Solution 1: Splitting criteria

a) • **Pen-and-paper solution**

1) Here, we have only one split variable x. We probe all splits of x in two groups, where thresholds are placed equidistant between the observed feature values (think about why this might help generalization):
   - (1) — $(2, 7, 10, 20)$ (split point 1.5)
   - $(1, 2)$ — $(7, 10, 20)$ (split point 4.5)
   - $(1, 2, 7)$ — $(10, 20)$ (split point 8.5)
   - $(1, 2, 7, 10)$ — (20) (split point 15)
2) For each split point, compute the sum of squares ($L2$ loss) in both groups.
3) Choose the point that splits both groups best w.r.t. empirical risk reduction.

A split point t leads to the following half-spaces:

$$\mathcal{N}_1(t) = \{(x, y) \in \mathcal{N} : x \leq t\} \text{ and } \mathcal{N}_2(t) = \{(x, y) \in \mathcal{N} : x > t\}.$$

Recall the corresponding minimization problem (only w.r.t. t as there is just one variable):

$$\min_t \left(\min_{c_1} \sum_{(x,y)\in\mathcal{N}_1(t)} (y - c_1)^2 + \min_{c_2} \sum_{(x,y)\in\mathcal{N}_2(t)} (y - c_2)^2 \right).$$

The inner minimizers are the respective mean target values in each node, $\hat{c}_1 = \bar{y}_1$ and $\hat{c}_2 = \bar{y}_2$, such that:

$$\min_t \left(\sum_{(x,y)\in\mathcal{N}_1(t)} (y - \bar{y}_1)^2 + \sum_{(x,y)\in\mathcal{N}_2(t)} (y - \bar{y}_2)^2 \right),$$

so the MSE of the parent is:

$$\rho_{\text{MSE}}(\mathcal{N}) = \frac{1}{|\mathcal{N}|} \sum_{(x,y)\in\mathcal{N}} (y - \bar{y})^2 = \frac{1}{5} \sum_{i=1}^{5} (y_i - 4.7)^2 = 22.56.$$

Calculate the risk $\mathcal{R}(\mathcal{N}, j, t)$ for each split point:

- $x \leq 1.5$

$$\begin{aligned}\mathcal{R}(\mathcal{N}, 1, 1.5) &= \tfrac{|\mathcal{N}_1|}{|\mathcal{N}|}\rho_{\text{MSE}}(\mathcal{N}_1) + \tfrac{|\mathcal{N}_2|}{|\mathcal{N}|}\rho_{\text{MSE}}(\mathcal{N}_2) = \\ &= \tfrac{1}{5} \cdot \left(\tfrac{1}{1}(1-1)^2\right) + \tfrac{4}{5} \cdot \left(\tfrac{1}{4}((1-5.625)^2 + (0.5-5.625)^2 + (10-5.625)^2 + (11-5.625)^2)\right) \\ &= 19.14\end{aligned}$$

- $x \leq 4.5 \implies \mathcal{R}(\mathcal{N}, 1, 4.5) = 13.43$
- $x \leq 8.5 \implies \mathcal{R}(\mathcal{N}, 1, 8.5) = 0.13$ **optimal**
- $x \leq 15 \implies \mathcal{R}(\mathcal{N}, 1, 15) = 12.64$

Proceeding accordingly for the monotonic, rank-preserving log transformation yields the same result:

- $\log x \leq 0.3 \implies \mathcal{R}(1, 0.3) = 19.14$
- $\log x \leq 1.3 \implies \mathcal{R}(1, 1.3) = 13.43$

  - $\log x \leq 2.1 \implies \mathcal{R}(1, 2.1) = 0.13$ **optimal**
  - $\log x \leq 2.6 \implies \mathcal{R}(1, 2.6) = 12.64$

- **R solution**

```r
x <- c(1, 2, 7, 10, 20)
y <- c(1, 1, 0.5, 10, 11)

compute_mse <- function (y) mean((y - mean(y))**2)

compute_total_mse <- function (yleft, yright) {
  num_left <- length(yleft)
  num_right <- length(yright)
  w_mse_left <- num_left / (num_left + num_right) * compute_mse(yleft)
  w_mse_right <- num_right / (num_left + num_right) * compute_mse(yright)
  w_mse_left + w_mse_right
}

split <- function(x, y) {

  # try out all unique points as potential split points and ...
  unique_sorted_x <- sort(unique(x))
  split_points <- head(unique_sorted_x, length(unique_sorted_x) - 1) +
    0.5 * diff(unique_sorted_x)

  node_mses <- lapply(
    split_points,
    function(i) {
      y_left <- y[x <= i]
      y_right <- y[x > i]
      # ... compute SS in both groups
      mse_split <- compute_total_mse(y_left, y_right)
      print(sprintf("split at %.1f: empirical risk = %.2f", i, mse_split))
      mse_split
  })

  # select the split point yielding the maximum impurity reduction
  split_points[which.min(node_mses)]

}

split(x, y) # 3rd obs is best split point
```

```
## [1] "split at 1.5: empirical risk = 19.14"
## [1] "split at 4.5: empirical risk = 13.43"
## [1] "split at 8.5: empirical risk = 0.13"
## [1] "split at 15.0: empirical risk = 12.64"
## [1] 8.5
```

```r
split(log(x), y) # again, 3rd obs wins
```

```
## [1] "split at 0.3: empirical risk = 19.14"
## [1] "split at 1.3: empirical risk = 13.43"
## [1] "split at 2.1: empirical risk = 0.13"
## [1] "split at 2.6: empirical risk = 12.64"
## [1] 2.124248
```

b) For regression trees, we usually identify *impurity* with *variance*. Here is why:

- It is reasonable to define impurity via the deviation between actual target values and the predicted constant – either using absolute or square distances to enforce symmetry of positive and negative residuals.
- Recall the constant $L2$ risk minimizer for a node $\mathcal{N}$:

$$\bar{y} = \arg\min_c \sum_{(x,y)\in\mathcal{N}} (y-c)^2,$$

because

$$\begin{aligned}
\min_c \sum_{(x,y)\in\mathcal{N}} (y-c)^2 &\iff \frac{\partial}{\partial c}\left(\sum_{(x,y)\in\mathcal{N}} (y-c)^2\right) = 0 \\
&\iff \frac{\partial}{\partial c}\left(\sum_{i=1}^{|\mathcal{N}|}\left({y^{(i)}}^2 - 2y^{(i)}c + c^2\right)\right) = 0 \\
&\iff \left(\sum_{i=1}^{|\mathcal{N}|}\left(-2y^{(i)} + 2c\right)\right) = 0 \\
&\iff |\mathcal{N}|\cdot c = \sum_{i=1}^{|\mathcal{N}|} y^{(i)} \\
&\implies \hat{c} = \frac{1}{|\mathcal{N}|}\sum_{i=1}^{|\mathcal{N}|} y^{(i)} = \bar{y}.
\end{aligned}$$

- Consequently, we also have

$$\bar{y} = \arg\min_c \frac{1}{|\mathcal{N}|}\sum_{(x,y)\in\mathcal{N}} (y-c)^2,$$

where the right hand side is the (biased) sample variance for sample mean c.

- Therefore, predicting the sample mean both minimizes risk under $L2$ loss and variance impurity.
- Since constant mean prediction is equivalent to an intercept LM (minimizing the sum of squared residuals!), regression trees with $L2$ loss perform piecewise constant linear regression.
- The same correspondence holds between impurity via absolute distances and $L1$ regression.

Solution 2: Impurity reduction

The target class proportion $\pi_k^{(\mathcal{N})}$ of class $k \in \mathcal{Y}$ in a node can be computed as

$$\pi_k^{(\mathcal{N})} = \frac{1}{|\mathcal{N}|}\sum_{(x^{(i)},y^{(i)})\in\mathcal{N}} [y^{(i)} = k].$$

We can define $n \in \mathbb{N}$ i.i.d. RV $Y^{(1)}, \ldots, Y^{(n)}$ that are distributed like the training data via the categorical distribution induced by the above class frequencies:

$$\mathbb{P}(Y^{(i)} = k|\mathcal{N}) = \pi_k^{(\mathcal{N})} \quad \forall i \in \{1,\ldots,n\}, \quad k \in \mathcal{Y}.$$

By design, the corresponding estimators $\hat{Y}^{(1)}, \ldots, \hat{Y}^{(n)}$ from the randomizing rule follow the same distribution:

$$\mathbb{P}(\hat{Y}^{(i)} = k|\mathcal{N}) = \pi_k^{(\mathcal{N})} \quad \forall i \in \{1,\ldots,n\}, \quad k \in \mathcal{Y}.$$

Then, we can define the MCE for predicting in a node with $n = |\mathcal{N}|$ observations distributed like the training data:

$$\rho_{\text{MCE}}(\mathcal{N}) = \frac{1}{n}\sum_{i=1}^{n}\left[Y^{(i)} \neq \hat{Y}^{(i)}\right].$$

Taking the expectation of this MCE leads to:

$$
\begin{aligned}
\mathbb{E}_{Y^{(1)},\ldots,Y^{(n)},\hat{Y}^{(1)},\ldots,\hat{Y}^{(n)}}\left(\rho_{\text{MCE}}(\mathcal{N})\right) &= \mathbb{E}_{Y^{(1)},\ldots,Y^{(n)},\hat{Y}^{(1)},\ldots,\hat{Y}^{(n)}}\left(\frac{1}{n}\sum_{i=1}^{n}\left[Y^{(i)} \neq \hat{Y}^{(i)}\right]\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{Y^{(i)},\hat{Y}^{(i)}}\left(\left[Y^{(i)} \neq \hat{Y}^{(i)}\right]\right) \quad \text{i.i.d. assumption} \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{Y^{(i)}}\left(\mathbb{E}_{\hat{Y}^{(i)}}\left(\left[Y^{(i)} \neq \hat{Y}^{(i)}\right]\right)\right) \quad \text{Fubini's theorem} \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{Y^{(i)}}\left(\sum_{k\in\mathcal{Y}}\pi_k^{(\mathcal{N})}\cdot\left[Y^{(i)} \neq k\right]\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{Y^{(i)}}\left(\sum_{k\in\mathcal{Y}\setminus\{Y^{(i)}\}}\pi_k^{(\mathcal{N})}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{Y^{(i)}}\left(1-\pi_{k=Y^{(i)}}^{(\mathcal{N})}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{g}\pi_k^{(\mathcal{N})}\cdot\left(1-\pi_k^{(\mathcal{N})}\right) \\
&= n\cdot\frac{1}{n}\sum_{k=1}^{g}\pi_k^{(\mathcal{N})}\cdot\left(1-\pi_k^{(\mathcal{N})}\right) \\
&= \sum_{k=1}^{g}\pi_k^{(\mathcal{N})}-\sum_{k=1}^{g}\left(\pi_k^{(\mathcal{N})}\right)^2 \\
&= 1-\sum_{k=1}^{g}\left(\pi_k^{(\mathcal{N})}\right)^2.
\end{aligned}
$$

This is precisely the Gini index CART use for splitting with Brier score.

Gini impurity can thus be viewed as the frequency with which a randomly chosen sample would be misclassified if we used randomized predictions according to the categorical distribution induced by the training class frequencies.

In other words, we minimize the expected rate of misclassification among random samples from data distributed like the training data if we predict according to the observed class probabilities.