Finding error bars/PDF of model parameters

Last Time

- We learned how to find the “best-fit” parameter
  - Using least squares fitting
  - calculating chi2, finding where chi2 is at its min
- Today
  - We’ll go further with our chi2 statistics
  - Find error PDFs / bars around our “best-fit”

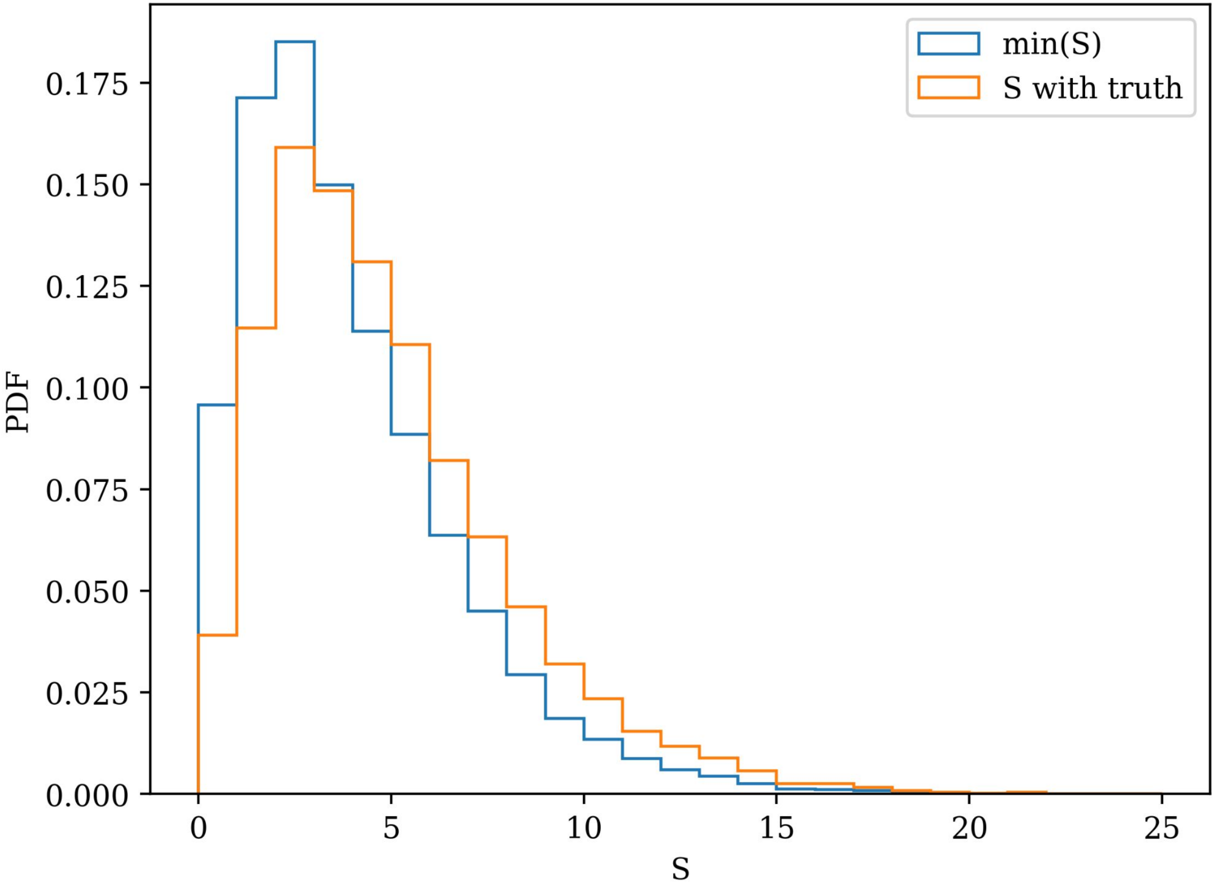
Previously we looked at the distribution of chi2 or S values assuming a = 10 m/s2, the true value

$S = \Sigma_i \left((v(t_i) - v_i) / \sigma_i \right)^2$

$v(t) = a*t$

What would happen if we didn't know the true value of a, and we instead used the best fit?

S got smaller

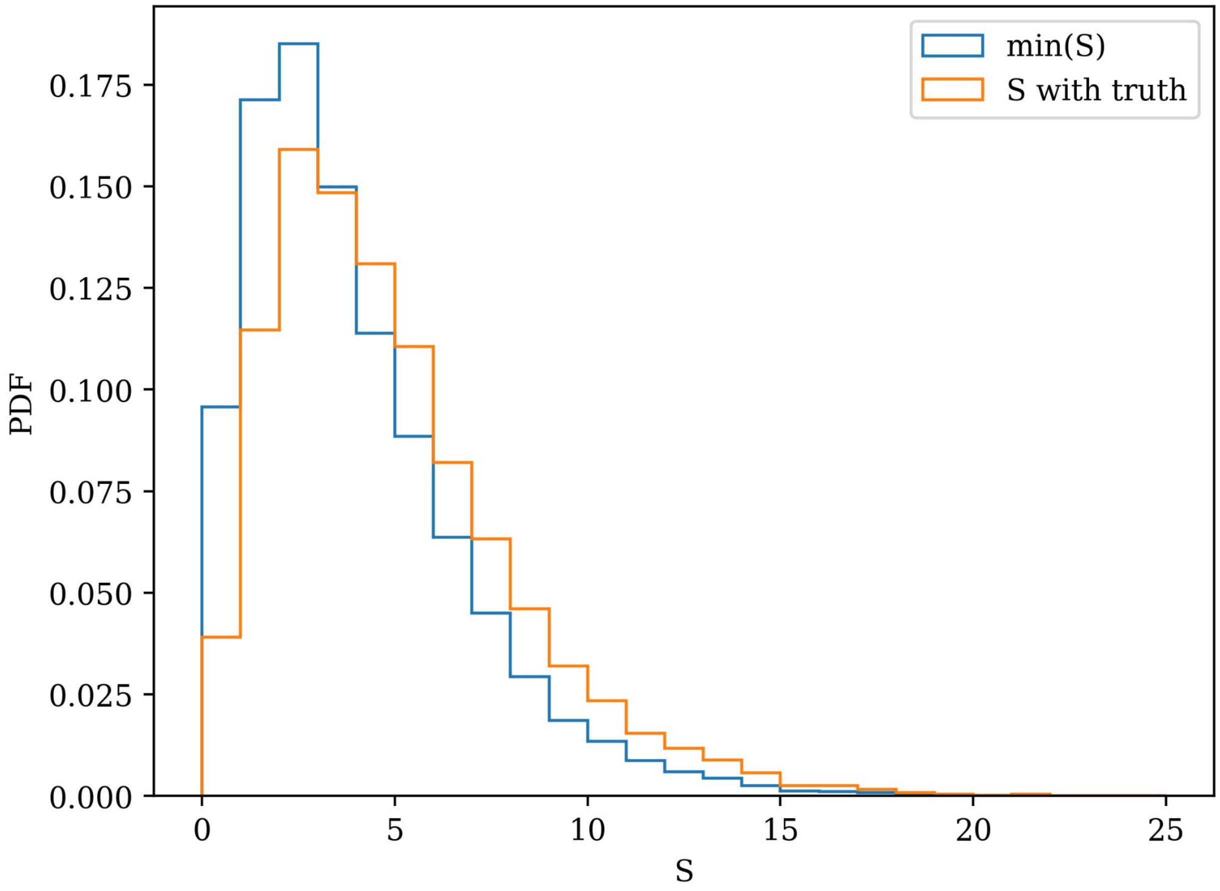


S got smaller

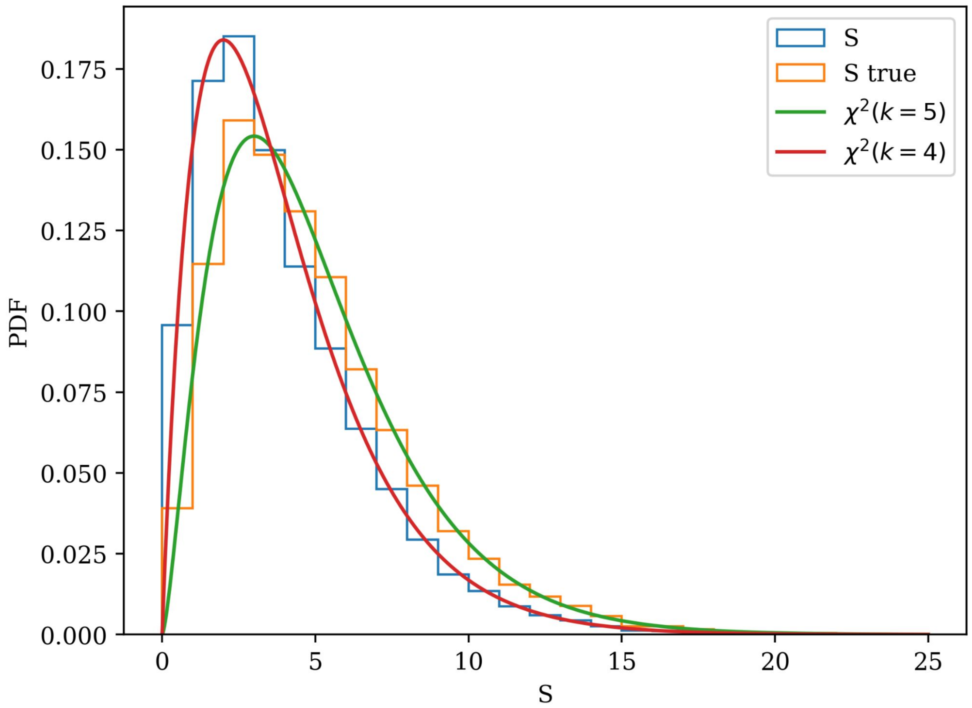
Makes sense

- we minimized it

But what happened to the distribution?

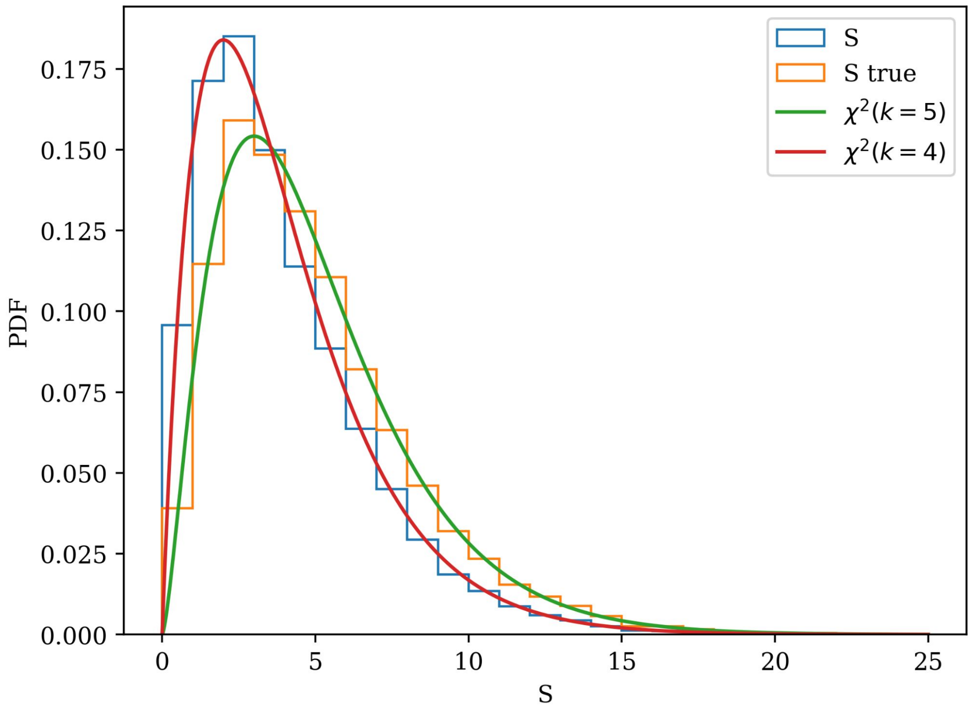


It now follows a chi2 with k = 4



It now follows a chi2 with k = 4

Why?



