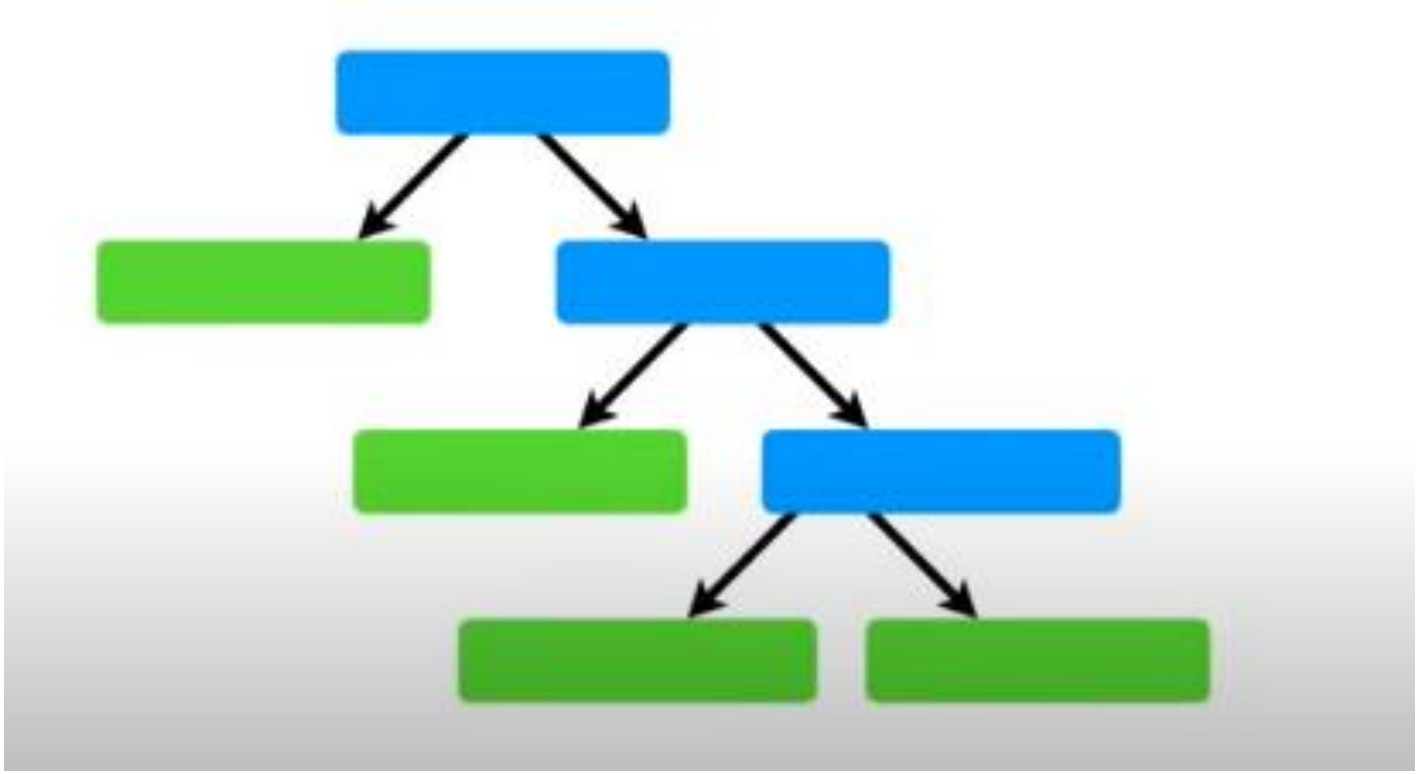
DecisionTree Regressor

DecisionTree Regressor



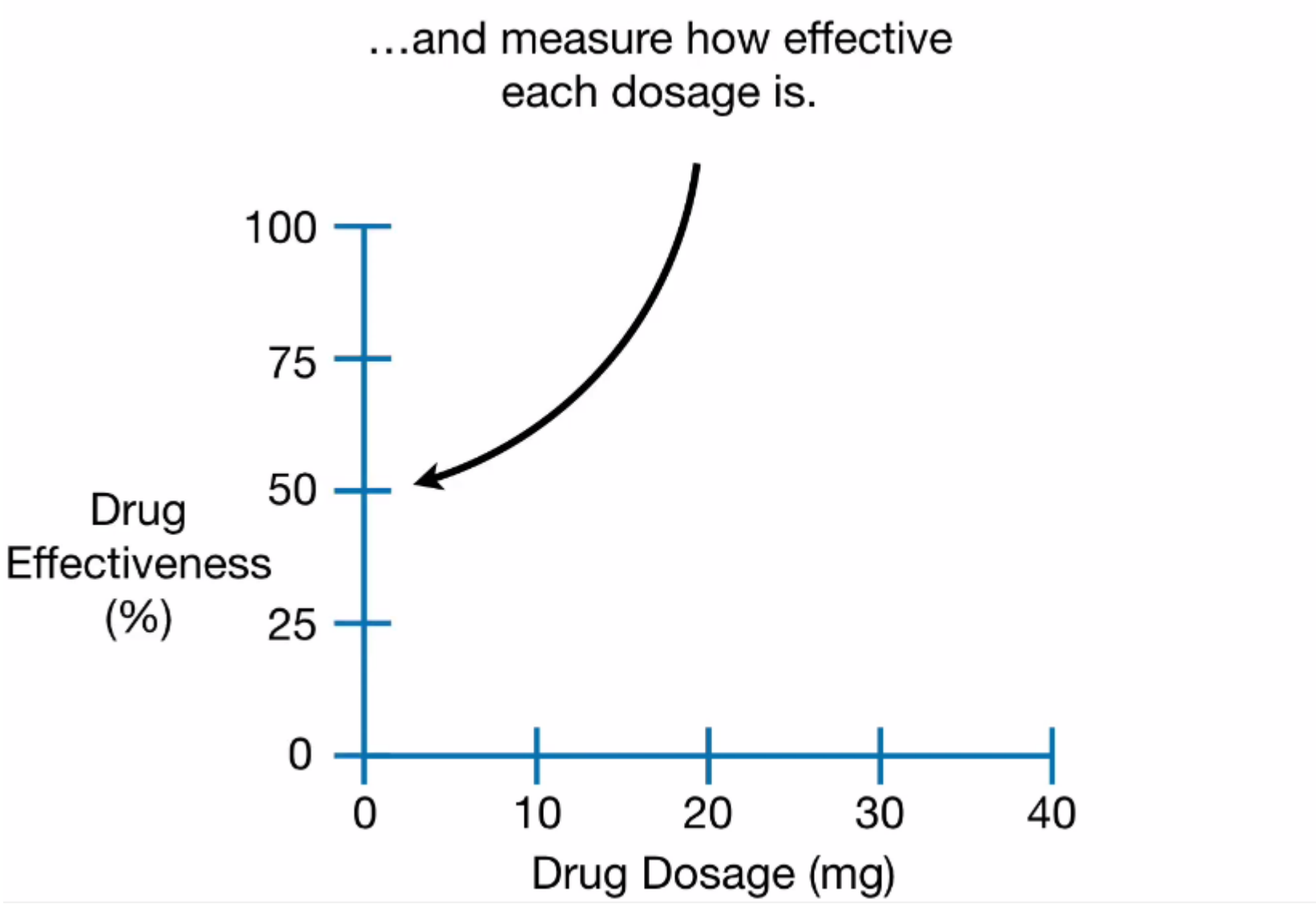
Imagine we developed a new drug…

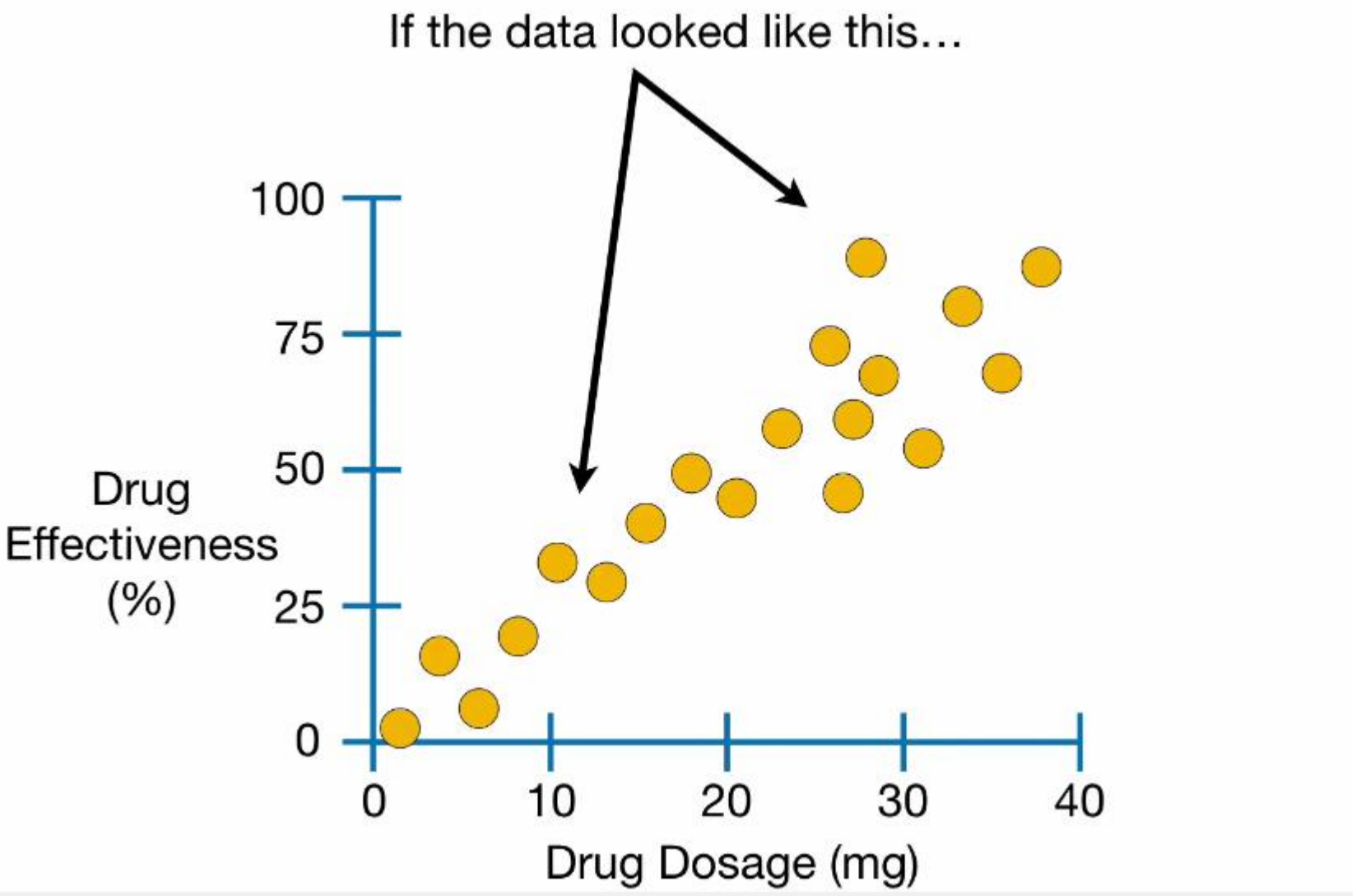
VS.

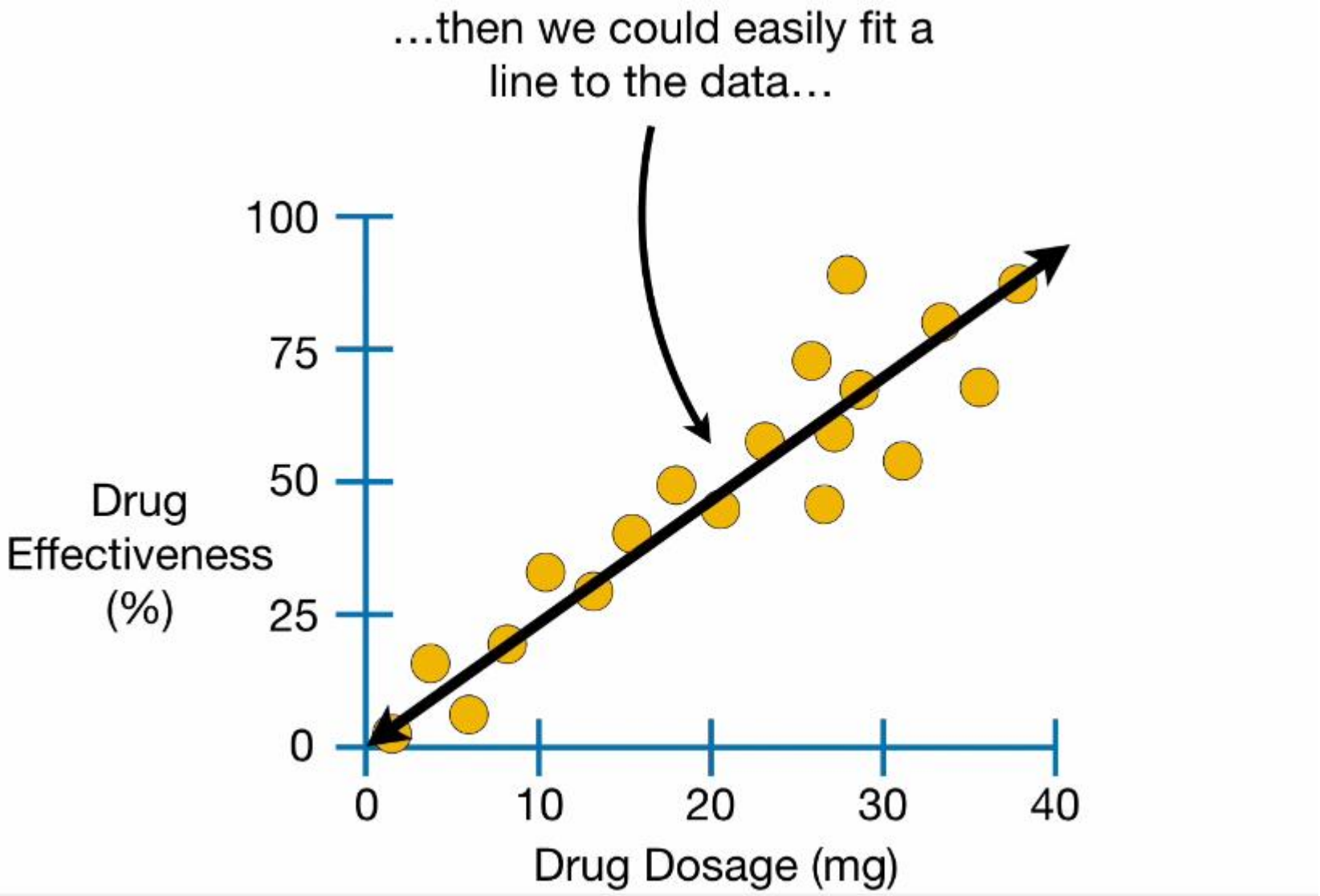
…to cure the common cold.

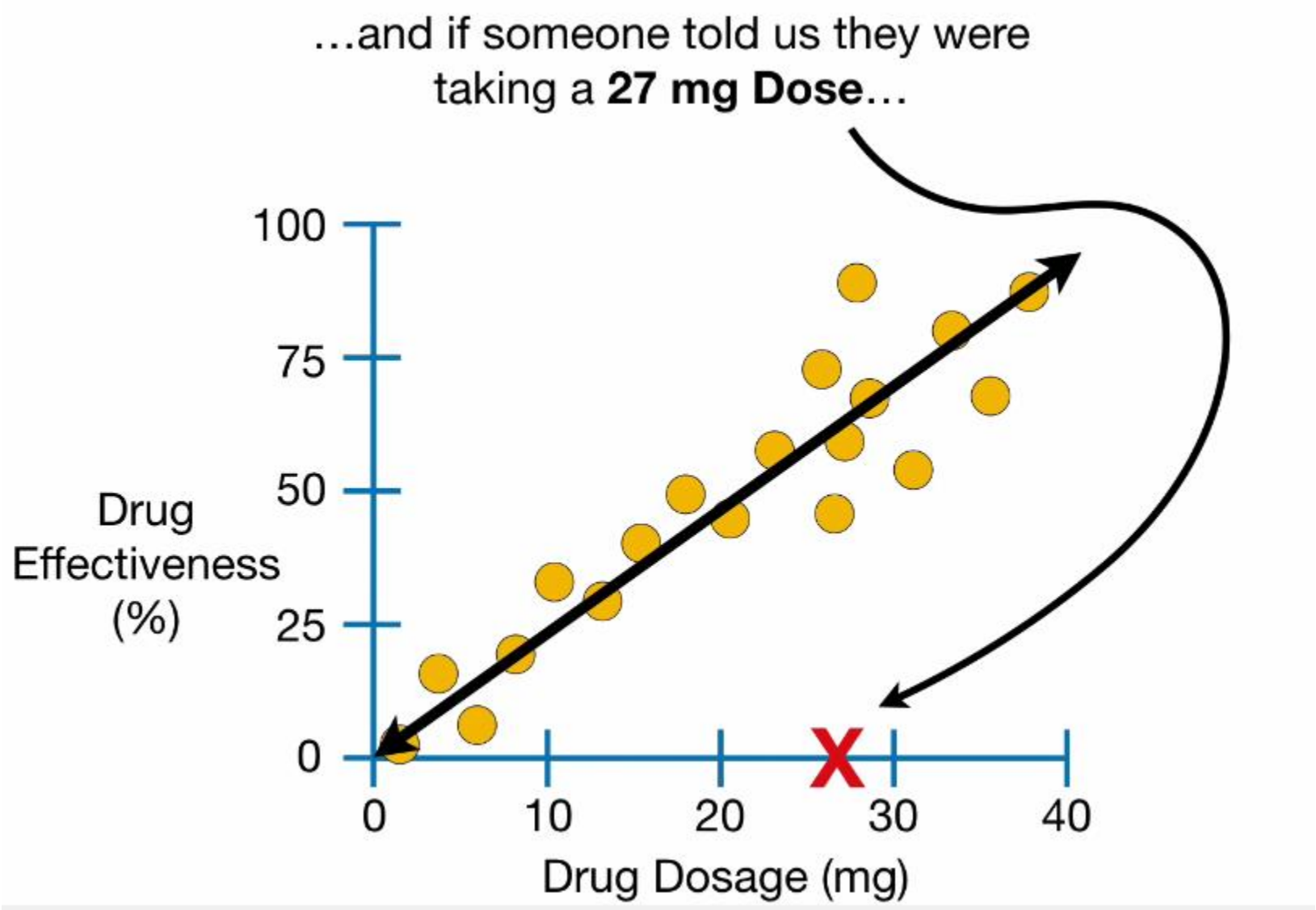
However, we don't know the optimal dosage to give to patients.

vs.

