Introduction to Machine Learning – Decision Trees

Dr. Ab Mosca (they/them)

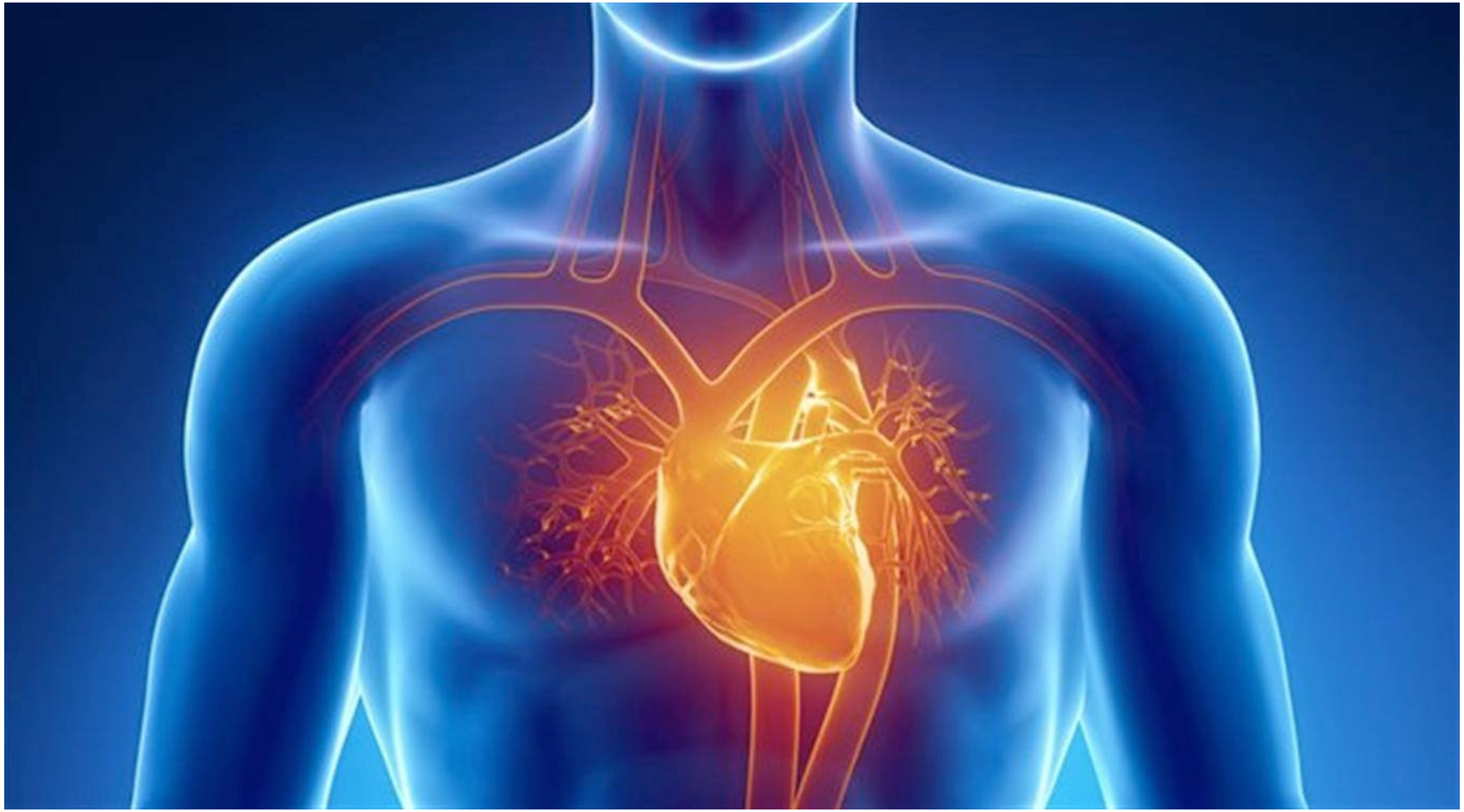
Slides based off slides courtesy of Jordan Crouser (https://jcrouser.github.io/)

Plan for Today

Basic mechanics of tree-based methods

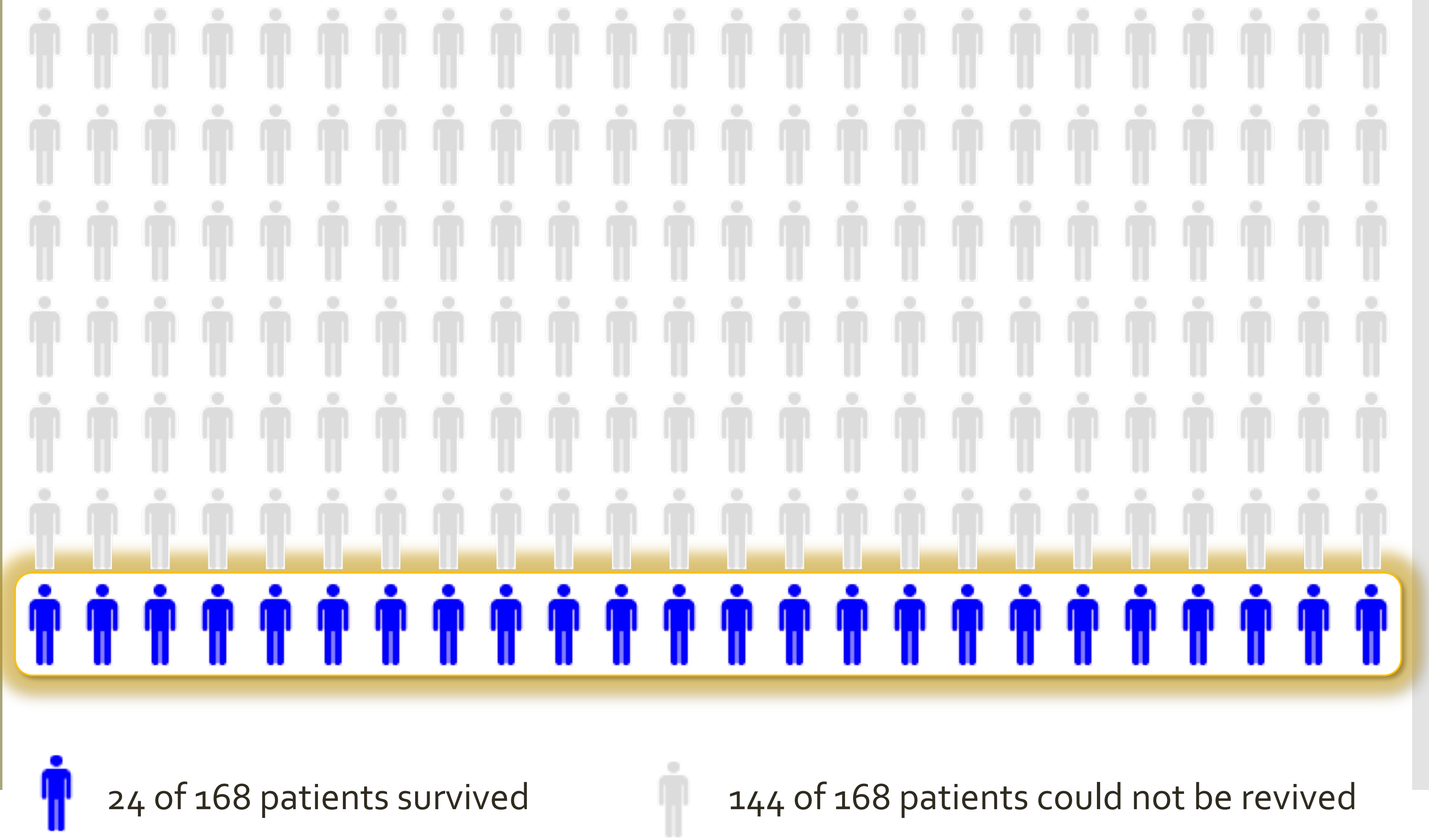
- Classification example
- Choosing good splits
- Pruning

