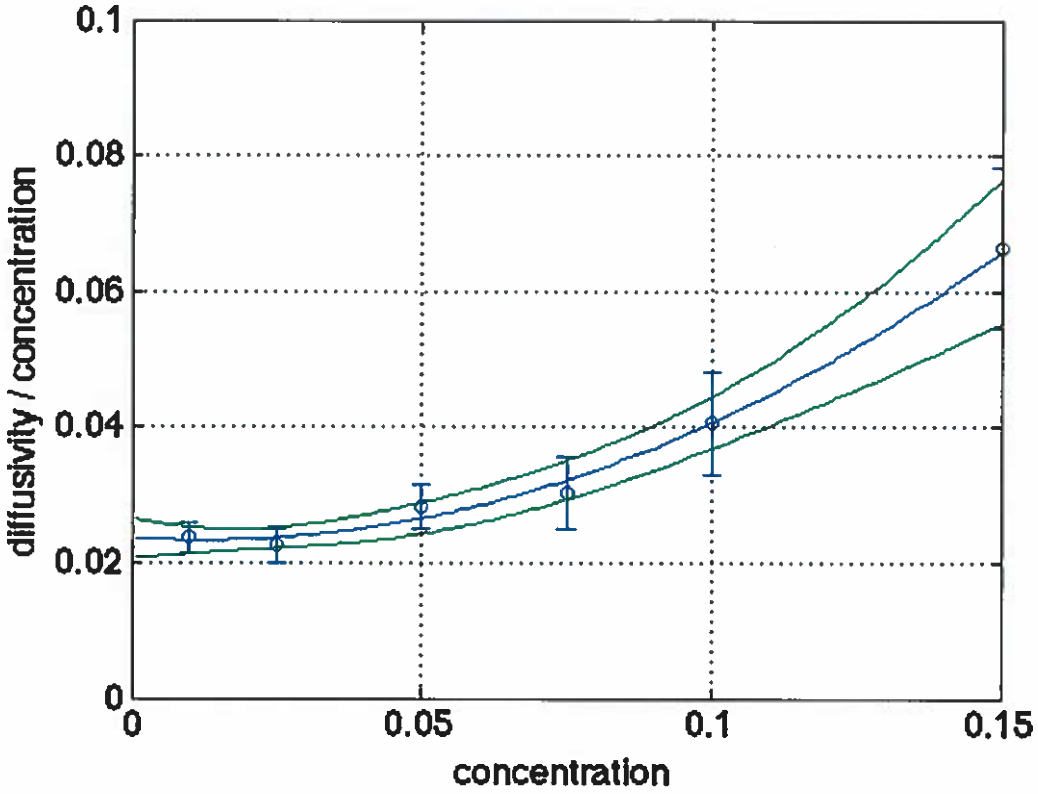
```
format compact
% In this problem we examine the behavior of a dilute
% suspension of particles which is being sheared.  As the
% suspension is sheared (the motion which is produced
% when a fluid is confined between two concentric cylinders
% and one of them is rotated) the particles will tumble over
% one another.  This will lead to a random walk of the
% particles which can be characterized by a diffusion
% coefficient, much like a molecular diffusivity.  We can
% measure this quantity - the shear-induced self-diffusivity -
% by examining the random walk of a single tracer particle in
% a suspension of otherwise identical particles.  Because the
% random motion is due to the interaction of the tracer with
% other particles, the diffusivity will be identically zero if the
% concentration of the other particles is zero.
pause
% In our lab we measured the shear-induced self-diffusivity in
% the dilute limit.  If the particles are perfect spheres, theory
% suggests that the leading order term in the diffusivity
% should go as c^2 where c is the concentration.  Other
% models suggest that if the spheres are not perfect, the
% diffusivity should be proportional to c to leading order.  More
% recently, models have suggested that even for smooth spheres the
% presence of bounding walls gives rise to an O(c) diffusivity.
%
% In this example we examine data to determine which model best
% describes the experiments:
%
% c       diffusivity     error
% 0.01    2.37e-4         2.2e-5
% 0.025   5.63e-4         6.5e-5
% 0.05    1.42e-3         1.6e-4
% 0.075   2.27e-3         4.0e-4
% 0.10    4.06e-3         7.6e-4
% 0.15    9.96e-3         1.8e-3
```

```
%
% The errors given above are the one standard deviation errors
% in the measured diffusivities calculated from the statistics
% governing the measurement process.
pause
% Using this data, we wish to fit a constitutive equation for the
% diffusivity of the form:
%
%         Diffusivity = c * x(1) + c^2 * x(2) + c^3 * x(3)
%
% using weighted linear regression, and determine the error
% in each of the fitting coefficients.  In particular, we need to
% determine if the difference between the O(c) coefficient x(1) and
% zero is statistically significant.
% First we put in the data:
c=[.01,.025,.05,.075,.1,.15]';
diff=[2.37e-4,5.63e-4,1.4153e-3,2.27e-3,4.055e-3,9.965e-3]';
sigma=[2.2e-5,6.491228e-5, 1.617643e-4,3.98208e-4,7.605929e-4,1.775587e-3]';
pause
%Let's plot this data up:
figure(1)
errorbar (c,diff,sigma)
set(gca,'FontSize',18)
xlabel('concentration')
ylabel('diffusivity')
grid on
%As you can see, it certainly goes to zero as the
%concentration goes to zero.
pause
%The asymptotic behavior can be visualized better
%by plotting diff/c rather than just diff:
figure(2)
errorbar(c,diff./c,sigma./c,sigma./c,'o')
set(gca,'FontSize',18)
xlabel('concentration')
ylabel('diffusivity / concentration')
grid on
hold on
%Here there is clearly a non-zero asymptotic value
%of diff/c as c goes to zero.  This is what we are
%trying to capture.
pause
%OK, now for the weighted linear regression.  The
%unweighted linear regression problem is of the
%form:
%
%   min || A x - b ||
%
%In the weighted regression problem we weight each
%data point by the inverse of its variance when forming
%the sum of the squares of the deviation.  Let's do this:
%
%First we set up the matrix of modeling functions.  We
%have a linear term, a quadratic term, and a cubic term:
a=[c,c.^2,c.^3]
a =
```

```
    0.0100    0.0001    0.0000
    0.0250    0.0006    0.0000
    0.0500    0.0025    0.0001
    0.0750    0.0056    0.0004
    0.1000    0.0100    0.0010
    0.1500    0.0225    0.0034
pause
%This is the weighting function:
vardiff=diag(sigma.^2);
weight=inv(vardiff)
weight =
   1.0e+09 *
    2.0661         0         0         0         0         0
         0    0.2373         0         0         0         0
         0         0    0.0382         0         0         0
         0         0         0    0.0063         0         0
         0         0         0         0    0.0017         0
         0         0         0         0         0    0.0003
pause
%We now define a matrix kw:
kw=inv(a'*weight*a)*a'*weight
kw =
   1.0e+04 *
    0.0115    0.0009   -0.0005   -0.0002   -0.0000    0.0000
   -0.3532    0.0251    0.0474    0.0126    0.0017   -0.0038
    2.1167   -0.3116   -0.3388   -0.0639    0.0127    0.0472
%which can be used the conventional way:
x=kw*diff
x =
    0.0237
   -0.0539
    2.2376
%which has a lead coefficient different from zero.
pause
%Now for the error:
varx=kw*vardiff*kw'
varx =
    0.0000   -0.0004    0.0024
   -0.0004    0.0195   -0.1500
    0.0024   -0.1500    1.3349
xerror=diag(varx).^.5;
%So the coefficients and errors are:
[x,xerror]
ans =
    0.0237    0.0030
   -0.0539    0.1395
    2.2376    1.1554
%and in particular,
x(1)/xerror(1)
ans =
    7.9682
%So the order c coefficient is statistically different from
%zero (about 8 standard deviations).
pause
%Let's finish off with a bit of graphics.  Let's plot up
%the model predictions together with its uncertainty.
```

```
cp=[.001:.001:.15]';
ap=[cp,cp.^2,cp.^3];
dp=ap*x;
perror=diag(ap*varx*ap').^.5;
figure(2)
plot(cp,dp./cp,'b')
plot(cp,(dp+perror)./cp,'g')
plot(cp,(dp-perror)./cp,'g')
axis([0 .15 0 .1]);
%where we have plotted the 1-sigma confidence interval
%of the model function, based on the given standard deviations.
hold off
echo off
```