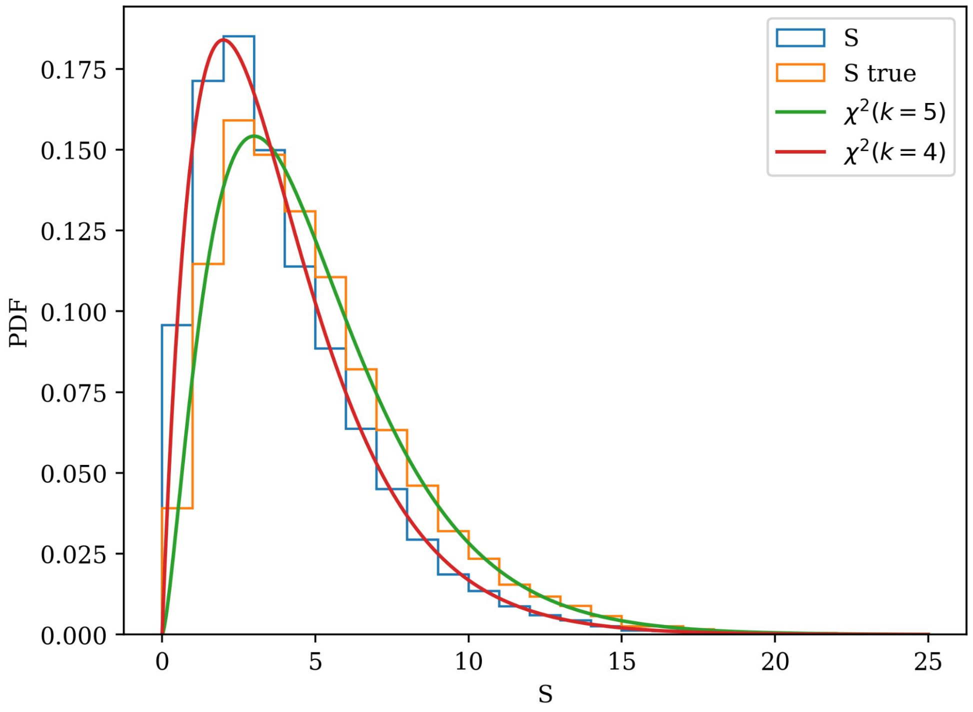
It now follows a chi2 with k = 4

Why?

S is a sum of random, independent normally distributed variables

If instead we first know the best fit slope,

It is no longer completely independent, we used the data to calculate m



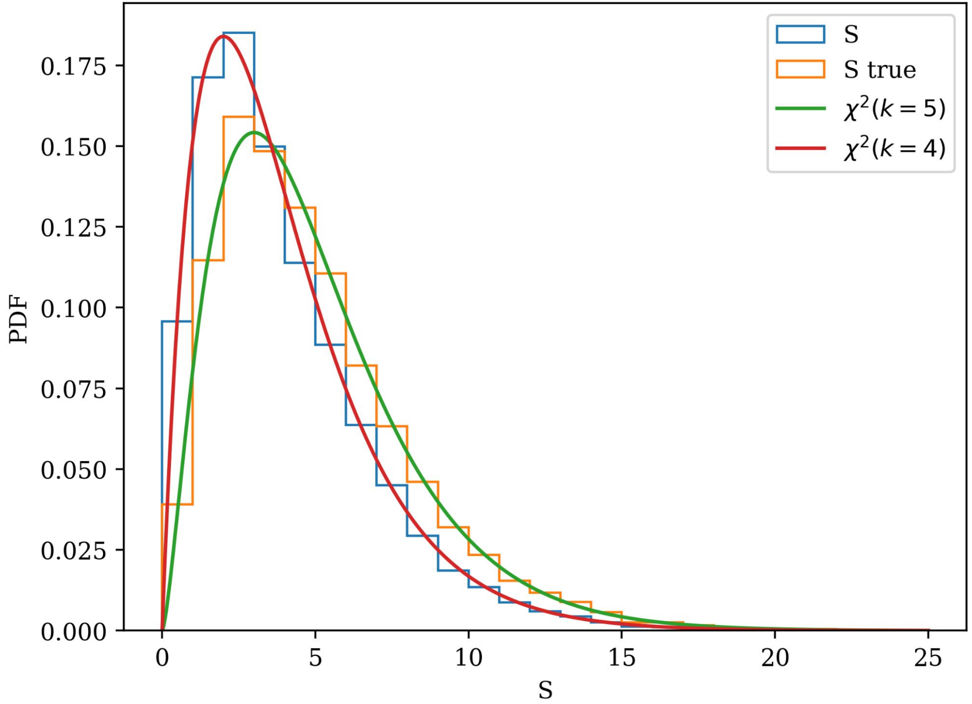
It now follows a chi2 with k = 4

Why?

S is a sum of random, independent normally distributed variables

If instead we first know the best fit slope,

It is no longer completely independent, we used the data to calculate m



If we already know best fit m, and the first 4 data point, could we calculate the 5th data point?

Statistical degrees of freedom

k is often called the degrees of freedom

If we optimize (minimize of chi2 or S) over free parameters we constrain the number of degrees of freedom

Statistical dof or k = N - m

N = independent data points

m = number of parameters we fit

Now we know chi2(a = Truth) and chi2(a = best a) will be different distributions

Best-fit a, is our best guess at what a is, but what can we say about the full distribution of where the true a could be

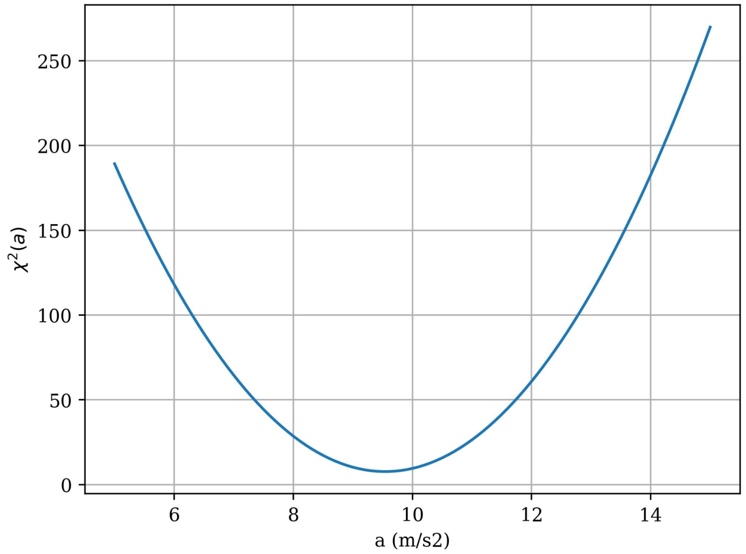
Now we know chi2(a = Truth) and chi2(a = best a) will be different distributions

Best-fit a, is our best guess at what a is, but what can we say about the full distribution of where the true a could be

Last time we had a chi2 curve as a function of a

chi2(a= best a) is at the min

chi2(a = Truth) is at some larger value



What if we looked at the difference

chi2(a = Truth) - min(chi2)

What if we looked at the difference

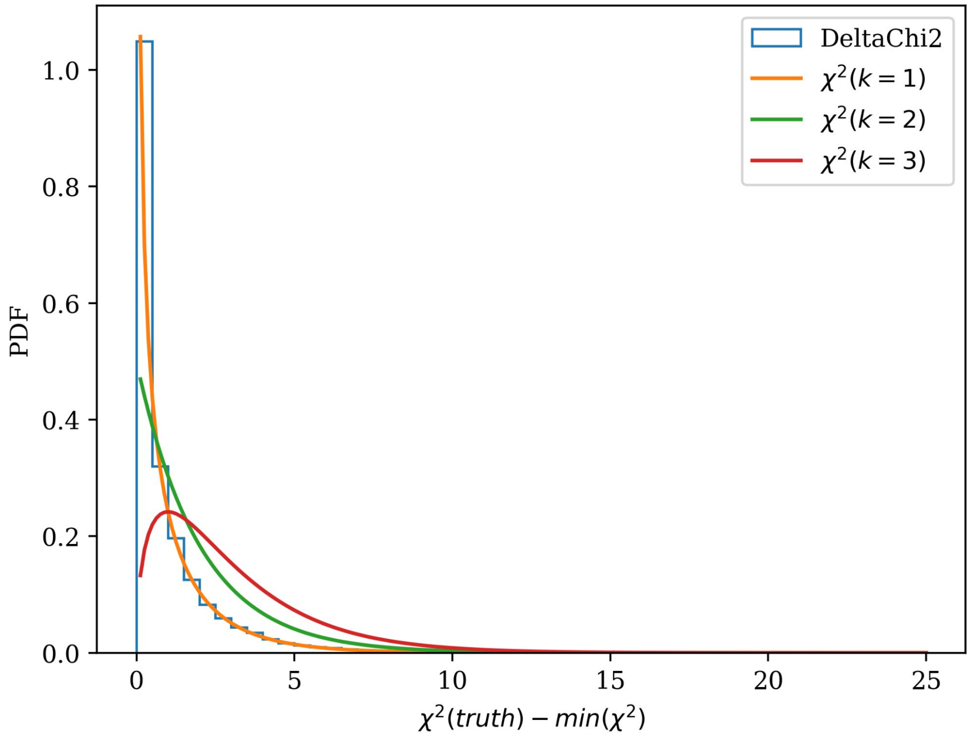
chi2(a = Truth) - min(chi2)

Run a bunch of the same experiment

What if we looked at the difference

chi2(a = Truth) - min(chi2)

It looks like a chi2 dist with k = 1!

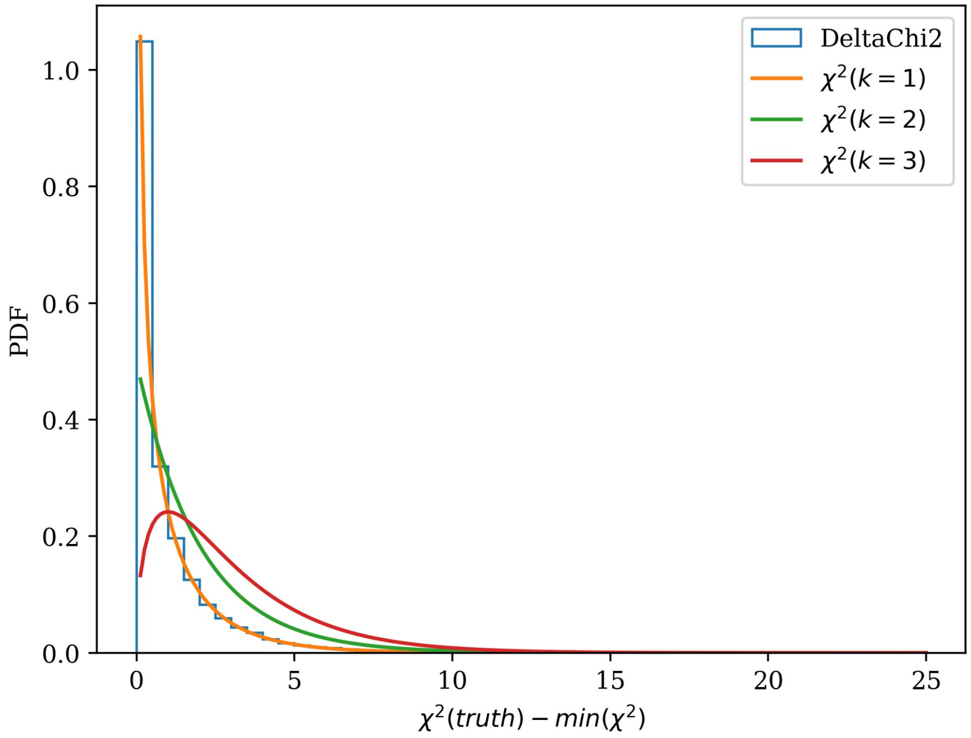


What if we looked at the difference

chi2(a = Truth) - min(chi2)

It looks like a chi2 dist with k = 1!

Why?