Example: surviving cardiac arrest



Example: surviving cardiac arrest

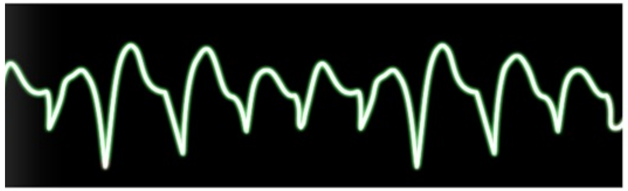
Full dataset: 168 patients



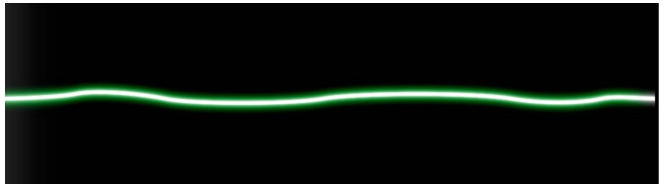
Crystal ball: best predictor

Different types of arrhythmia

Normal



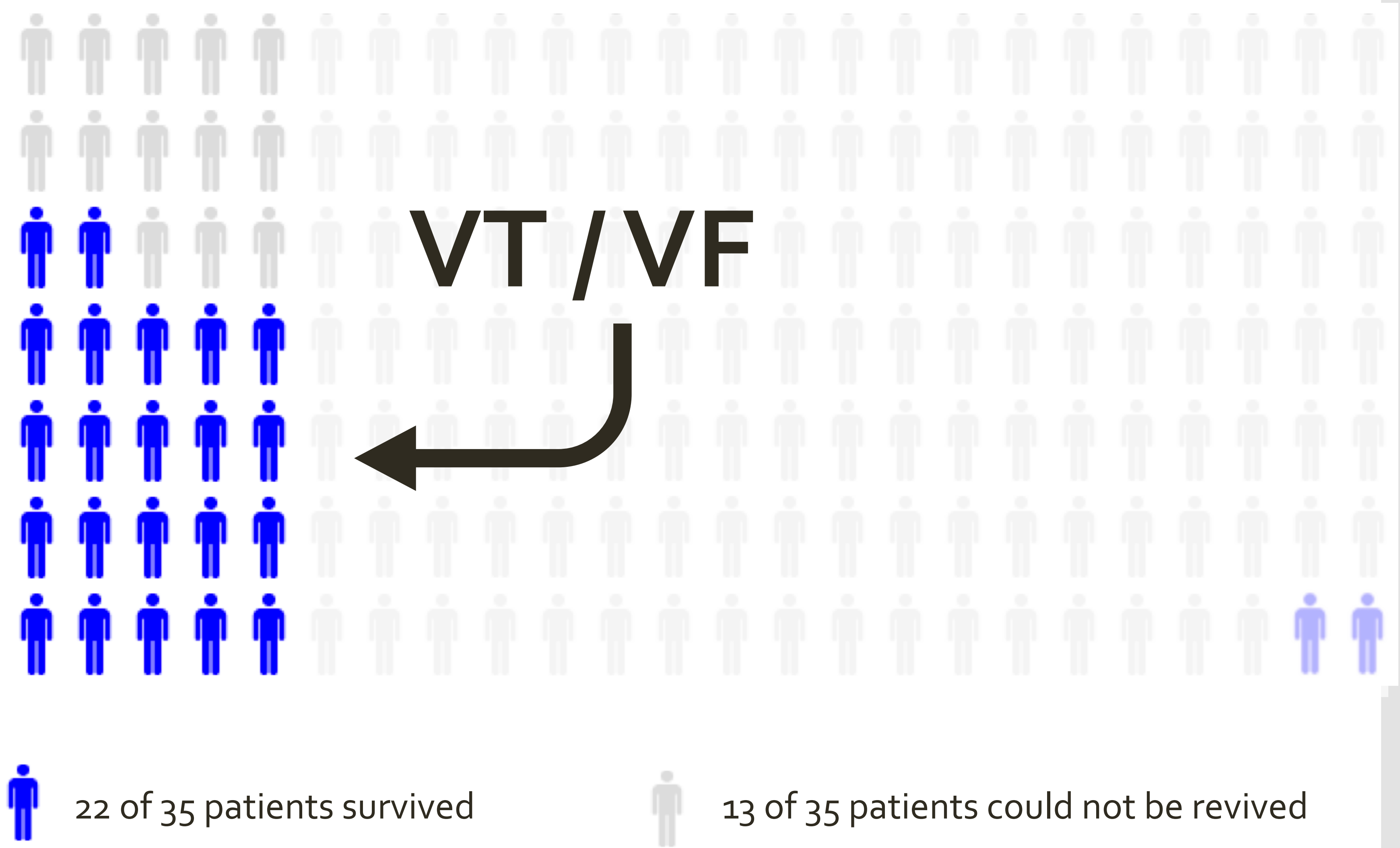
Ventricular Tachycardia (VT) / Ventricular Fibrillation (VF)