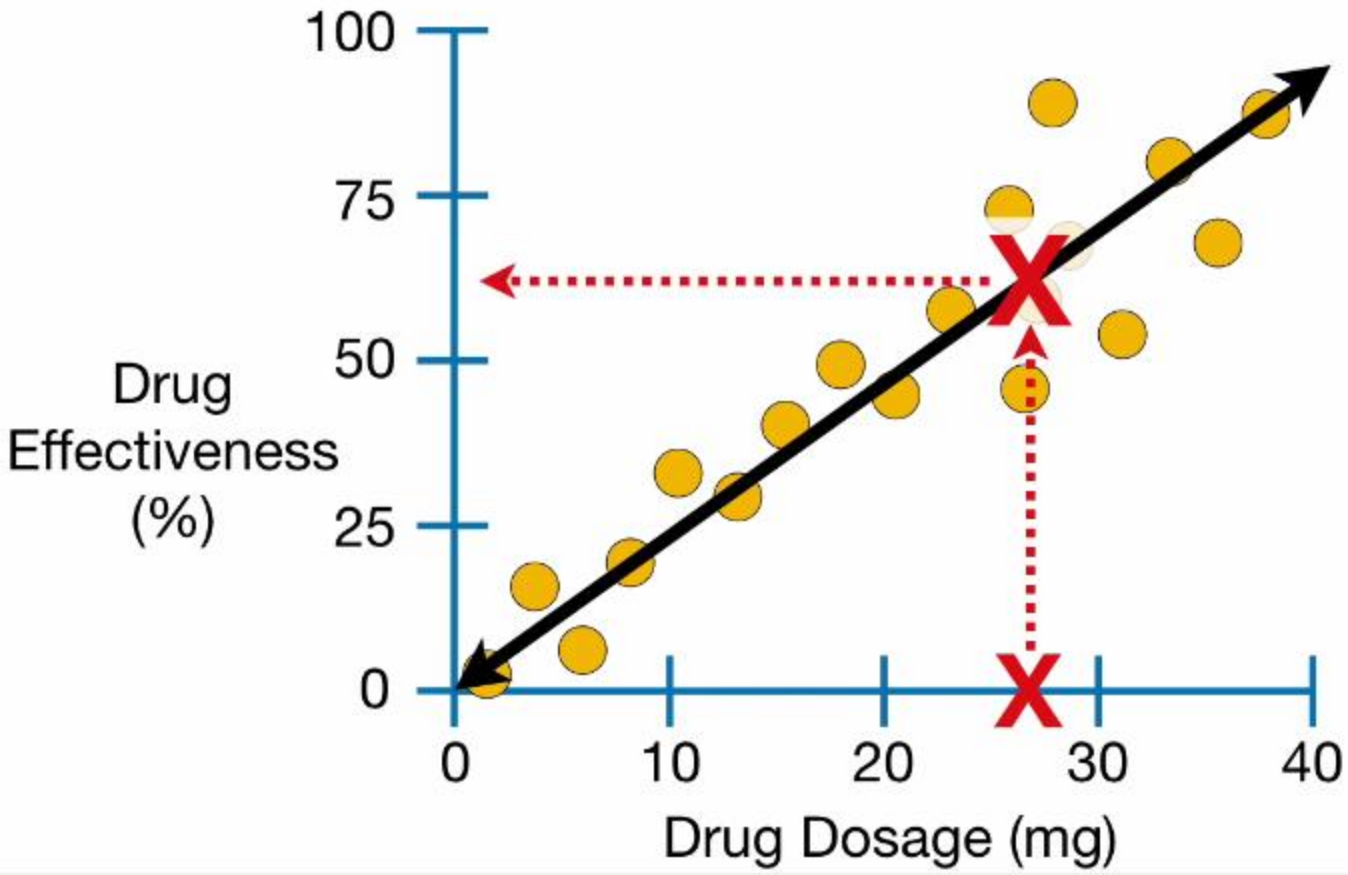

…and measure how effective each dosage is.
100
75
50
25
0
Drug Effectiveness (%)
0
10
20
30
40
Drug Dosage (mg)


If the data looked like this…
100
75
50
25
0
Drug Effectiveness (%)
0
10
20
30
40
Drug Dosage (mg)

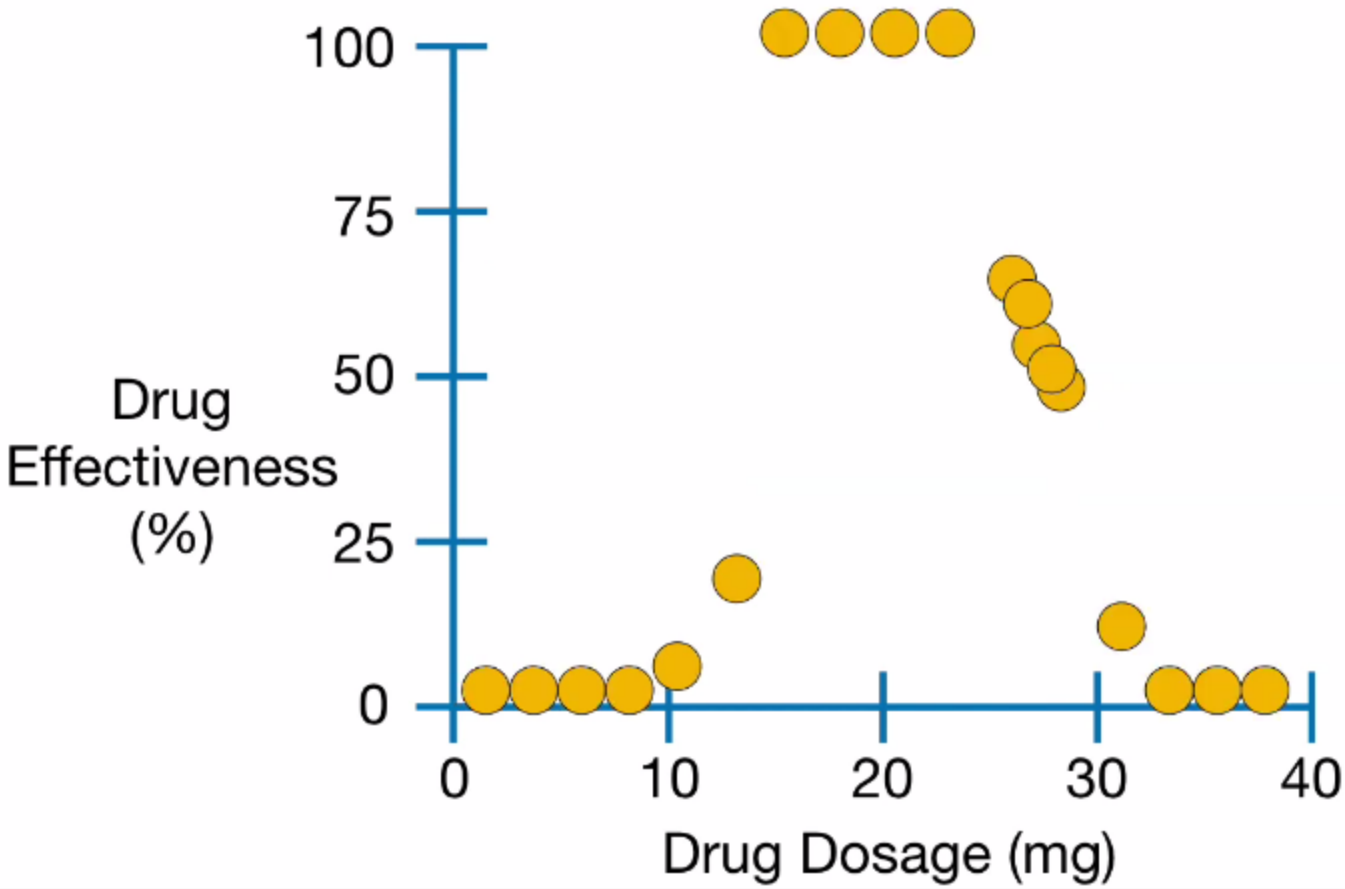

…then we could easily fit a line to the data…
100
75
50
25
0
Drug Effectiveness (%)
0
10
20
30
40
Drug Dosage (mg)

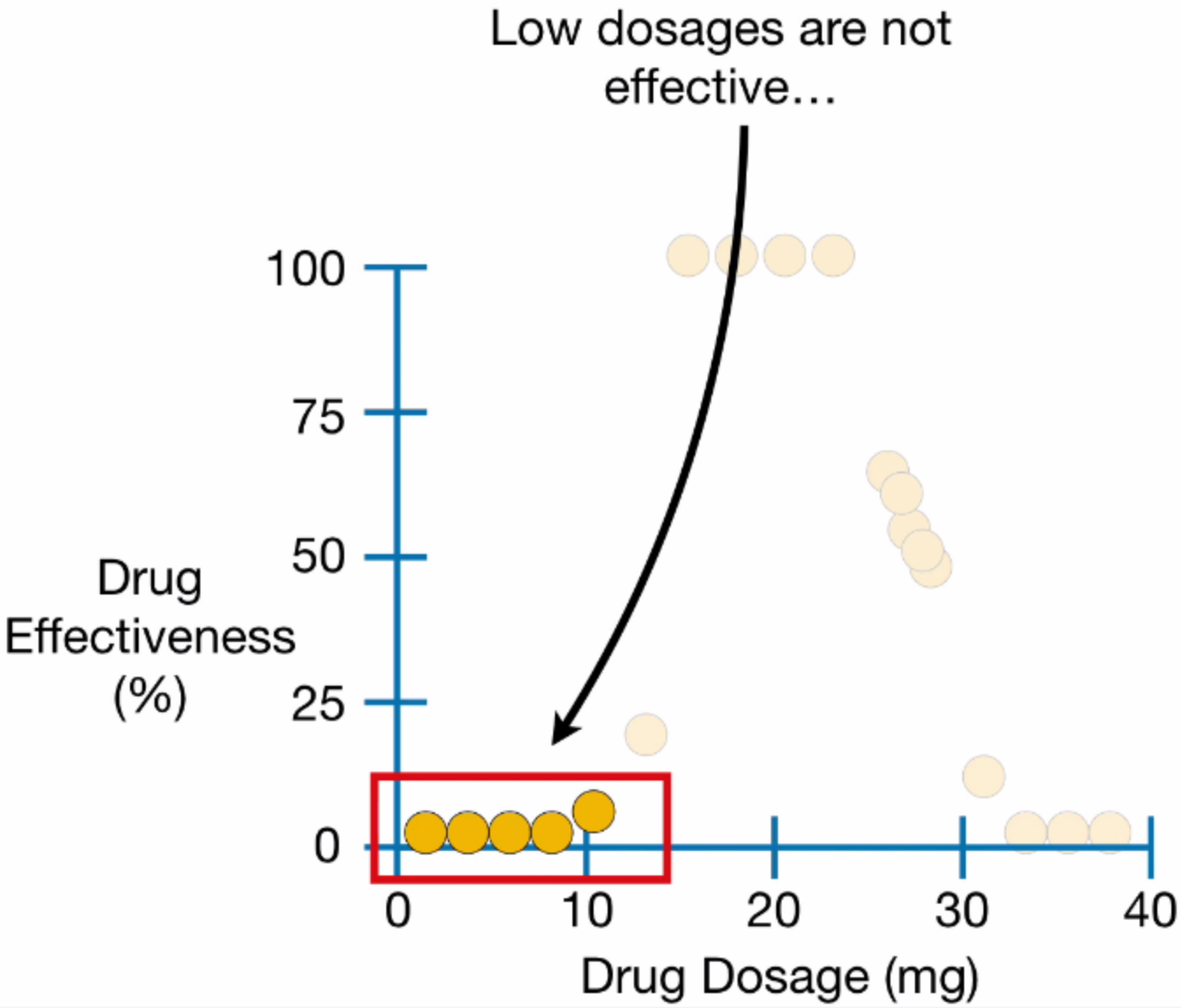

…and if someone told us they were taking a **27 mg Dose**…
100
75
50
25
0
Drug Effectiveness (%)
0
10
20
30
40
Drug Dosage (mg)
X

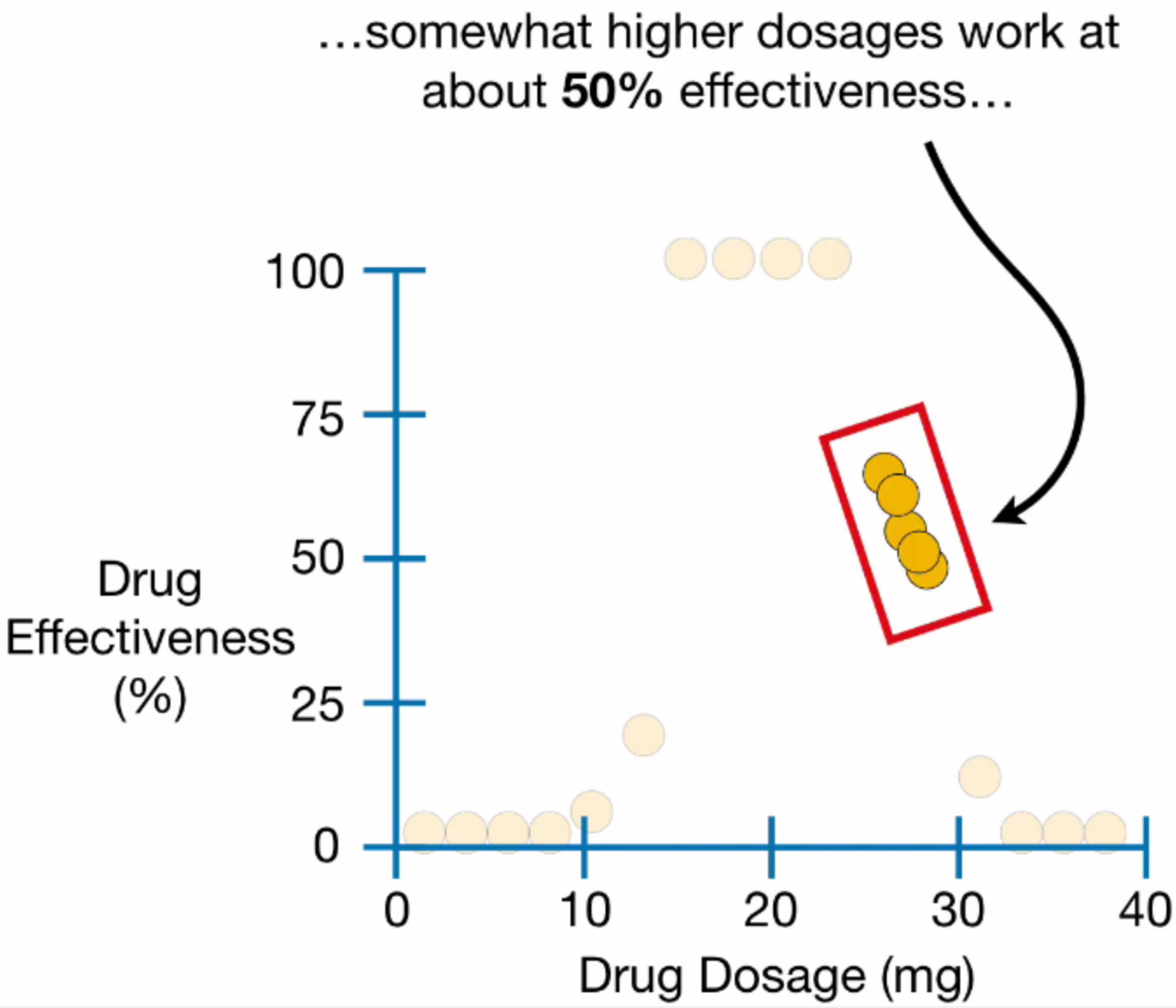
…we could use the line to predict that a **27 mg Dose** should be **62% Effective**.

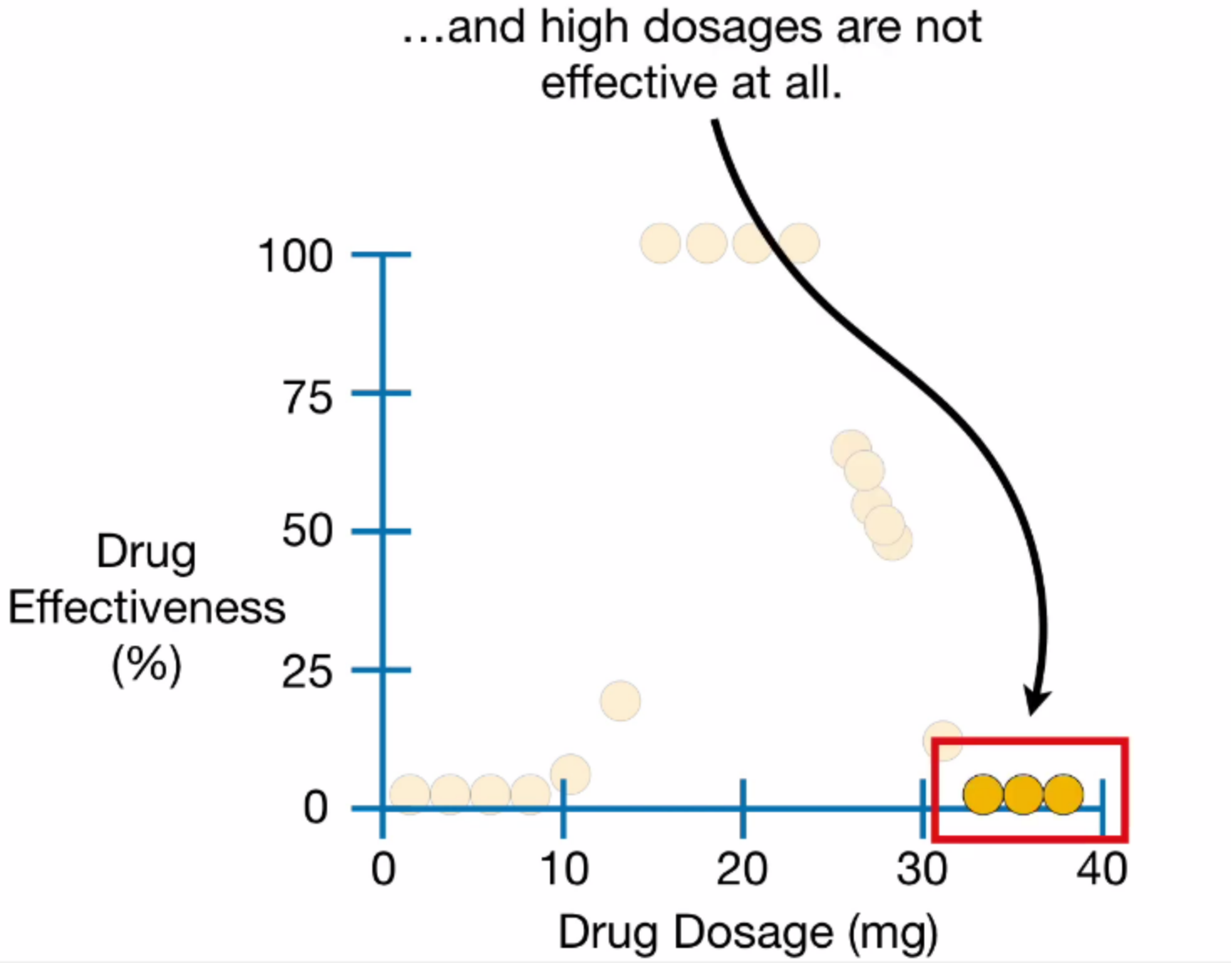


However, what if the data looked like this?