Published with MATLAB® 7.12

Ok, this works if you know σ_b - but what if you don't?

This often occurs in model transformations when you are linearizing the model - Even if all the σ_b's are the same before the start, you don't have them the same afterwards!

If you can estimate the way error varies for meas., you can use error propagation to improve linear regression!

Let's look at Michaelis-Menten kinetics:

$$v = \frac{V_{max}[S]}{K_m + [S]}$$ ← substrate conc.

V_{max} ≡ max rate

K_m ≡ Michaelis Menten Cst
- measure of protein binding cst

The classic way of linearizing this is to take the inverse!

$$\frac{1}{v} = \frac{K_M + [S]}{V_{max}[S]}$$

$$= \underbrace{\frac{1}{V_{max}}}_{x_1} + \underbrace{\frac{K_M}{V_{max}}}_{x_2} \frac{1}{[S]}$$

This is known as a Lineweaver-Burk plot. It's very convenient but it really distorts the data!

Let $b = \frac{1}{v}$

Suppose we have some σ_v. What is σ_b? => Use error propagation:

$$\sigma_b^2 = \left(\frac{\partial b}{\partial v}\right)^2 \sigma_v^2$$

$$\frac{\partial b}{\partial v} = \frac{-1}{v^2} \quad \text{so} \quad \sigma_b^2 = \frac{\sigma_v^2}{v^4}$$

A huge dependence! How can

we avoid this? We use weighted regression!

Assume σ_w^2 is constant

$$b_i = \frac{1}{w_c}$$

$$\therefore \underset{\approx}{\Sigma}_b^2 = \sigma_w^2 \, \mathrm{diag}\left(\frac{1}{\underset{\sim}{w}^4}\right)$$

And:

$$\underset{\approx}{K_W} = \left\{\underset{\approx}{A}^T \left(\mathrm{diag}(\underset{\sim}{w}^4)\right) \underset{\approx}{A}\right\}^{-1} \underset{\approx}{A}^T \mathrm{diag}(\underset{\sim}{w}^4)$$

Note that multiplying $\underset{\approx}{\Sigma}_b^2$ by any constant doesn't affect $\underset{\approx}{K_W}$ (cancels out) - so you don't need to know σ_w, just assume it's constant!

This weights meas. at high rxn rate more, as they are more accurate! → in Lineweaver Burk plot