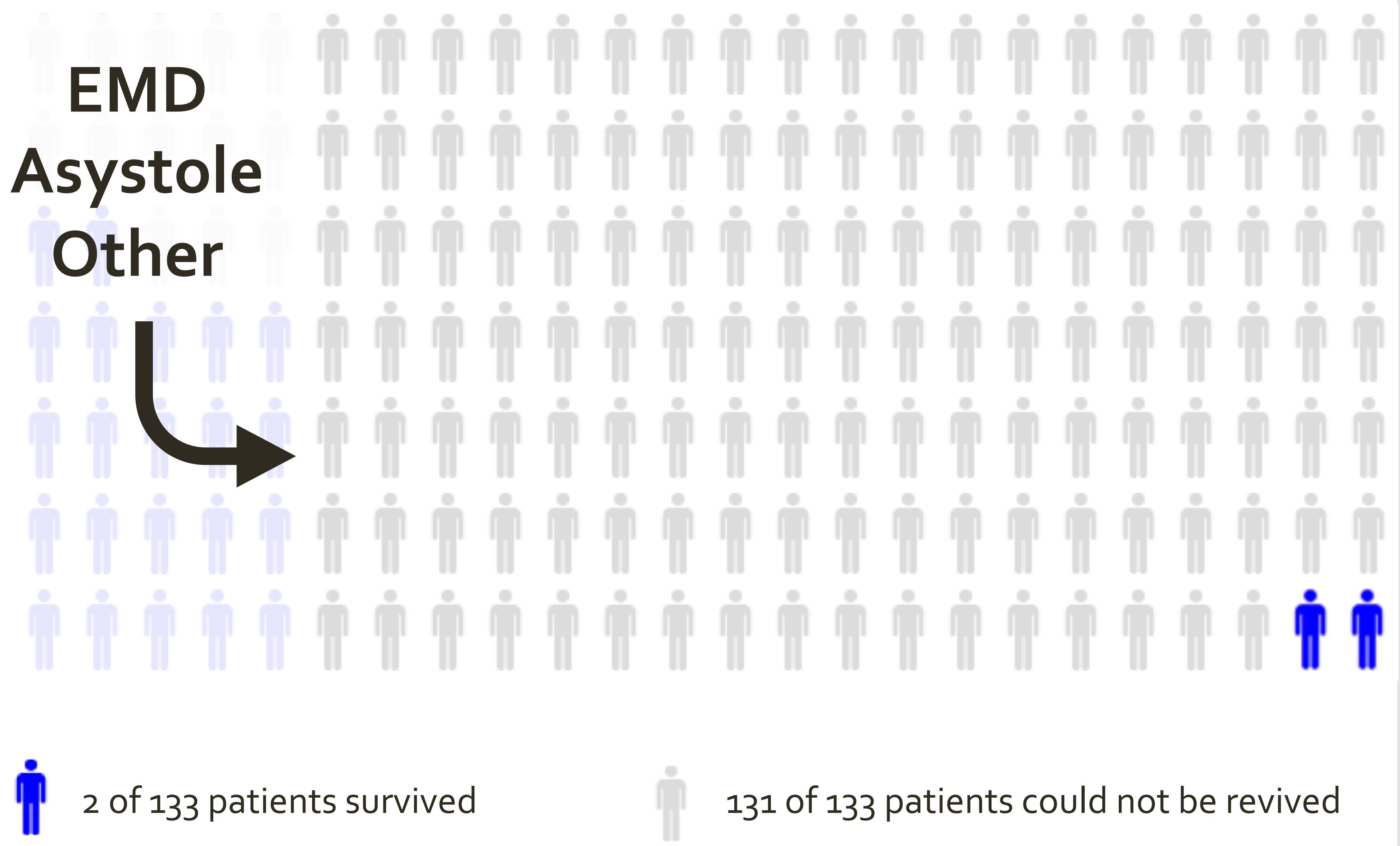
EMD / Asystole / Other

Images courtesy of Wikipedia

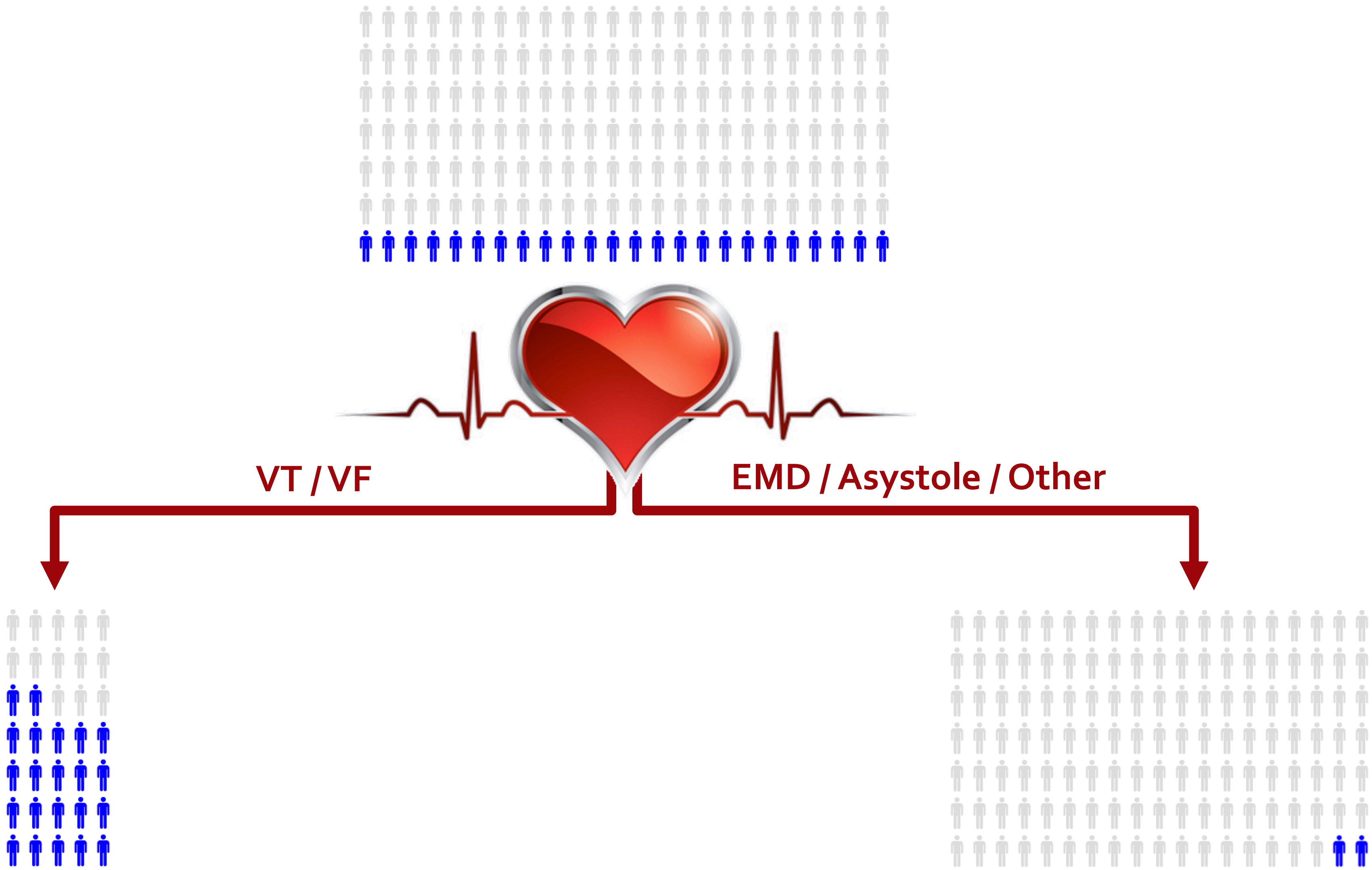
First Split: Initial Heart Rhythm



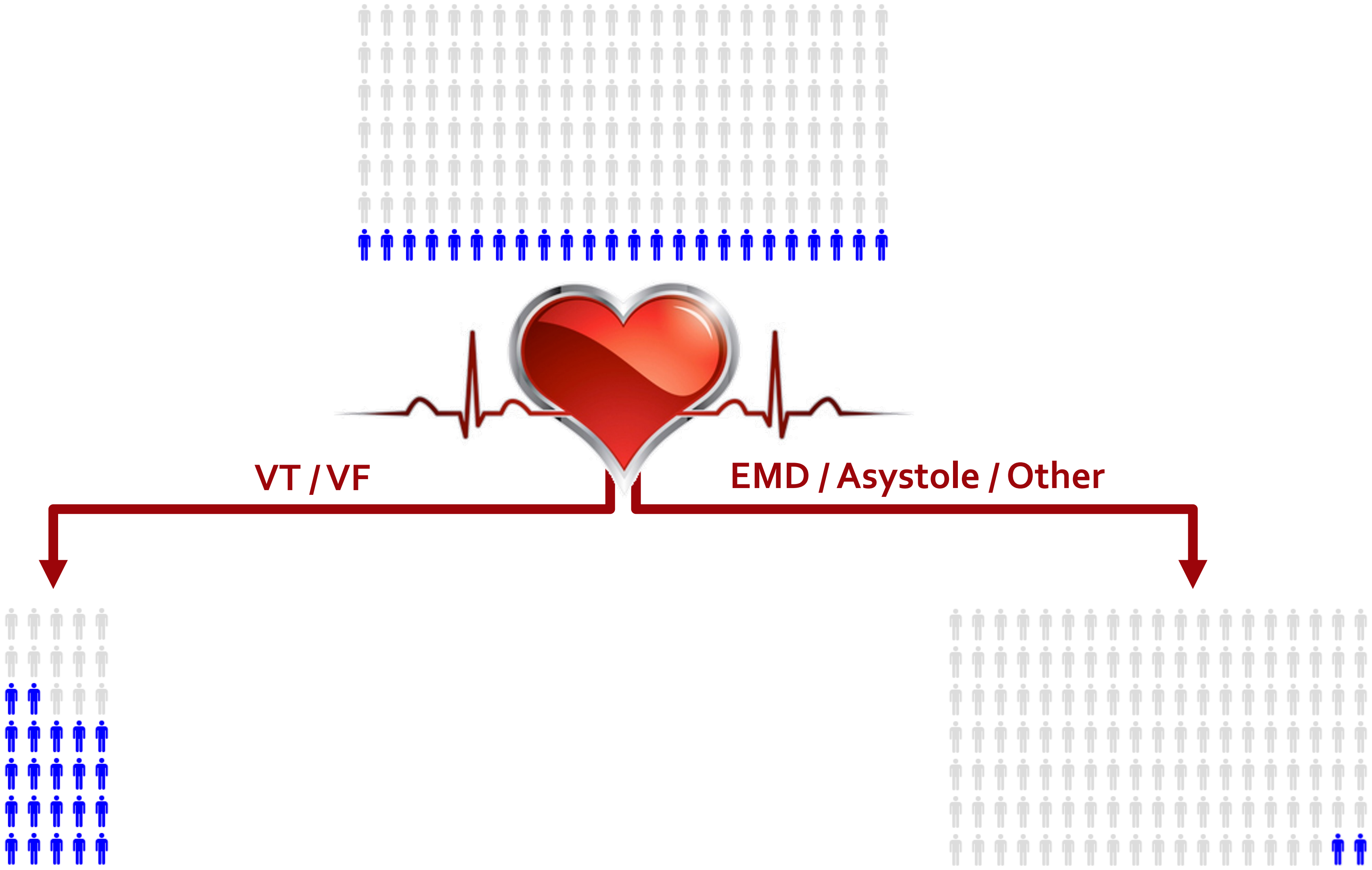