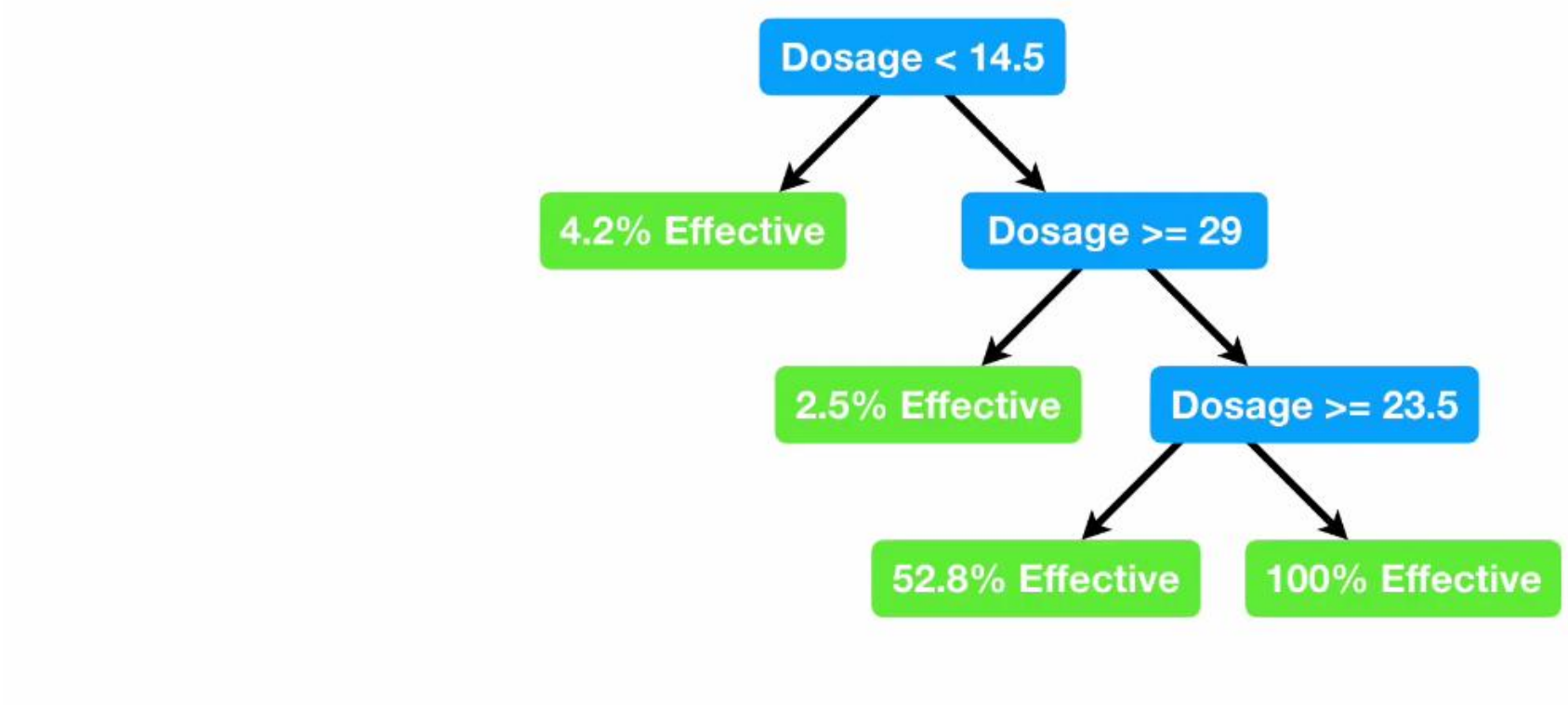



Low dosages are not effective…
100
75
50
25
0
Drug Effectiveness (%)
0
10
20
30
40
Drug Dosage (mg)

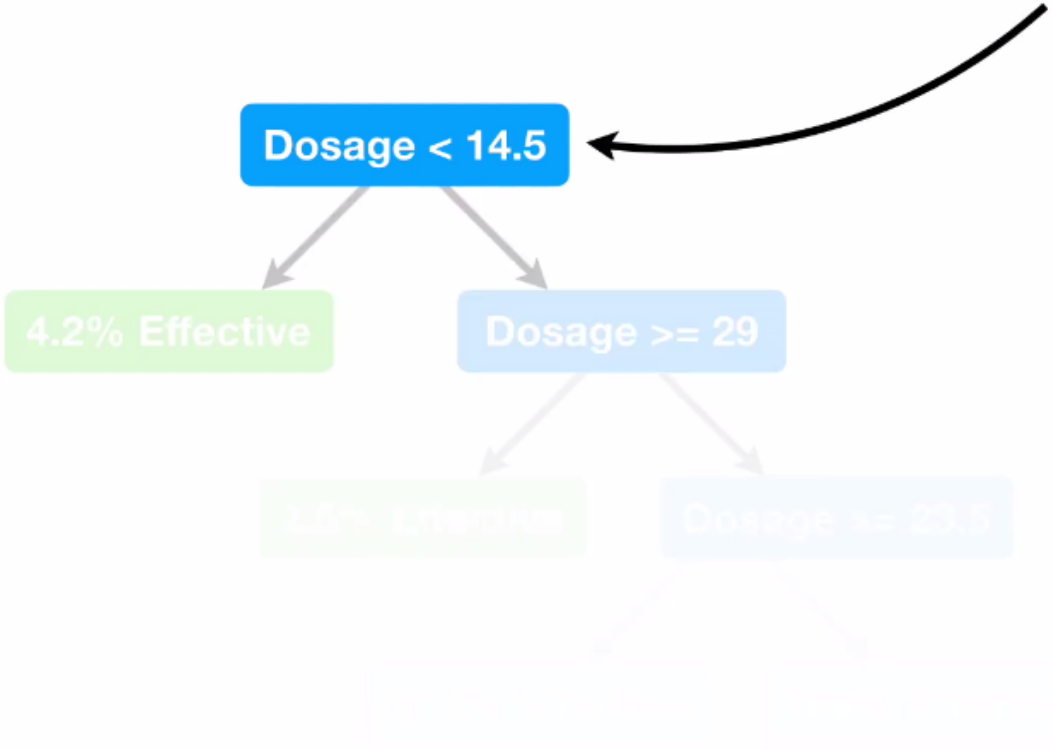

…somewhat higher dosages work at about 50% effectiveness…
100
75
50
25
0
Drug Effectiveness (%)
0
10
20
30
40
Drug Dosage (mg)


…and high dosages are not effective at all.
100
75
50
25
0
Drug Effectiveness (%)
0
10
20
30
40
Drug Dosage (mg)

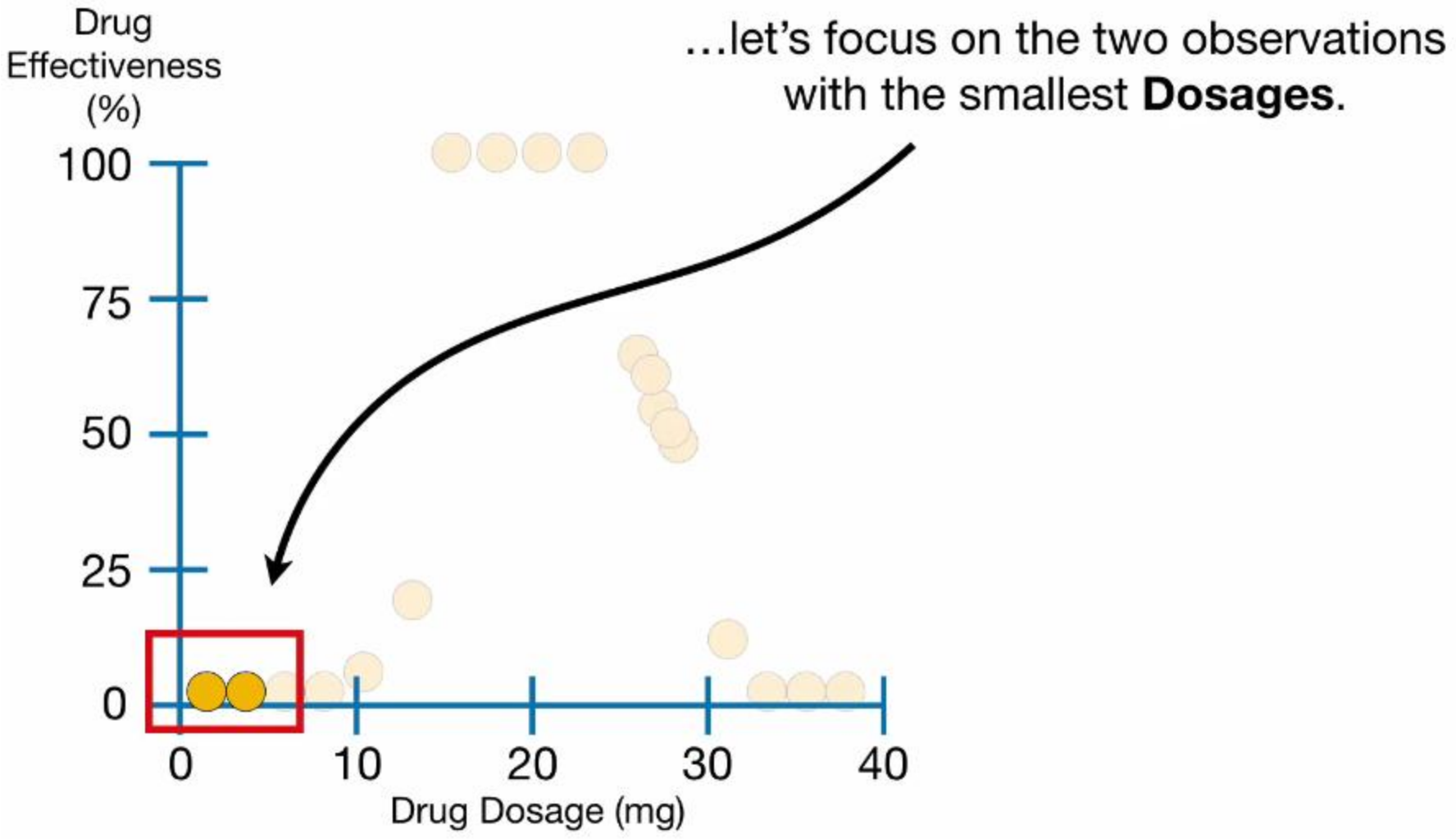
DecisionTree Building



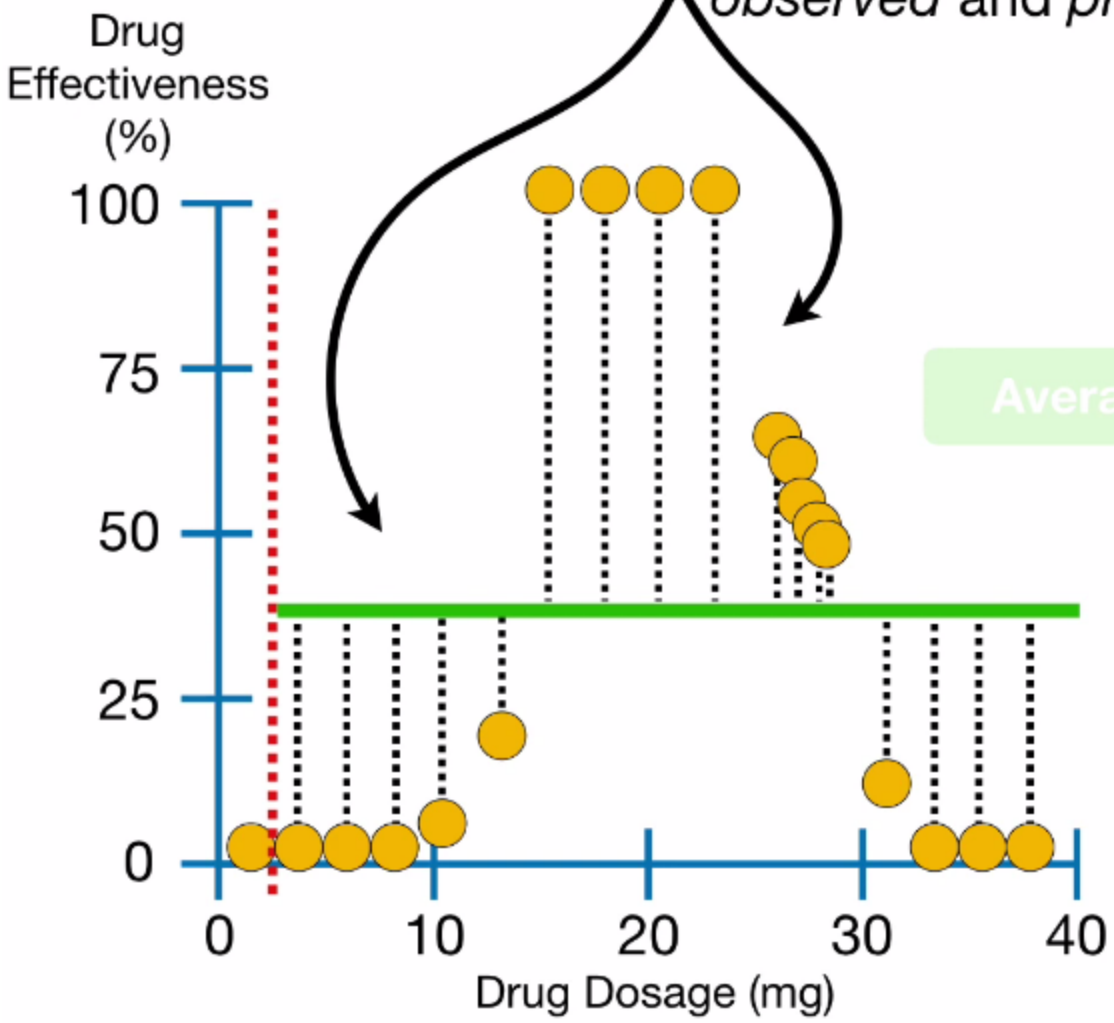
…the first thing we do is figure out why we start by asking if **Dosage < 14.5**.

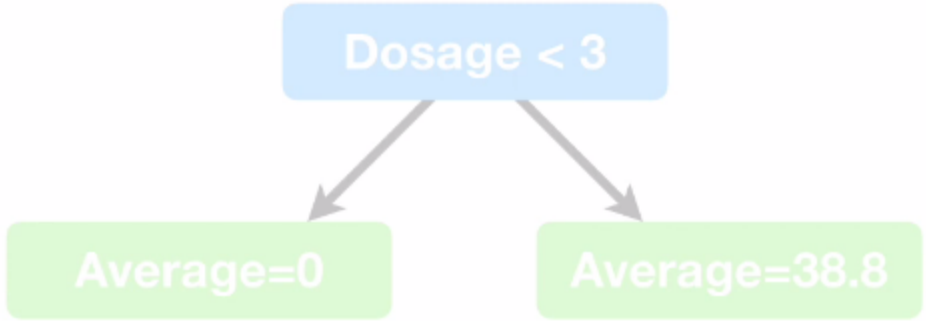


Dosage	Drug Effect.
10	58
20	60
35	57
5	44
etc…	etc…

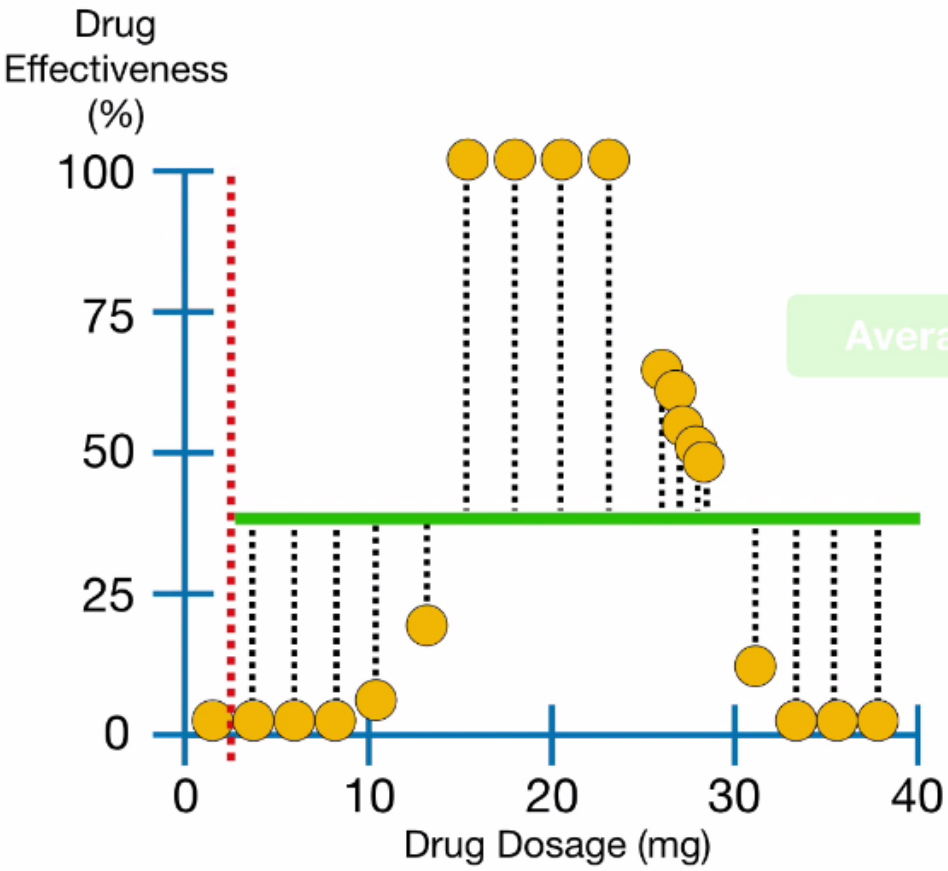

Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
…let's focus on the two observations with the smallest **Dosages**.

