

Problem_1

Problem 1: Predicting healthcare charges, in USD, using a patient's age and BMI as features

```
library(readr)
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.3

library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v stringr    1.5.1
## v forcats    1.0.0      v tibble     3.2.1
## v lubridate  1.9.3      v tidyr      1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
  ↪ to become errors

library(rpart) # this is what we use to make the decision tree

## Warning: package 'rpart' was built under R version 4.3.3

library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.3.3

library(dplyr)

df <- read_csv("./insurance_charges.csv")

## Rows: 1338 Columns: 5
## -- Column specification -----
## Delimiter: ","
## dbf (5): age, bmi, charges, f_bmi, cardiovascular_care_cost
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

head(df)

## # A tibble: 6 x 5
##   age  bmi charges f_bmi cardiovascular_care_cost
##   <dbl> <dbl>   <dbl> <dbl>               <dbl>
## 1   19  3.90  16885.  0.591             1876.
## 2   18  5.19   1726.  0.716             2466.
```

```
## 3    28  5.00   4449.  0.699                2473.
## 4    33  3.03  21984.  0.481                1656.
## 5    32  4.09   3867.  0.612                1933.
## 6    31  3.51   3757.  0.545                1548.
```

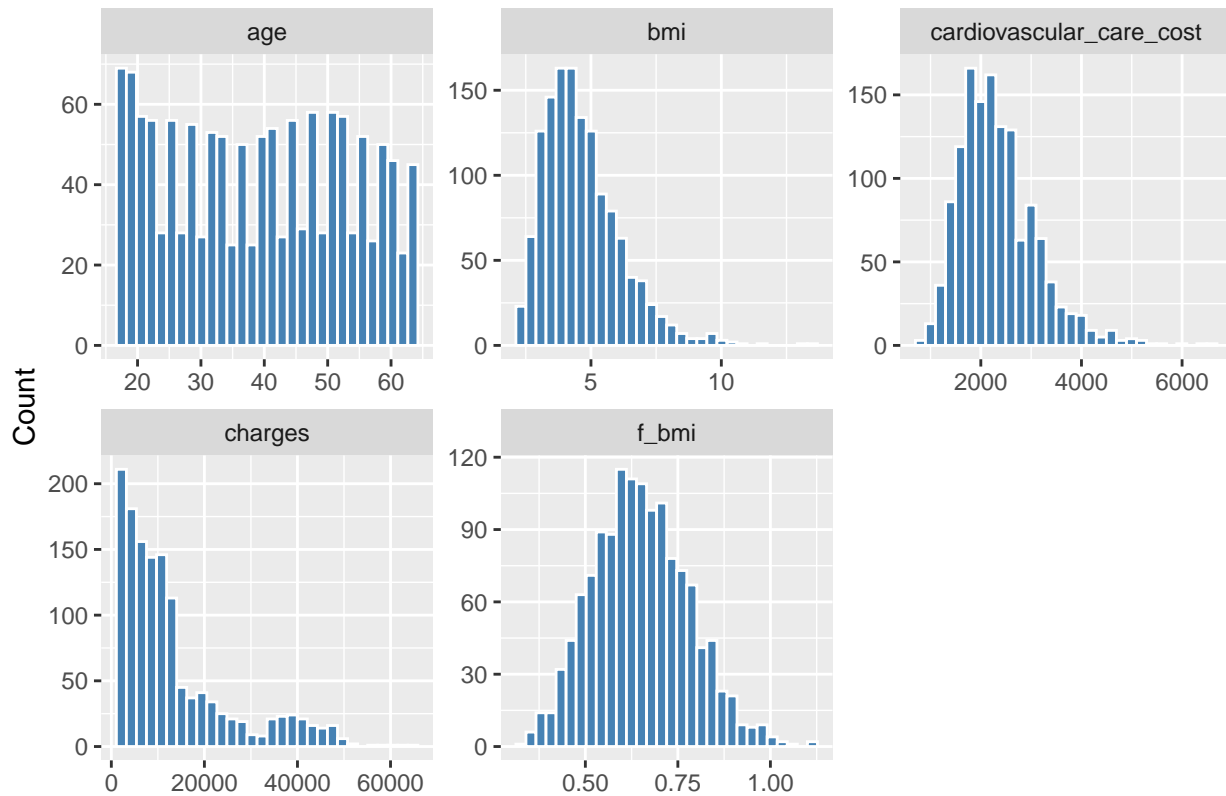
```
summary(df)
```

```
##      age          bmi          charges          f_bmi
##  Min.   :18.00   Min.    : 2.179   Min.     : 1122   Min.     :0.3382
##  1st Qu.:27.00   1st Qu.: 3.607   1st Qu.: 4740   1st Qu.:0.5572
##  Median :39.00   Median : 4.407   Median : 9382   Median :0.6442
##  Mean   :39.21   Mean    : 4.672   Mean    :13270   Mean    :0.6497
##  3rd Qu.:51.00   3rd Qu.: 5.434   3rd Qu.:16640   3rd Qu.:0.7351
##  Max.   :64.00   Max.    :13.359   Max.     :63770   Max.     :1.1258
##  cardiovascular_care_cost
##  Min.     : 827.1
##  1st Qu.:1784.1
##  Median :2204.5
##  Mean    :2338.0
##  3rd Qu.:2720.7
##  Max.     :6621.8
```