First Split: Initial Heart Rhythm



Another view: partitioning

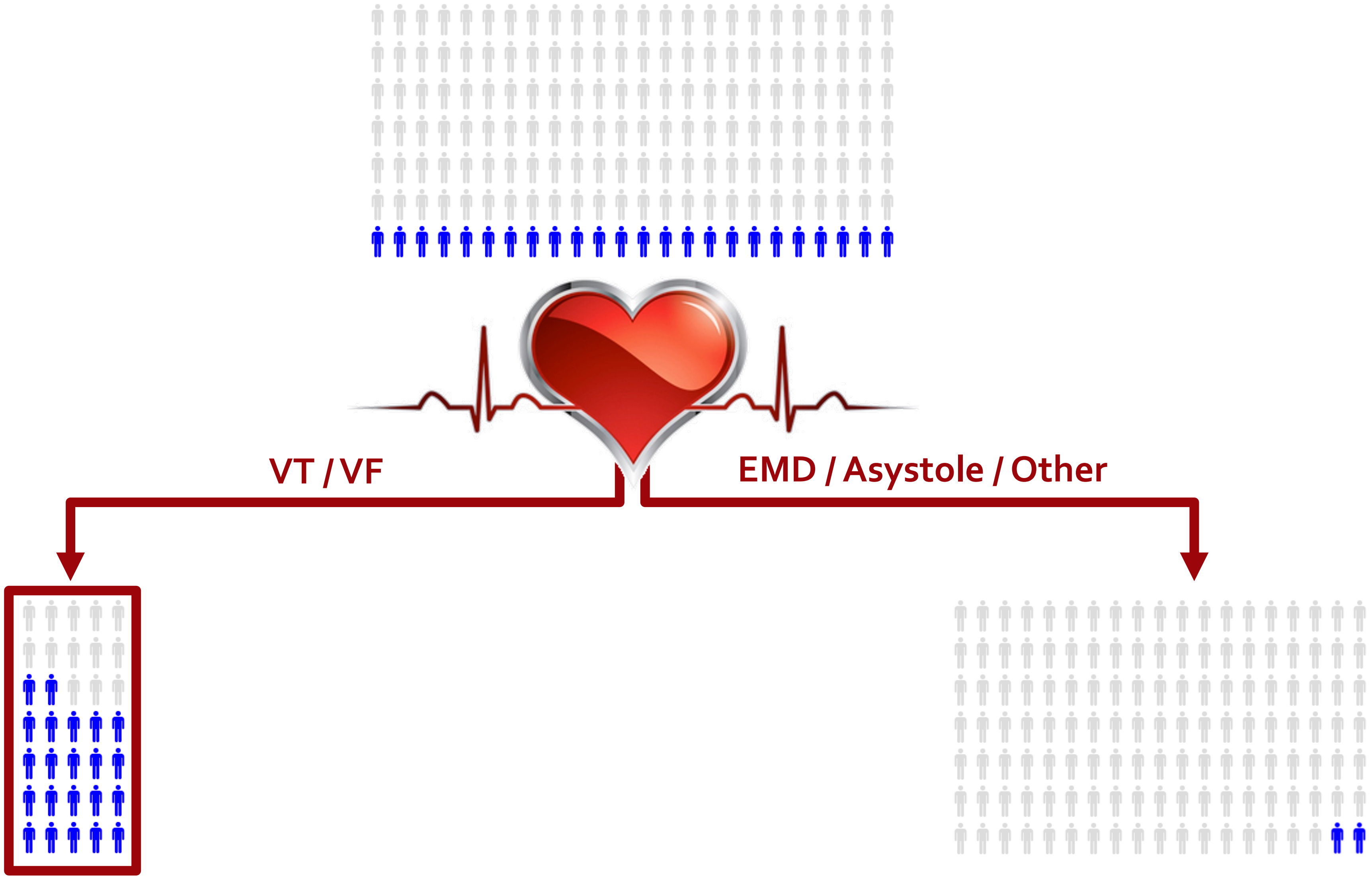


Now what do we do?

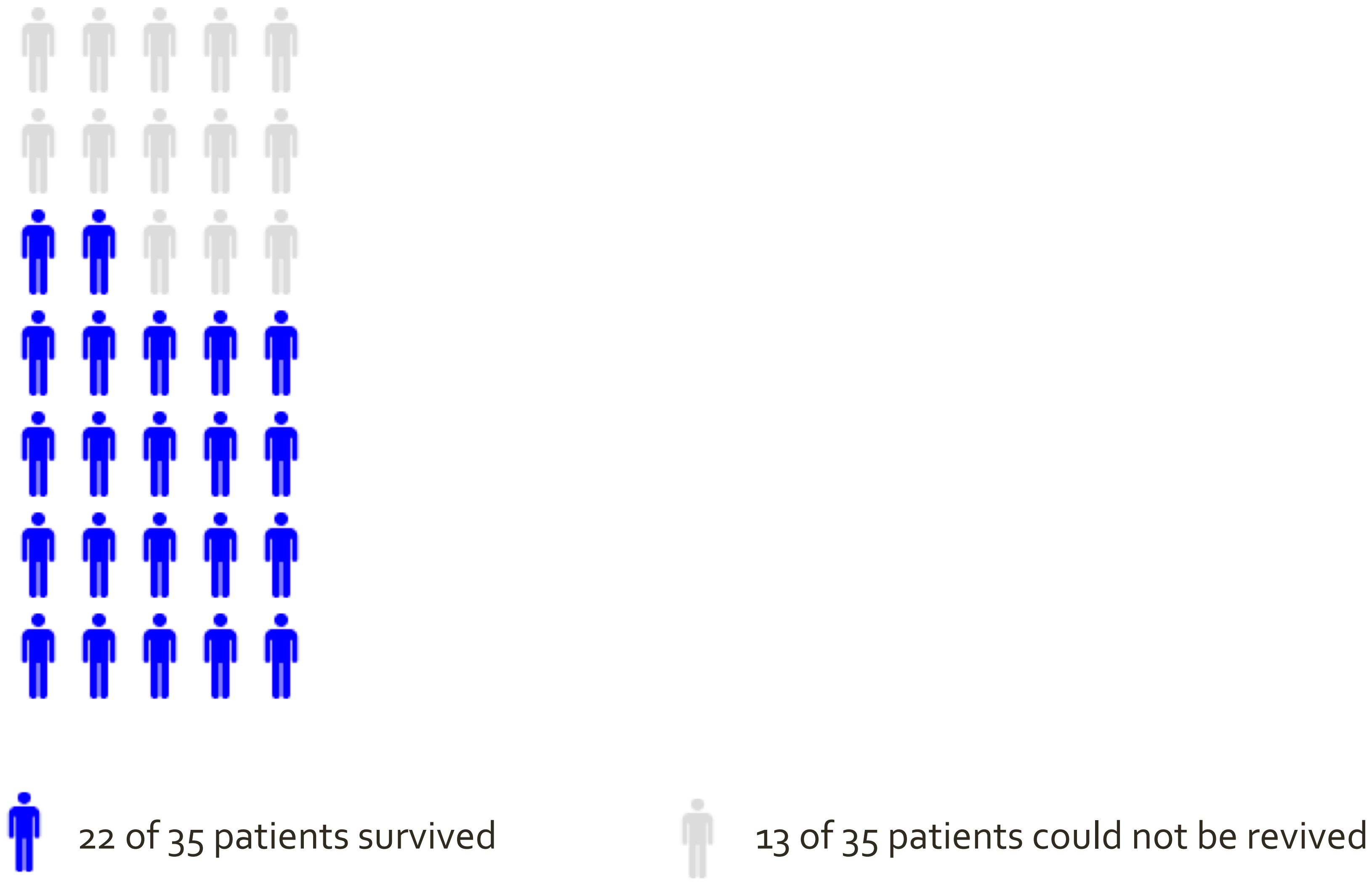


Ideas?

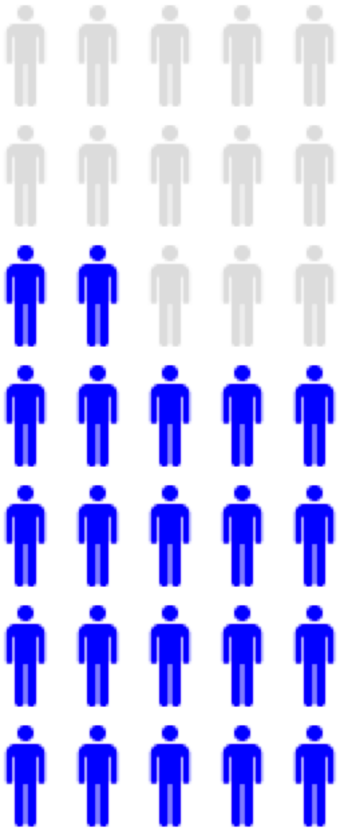
Recursion!