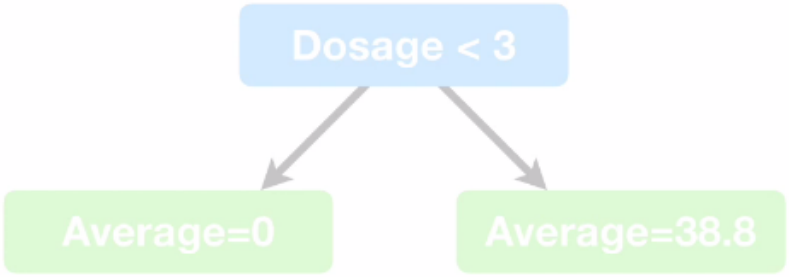
For each point in the data, we can draw its **residual**, the difference between the *observed* and *predicted* values…





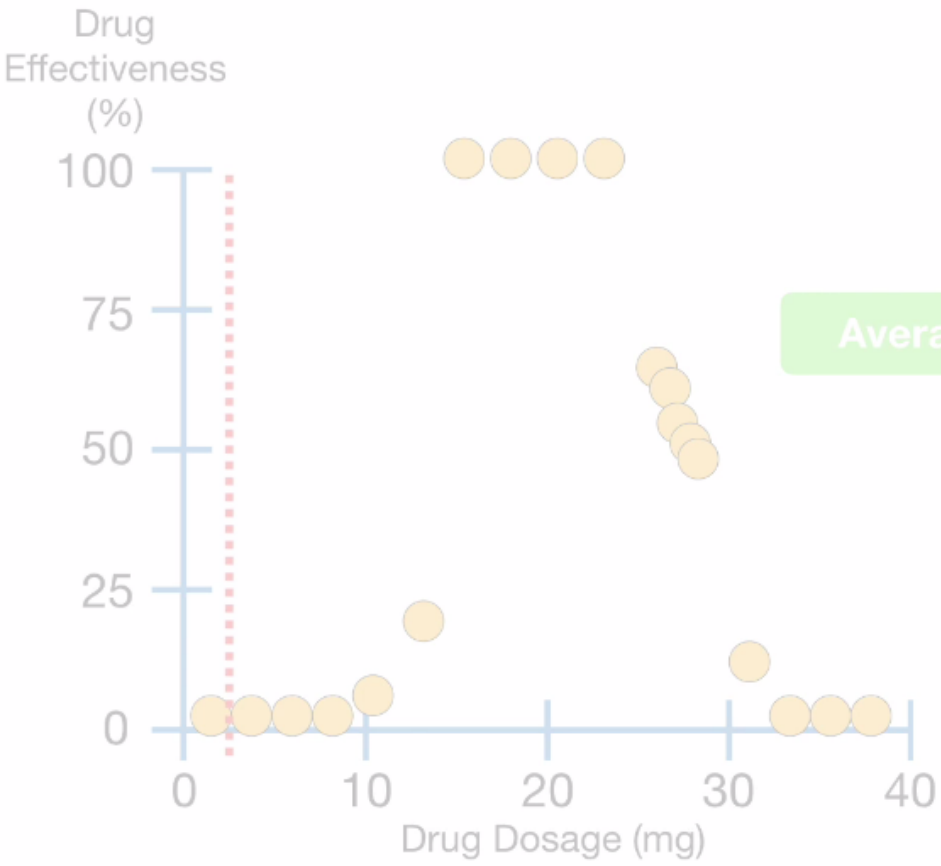
…and get **27,468.5**.

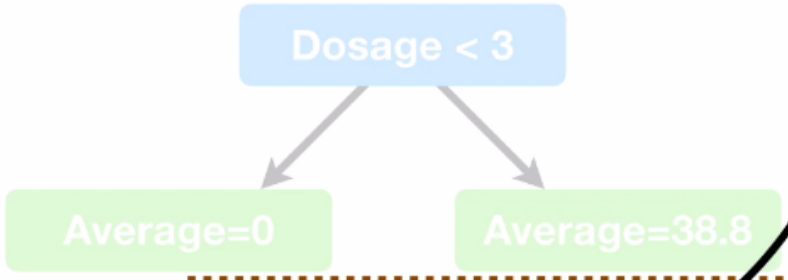


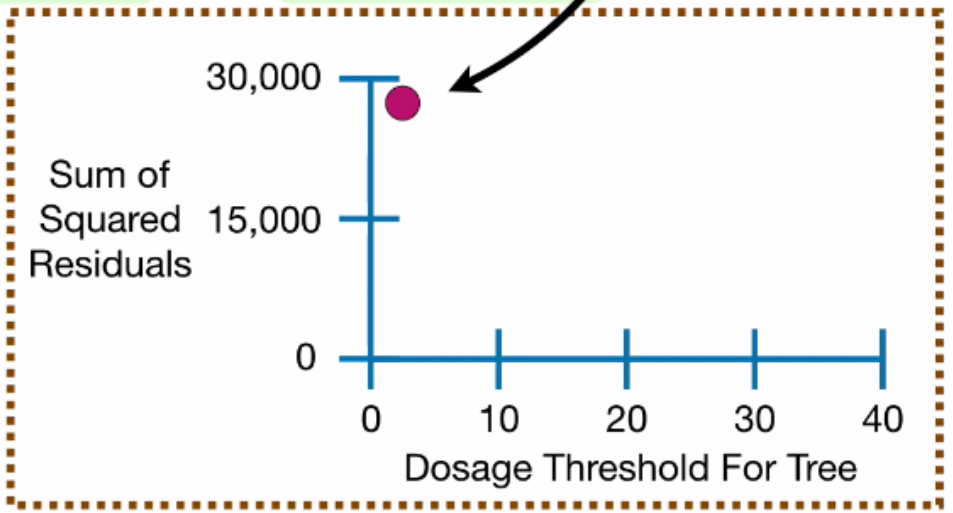


$$(0 - 0)^2 + (0 - 38.8)^2 + (0 - 38.8)^2 + (0 - 38.8)^2$$
$$+ (5 - 38.8)^2 + (20 - 38.8)^2 + (\ 00 - 38.8)^2$$
$$+ (100 - 38.8)^2 + \ldots + (0 - 38.8)^2$$
$$= 27{,}468.5$$

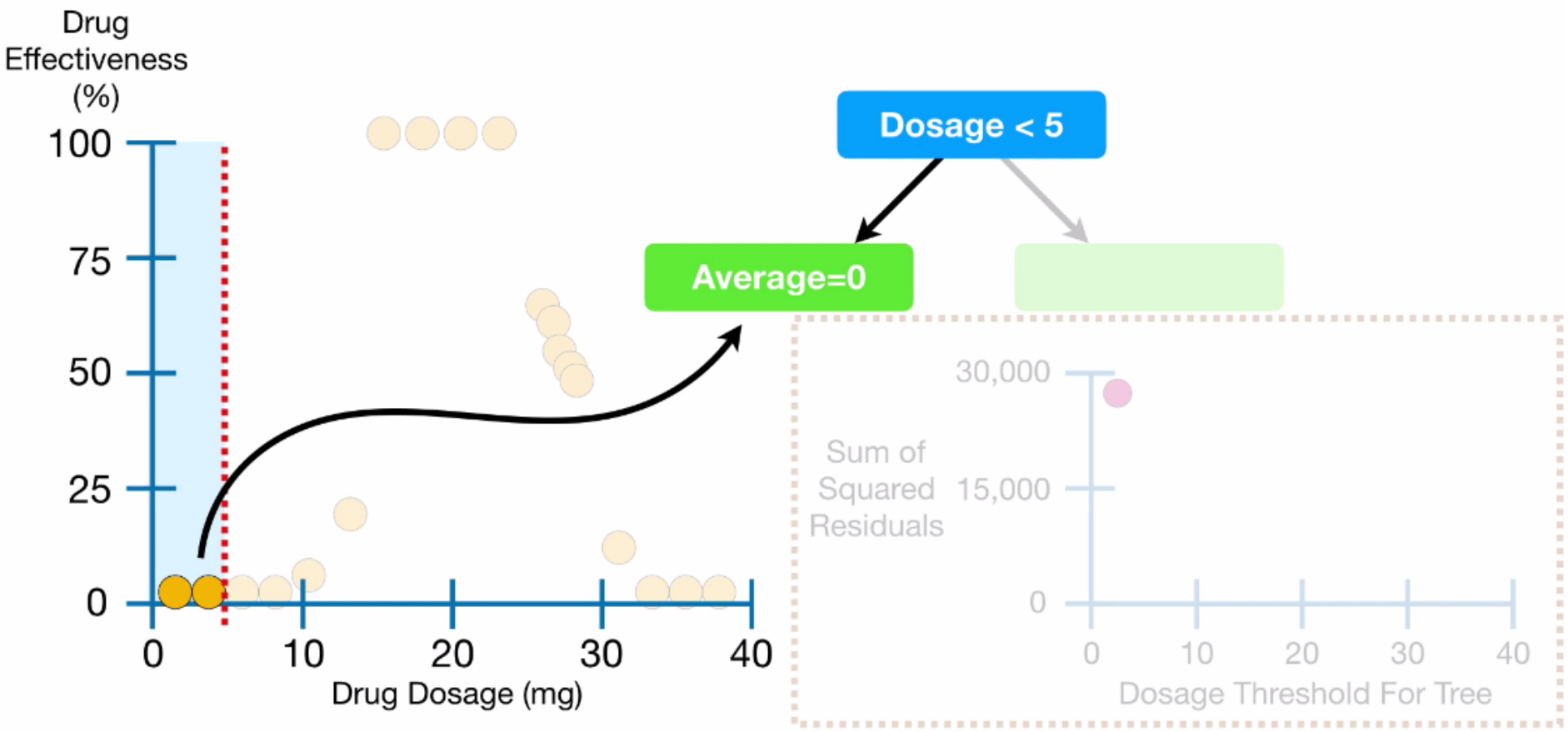
NOTE: We can plot the sum of squared residuals on this graph.



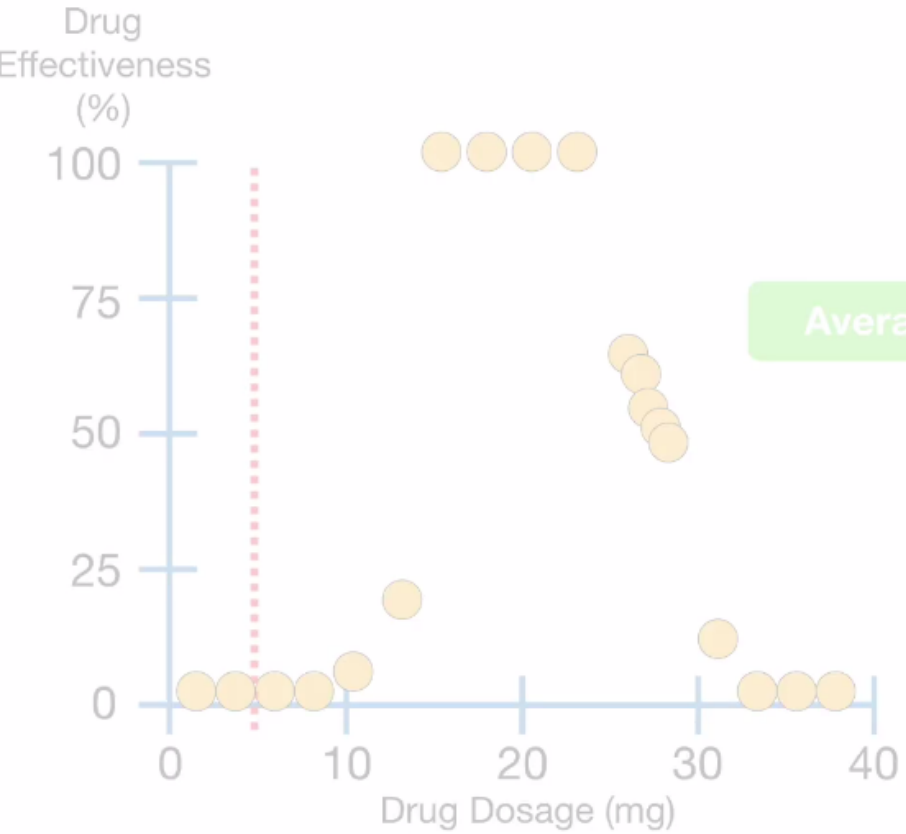


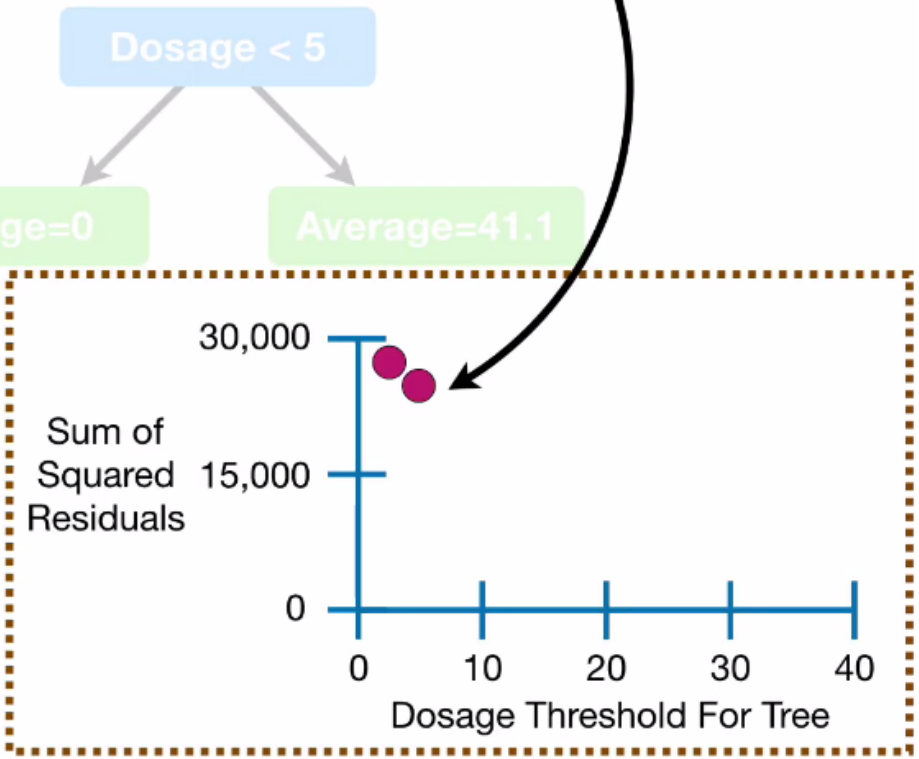


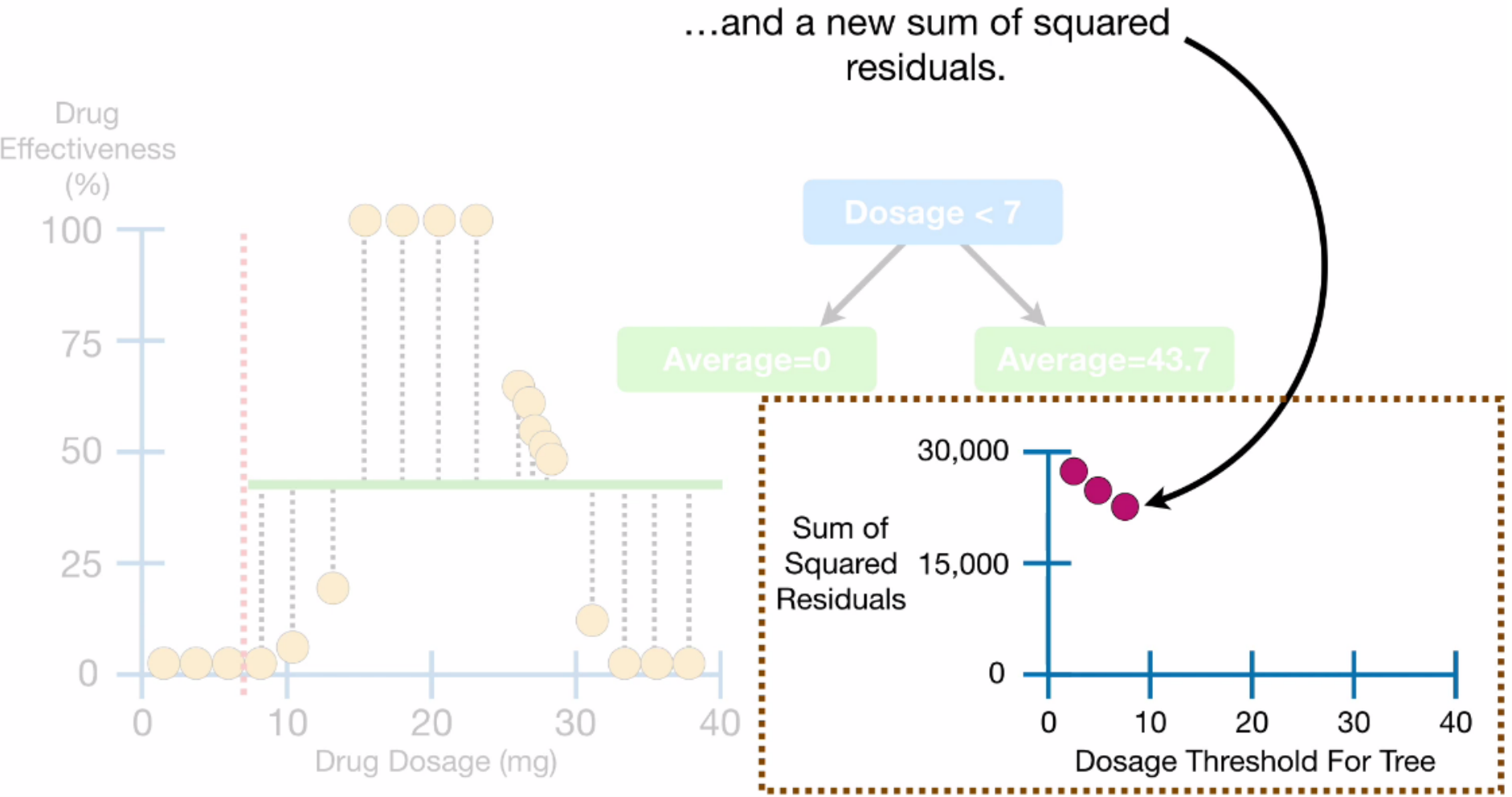
Using **Dosage < 5** gives us new predictions…

