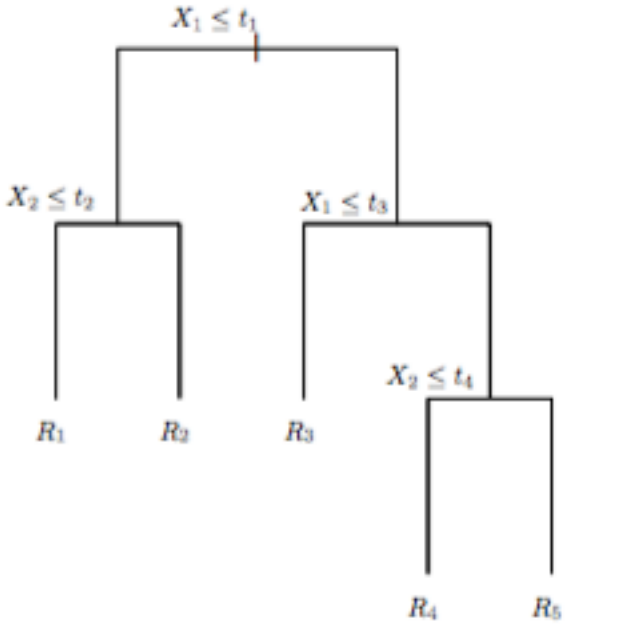
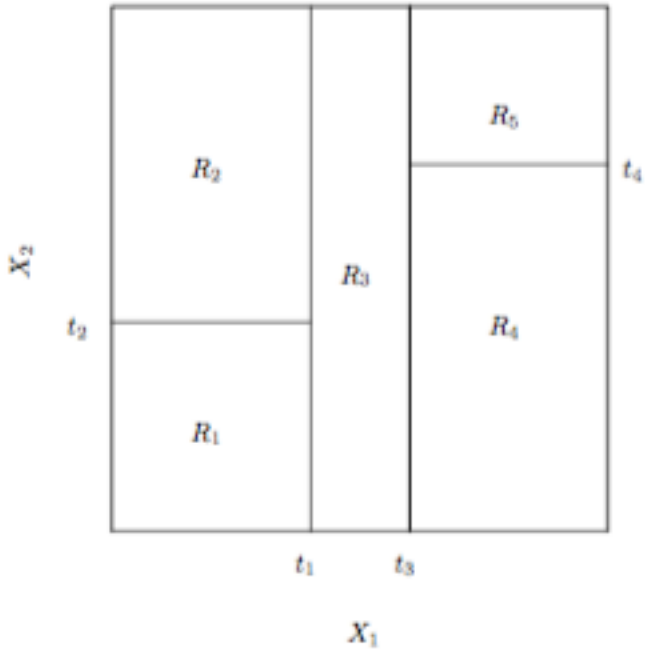
# Regression Trees

- Same as classification trees except we predict a numeric value
- Slight complication with the entropy calculation for the splits

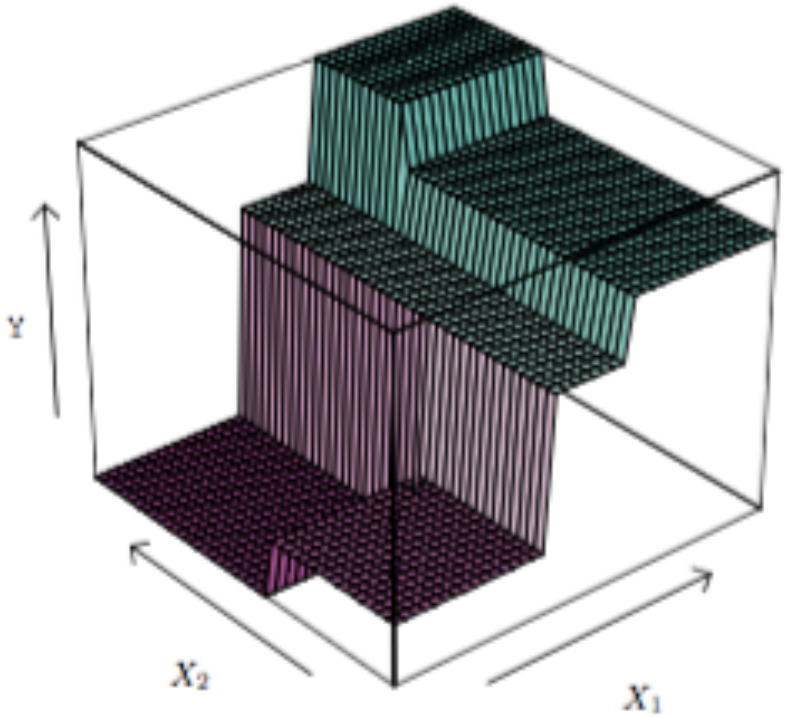
# Regression Trees

- Consider a data set with X1 and X2 as independent variables and Y as dependent variable with values R1, R2, R3, R4, and R5.
- The right figure represents a partitioning of the space according to the induced tree.