Let us start with really basic exploratory data analysis to gain some understanding on our data.

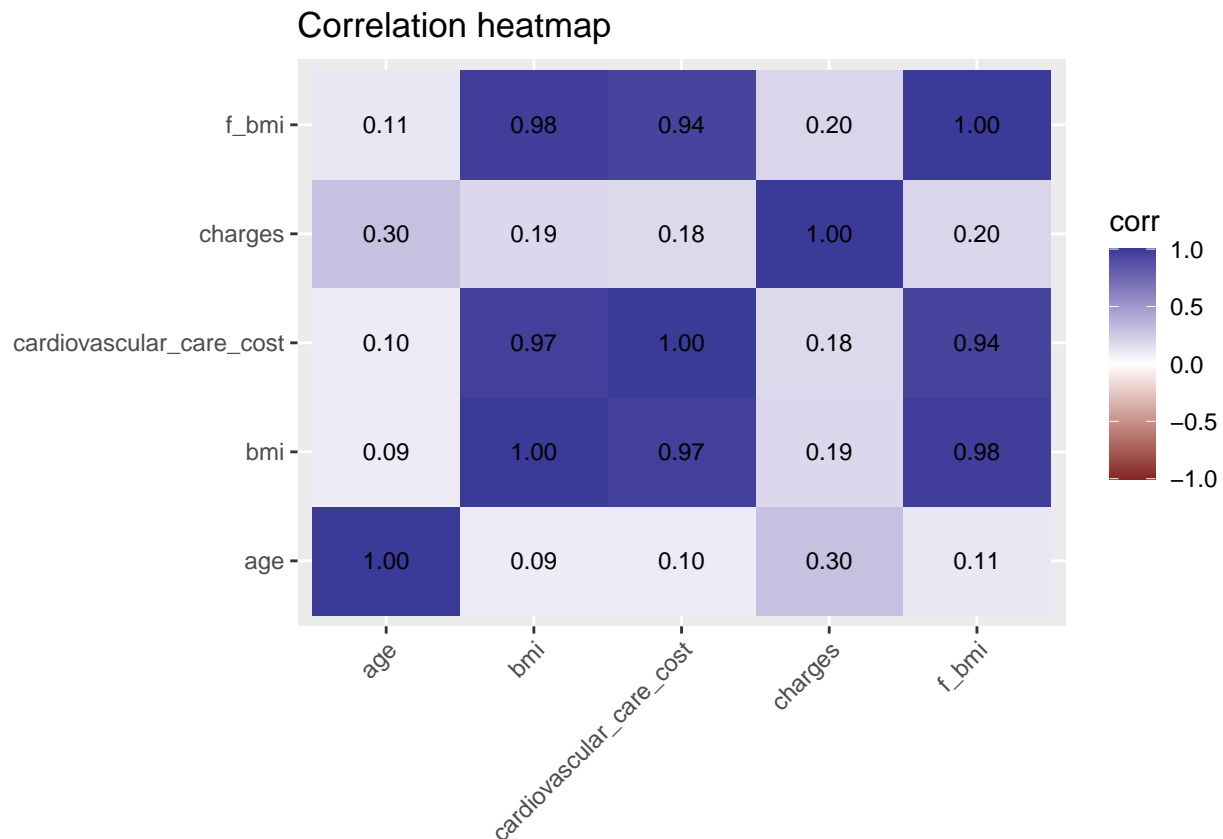
```
# Distributions of the numeric columns
df |>
  select(where(is.numeric)) |>
  pivot_longer(everything(), names_to = "var", values_to = "val") |>
  ggplot(aes(val)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "white") +
  facet_wrap(~ var, scales = "free") +
  labs(title = "Distributions of numeric variables", x = NULL, y = "Count")
```