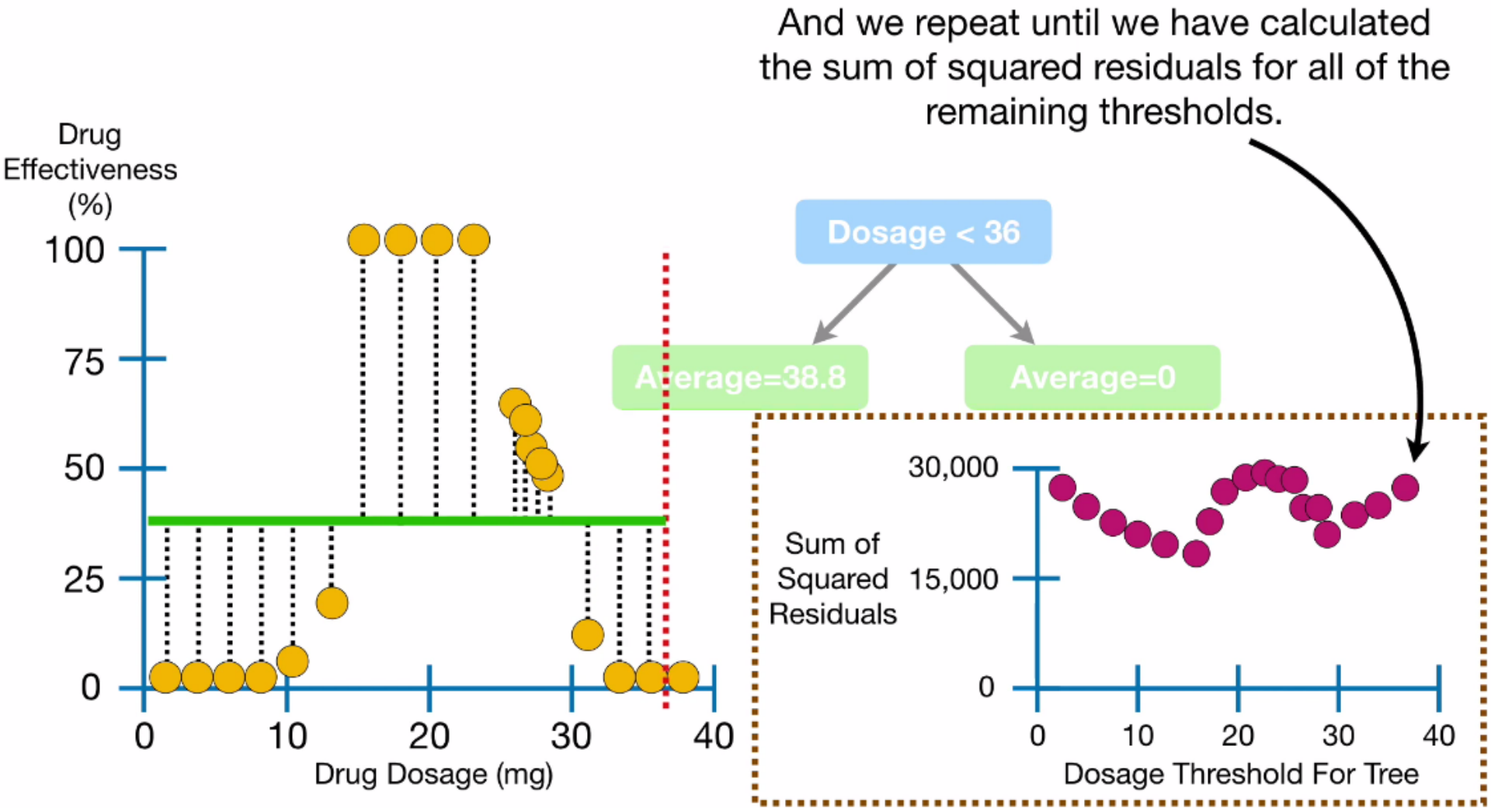


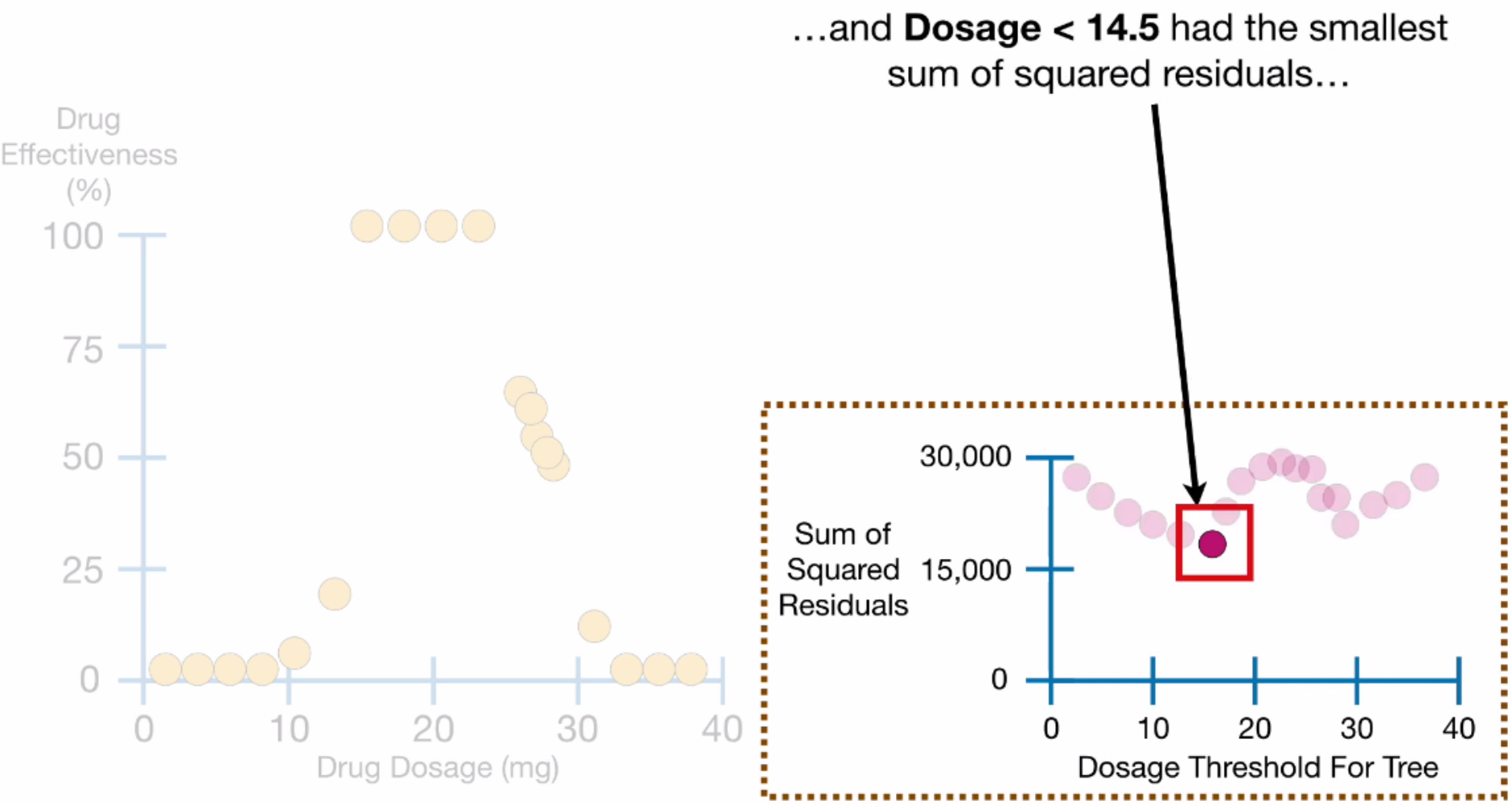
…and that means we can add a new sum of squared residuals to our graph.



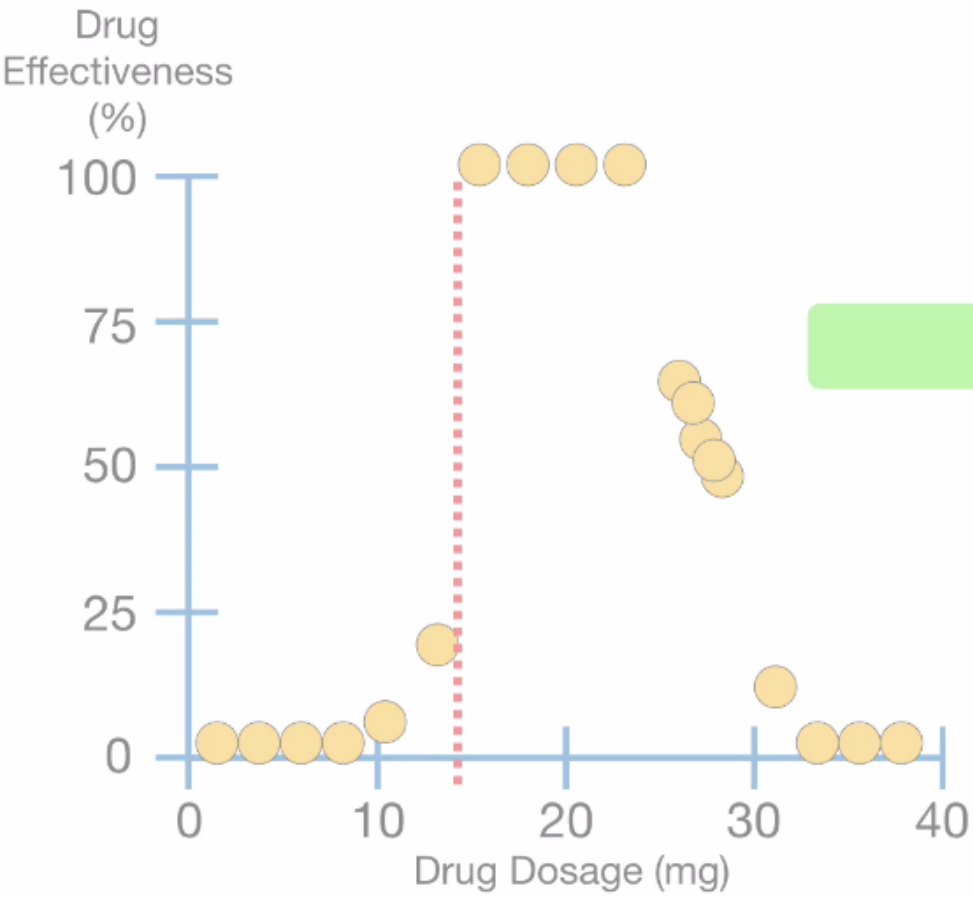


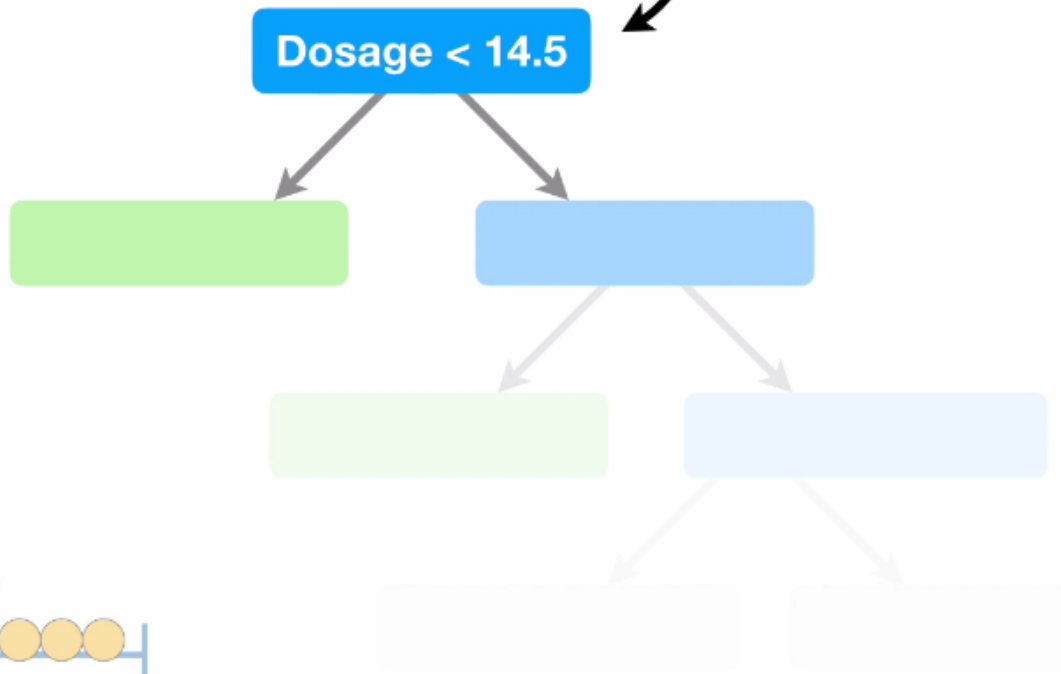

…and a new sum of squared residuals.
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 7
Average=0
Average=43.7
30,000
15,000
0
Sum of Squared Residuals
0
10
20
30
40
Dosage Threshold For Tree

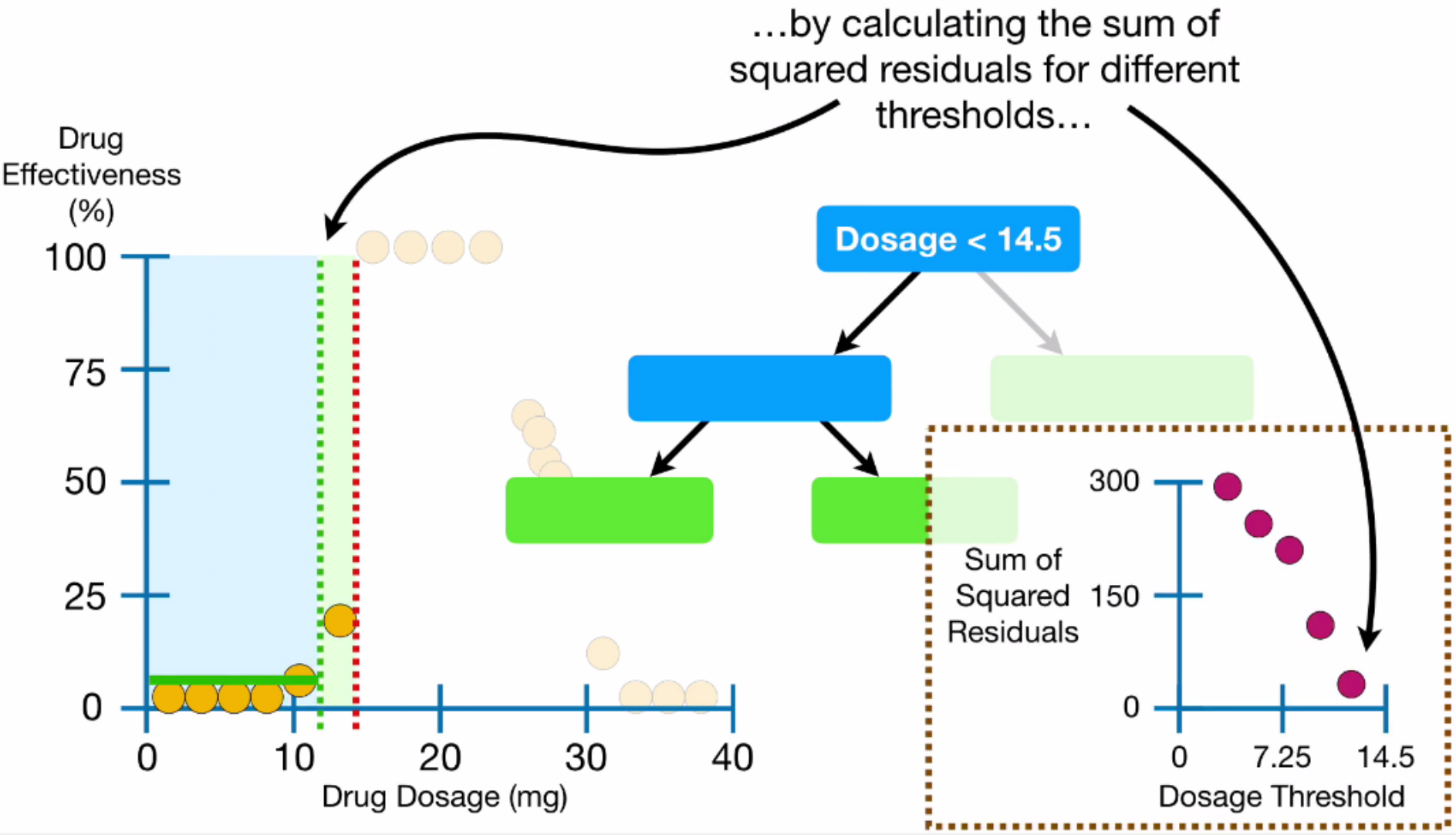

And we repeat until we have calculated the sum of squared residuals for all of the remaining thresholds.
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 36
Average=38.8
Average=0
Sum of Squared Residuals
30,000
15,000
0
0
10
20
30
40
Dosage Threshold For Tree


…and **Dosage < 14.5** had the smallest sum of squared residuals…
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Sum of Squared Residuals
30,000
15,000
0
0
10
20
30
40
Dosage Threshold For Tree

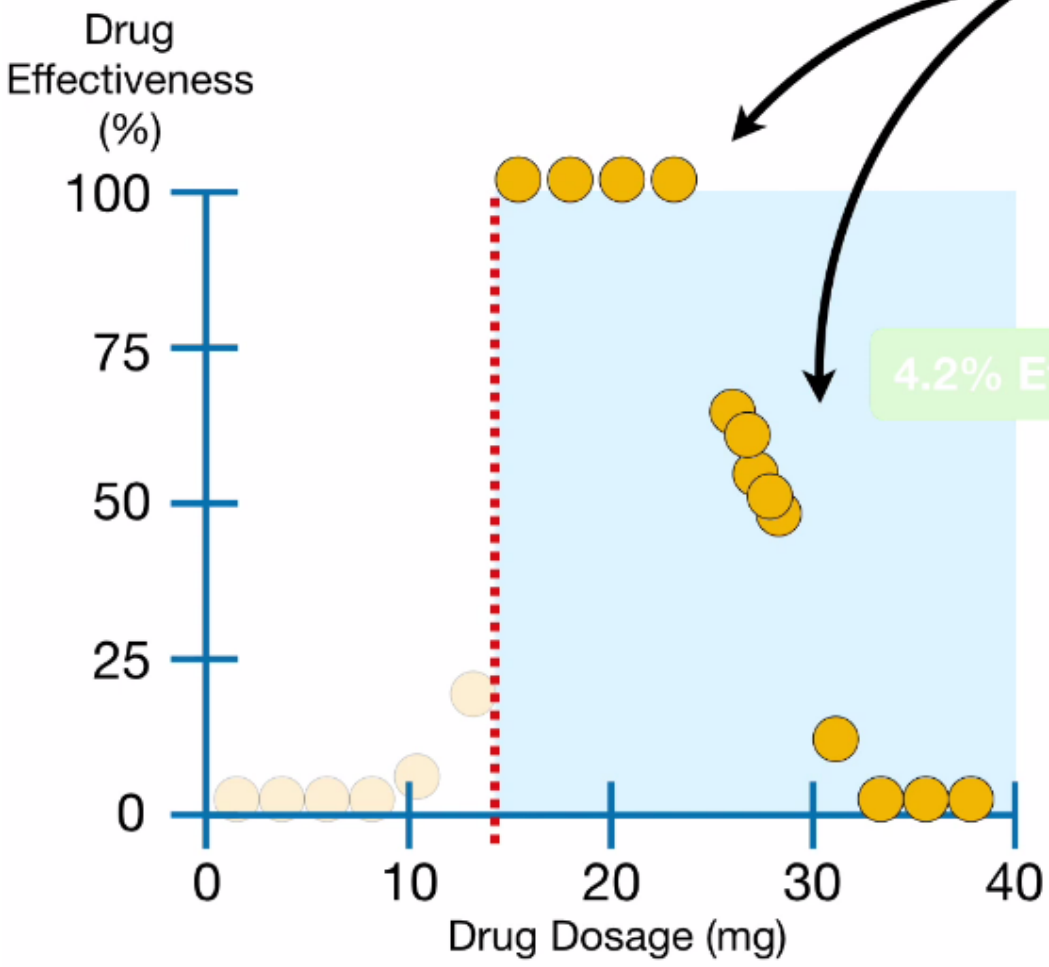
…so **Dosage < 14.5** will be root of the tree.






