Solution 1:

1) The conclusion is incorrect. Since the tree structure is built recursively, the algorithm does not necessarily identify the optimal tree with lowest empirical risk on the training data. This lies in the nature of greedy optimization procedures. Empirical risk minimization (ERM) is only performed to identify *the next* splitting rule, and not entire sets of subsequent splitting rules.

2) CART does automatically select features for splitting nodes if they lead to an expected reduction in empirical risk. Irrelevant features are therefore more likely to be picked less often for split rules in model construction. (Of course, the subject of assessing feature importance is left for the chapter on random forests.) However, one could gain a rough understanding of a feature's relevance by looking at how often it was picked for splitting a node. However, this kind of "split rule selection frequency" does not necessarily relate to a feature variable's contribution to ERM.

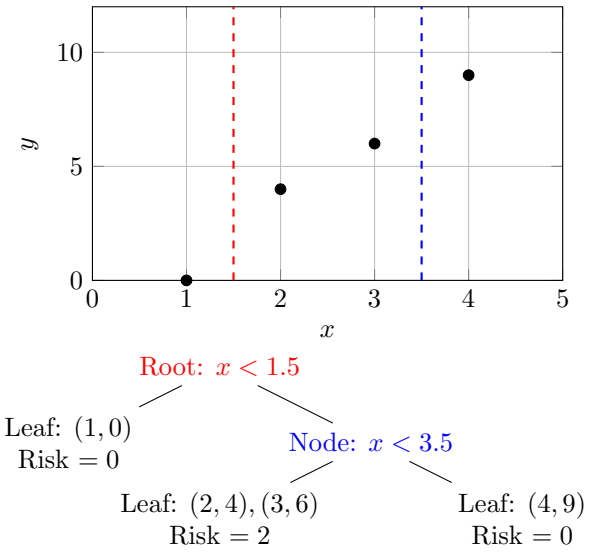
3) CART can perform automatic feature selection by remembering surrogate splits in an extra step in model construction. Per default, the `rpart` package retains up to 5 surrogate splits. For each split rule, a surrogate split rule that leads to sorting observations into child nodes in a similar way is retained. These surrogate splits can then be used to "guide" observations through the tree even if they have some missing feature values. Therefore, CART is generally-speaking well-suited to handle missing observations.

4) The number of possible split points evaluated per feature variable is equal to the number of different values the respective feature has in the training data minus 1, e.g., a numerical or categorical variable with 4 different values in the training data has 3 potential split points. (Actually, for a continuous feature there is an infinite amount of possible split points, but there are just 3 which lead to different results for the training data.) As each feature variable can be used for the split point, one needs to sum over the feature variables in the data set.

$$\text{Number of possible split points} = \sum_{j=1}^{p} (\text{number of different values in the training data}_j - 1) \leq 3 \cdot (n-1)$$

Solution 2:

(a) **Scatter plot and perfect 2-split tree:**



- Each leaf contains:
  - Left: obs 1 ($Y = 0$), pure, risk $= 0$
  - Middle: obs 2, 3 ($Y = 4, 6$), mean $= 5$, risk $= (4-5)^2 + (6-5)^2 = 1 + 1 = 2$
  - Right: obs 4 ($Y = 9$), pure, risk $= 0$
- **Total risk** $= 0 + 2 + 0 = 2$ (very low since leaves are nearly pure)
- **Caution:** Such a perfect tree will *overfit* (just memorizes training data). Real-world trees need stopping rules (max depth, min samples per leaf, etc.) to avoid this.
- **Take-away:** Only *two* splits are allowed, so not all leaves can be pure. The best achievable tree has three leaves: two pure ($\{1\}, \{4\}$), one with two samples ($\{2, 3\}$), with total risk 2.

(b) **CART's greedy search on Data A:**

(i) **Candidate split points for X_1:** mid-points of $\{1, 2, 3, 4\}$ are $\{1.5, 2.5, 3.5\}$.