Distributions of numeric variables



```
# Correlation matrix heatmap
corr_mat <- cor(df |> select(where(is.numeric)), use = "pairwise.complete.obs")

corr_mat |>
  as.data.frame() |>
  tibble::rownames_to_column("var1") |>
  pivot_longer(-var1, names_to = "var2", values_to = "corr") |>
  ggplot(aes(var1, var2, fill = corr)) +
  geom_tile() +
  geom_text(aes(label = sprintf("%.2f", corr)), size = 3) +
  scale_fill_gradient2(limits = c(-1, 1)) +
  labs(title = "Correlation heatmap", x = NULL, y = NULL) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

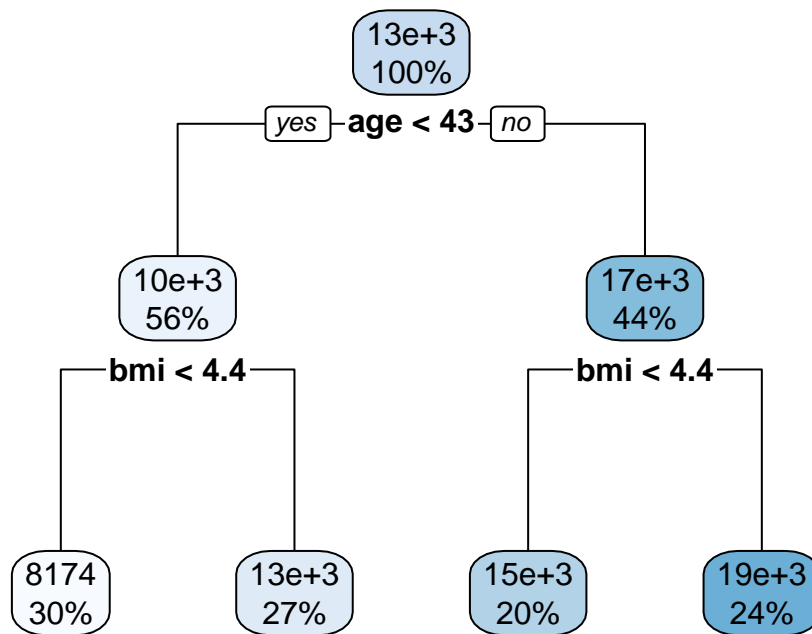


From the EDA, we can see that charges are right-skewed with a few high-cost outliers, while age is fairly uniform across the dataset. The correlation heatmap shows that BMI and f_bmi are almost perfectly correlated (0.98), while their relationship to charges is only moderate. Because CART models split on orderings, monotone transforms like f_bmi usually yield nearly identical trees to BMI.

a)

```
# Decision tree predicting charges based on bmi and age.
model1 <- rpart(charges ~ age + bmi, data = df)