…by calculating the sum of squared residuals for different thresholds…
Drug Effectiveness (%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)
Dosage < 14.5
Sum of Squared Residuals
300
150
0
0
7.25
14.5
Dosage Threshold

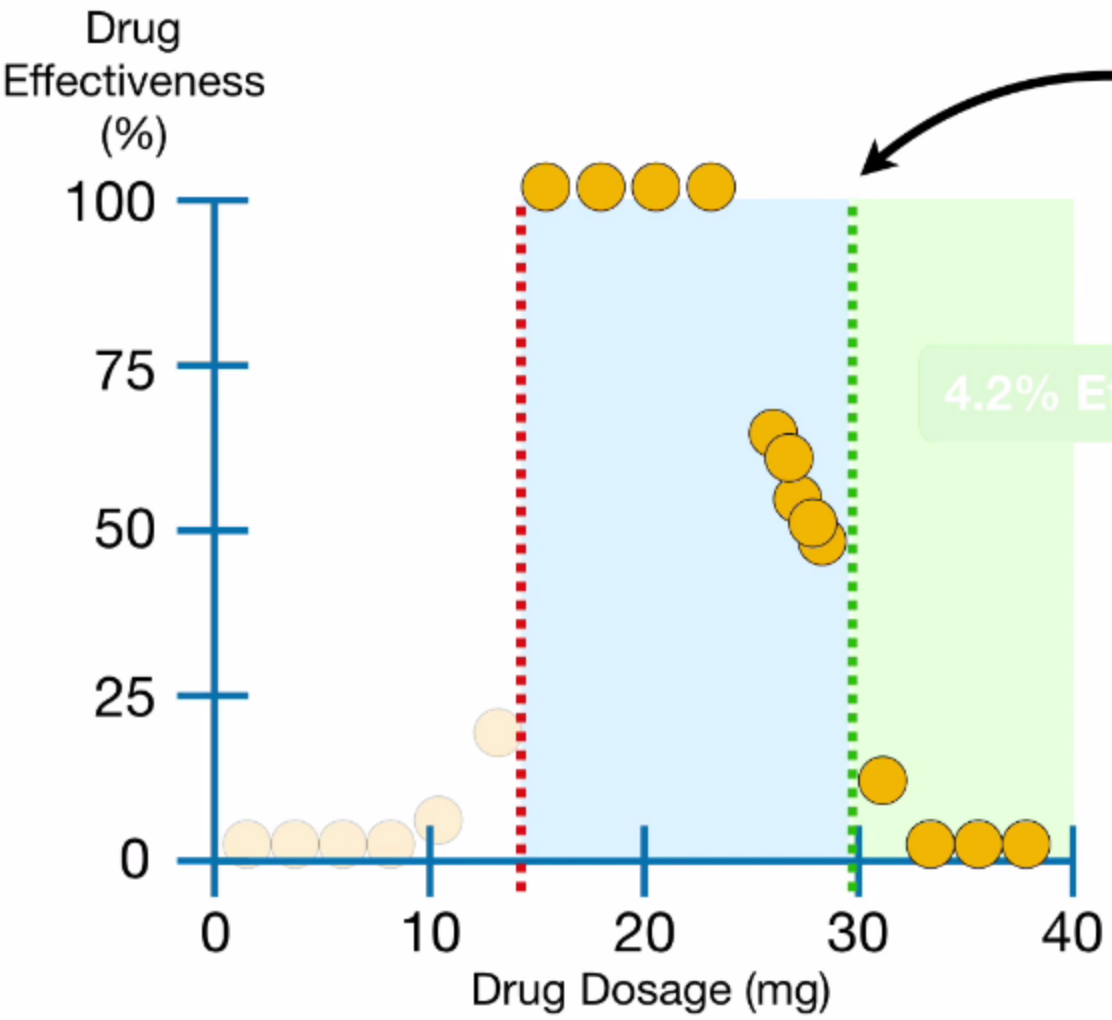
Now we need to figure out what to do with the remaining **13** observations with **Dosages >= 14.5**.

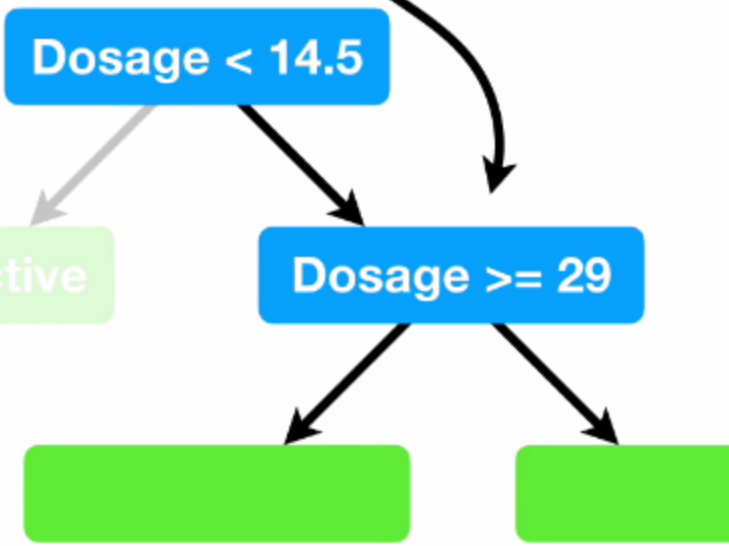


Dosage < 14.5

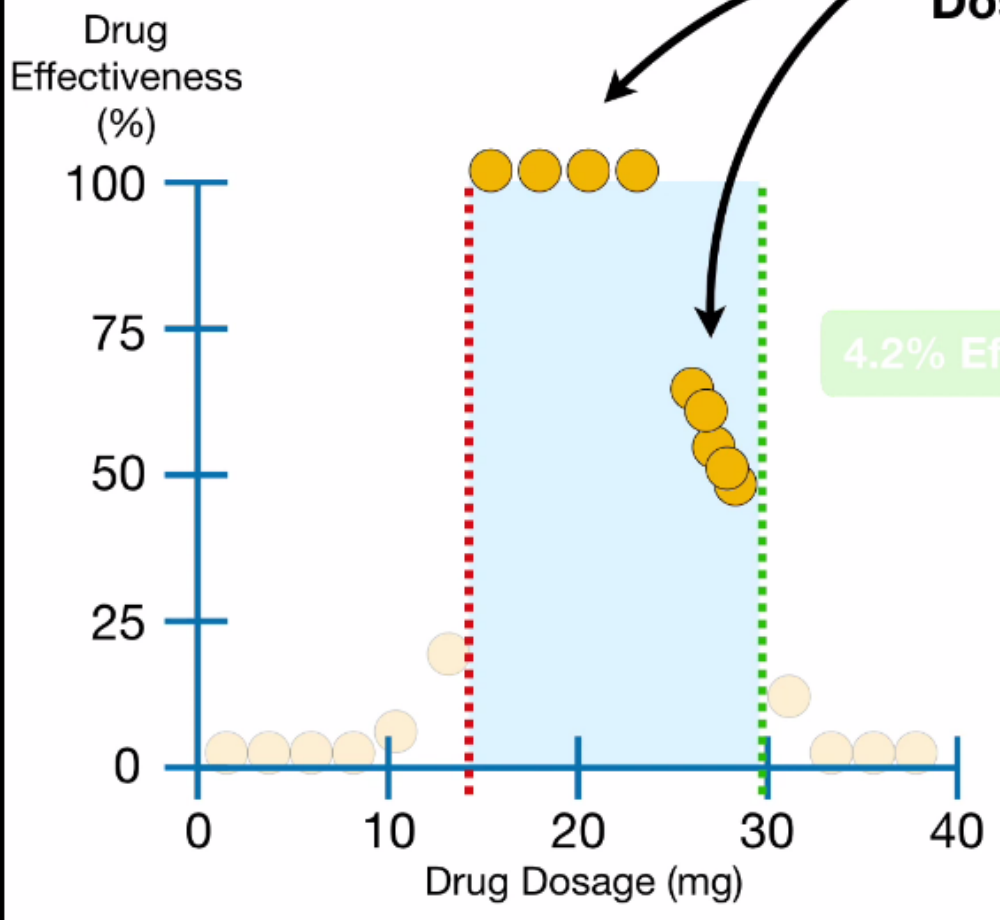
4.2% Effective

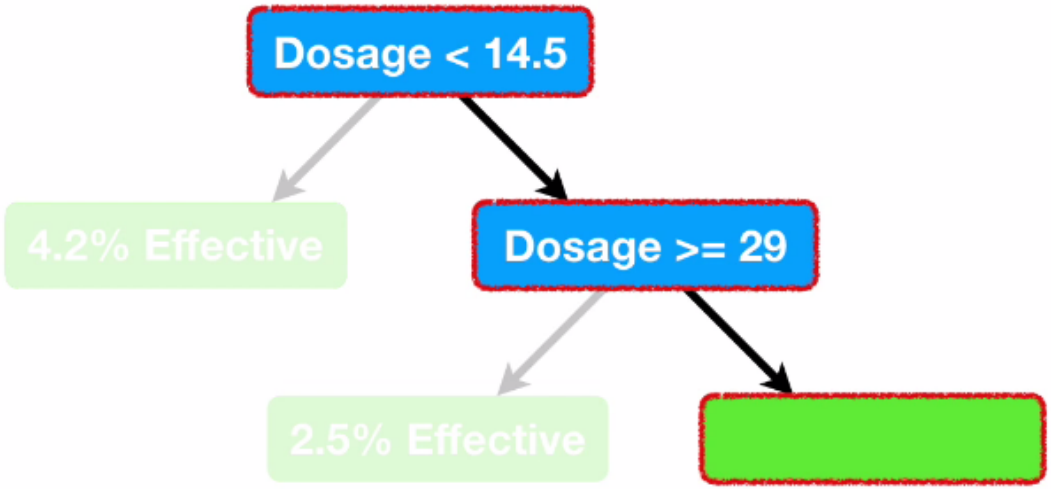
…and we do that by finding the threshold that gives us the smallest sum of squared residuals.



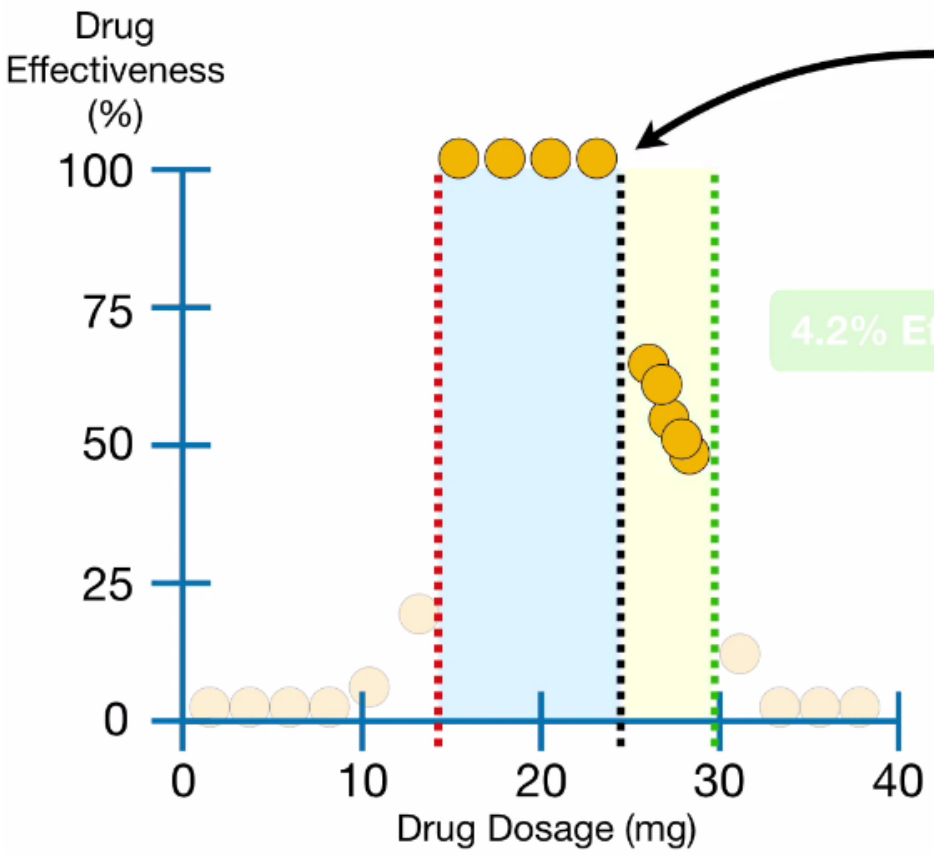


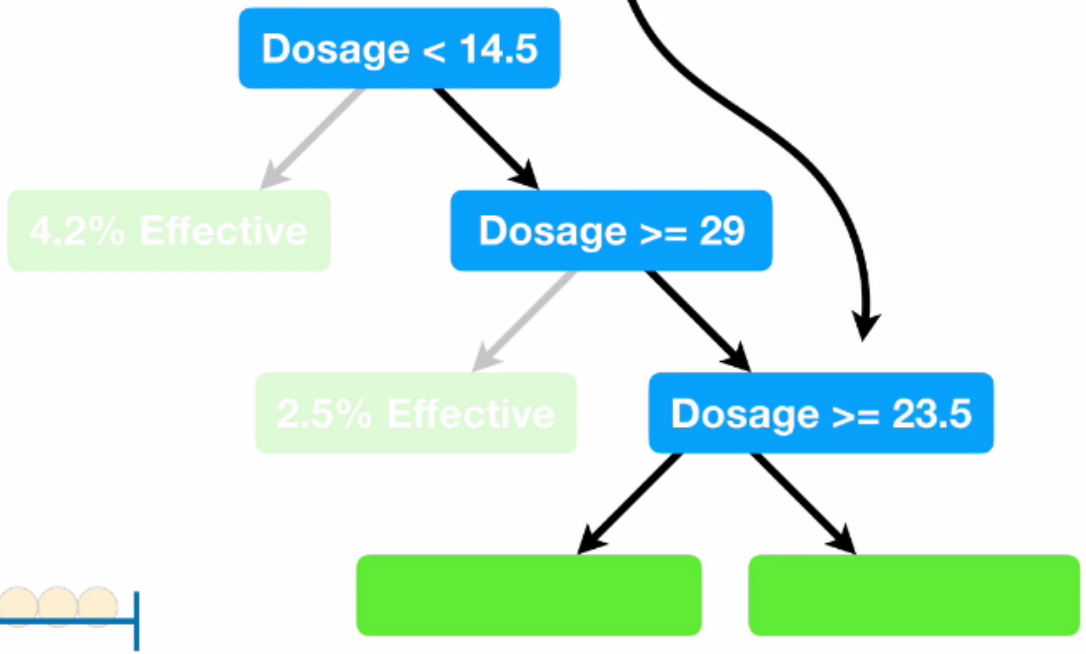
Now we need to figure out what to do with the **9** observations with **Dosages** between **14.5** and **29**.

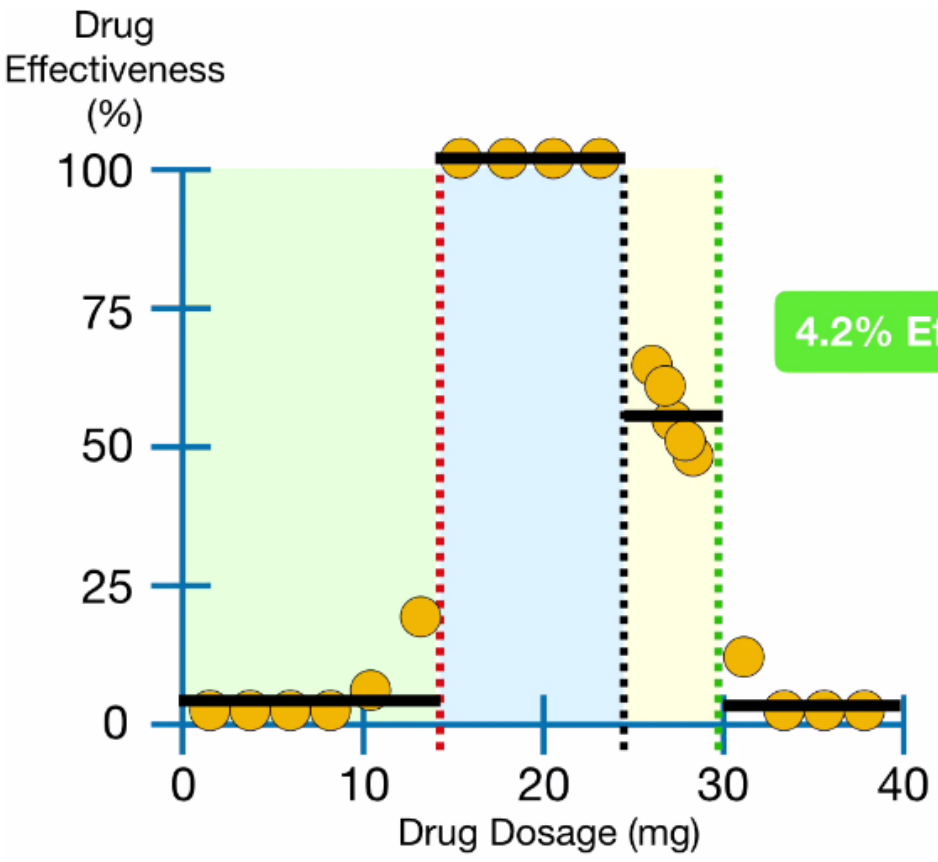


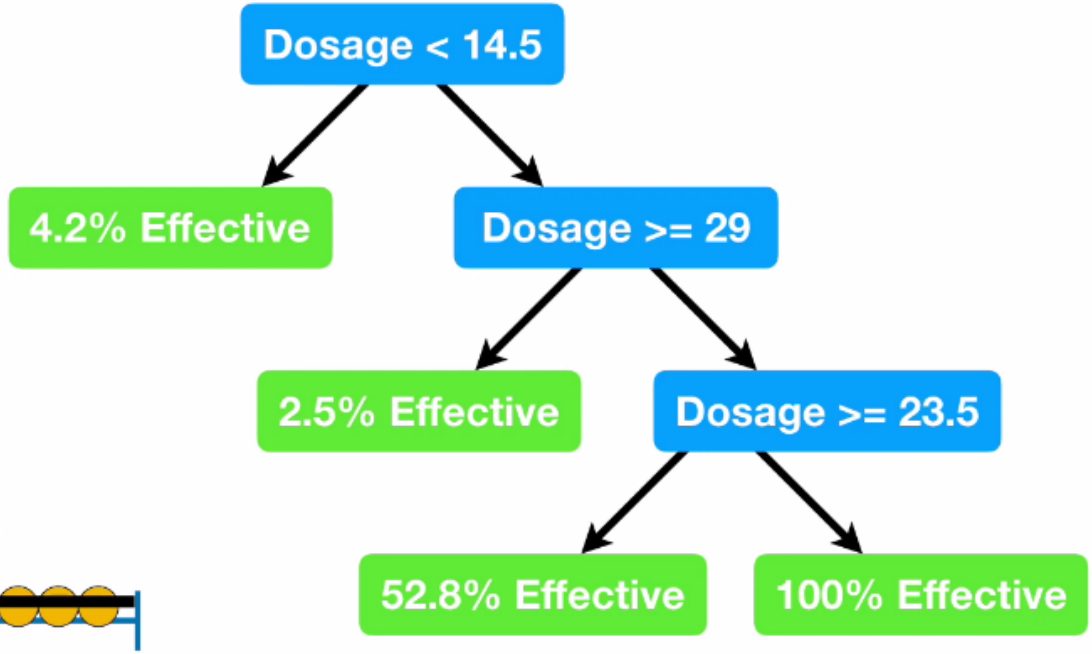


…by finding the threshold that gives us the minimum sum of squared residuals.