(ii) **Computing SSE for each split:**

Split	Left: obs, Y	Right: obs, Y	$\bar{y}_L$	$\bar{y}_R$	$\text{SSE}_{\text{split}}$
$X_1 < 1.5$	$\{1\}, Y = \{0\}$	$\{2, 3, 4\}, Y = \{4, 6, 9\}$	0	$19/3 \approx 6.33$	12.67
$X_1 < 2.5$	$\{1, 2\}, Y = \{0, 4\}$	$\{3, 4\}, Y = \{6, 9\}$	2	7.5	12.5
$X_1 < 3.5$	$\{1, 2, 3\}, Y = \{0, 4, 6\}$	$\{4\}, Y = \{9\}$	$10/3 \approx 3.33$	9	18.67

Detailed calculations:

- **Split at** 1.5:
  Left risk $= (0-0)^2 = 0$;
  Right risk $= (4-6.33)^2 + (6-6.33)^2 + (9-6.33)^2 = 38/3$;
  Total $= 38/3 \approx 12.67$.
- **Split at** 2.5: Left risk $= (0-2)^2 + (4-2)^2 = 8$;
  Right risk $= (6-7.5)^2 + (9-7.5)^2 = 4.5$;
  Total $= 12.5 \Rightarrow$ **Best split:** $X_1 < 2.5$ (lowest SSE).
- **Split at** 3.5:
  Left risk $= (0-3.33)^2 + (4-3.33)^2 + (6-3.33)^2 = 56/3$;
  Right risk $= (9-9)^2 = 0$;
  Total $= 56/3 \approx 18.67$.

2nd split after first split at $X_1 < 2.5$:

Child	Observations	Mean	Risk (SSE)
Left	$\{1,2\}, Y = \{0,4\}$	2	$(0-2)^2 + (4-2)^2 = 8$
Right	$\{3,4\}, Y = \{6,9\}$	7.5	$(6-7.5)^2 + (9-7.5)^2 = 4.5$

Second split options:

Option	Leaves	Leaf risks	Total risk
Split left at $X_1 < 1.5$	$\{1\}(\bar{y}=0), \{2\}(\bar{y}=4), \{3,4\}(\bar{y}=7.5)$	$0, 0, 4.5$	4.5
Split right at $X_1 < 3.5$	$\{1,2\}(\bar{y}=2), \{3\}(\bar{y}=6), \{4\}(\bar{y}=9)$	$8, 0, 0$	8

Best second split: Split the left child ($\{1,2\}$) at $X_1 < 1.5$, yielding leaves $\{1\}$, $\{2\}$, $\{3,4\}$, with total risk = 4.5.

Final greedy 2-split tree: Root split at $X_1 < 2.5$; left child split at $X_1 < 1.5$. Leaves: $\{1\}$ (risk 0), $\{2\}$ (risk 0), $\{3,4\}$ (risk 4.5). Total risk = 4.5.

(c) **Comparison:** The greedy 2-split tree (total risk = 4.5) is *not* the same as the optimal 2-split tree (total risk = 2). The greedy approach picks $X_1 < 2.5$ as the first split, which leads to a suboptimal final tree. The optimal tree uses splits at $X_1 < 1.5$ and $X_1 < 3.5$, achieving lower risk by creating leaves $\{1\}$, $\{2,3\}$, $\{4\}$ instead of $\{1\}$, $\{2\}$, $\{3,4\}$.

Solution 3:

(a) **Surrogate split for primary split $X_1 < 2.5$:**

The actual CART picks the root split $X_1 < 2.5$. To mimic this primary split with a surrogate that only uses X_2, we fit a stump where the target is whether $X_1 < 2.5$ holds.

The surrogate dataset becomes:

Obs	X_2	$X_1 < 2.5 (Y_{\text{new}})$
1	2.4	1
2	2.6	1
3	3.0	0
4	4.0	0

Now we find the best split on X_2 to predict Y_{new}. Candidate split points for X_2: $\{2.5, 2.8, 3.5\}$.

For each split on X_2, we evaluate agreement with the primary split $X_1 < 2.5$:

Split on X_2	Agreement	Notes
$X_2 < 2.5$	$3/4 = 0.75$	obs 2 misrouted
$X_2 < 2.8$	$4/4 = 1.0$	perfect
$X_2 < 3.5$	$3/4 = 0.75$	obs 3 misrouted

Best surrogate rule: $\boxed{X_2 < 2.8}$ with perfect agreement = 1.0.

(b) **Prediction with missing X_1 using surrogate + CART check:**

New observation: $X_1 = \text{NA}$, $X_2 = 2.30$.

- Primary split (actual CART): $X_1 < 2.5$. Missing X_1 triggers surrogate.
- Best surrogate: $X_2 < 2.8$. Since $2.30 < 2.8$, route to the *left* child of the root (same side as obs 1,2).
- Prediction (left child mean) is $\hat{y} \approx 2$ (mean of obs 1,2).

Surrogate mechanism (brief):

- (i) Pick primary split on available data.
- (ii) Create surrogates on other features that mimic the primary partition.
- (iii) Rank by agreement.
- (iv) At prediction, if primary feature is missing, use best available surrogate to route.