# Plots the decision tree
rpart.plot(model1)
```



```
# Returns a summary of the model
summary(model1)
```

```
## Call:
## rpart(formula = charges ~ age + bmi, data = df)
##   n= 1338
##
##           CP nsplit rel error   xerror   xstd
## 1 0.07793137      0 1.0000000 1.0008480 0.05186892
## 2 0.01920458      1 0.9220686 0.9267670 0.04927575
## 3 0.01507422      2 0.9028640 0.9189620 0.04638781
## 4 0.01000000      3 0.8877898 0.9203125 0.04520415
##
## Variable importance
## age bmi
## 68 32
##
## Node number 1: 1338 observations,    complexity param=0.07793137
## mean=13270.42, MSE=1.465428e+08
## left son=2 (755 obs) right son=3 (583 obs)
## Primary splits:
##   age < 42.5      to the left,  improve=0.07793137, (0 missing)
##   bmi < 4.357944 to the left,  improve=0.04212369, (0 missing)
## Surrogate splits:
##   bmi < 5.728587 to the left,  agree=0.582, adj=0.041, (0 split)
##
## Node number 2: 755 observations,    complexity param=0.01920458
## mean=10300.81, MSE=1.313497e+08
## left son=4 (396 obs) right son=5 (359 obs)
## Primary splits:
##   bmi < 4.357944 to the left,  improve=0.03797077, (0 missing)
##   age < 26.5      to the left,  improve=0.01289901, (0 missing)
## Surrogate splits:
```

```
##      age < 18.5      to the right, agree=0.534, adj=0.019, (0 split)
##
## Node number 3: 583 observations,      complexity param=0.01507422
##   mean=17116.14, MSE=1.400084e+08
##   left son=6 (262 obs) right son=7 (321 obs)
##   Primary splits:
##     bmi < 4.392632 to the left,  improve=0.03621035, (0 missing)
##     age < 58.5      to the left,  improve=0.03075757, (0 missing)
##   Surrogate splits:
##     age < 47.5      to the left,  agree=0.566, adj=0.034, (0 split)
##
## Node number 4: 396 observations
##   mean=8174.444, MSE=5.228144e+07
##
## Node number 5: 359 observations
##   mean=12646.34, MSE=2.080781e+08
##
## Node number 6: 262 observations
##   mean=14623.87, MSE=5.261606e+07
##
## Node number 7: 321 observations
##   mean=19150.33, MSE=2.021303e+08
```

```
# R-squared ( from training data)
pred <- predict(model1, df)
rss <- sum((df$charges - pred)^2)      # residual sum of squares
tss <- sum((df$charges - mean(df$charges))^2) # total sum of squares
rsq <- 1 - rss/tss

cat("The Training R-squared is:", round(rsq, 3), "\n")
```

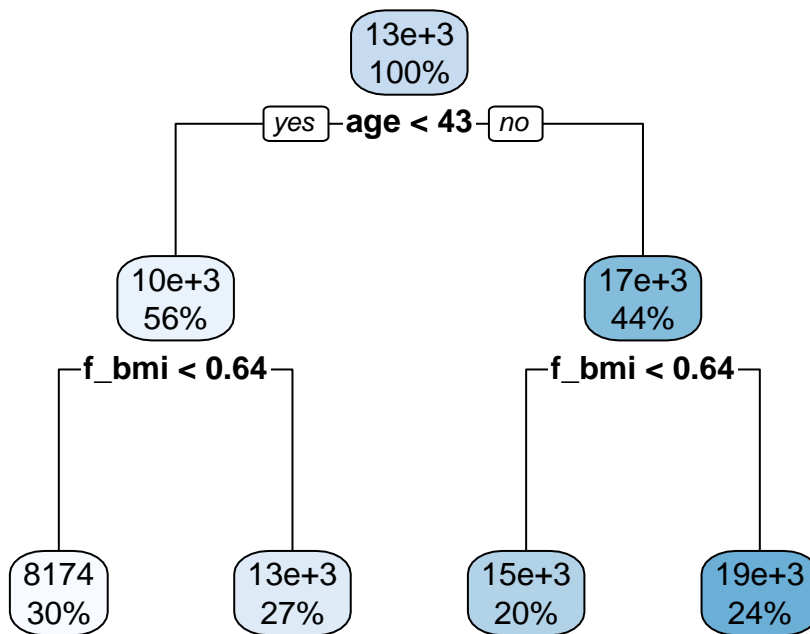
```
## The Training R-squared is: 0.112
```

The training R^2 is 11.2%, and the cross-validated relative error from printcp is ~0.92, meaning the tree only modestly improves over predicting the mean. Age contributes most of the predictive power, with importance 68 compared to 32 for BMI.

b)