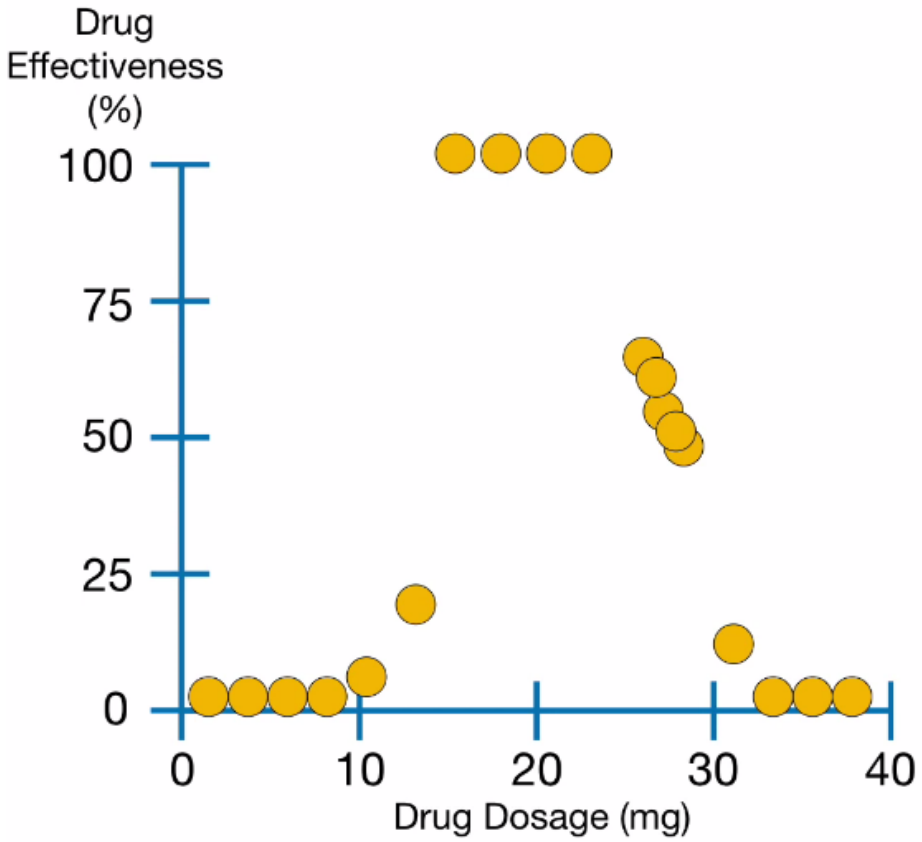





Drug
Effectiveness
(%)
100
75
50
25
0
0
10
20
30
40
Drug Dosage (mg)


Dosage < 14.5
4.2% Effective
Dosage >= 29
2.5% Effective
Dosage >= 23.5
52.8% Effective
100% Effective

So far we have built a tree using a single predictor, **Dosage**, to predict **Drug Effectiveness**.

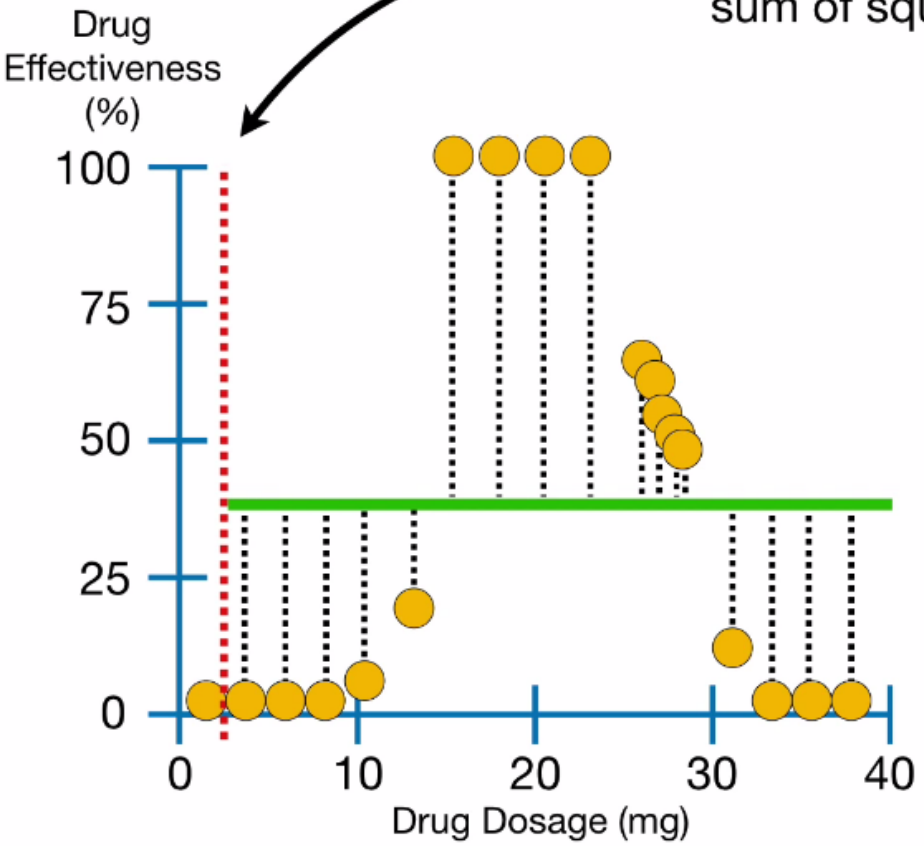


Dosage	Drug Effect.
10	58
20	60
35	57
5	44
etc…	etc…

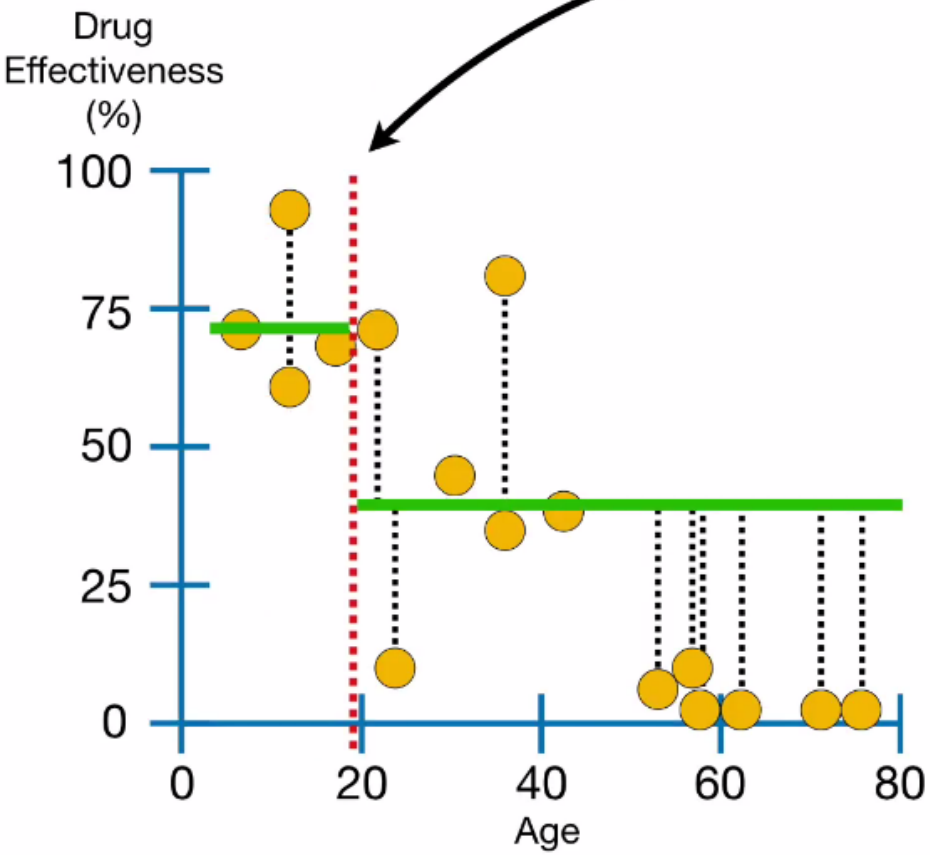
Now let's talk about how to build a tree to predict **Drug Effectiveness** using a bunch of predictors.

Dosage	Age	Gender	Drug Effect.
10	25	Female	98
20	73	Male	0
35	54	Female	6
5	12	Male	44
etc…	etc…	etc…	etc…

Thus, just like before, we will try different thresholds for **Dosage** and calculate the sum of squared residuals at each step…



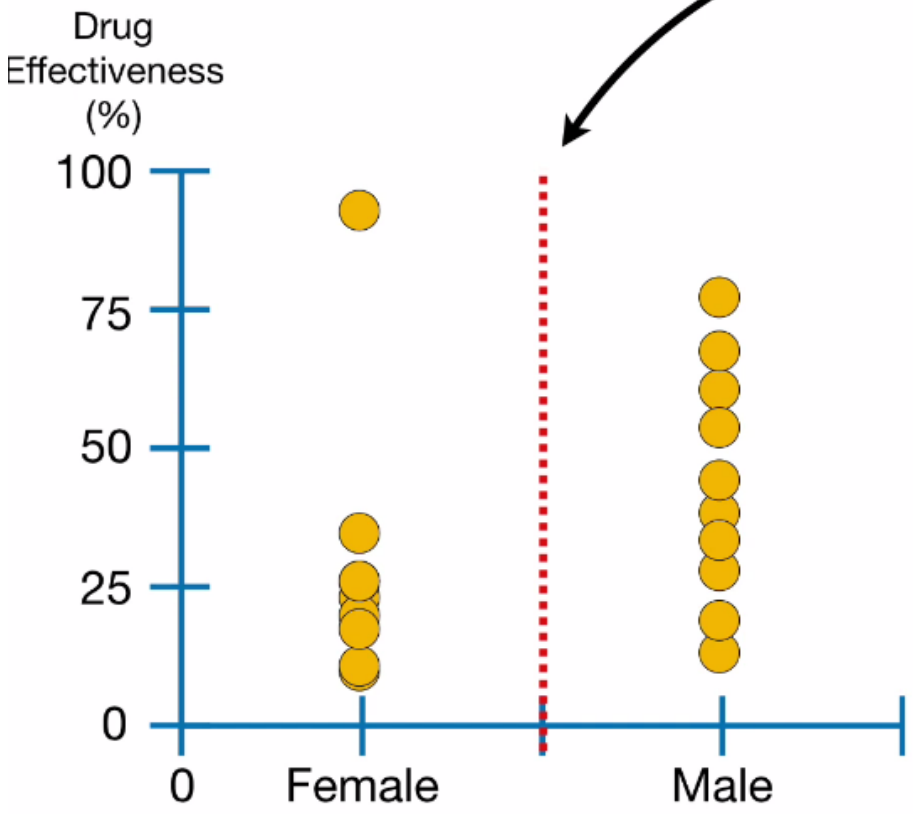
Dosage	Age	Gender	Drug Effect.
10	25	Female	98
20	73	Male	0
35	54	Female	6
5	12	Male	44
etc…	etc…	etc…	etc…



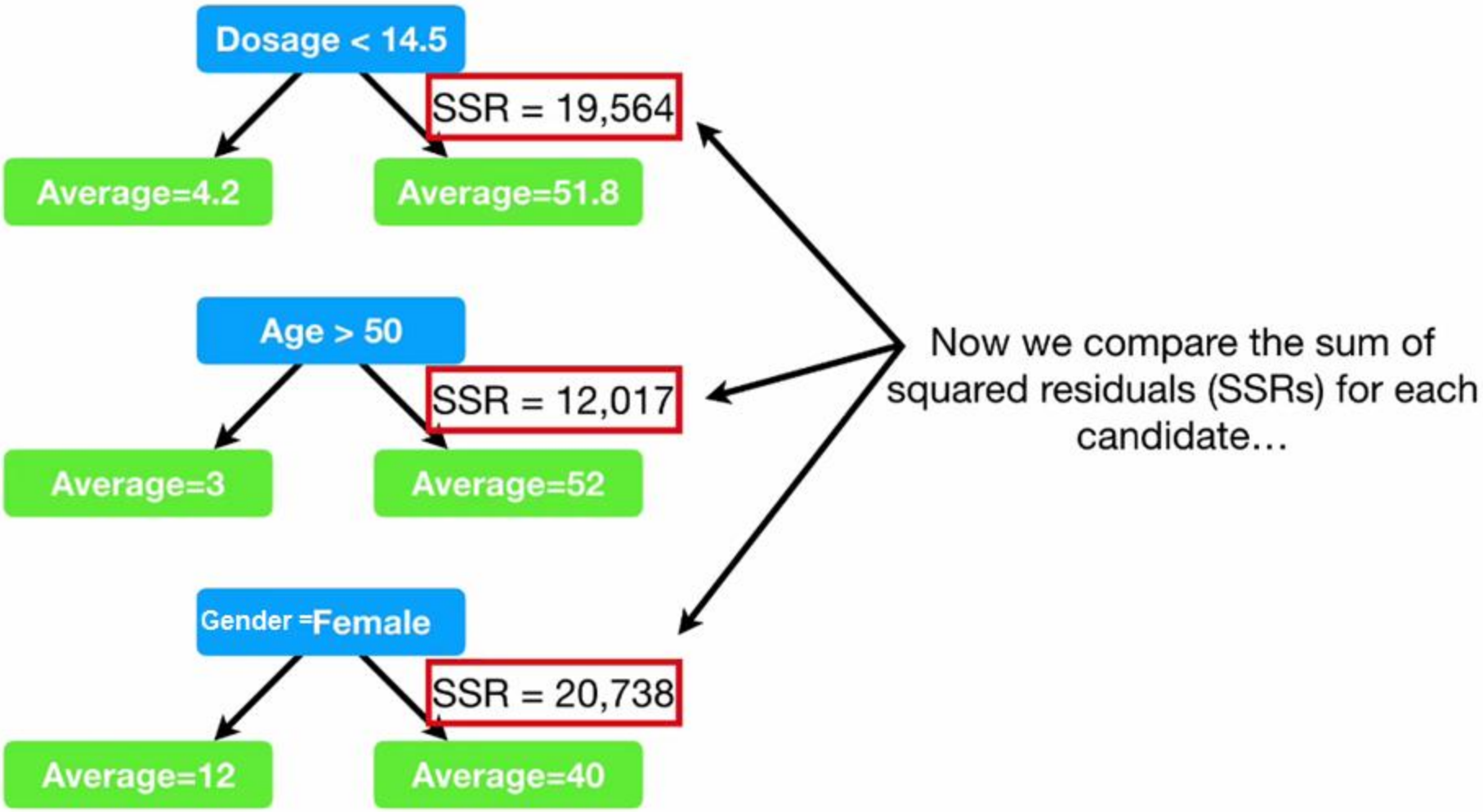
Just like with **Dosage**, we try different thresholds for **Age** and calculate the sum of squared residuals at each step…

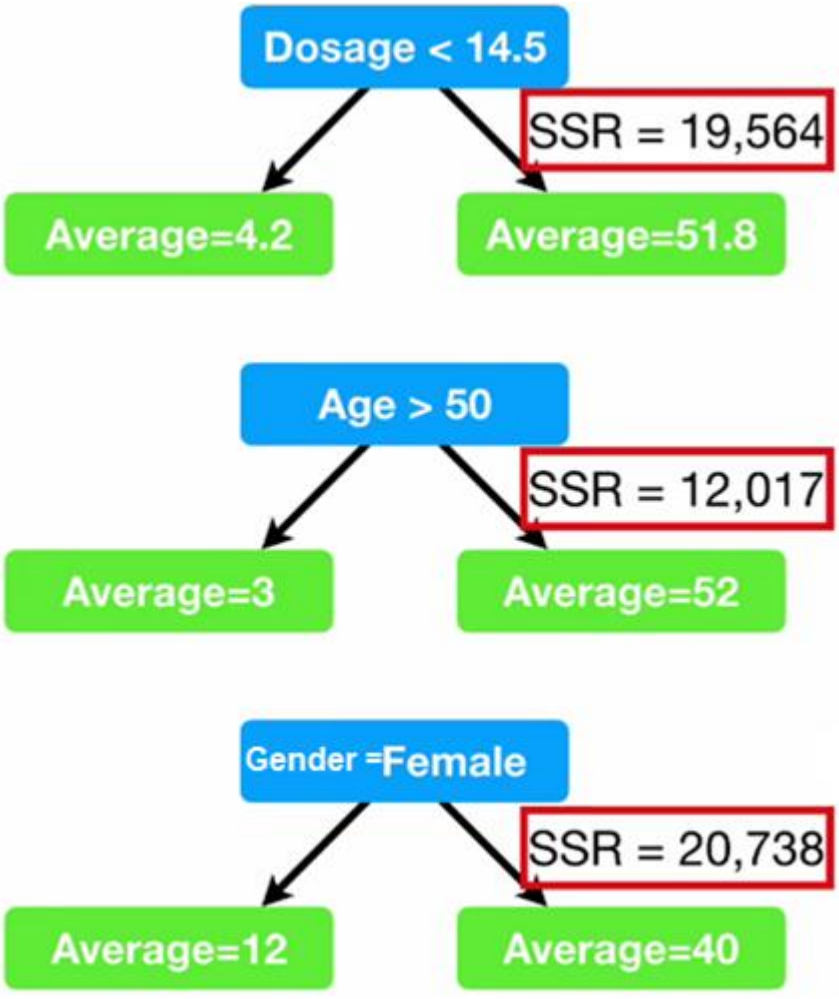
Dosage	Age	Gender	Drug Effect.
10	25	Female	98
20	73	Male	0
35	54	Female	6
5	12	Male	44
etc…	etc…	etc…	etc…

there is only one
threshold to try…



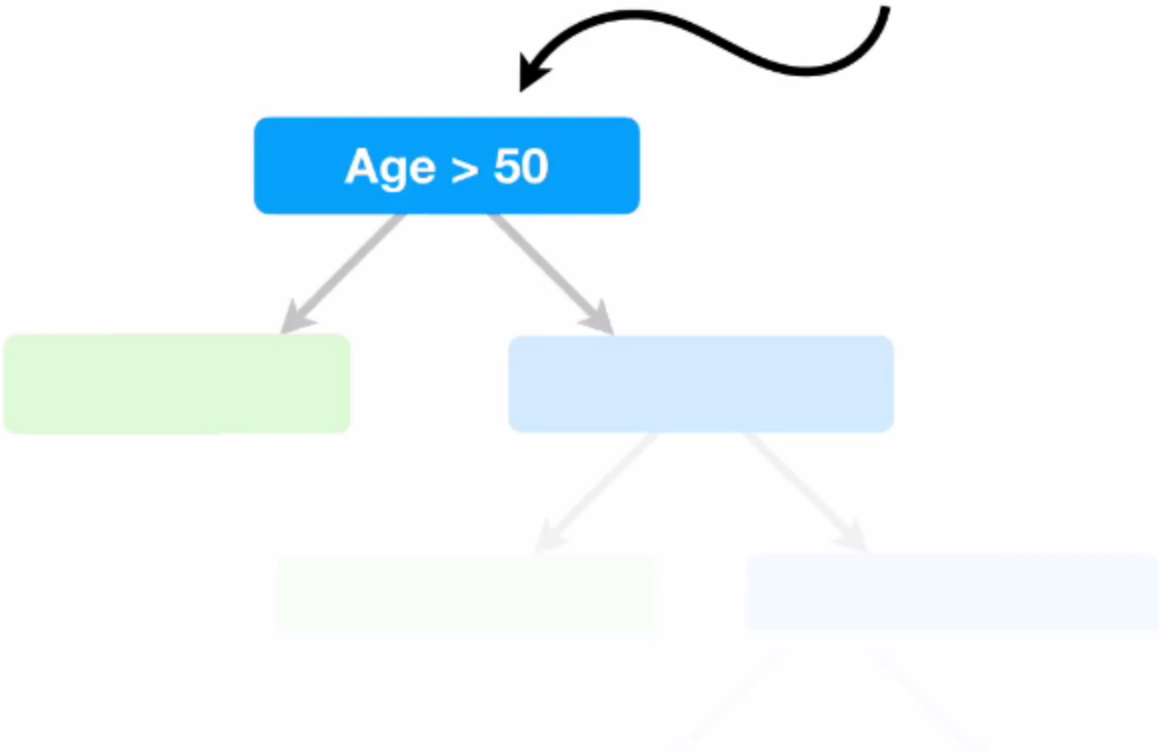
Dosage	Age	Gender	Drug Effect.
10	25	Female	98
20	73	Male	0
35	54	Female	6
5	12	Male	44
etc…	etc…	etc…	etc…