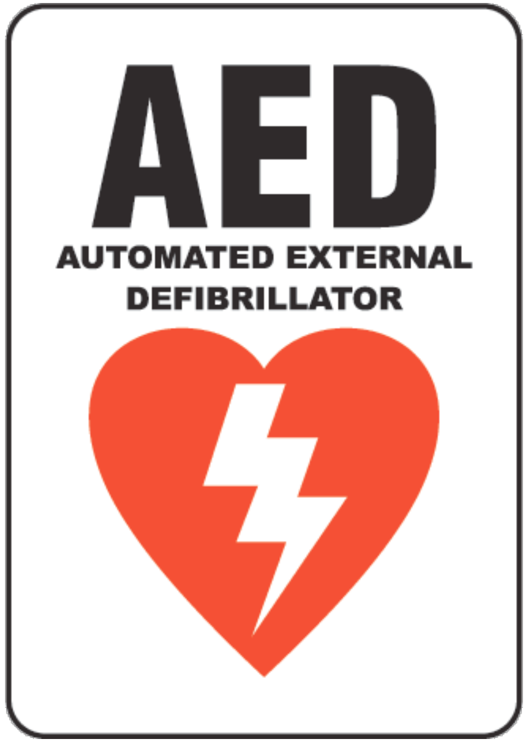


VT / VF group only



Next split: response to defibrillation

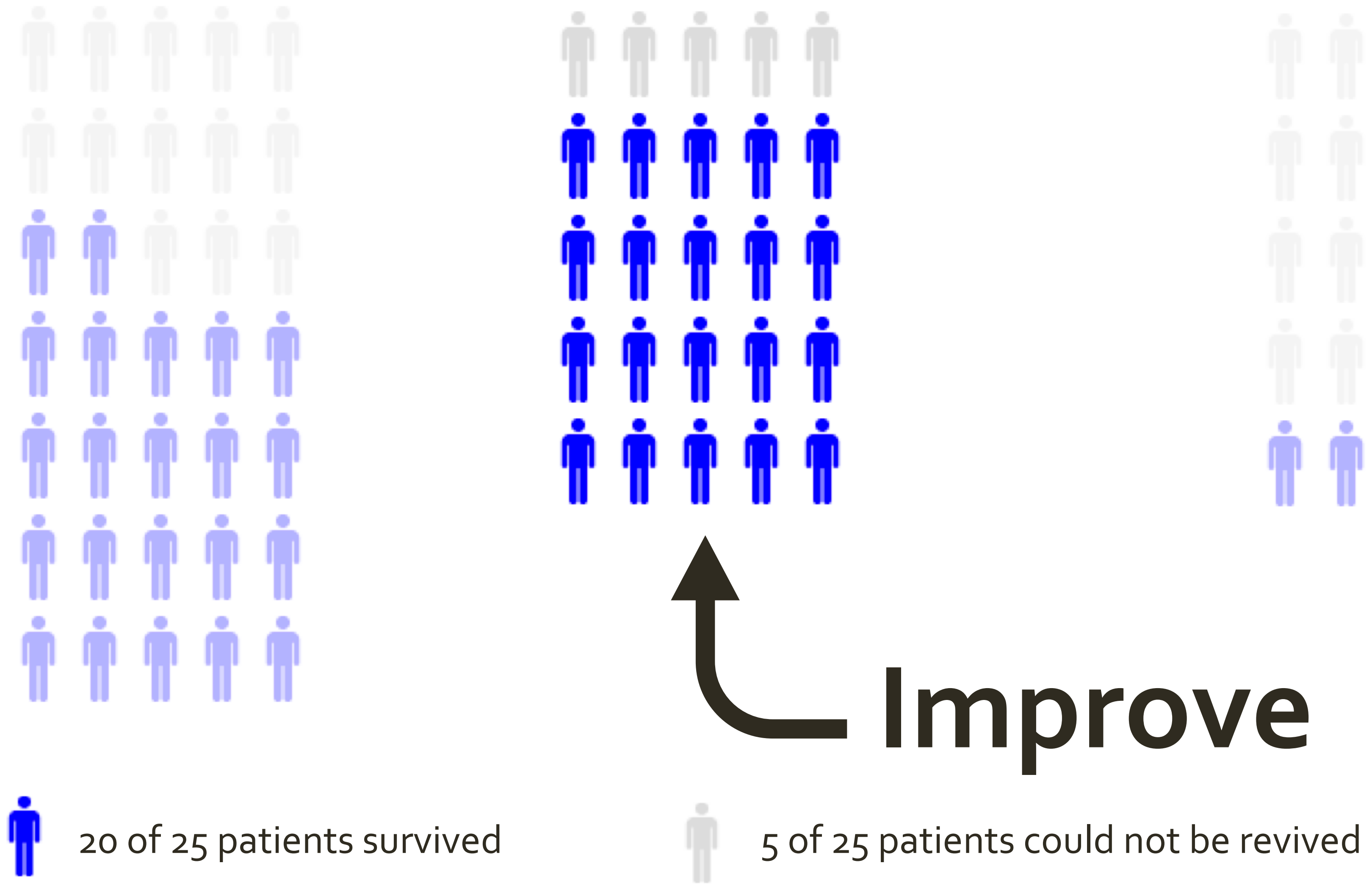




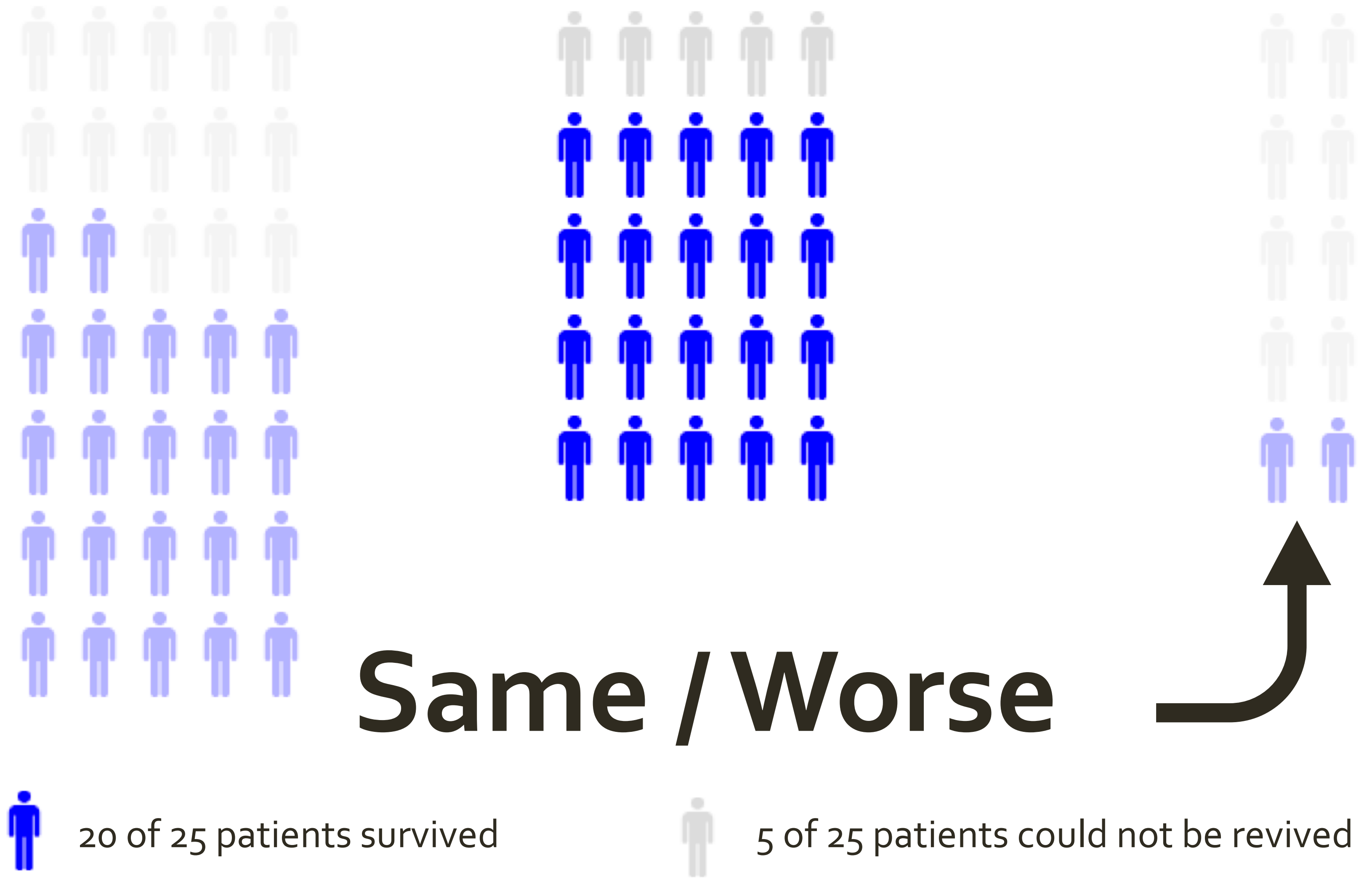
22 of 35 patients survived

13 of 35 patients could not be revived

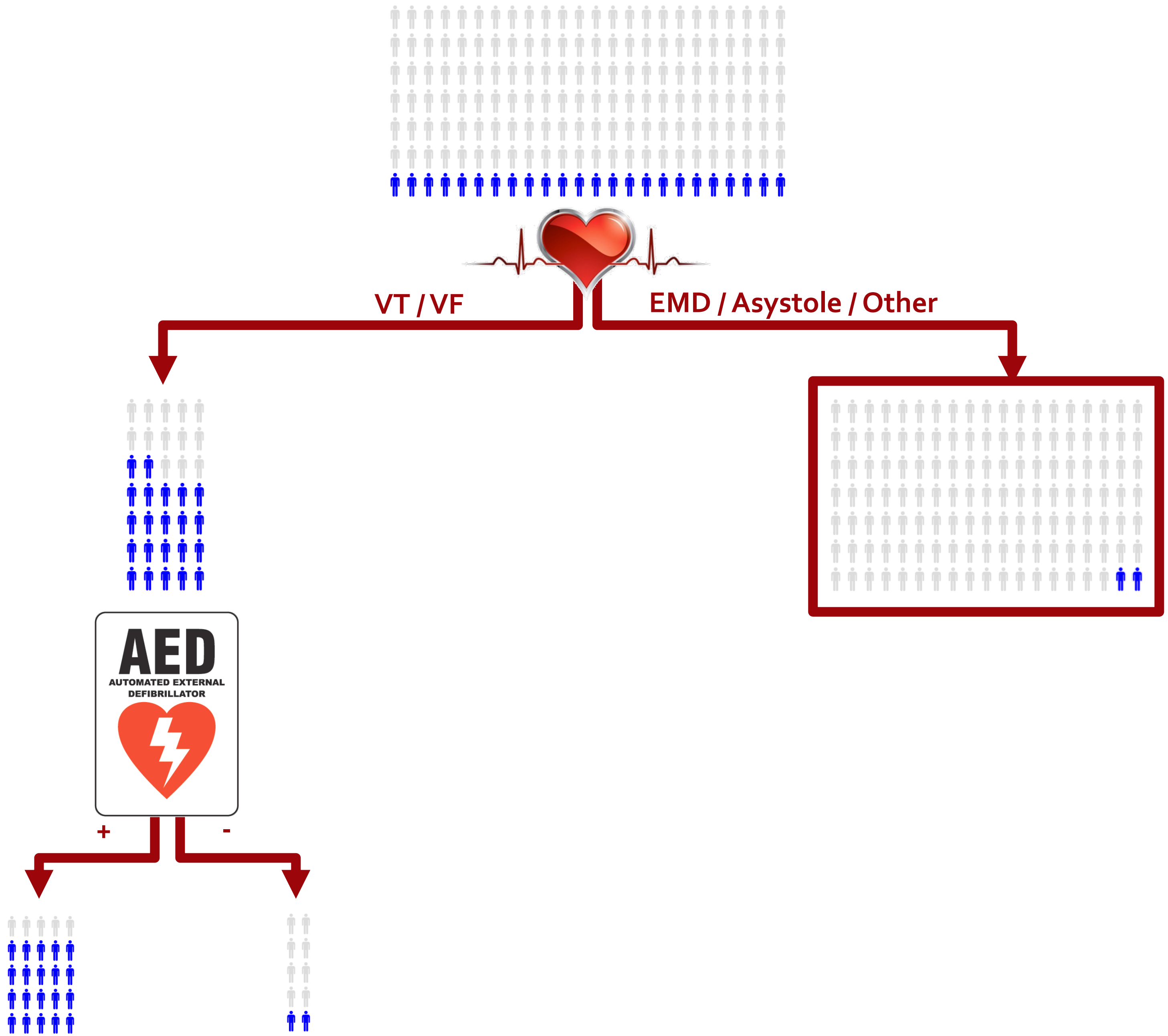
Next split: response to defibrillation



Next split: response to defibrillation

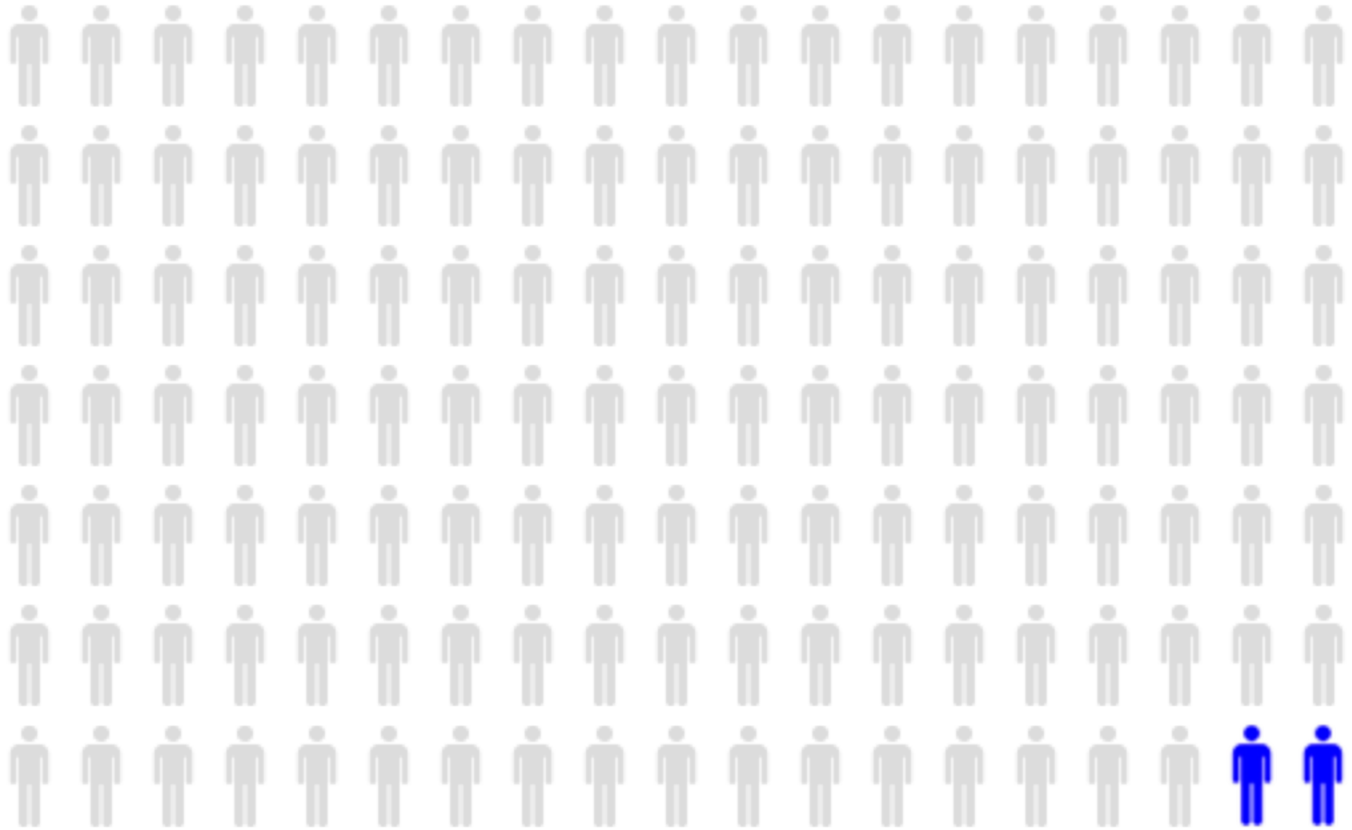


Partition view



Next split: response to ~~defibrillation~~ medication

Next split: response to defibrillation

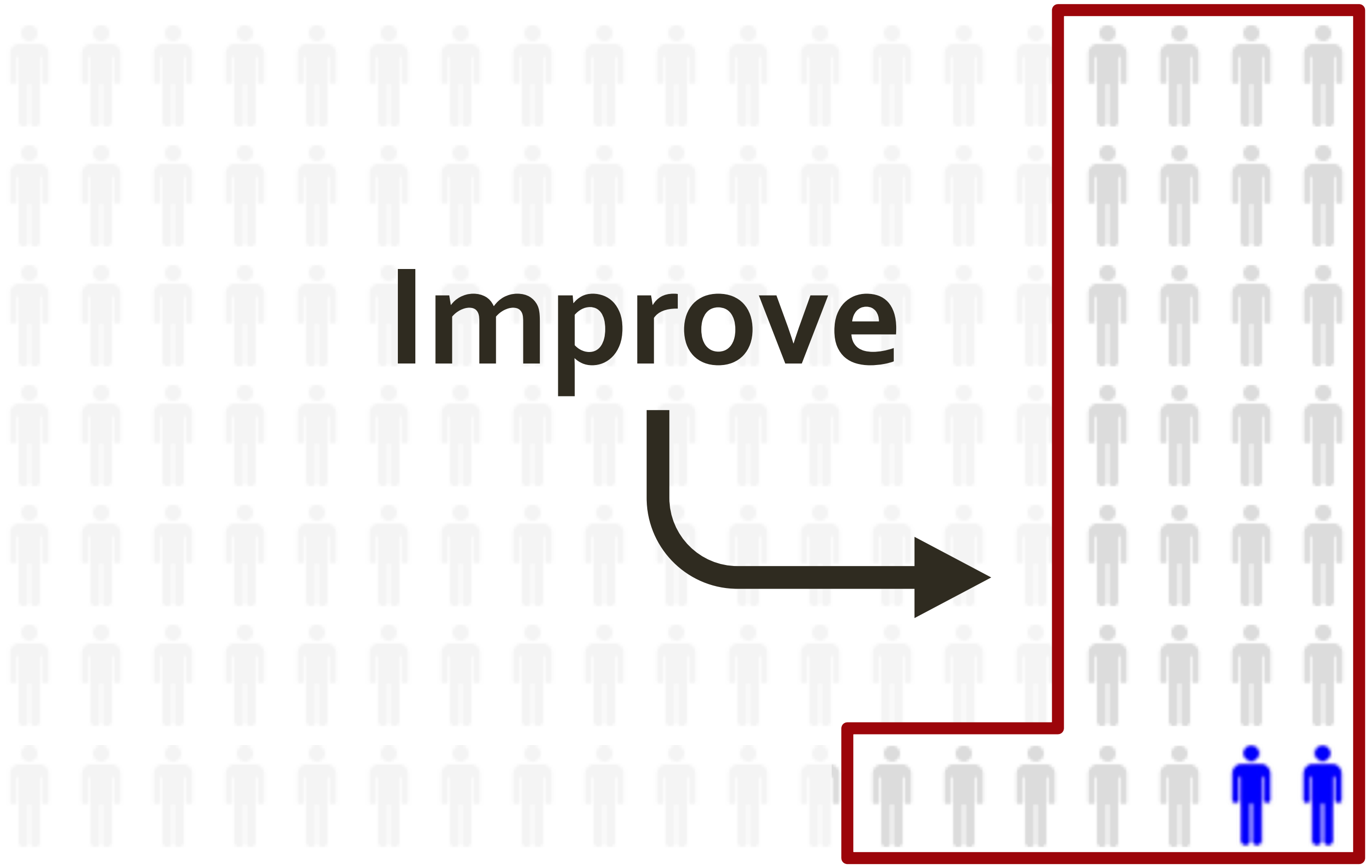


2 of 133 patients survived

131 of 133 patients could not be revived

Next split: response to ~~defibrillation~~ medication

Next split: response to defibrillation

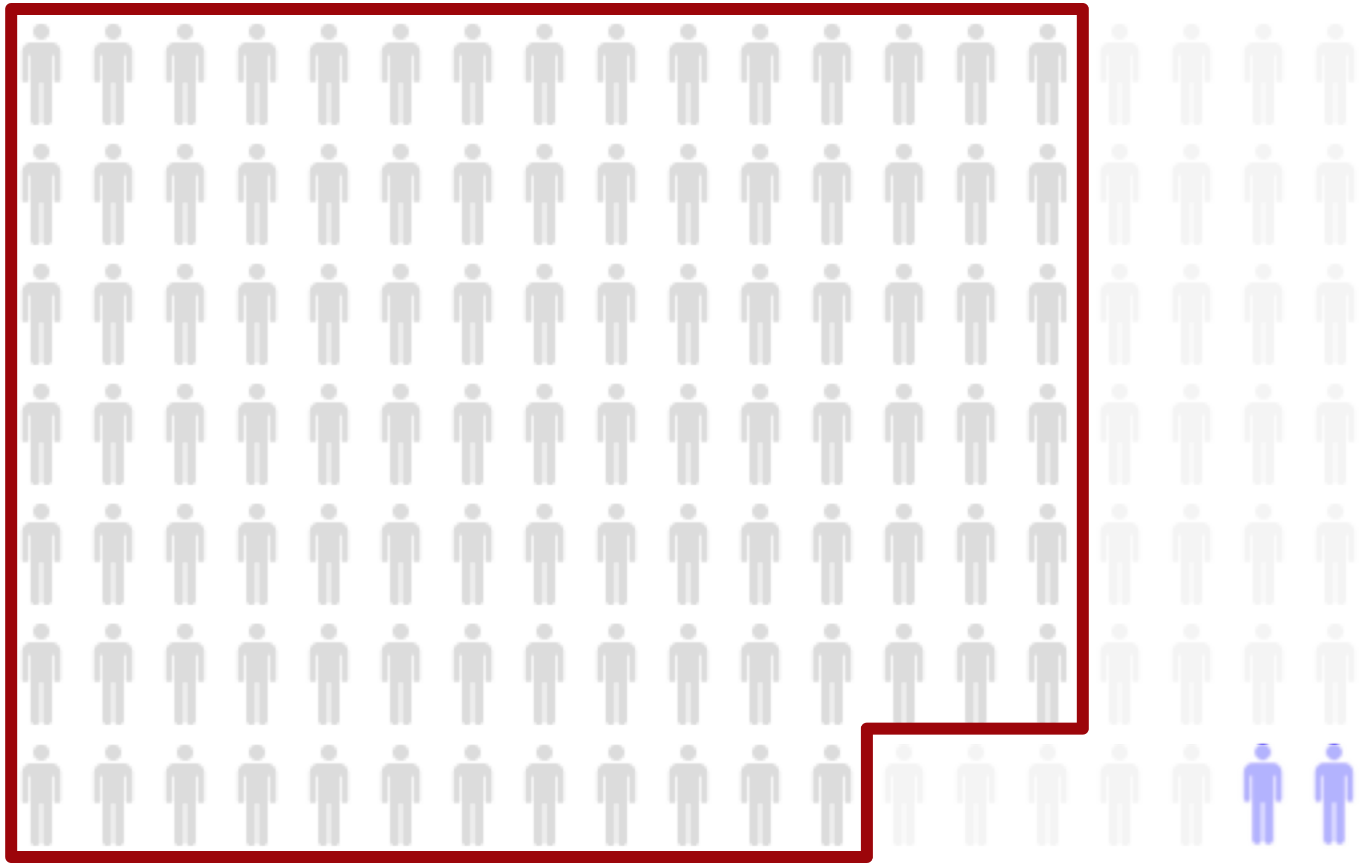


2 of 31 patients survived

29 of 31 patients could not be revived

Next split: response to ~~defibrillation~~ medication

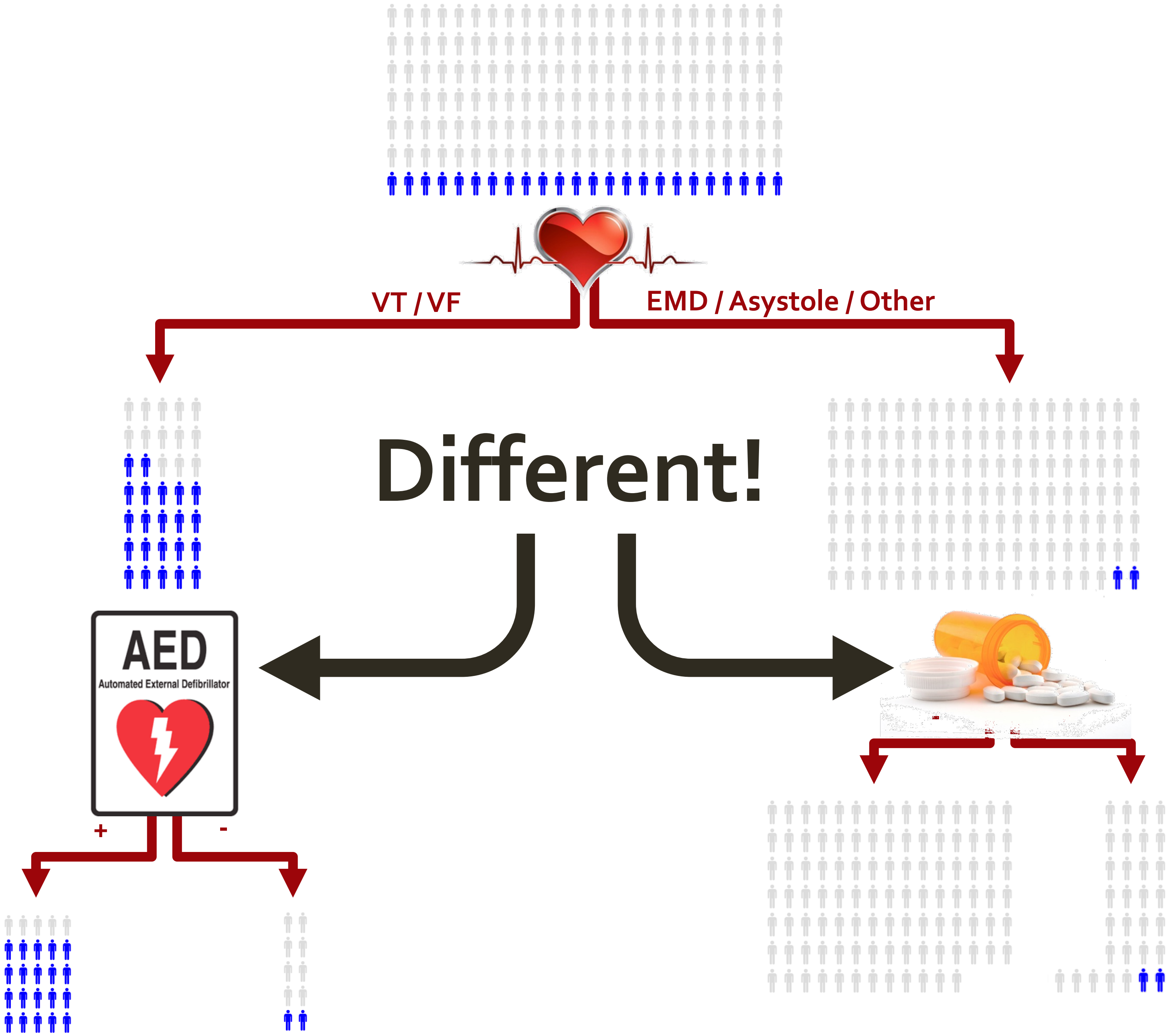
Next split: response to defibrillation



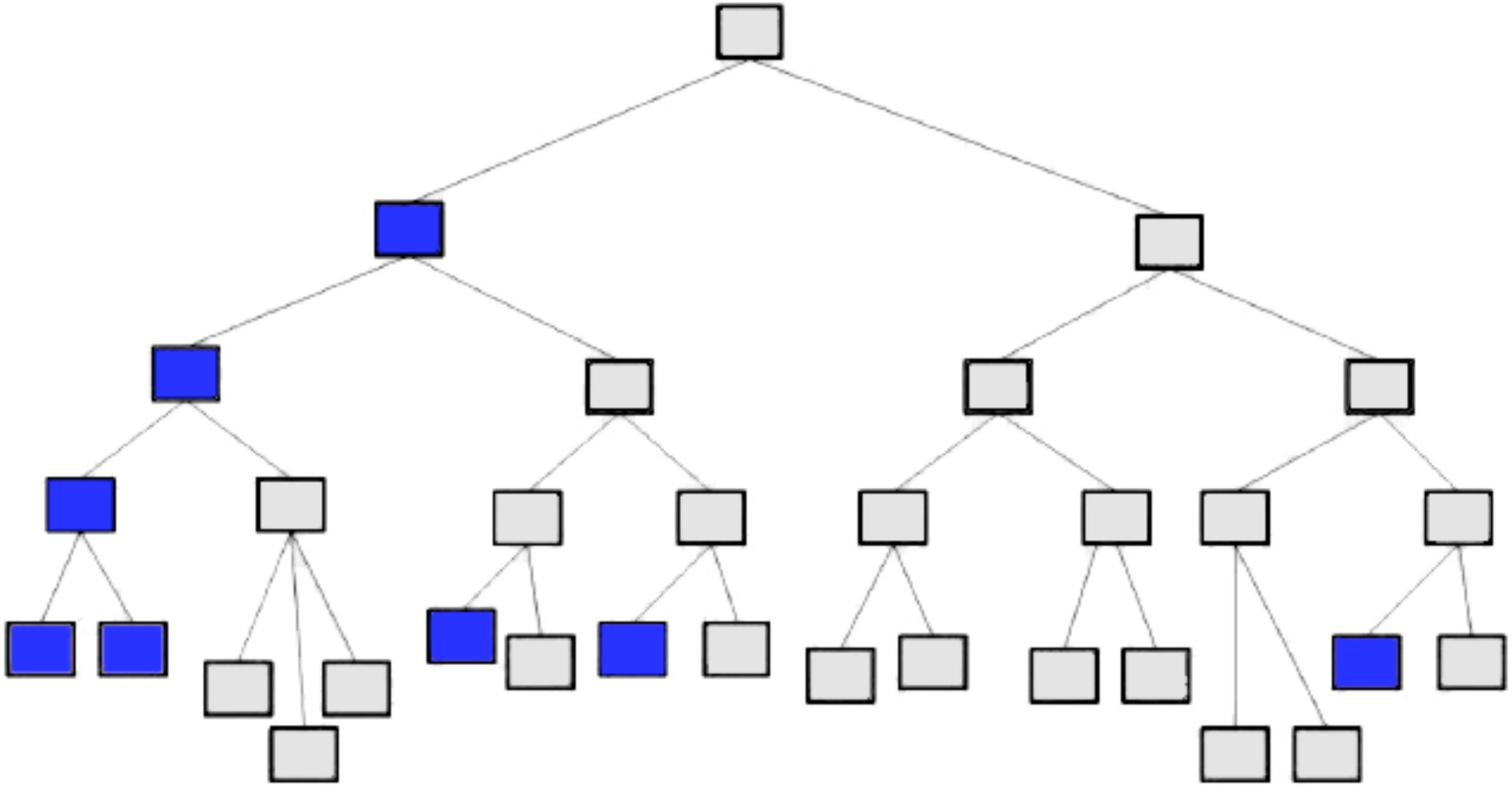
0 of 102 patients survived