What if we looked at the difference

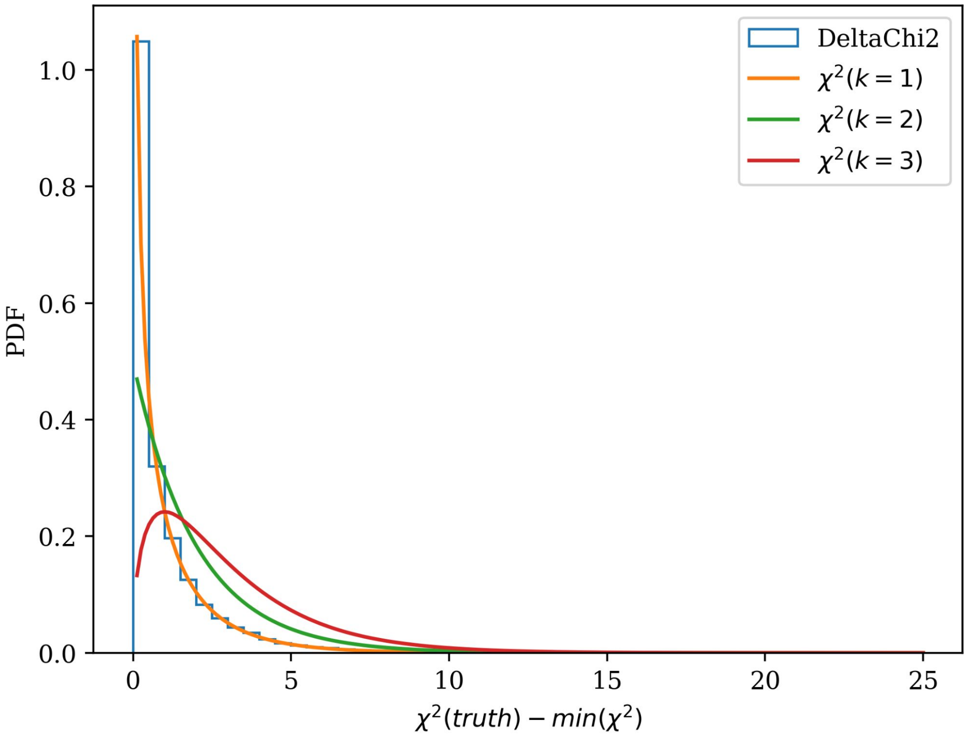
chi2(a = Truth) - min(chi2)

It looks like a chi2 dist with k = 1!

Why?

Both chi2’s are calculated using the same data

the only remaining degree of freedom is the best fit a



Cool, what do we do with that information?

Cool, what do we do with that information?

Given the min chi2, we now know the probabilistic distribution of offset to chi2(truth)

P(chi2 < 1, k=1) = 0.68

Cool, what do we do with that information?

Given the min chi2, we now know the probabilistic distribution of offset to chi2(truth)

P(chi2 < 1, k=1) = 0.68

- Can find this with stats.ppf(0.68, 1)

68% chance chi2(truth) is within (min(chi2), min(chi2) + 1)

Let’s look at that on our plot

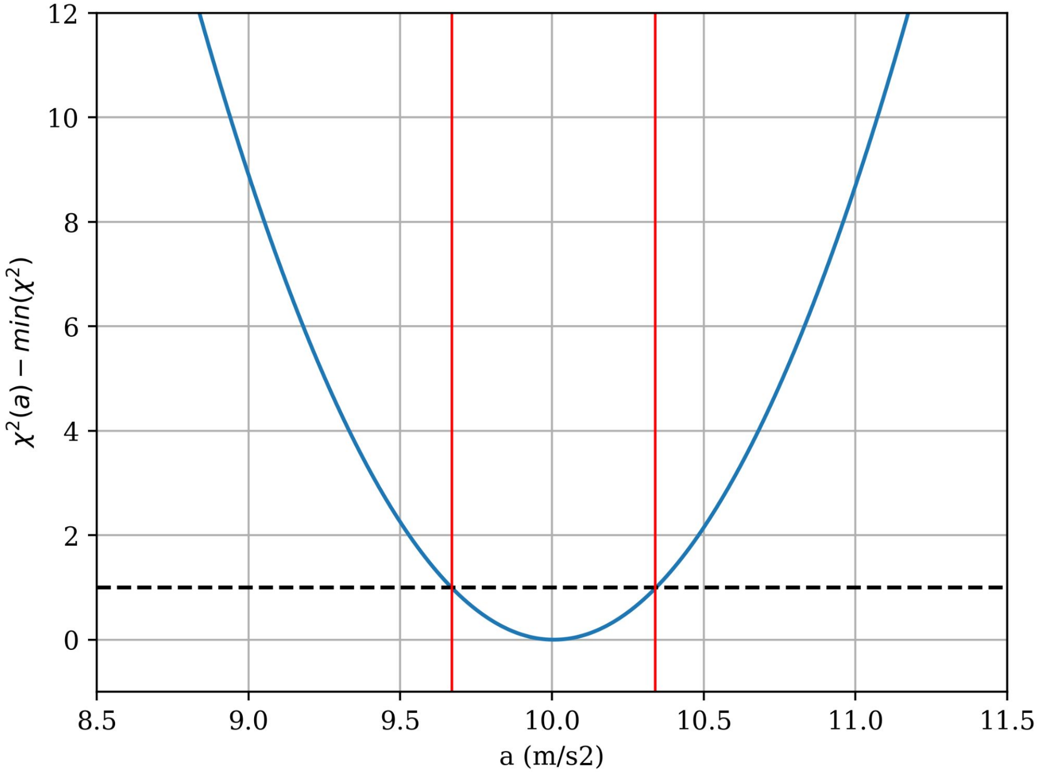
Then prob that a is in the region where

chi2 <= min(chi2) + 1

is also 68%

68% confidence that a is between 9.67 and 10.34 m/s2

a = 10 +/- 0.33 m/s2



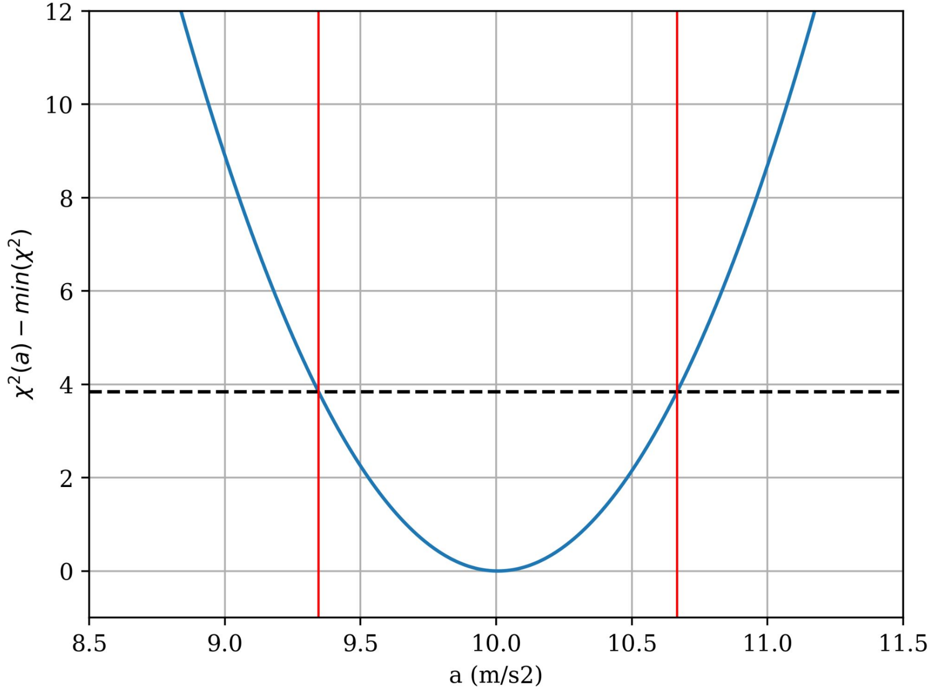
Different data than last lecture
Best-fit a ~ 10 m/s2

How about 95%?

stats.chi2.ppf(0.95, 1) = 3.84

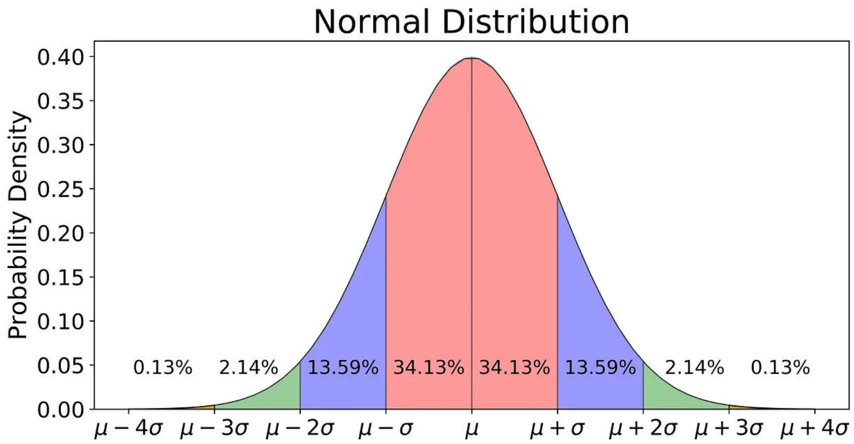
95% confidence 9.34 - 10.67

a = 10 +/- 0.67 m/s2



68% error -> 1 sigma confidence

95% error -> 2 sigma confidence



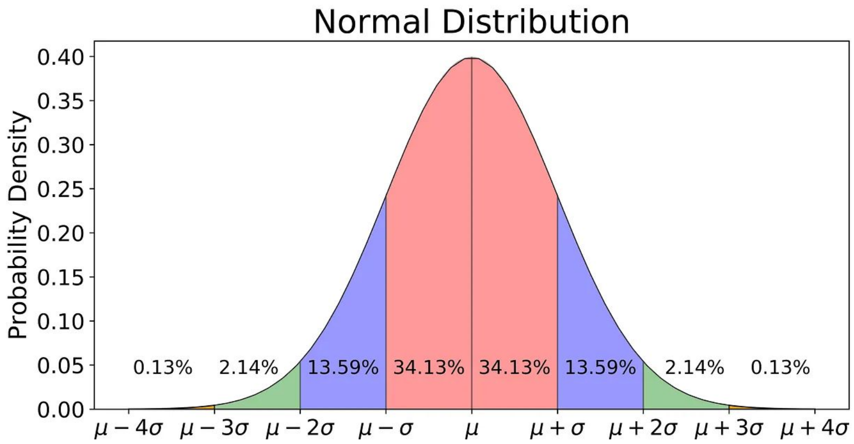
68% error -> 1 sigma confidence

95% error -> 2 sigma confidence

Our 68% error = +/- 0.33 m/s2

Our 95% error = +/- 0.66 m/s2

Scaling like a Gaussian!

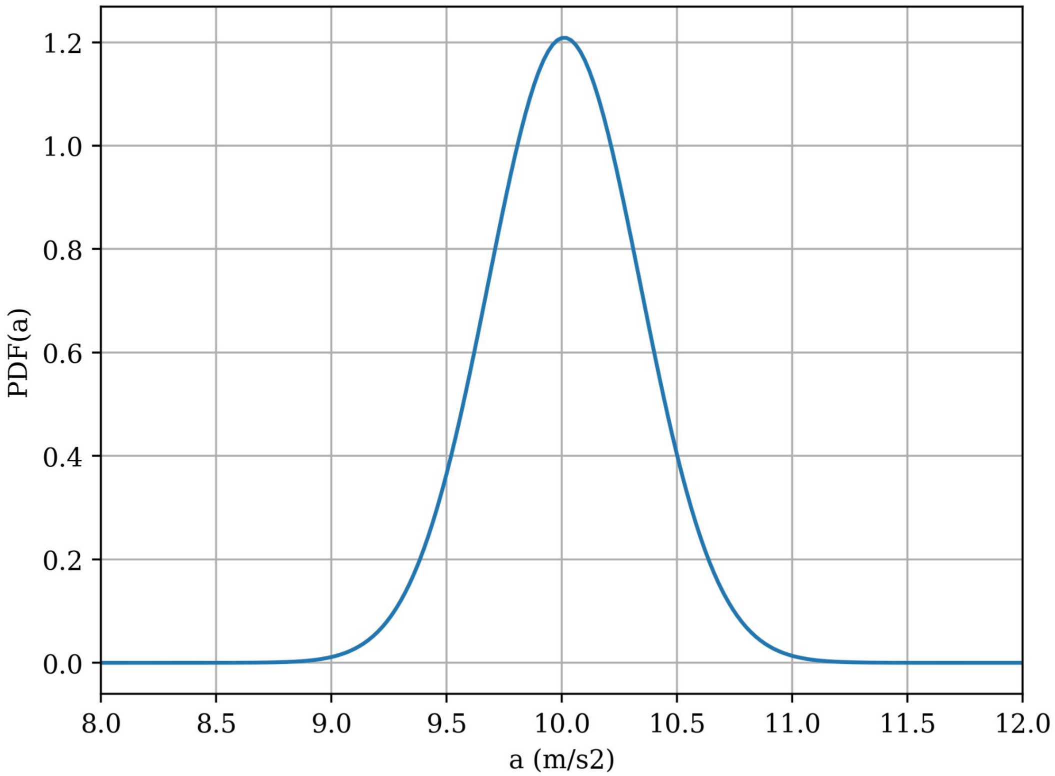


Sigma can be found from chi2(a)

Error PDF can than be a Gaussian with mu = best a

This will not always be the case, but is safe enough to assume with

- enough data
- Gaussian data errors



What if we don't know for sure that $v0 = 0$ m/s?