```
# Decision tree predicting charges based on f_bmi and age.
model2 <- rpart(charges ~ age + f_bmi, data = df)

# Plots the decision tree
rpart.plot(model2)
```



Returns a summary of the model
`summary(model2)`

```
## Call:
## rpart(formula = charges ~ age + f_bmi, data = df)
##   n= 1338
##
##           CP nsplit rel error   xerror   xstd
## 1 0.07793137      0 1.0000000 1.0012758 0.05190793
## 2 0.01920458      1 0.9220686 0.9261661 0.04929508
## 3 0.01507422      2 0.9028640 0.9219147 0.04581099
## 4 0.01000000      3 0.8877898 0.9193190 0.04516877
##
## Variable importance
##   age f_bmi
##   68   32
##
## Node number 1: 1338 observations,    complexity param=0.07793137
## mean=13270.42, MSE=1.465428e+08
## left son=2 (755 obs) right son=3 (583 obs)
## Primary splits:
##   age < 42.5      to the left, improve=0.07793137, (0 missing)
##   f_bmi < 0.6392812 to the left, improve=0.04212369, (0 missing)
## Surrogate splits:
##   f_bmi < 0.7580472 to the left, agree=0.582, adj=0.041, (0 split)
##
## Node number 2: 755 observations,    complexity param=0.01920458
## mean=10300.81, MSE=1.313497e+08
## left son=4 (396 obs) right son=5 (359 obs)
## Primary splits:
##   f_bmi < 0.6392812 to the left, improve=0.03797077, (0 missing)
##   age < 26.5      to the left, improve=0.01289901, (0 missing)
## Surrogate splits:
```

```
##      age < 18.5      to the right, agree=0.534, adj=0.019, (0 split)
##
## Node number 3: 583 observations,      complexity param=0.01507422
##   mean=17116.14, MSE=1.400084e+08
##   left son=6 (262 obs) right son=7 (321 obs)
##   Primary splits:
##     f_bmi < 0.6427244 to the left,  improve=0.03621035, (0 missing)
##     age < 58.5      to the left,  improve=0.03075757, (0 missing)
##   Surrogate splits:
##     age < 47.5      to the left,  agree=0.566, adj=0.034, (0 split)
##
## Node number 4: 396 observations
##   mean=8174.444, MSE=5.228144e+07
##
## Node number 5: 359 observations
##   mean=12646.34, MSE=2.080781e+08
##
## Node number 6: 262 observations
##   mean=14623.87, MSE=5.261606e+07
##
## Node number 7: 321 observations
##   mean=19150.33, MSE=2.021303e+08

# R-squared ( from training data)
pred <- predict(model2, df)
rss <- sum((df$charges - pred)^2)      # residual sum of squares
tss <- sum((df$charges - mean(df$charges))^2) # total sum of squares
rsq <- 1 - rss/tss