102 of 102 patients could not be revived

Partition view



Continue…



Discussion

What could go wrong with this approach?

Growing (and pruning) trees

Big idea: build a big tree, then cut off (“prune”) the branches that aren’t improving performance

Growing (and pruning) trees

Big idea: build a big tree, then cut off ("prune") the branches that aren't improving performance

Why not just build a smaller tree to begin with?

Flashback: the lasso

- **Big idea**: minimize RSS plus an additional penalty that rewards small (sum of) **coefficient values**

$$\overbrace{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)}^{\text{RSS}} + \underbrace{\lambda}_{\text{Tuning parameter}} \overbrace{\sum_{j=1}^{p}\beta_j^2}^{\text{Shrinkage penalty}}$$

Cost complexity pruning

- **Big idea**: minimize RSS plus an additional penalty that rewards small **trees**

$$\overbrace{\sum_{m=1}^{|T|} \sum_{i: x_i \in R_m} \left(y_i - \hat{y}_{R_m}\right)^2}^{\text{RSS}} + \overbrace{\alpha |T|}^{\text{Shrinkage penalty}}$$

Loop over all terminal nodes

Then consider only the relevant predictors

Difference b/t true & predicted values

Rewards trees with fewer terminal nodes

Cost complexity pruning

- **Big idea**: minimize RSS plus an additional penalty that rewards small **trees**

$$\underbrace{\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} \left(y_i - \hat{y}_{R_m}\right)^2}_{\text{RSS}} + \underbrace{\boxed{\alpha}|T|}_{\text{Shrinkage penalty}}$$

- **Fun fact:** as we increase α, branches get pruned in a nice, predictable (<u>nested</u>) fashion (why is this useful?)

Tree variation of backward selection

Start by growing some big tree on the training data

1. Use cost complexity pruning to get a sequence of "best subtrees" (as a function of α)
2. Select a single "best" α using cross-validated prediction error or something similar
3. Return the associated tree

Discussion

- The minimization we just saw would help us find the best **regression** tree, but our example was about **classification**
- **Question**: what needs to change?

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} \boxed{\left(y_i - \hat{y}_{R_m}\right)^2} + \alpha|T|$$

- **Answer**: just like in previous classification settings, we can't use RSS

Trouble in paradise…

- Usual approach (minimizing classification error) isn't sensitive enough to **build** good trees
- Alternative 1: *Gini index of each node*

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

- Alternative 2: *cross-entropy of each node*

$$D = \sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

- Both are measures of **purity***

*small values → node contains mostly observations from the same class

Discussion

What advantages / disadvantages might decision trees have when compared to other methods?

Activity

Use the Hitters dataset (in the ISLP package)

Consider these variables:

- Years, Hits, RBI

We will use these three variables to create a decision tree that predicts Salary.

Start by dropping any players with NA salary.

Then, decide which variable to use for your first split, and what value of that variable to split around.

Then repeat until you have 4-6 terminal nodes.

Draw your decision tree.