What if we don’t know for sure that $v0 = 0$ m/s?

Then we have another free parameter

$v(t) = a*t + v0$

What if we don’t know for sure that $v0 = 0$ m/s?

Then we have another free parameter

$v(t) = a*t + v0$

How do we find the best solution and errors now?

What if we don’t know for sure that v0 = 0 m/s?

Then we have another free parameter

v(t) = a*t + v0

How do we find the best solution and errors now?

chi2(a, v0)

Now it’s a 2D parameter space

To find the chi2 min we now have to scan over 2 dimensions

We can still brute force this, but this gets harder and harder to do in higher dimensions

To find the chi2 min we now have to scan over 2 dimensions

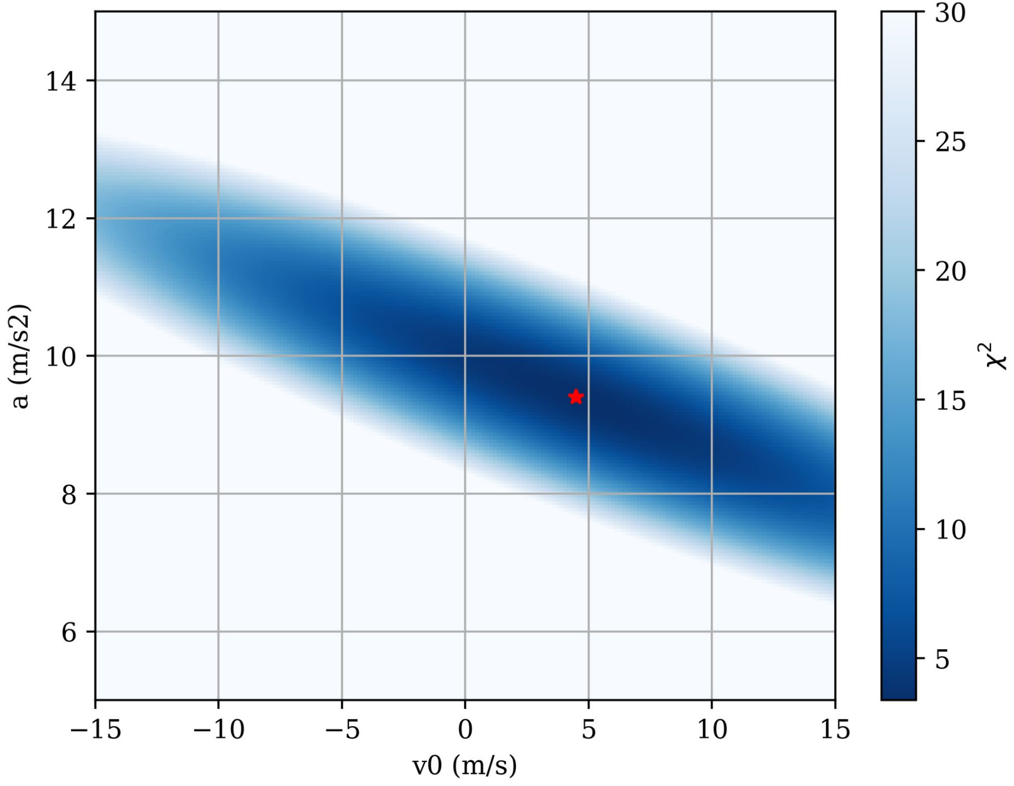
We can still brute force this, but this gets harder and harder to do in higher dimensions

- Need a 2D grid of a and v0 values
- Calculate chi2 at each grid point
- Find where the min chi2 is

To find the chi2 min we now have to scan over 2 dimensions

We can still brute force this, but this gets harder and harder to do in higher dimensions

- Need a 2D grid of a and v0 values
- Calculate chi2 at each grid point
- Find where the min chi2 is



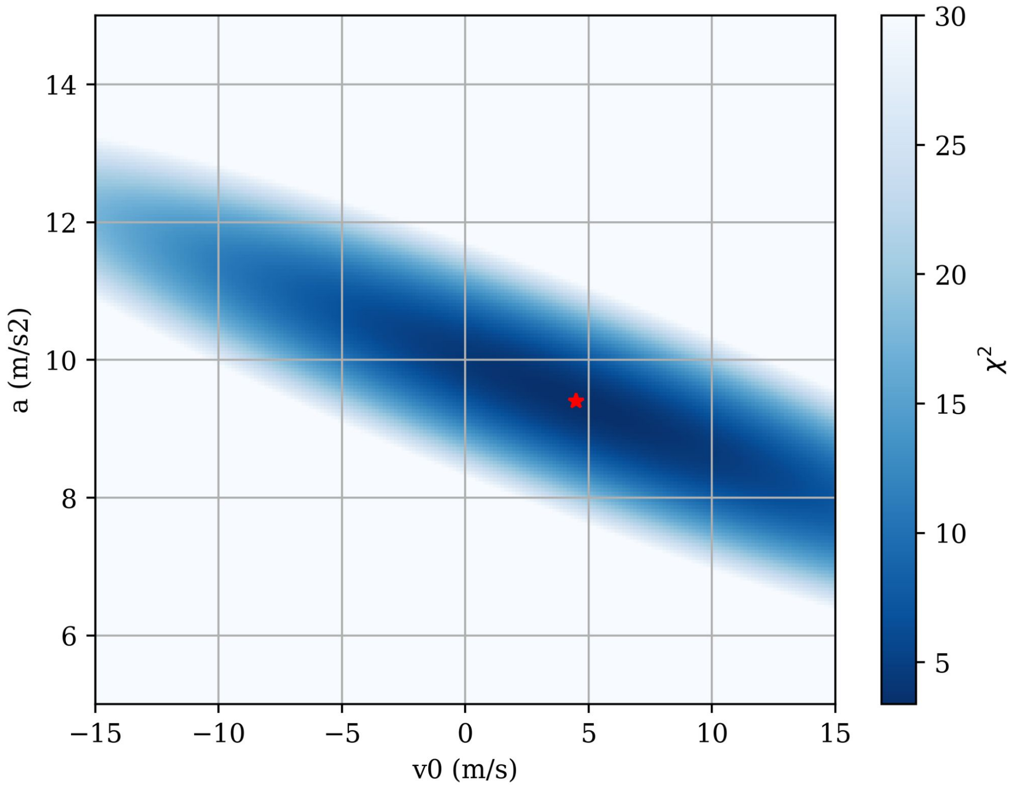
To find the chi2 min we now have to scan over 2 dimensions

We can still brute force this, but this gets harder and harder to do in higher dimensions

- Need a 2D grid of a and v0 values
- Calculate chi2 at each grid point
- Find where the min chi2 is

Best a = 9.4 m/s2

Best v0 = 4.5 m/s



What about the error PDF or confidence levels?

What about the error PDF or confidence levels?

We can still use Δchi2!

What about the error PDF or confidence levels?

We can still use Δchi2!

Though now we have 2 free parameters

- so 2 degrees of freedom ($k = 2$)

What about the error PDF or confidence levels?

We can still use Δchi2!

Though now we have 2 free parameters

- so 2 degrees of freedom (k = 2)

But how do we map the 1D bounds to 2D?

What about the error PDF or confidence levels?

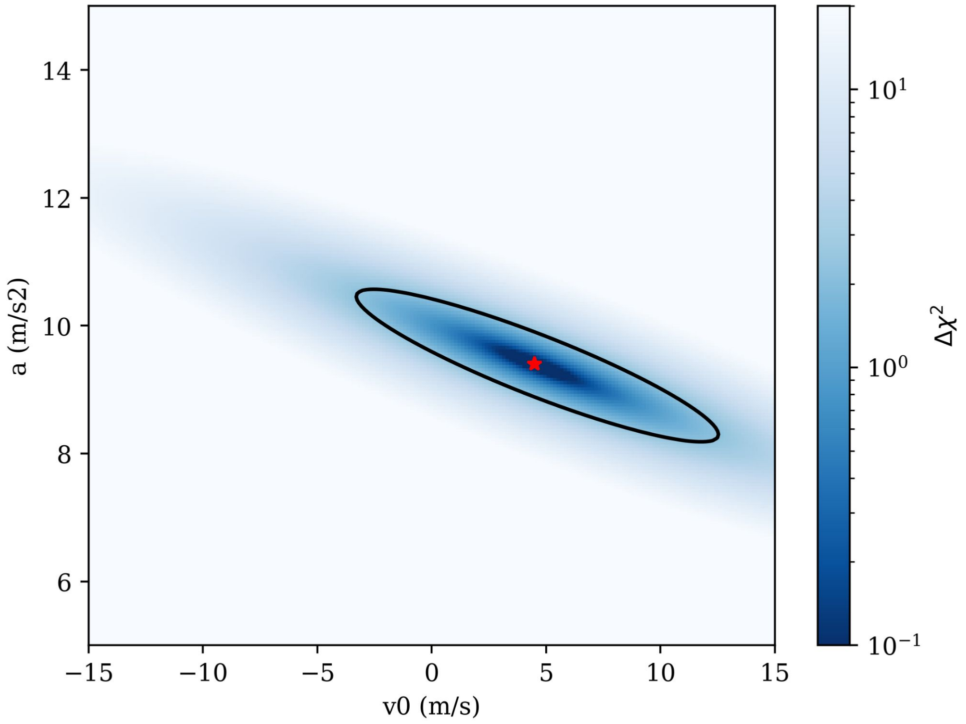
We can still use Δchi2!

Though now we have 2 free parameters

- so 2 degrees of freedom (k = 2)

But how do we map the 1D bounds to 2D?

- A 2D contour!

