# Outline of this talk

* Today I’m going to talk about how my R package, vtree, can be used to discover hidden patterns in clinical data.
* In the first part of my talk, I’ll introduce the basic functionality of vtree by showing *step by step* how to draw a CONSORT diagram.
* Then, in the second part of the talk, I’ll show how vtree can be used for data exploration.

# Part 1: Drawing a CONSORT flow diagram

* RCTs are often considered to be gold standard of clinical evidence.
* To live up to this to this billing, an RCT has to be done *well* and be transparently reported.
* A major step forward was the Consolidated Standards of Reporting Trials (the CONSORT statement), originally published in 1996, a key element of which is the CONSORT diagram.
* Some other tools are available for drawing CONSORT diagrams, but vtree is unique in two ways:
  1. it can be used for a much more general class of tasks (as we’ll see in part 2 of this talk) and,
  2. it is reproducible, so it can be easily updated and reproduced

Does treatment of COVID-19 with paxlovid reduce hospitalization or death?

* As an example, I’m going to use a randomized trial that was recently published in the New England Journal of Medicine.
* Paxlovid is an oral antiviral
* This is the drug that President Biden was recently treated with.

# [CONSORT diagram for paxlovid study]

* As I was preparing this talk, I noticed an apparent error in the published CONSORT diagram:
* According to the diagram, 2396 patients were assessed for eligibility and then 137 were excluded (which would leave **2259**) but the diagram indicates that only **2246** were randomized
* I will assume 2396 is a misprint, and that in fact 2383 were assessed for eligibility
* This is a good illustration of the risk of error in preparing a CONSORT diagram and the importance of reproducibility

# The paxlovid CONSORT diagram

# A single layer tree

* vtree stands for “variable tree” because it uses a simple tree model
* Each variable in the tree generates a new layer.

# A two-layer tree

* This two-layer tree shows the general structure of variable trees: the nodes represent nested subsets of the data frame.
* This tree includes a missing value node for **group**, because for those patients who were not randomized, there is of course no group.
* By default vtree shows missing values.
* Other tools sometimes hide missing values.
* (In some cases there is more missing data than non-missing data!)

# Pruning a node

* vtree has several different ways to specify which nodes to prune.
* Here we’re specifying that only the randomized patients should be followed.
* With each additional variable, the tree gets larger and larger. One way to keep a tree manageable is by pruning.

# A three-layer tree

# Adding labels

# Summary text + some finishing touches

# Part 2: Data Exploration

* I originally designed vtree to draw inclusion-exclusion flow diagrams like the CONSORT diagram.
* These diagrams are very useful for all kinds of study designs.
* But vtree's generality makes it good for lots of other data exploration tasks.
* In my experience people often have data sets but don’t really know much about the data itself.
* I think this is partly because, beyond just visually examining a spreadsheet of data, people often don’t have easy-to-use tools to explore their data
* vtree makes it easy to find patterns in data that might otherwise be overlooked.

# A retrospective cohort study

# Frequencies

# For convenience, I'll relabel some variables

# Patterns

* An alternative to an ordinary variable tree is what I call a pattern tree (but maybe it should be called a jelly bean tree)
* It displays a separate row for each combination of values.
* For convenience, pattern trees are arranged so that the most frequent combinations appear at the bottom.
* This can also be used to examine patterns of missing values.

# CMV reactivation (R) post-transplant

# Quantitative variables

* Finally, as flexible as it is, vtree is not suited to all types of data exploration.
* Vtree focuses on discrete variables.
* For quantitative variables there are many tools that are better suited to the task.
* Nevertheless, vtree does have some support for quantitative variables.

# Dichotomizing a quantitative variable

* Vtree has lots more features and all kinds of uses.
* In some situations it provides information that is similar to existing tools, like contingency tables or bar plots
* But in other situations, it provides useful information that would be very difficult to obtain in other ways
* And it does this using a simple model.