Dosage < 14.5
SSR = 19,564
Average=4.2
Average=51.8
Age > 50
SSR = 12,017
Average=3
Average=52
Gender =Female
SSR = 20,738
Average=12
Average=40
Now we compare the sum of squared residuals (SSRs) for each candidate…

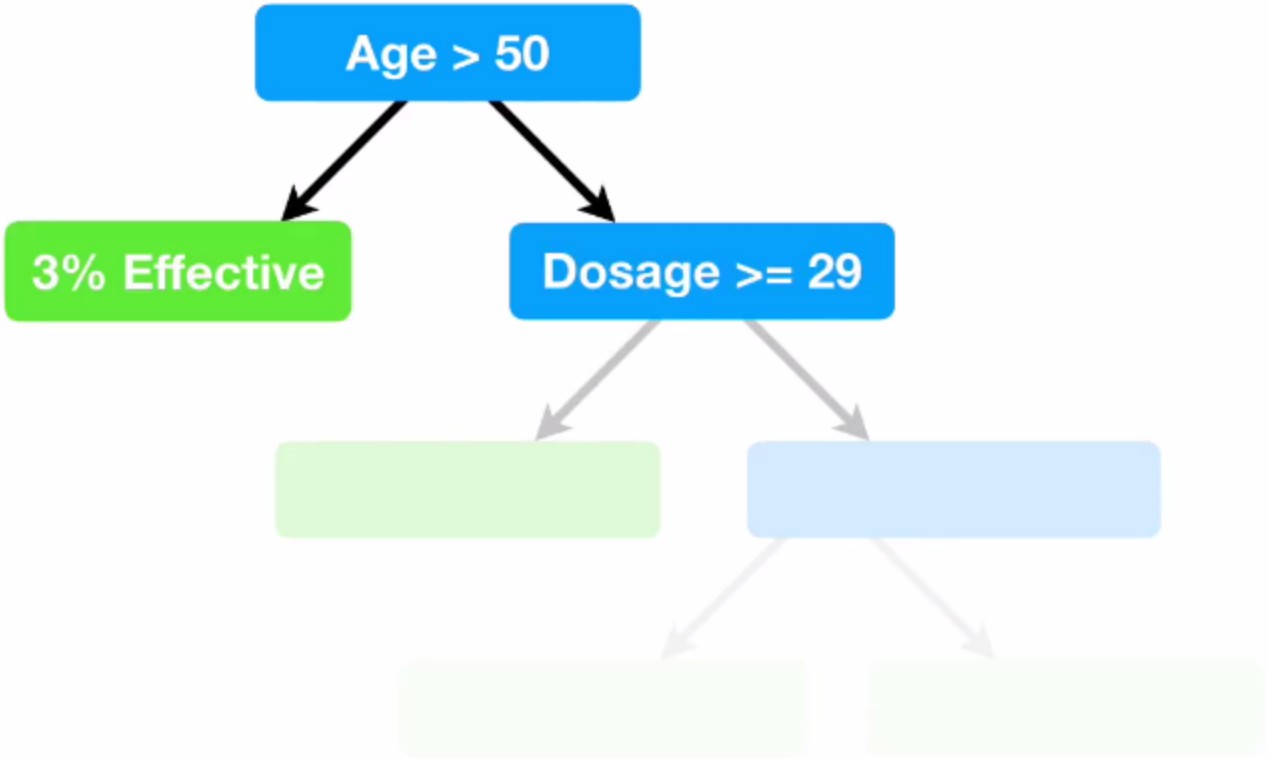


…and pick the candidate with the lowest value.

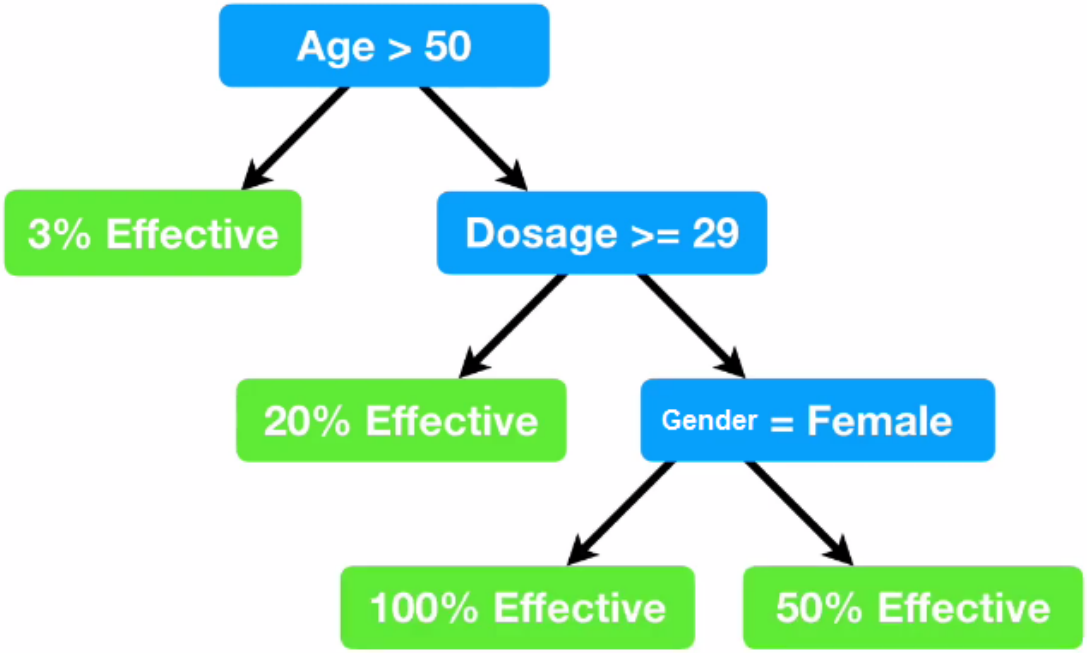
Since **Age > 50** had the lowest sum of squared residuals, it becomes the root of the tree.



Then we grow the tree just like before, except now we compare the lowest sum of squared residuals from each predictor.

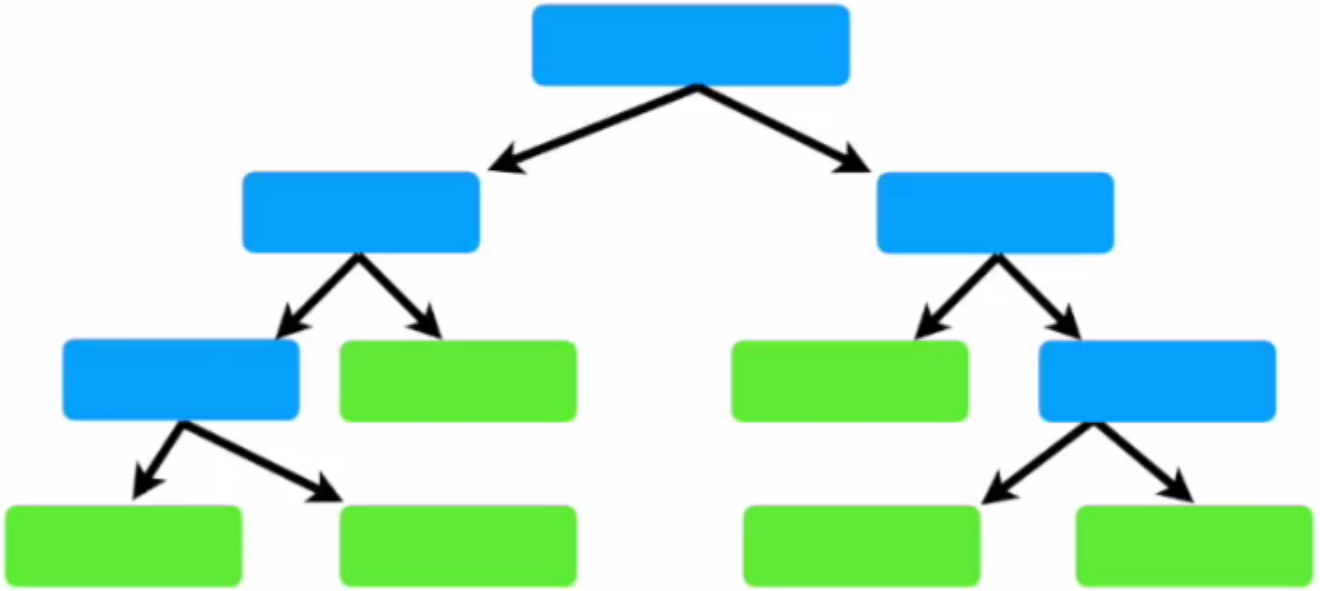


Then we grow the tree just like before, except now we compare the lowest sum of squared residuals from each predictor.

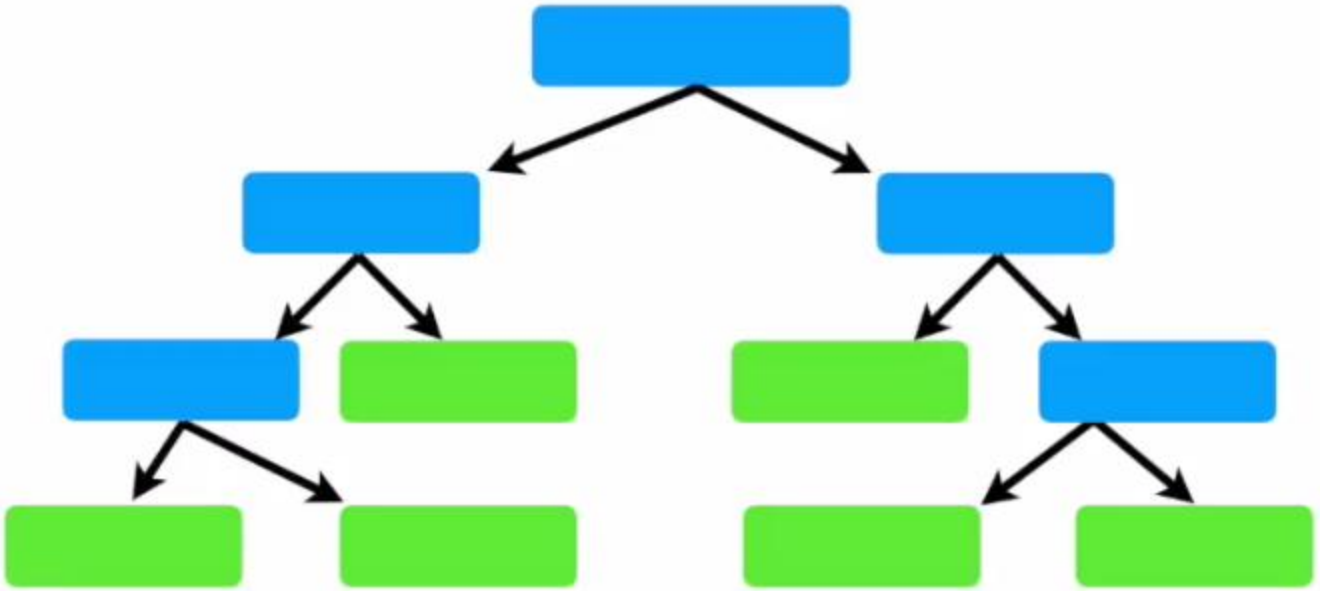


Dosage	Age	Gender	Drug Effect.
10	25	Female	98
20	73	Male	0
35	54	Female	6
5	12	Male	44
etc…	etc…	etc…	etc…

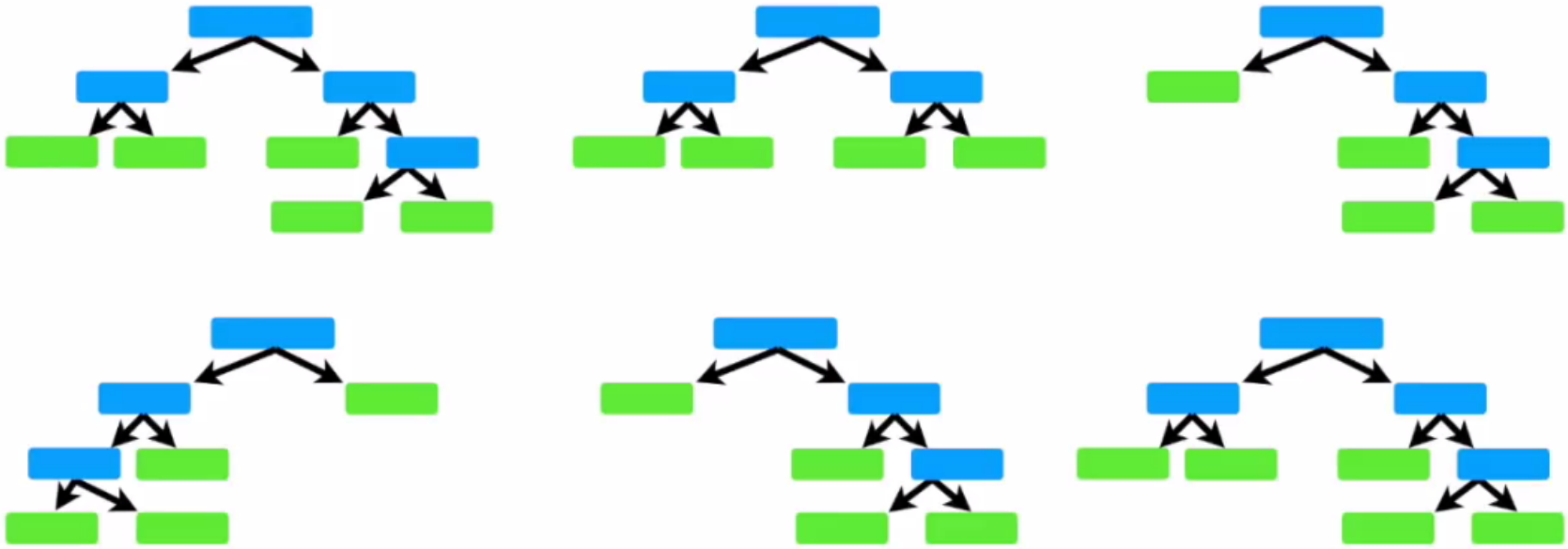
Decision Trees are easy to build, easy to use and easy to interpret…



"Trees have one aspect that prevents them from being the ideal tool for predictive learning, namely **inaccuracy**."



Random Forests combine
the simplicity of decision trees with flexibility
resulting in a vast improvement in accuracy.



Original Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	Yes	167	Yes

Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease

To create a bootstrapped dataset that is the same size as the original, we just randomly select samples from the original dataset.

Original Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	Yes	167	Yes

Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

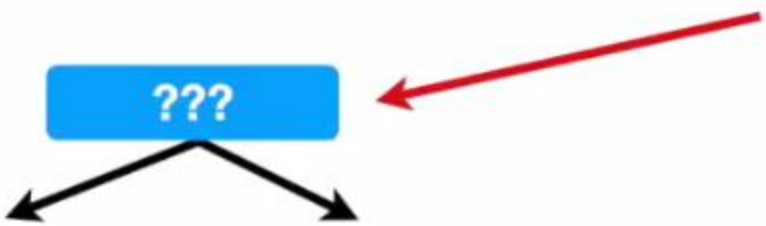
Create a decision tree using the bootstrapped dataset, but only use a random subset of variables (or columns) at each step.

In this example, we will only consider 2 variables (columns) at each step.

Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

Thus, instead of considering all 4 variables to figure out how to split the root node…



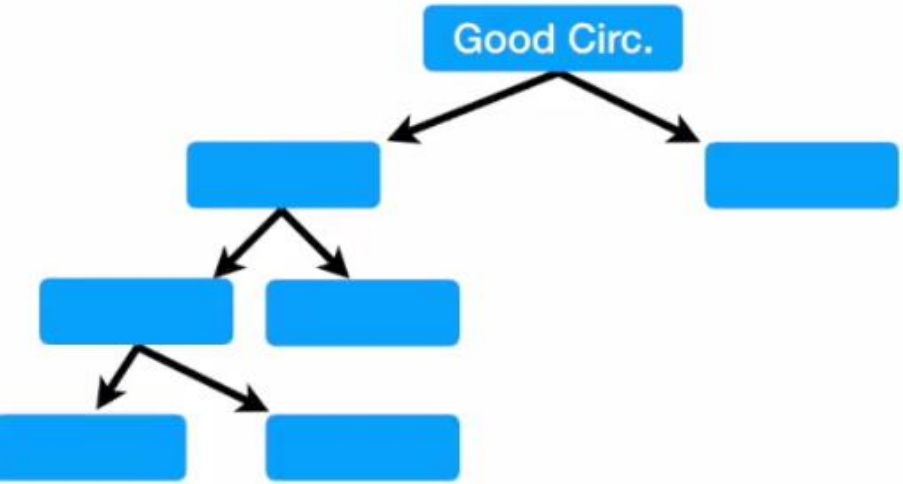
Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

…we randomly select 2.

Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes



Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

And we just build the tree as usual, but only considering a random subset of variables at each step.

Now go back to Step 1 and repeat: Make a new bootstrapped dataset and build a tree considering a subset of variables at each step.