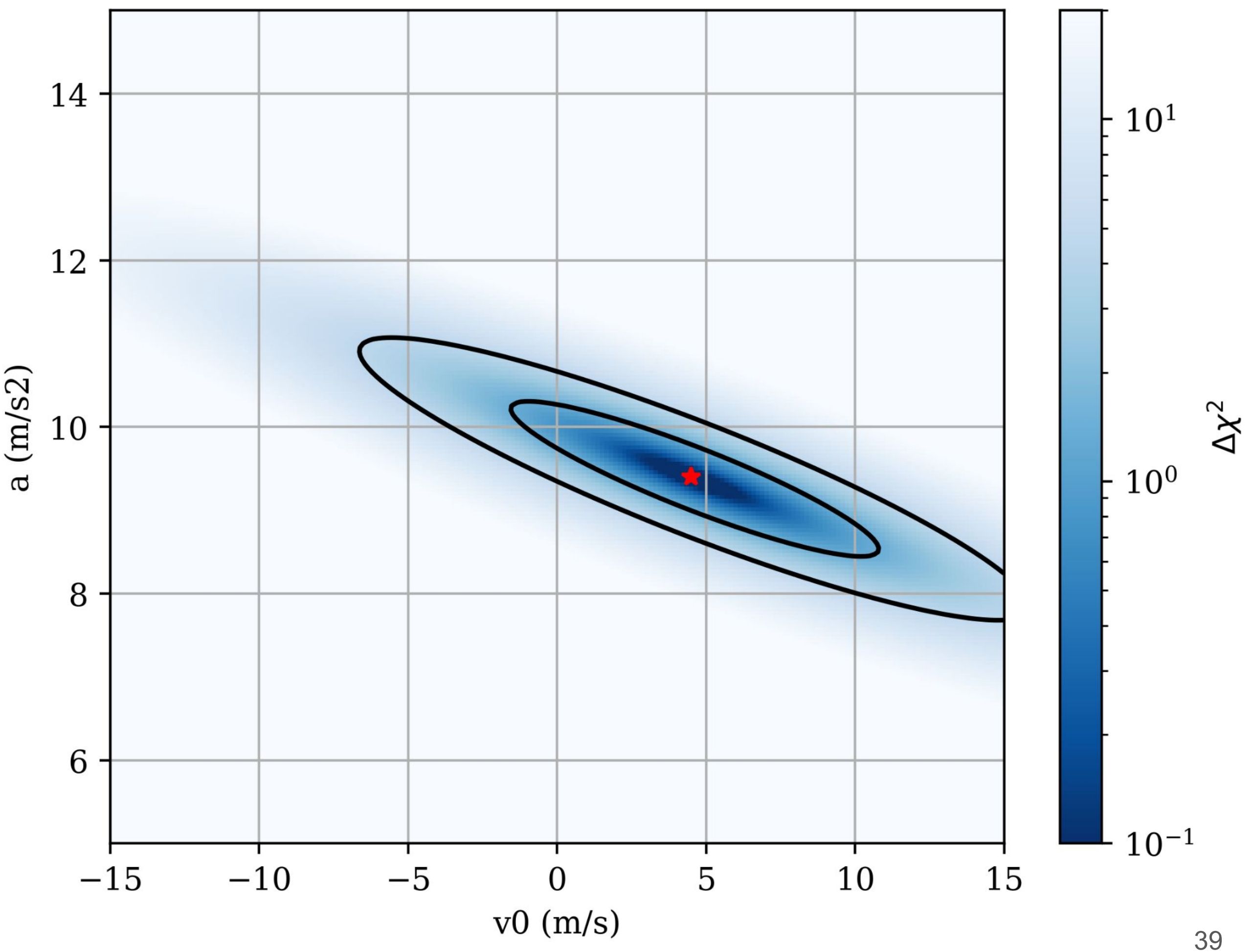


68% contour

stats.chi2.ppf(0.5, 2) = 1.39

stats.chi2.ppf(0.9, 2) = 4.6

90% probability true a and v0 is inside the 90% contour

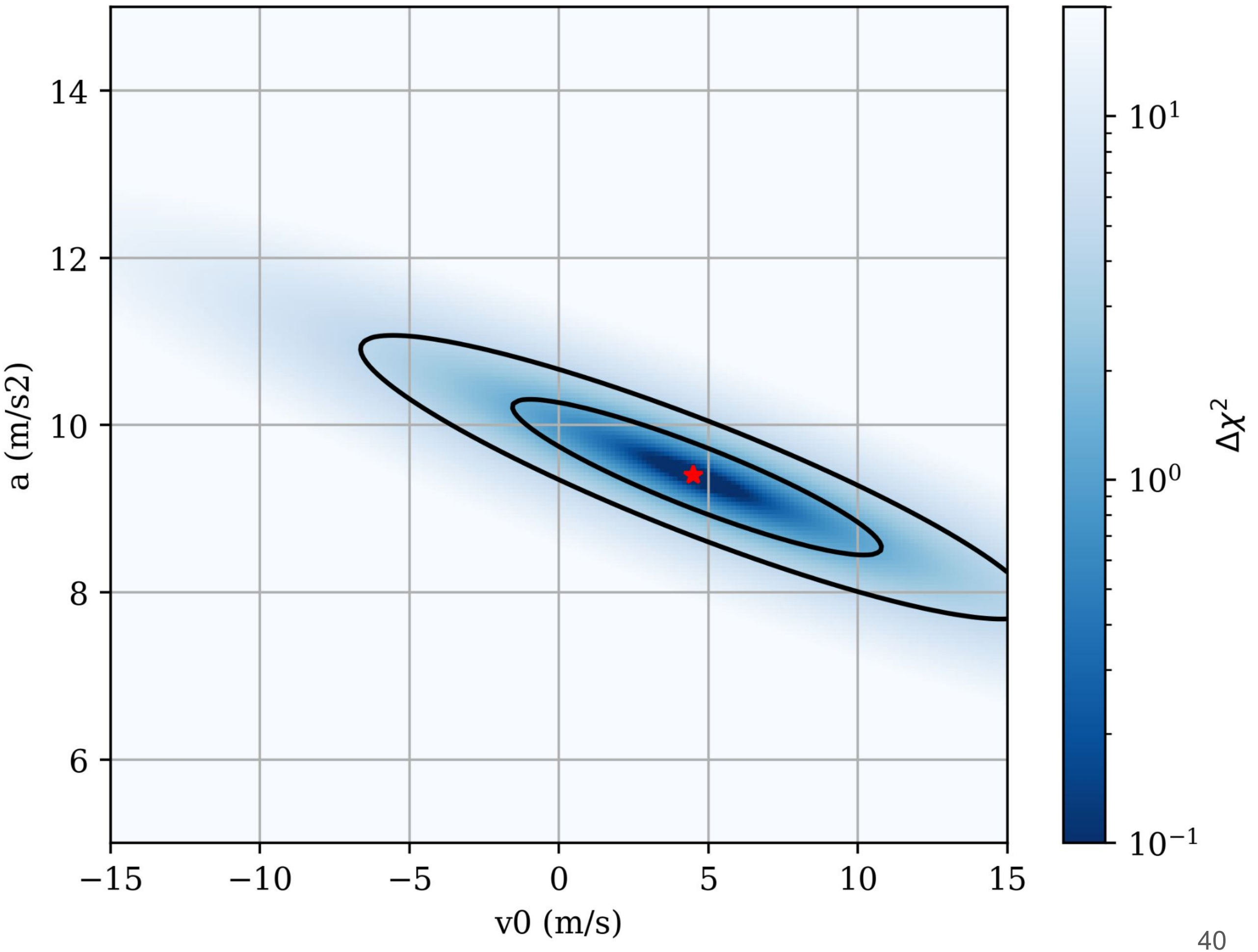


stats.chi2.ppf(0.5, 2) = 1.39

stats.chi2.ppf(0.9, 2) = 4.6

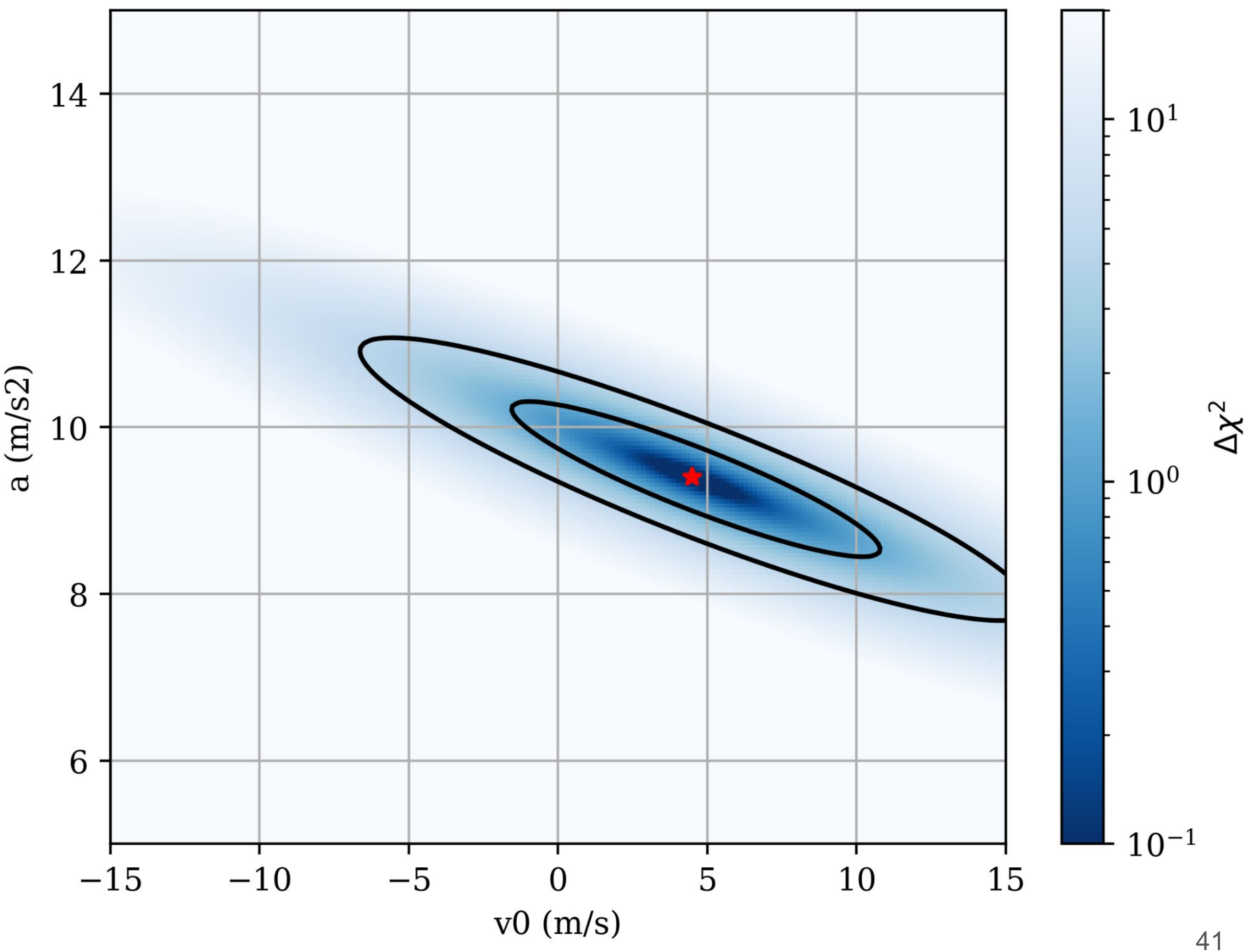
90% probability true a and v0 is inside the 90% contour

So how do we get an error bar from this?



So how do we get an error bar from this?

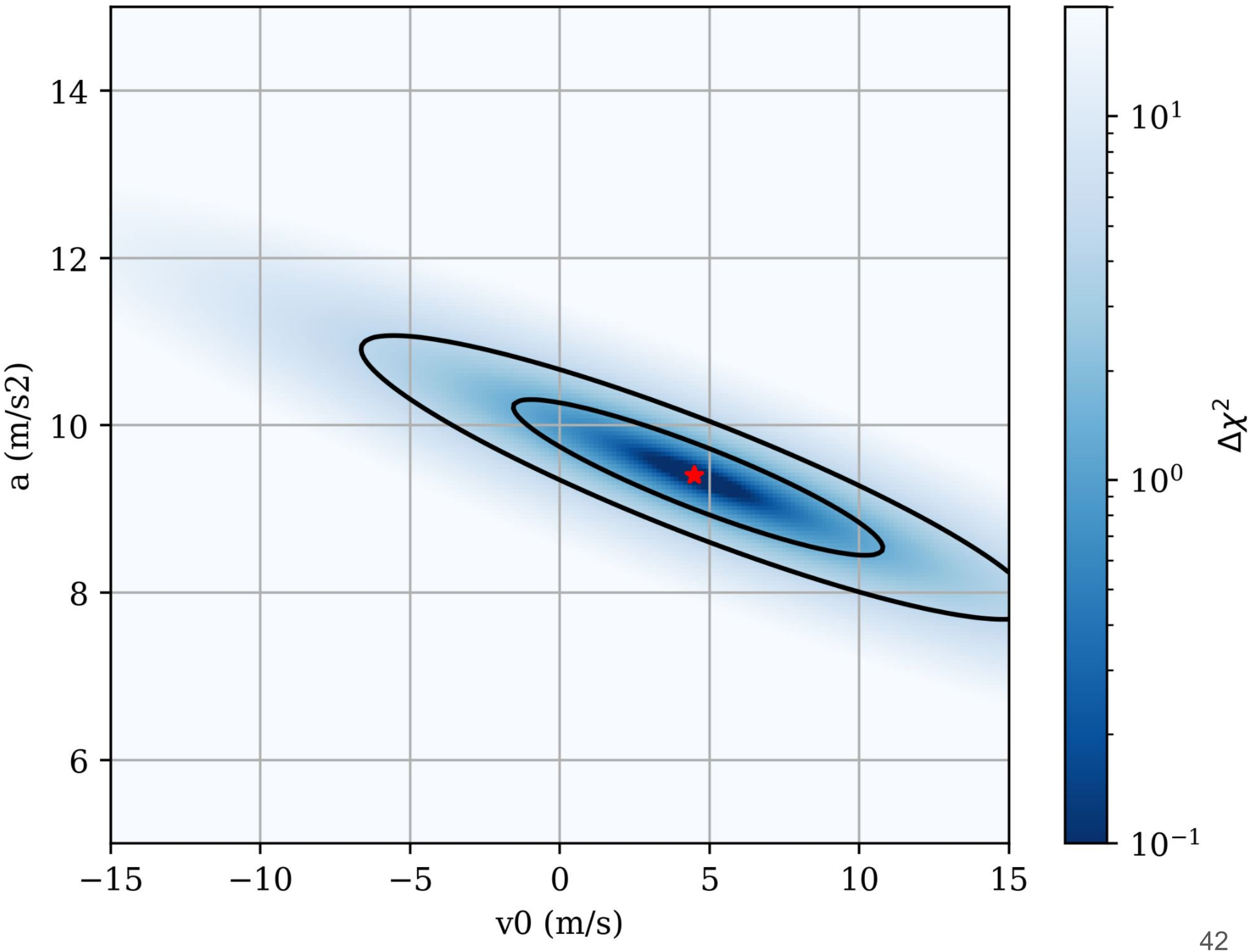
The a and v0 values that give lines that can describe the data are correlated



So how do we get an error bar from this?