# Regression Trees



# Regression Trees

The basic regression-tree-growing algorithm then is as follows:

1. Start with a single node containing all points. Calculate $m_c$ and $S$.
2. If all the points in the node have the same value for all the independent variables, stop. Otherwise, search over all binary splits of all variables for the one which will reduce $S$ as much as possible. If the largest decrease in $S$ would be less than some threshold $\delta$, or one of the resulting nodes would contain less than $q$ points, stop. Otherwise, take that split, creating two new nodes.
3. In each new node, go back to step 1.

with $$S = \sum_{c \in \text{leaves}(T)} \sum_{i \in C} (y_i - m_c)^2$$

where $m_c = \frac{1}{n_c} \sum_{i \in C} y_i$, the prediction for leaf $c$.

# Regression Trees

- Note: We do not use entropy but minimize the sum squared error at the leaves!

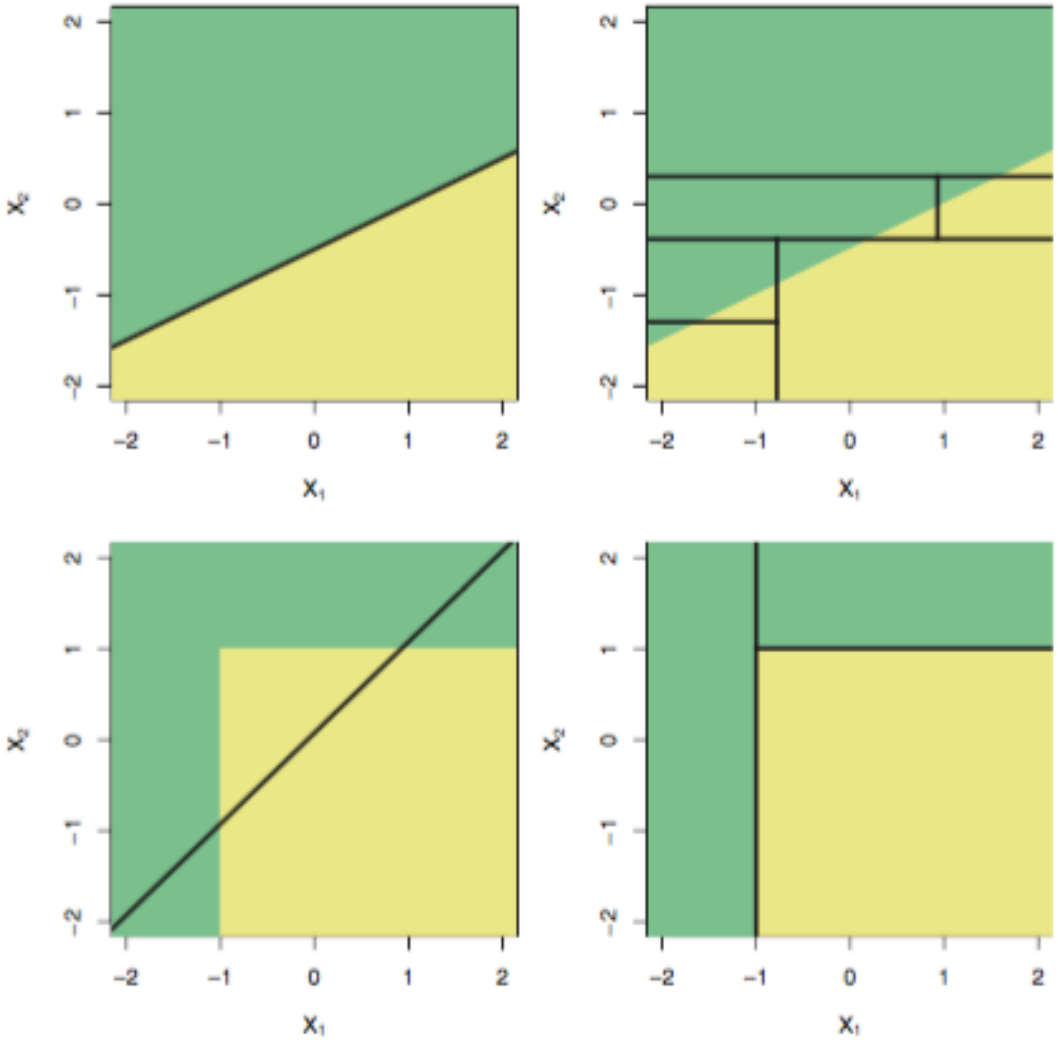



**FIGURE 8.7.** Top Row: *A two-dimensional classification example in which the true decision boundary is linear, and is indicated by the shaded regions. A classical approach that assumes a linear boundary (left) will outperform a decision tree that performs splits parallel to the axes (right).* Bottom Row: *Here the true decision boundary is non-linear. Here a linear model is unable to capture the true decision boundary (left), whereas a decision tree is successful (right).*