The a and v0 values that give lines that can describe the data are correlated

The possible values of a, depend on v0

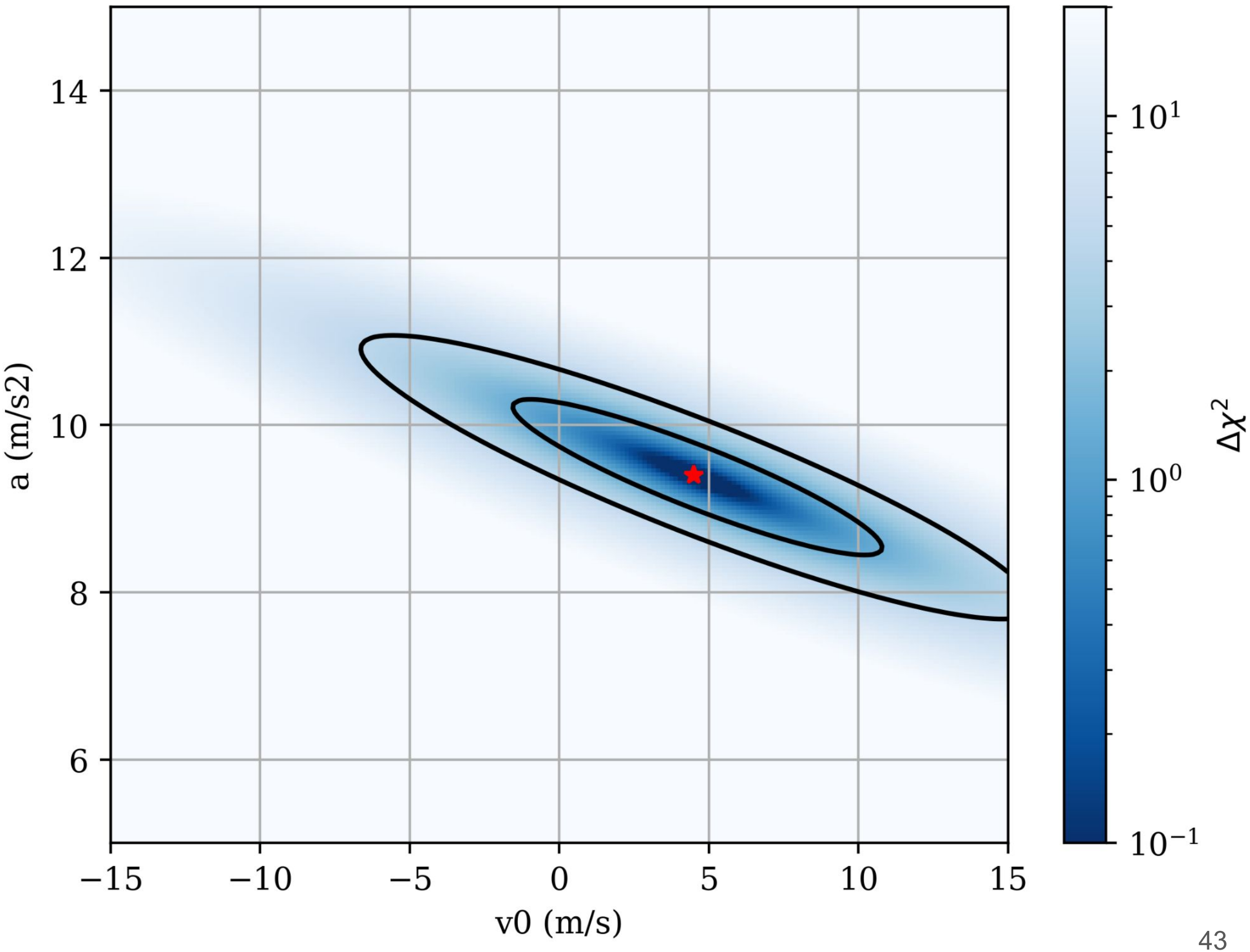


So how do we get an error bar from this?

The a and v0 values that give lines that can describe the data are correlated

The possible values of a, depend on v0

How to present this depends on what you want to know



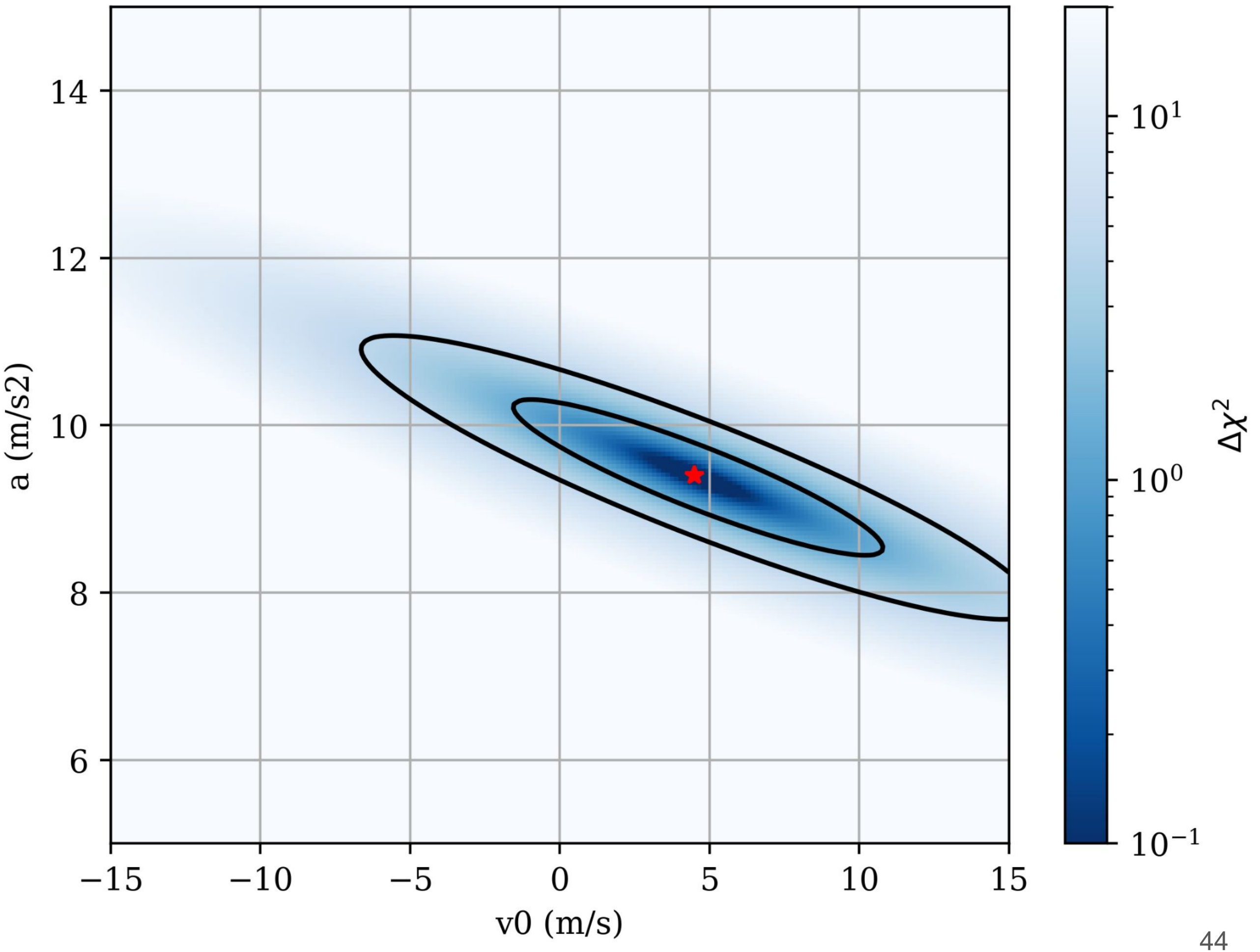
So how do we get an error bar from this?

The a and v0 values that give lines that can describe the data are correlated

The possible values of a, depend on v0

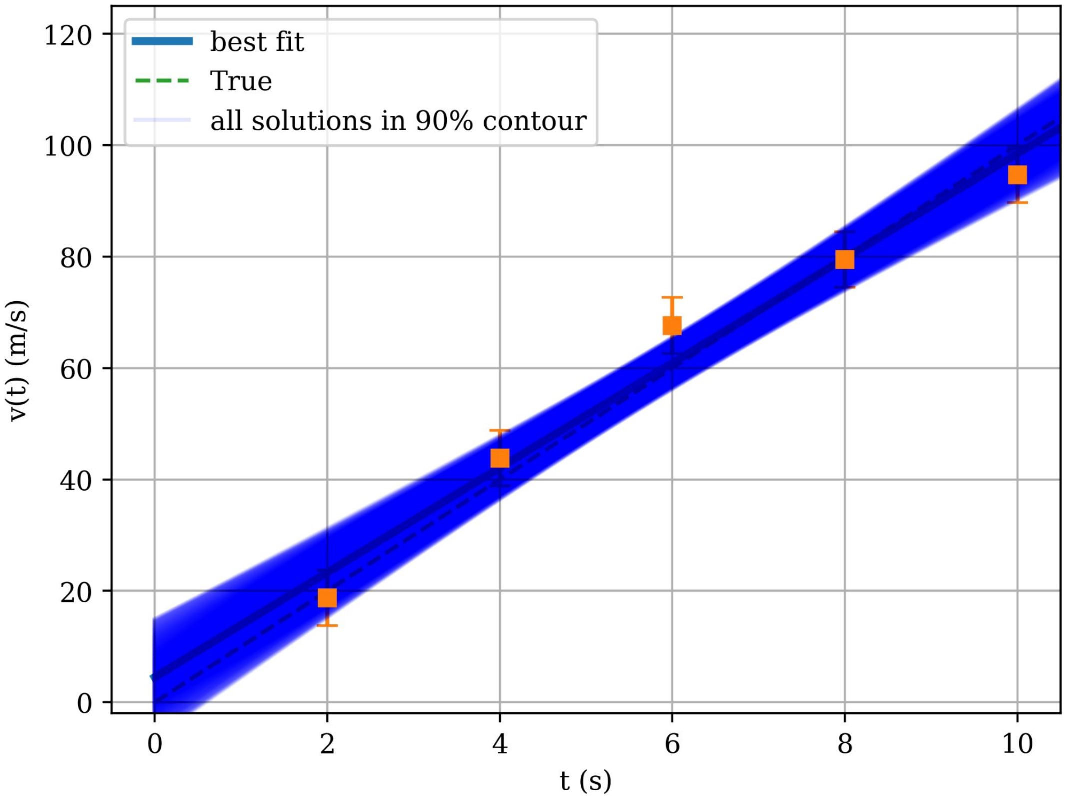
How to present this depends on what you want to know

This shows us all the allowed parameter space



We can also map this to our v(t) vs t plot

What are the allowed v values as a function of t



What if we instead wanted to know the error bar or PDF of 1 of the parameters

What if we instead wanted to know the error bar or PDF of 1 of the parameters

Say we don't care what a is, we only care about what v0 is

- Did that dragster have a rolling start?

What if we instead wanted to know the error bar or PDF of 1 of the parameters

Say we don't care what a is, we only care about what v0 is

- Did that dragster have a rolling start?

In this case a is what's called a **nuisance parameter**

- An unknown free parameter in our model that we are not interested in

Getting rid of a nuisance parameter

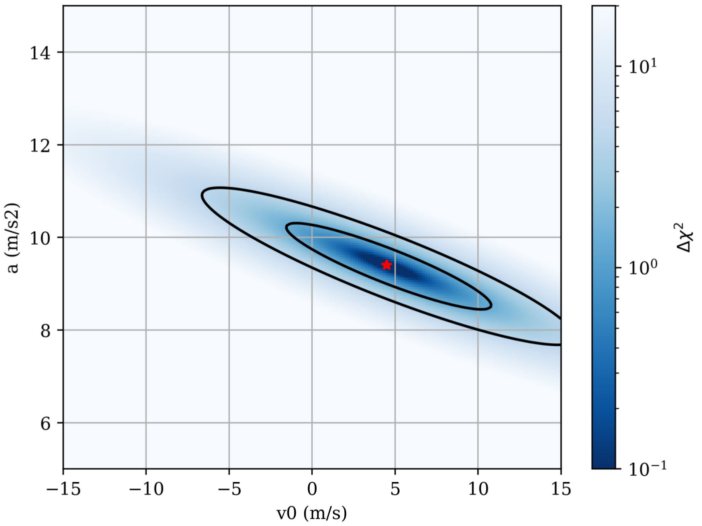
To “get rid” of a nuisance parameter you do what’s called profiling.

You have a 2D parameter space, but you can reduce that by minimizing chi2 over a, for each v0.

Getting rid of a nuisance parameter

To “get rid” of a nuisance parameter you do what’s called profiling.

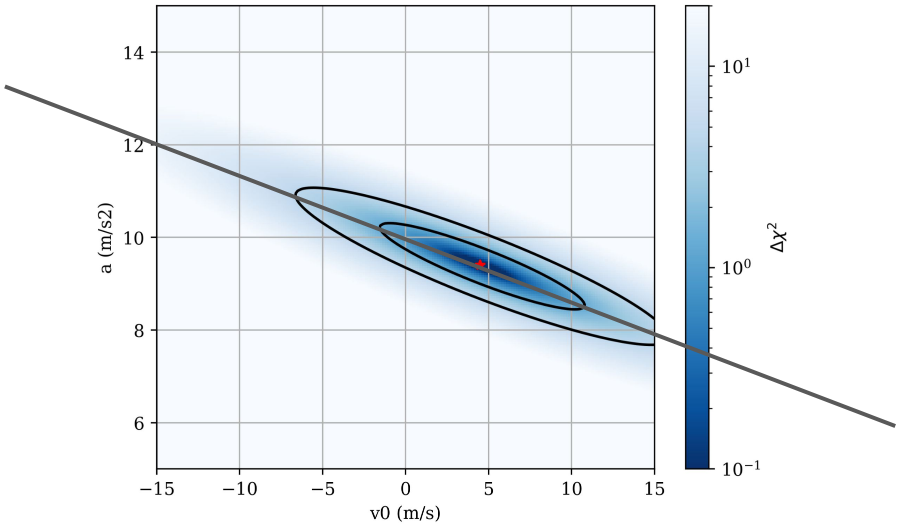
You have a 2D parameter space, but you can reduce that by minimizing chi2 over a, for each v0.



Getting rid of a nuisance parameter

To “get rid” of a nuisance parameter you do what’s called profiling.

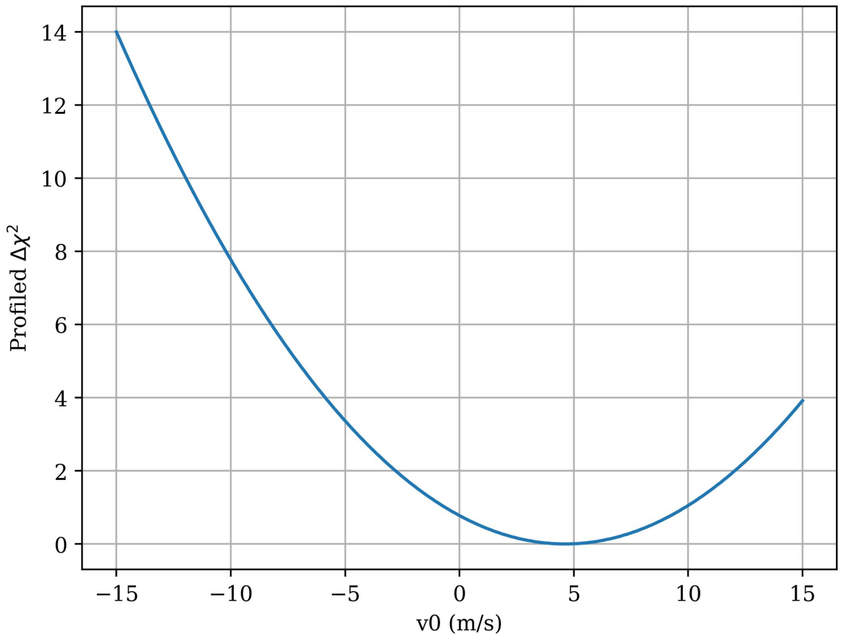
You have a 2D parameter space, but you can reduce that by minimizing chi2 over a, for each v0.



Getting rid of a nuisance parameter

To “get rid” of a nuisance parameter you do what’s called profiling.

You have a 2D parameter space, but you can reduce that by minimizing chi2 over a, for each v0.

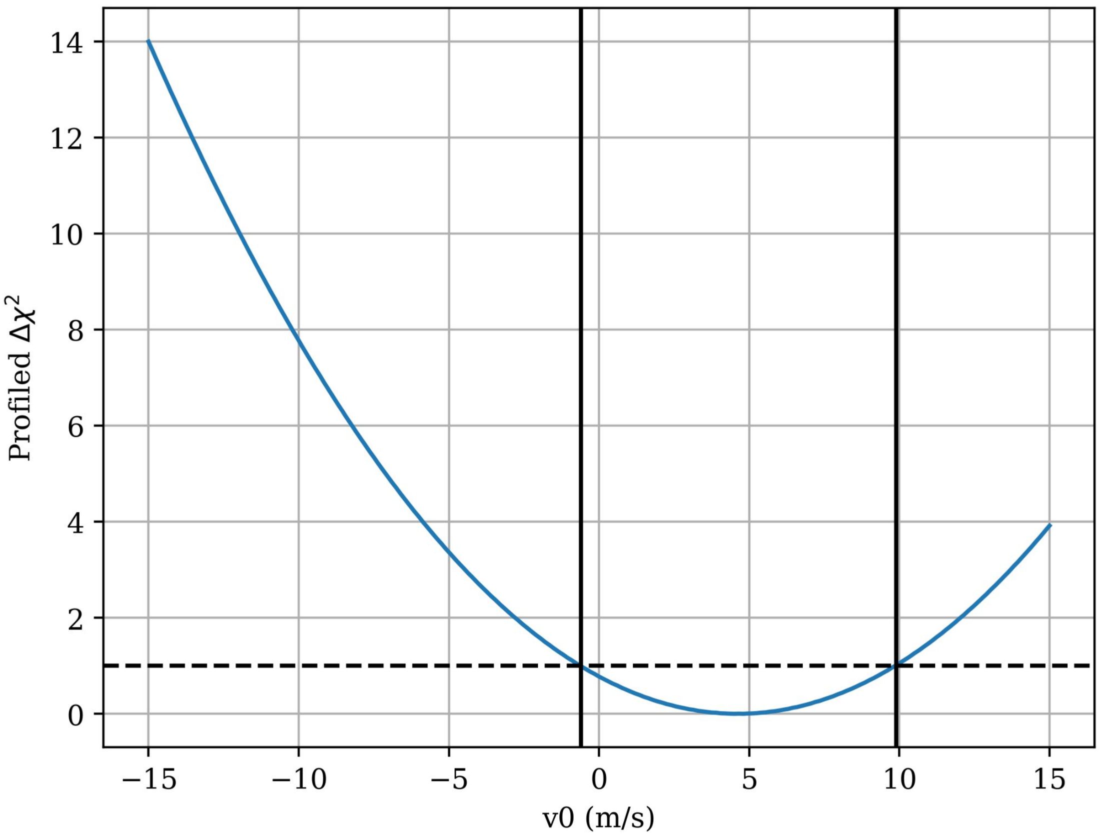


Getting rid of a nuisance parameter

We can use our chi2 with 1 dof again to find our 1D 68% error bar

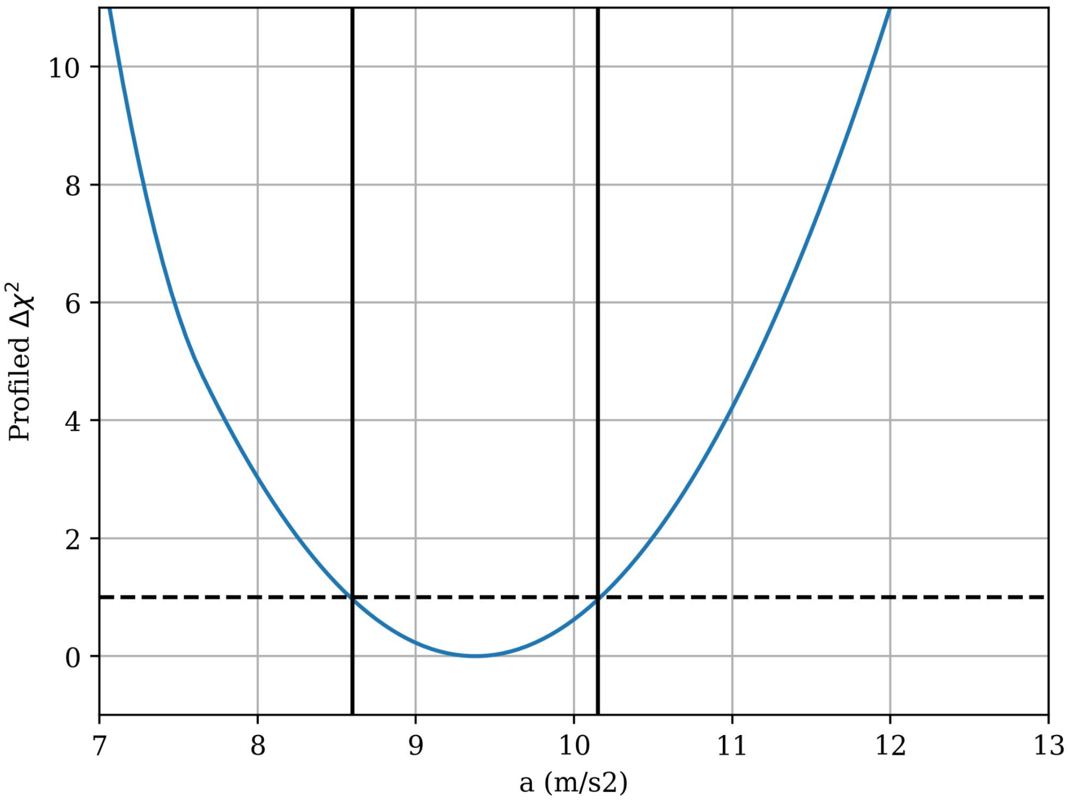
- We reduced the dof by constraining 1 of them

v0 = 4.5 +/- 5.2 m/s



We can do the same with a

a = 9.4 +/- 0.8 m/s2

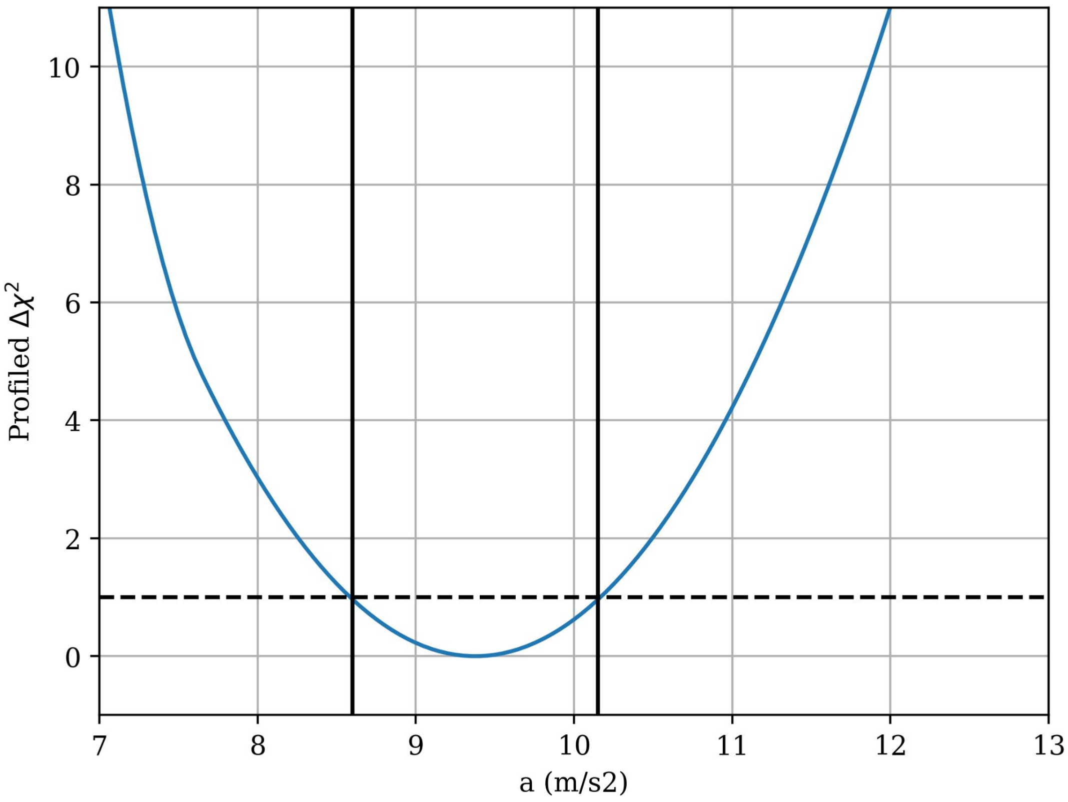


We can do the same with a

a = 9.4 +/- 0.8 m/s2

Previously with v0=0 we got

a = 10 +/- 0.33 m/s2



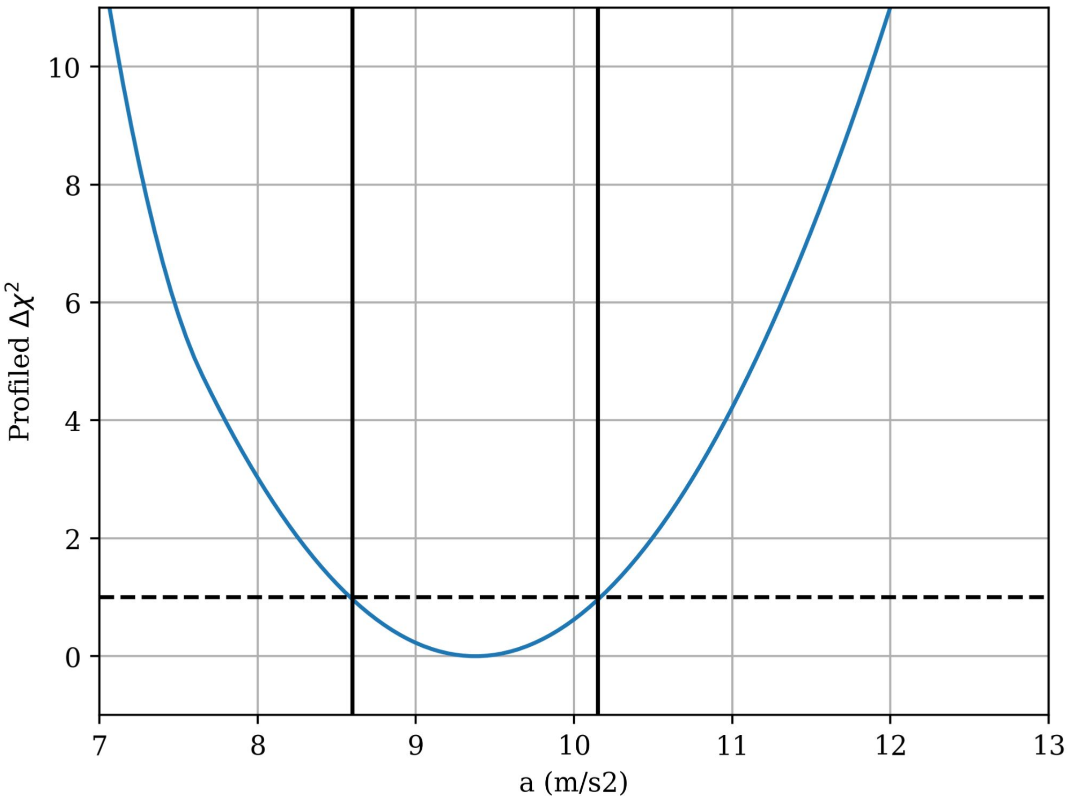
We can do the same with a

a = 9.4 +/- 0.8 m/s2

Previously with v0=0 we got

a = 10 +/- 0.33 m/s2

Why did the error bar get bigger?



We can do the same with a

a = 9.4 +/- 0.8 m/s2

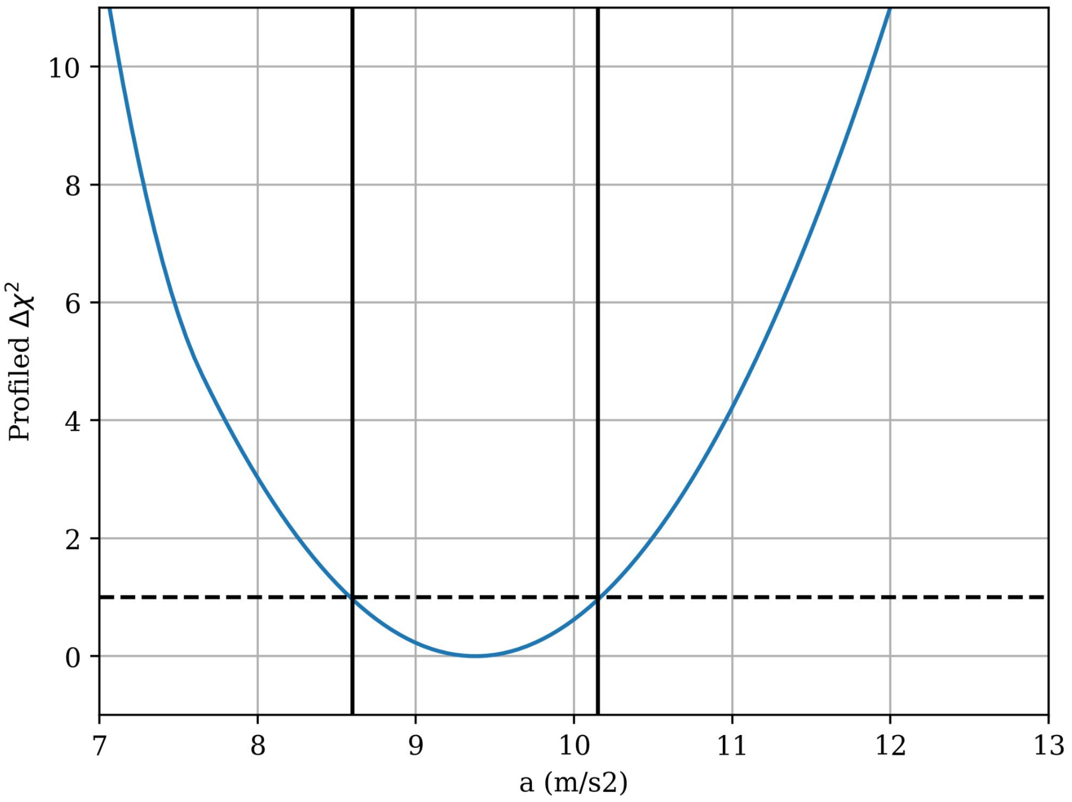
Previously with v0=0 we got

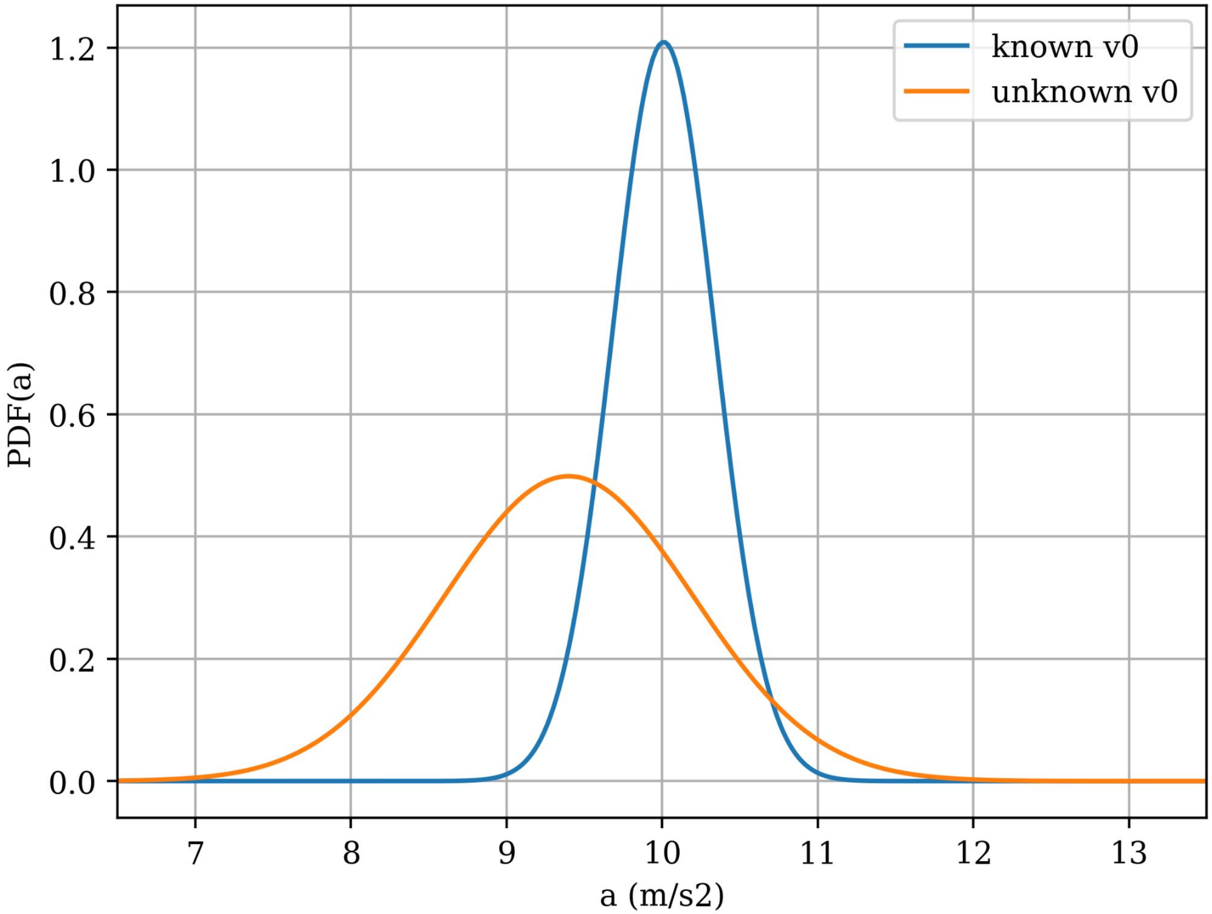
a = 10 +/- 0.33 m/s2

Why did the error bar get bigger?

We have less information

- v0 could be anything





known v0
unknown v0
PDF(a)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
7
8
9
10
11
12
13
a (m/s2)