cat("The Training R-squared is:", round(rsq, 3), "\n")
```

```
## The Training R-squared is: 0.112
```

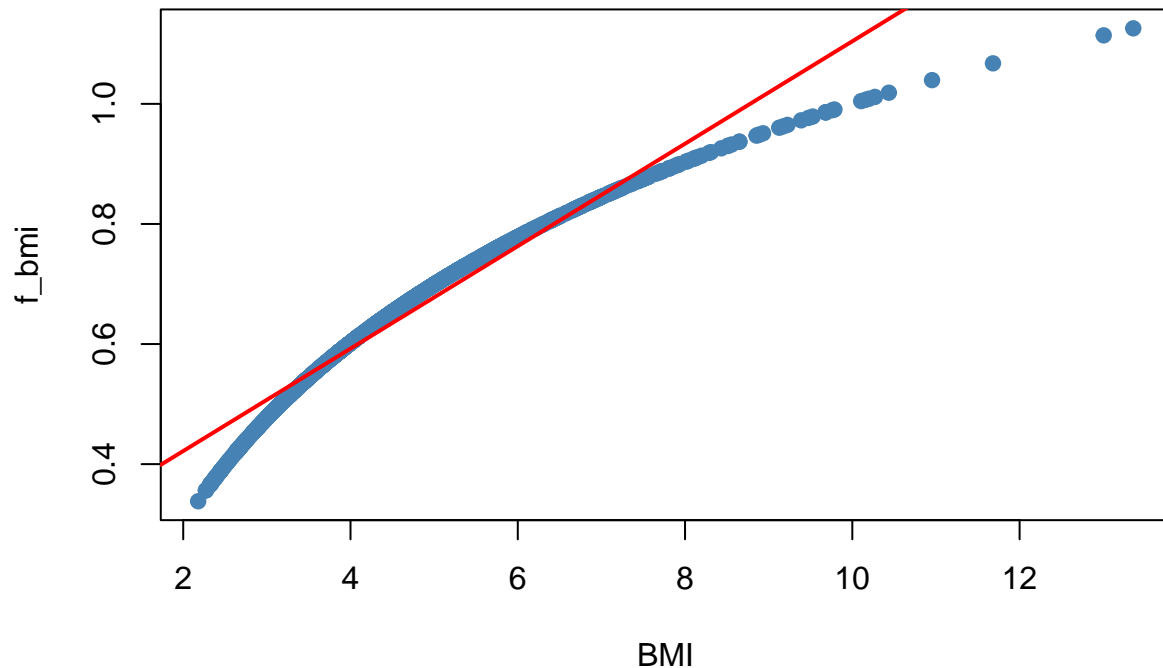
The training R^2 is again 11.2%. The tree structure and splits are almost identical to model (a), which is expected since f_bmi is a monotone transform of BMI. Age still dominates in variable importance (68 vs 32)

c)

Both trees have nearly identical splits (root split on $age < 42.5$, secondary splits on BMI 4.36 or f_bmi 0.64). Their training R^2 values are essentially the same (11.2%). This outcome is expected because f_bmi is a near-monotone transform of BMI (correlation 0.98), which preserves ordering. CART models base splits on order thresholds, so any monotone transform will yield similar partitions and performance.

```
# Plot f_bmi as a function of bmi
plot(df$bmi, df$f_bmi,
     xlab = "BMI",
     ylab = "f_bmi",
     main = "f_bmi as a function of BMI",
     pch = 19, col = "steelblue")
abline(lm(f_bmi ~ bmi, data = df), col = "red", lwd = 2)
```


f_bmi as a function of BMI



d)

e)

Firstly, we should define clearly which dependent variables we would like to keep. We will not use bmi since it is highly correlated to f_bmi, as shown above. We will keep age, and add charges, since it is not a value we aim to predict anymore.

```
# Grid of cp values
cps <- c(0, 0.02, 0.04, 0.06, 0.08, 0.1)

# Fit tree at a given cp and compute training metrics
fit_one <- function(cp) {

  fit <- rpart(cardiovascular_care_cost ~ age + f_bmi + charges,
               data = df,
               control = rpart.control(cp = cp)) # keep other defaults

  y <- df$cardiovascular_care_cost
  pred <- predict(fit, df)
  rss <- sum((y - pred)^2); tss <- sum((y - mean(y))^2)
  r2 <- 1 - rss/tss
  rmse <- sqrt(mean((y - pred)^2))

  leaves <- sum(fit$frame$var == "<leaf>")

  depth <- max(rpart:::tree.depth(as.numeric(row.names(fit$frame))))

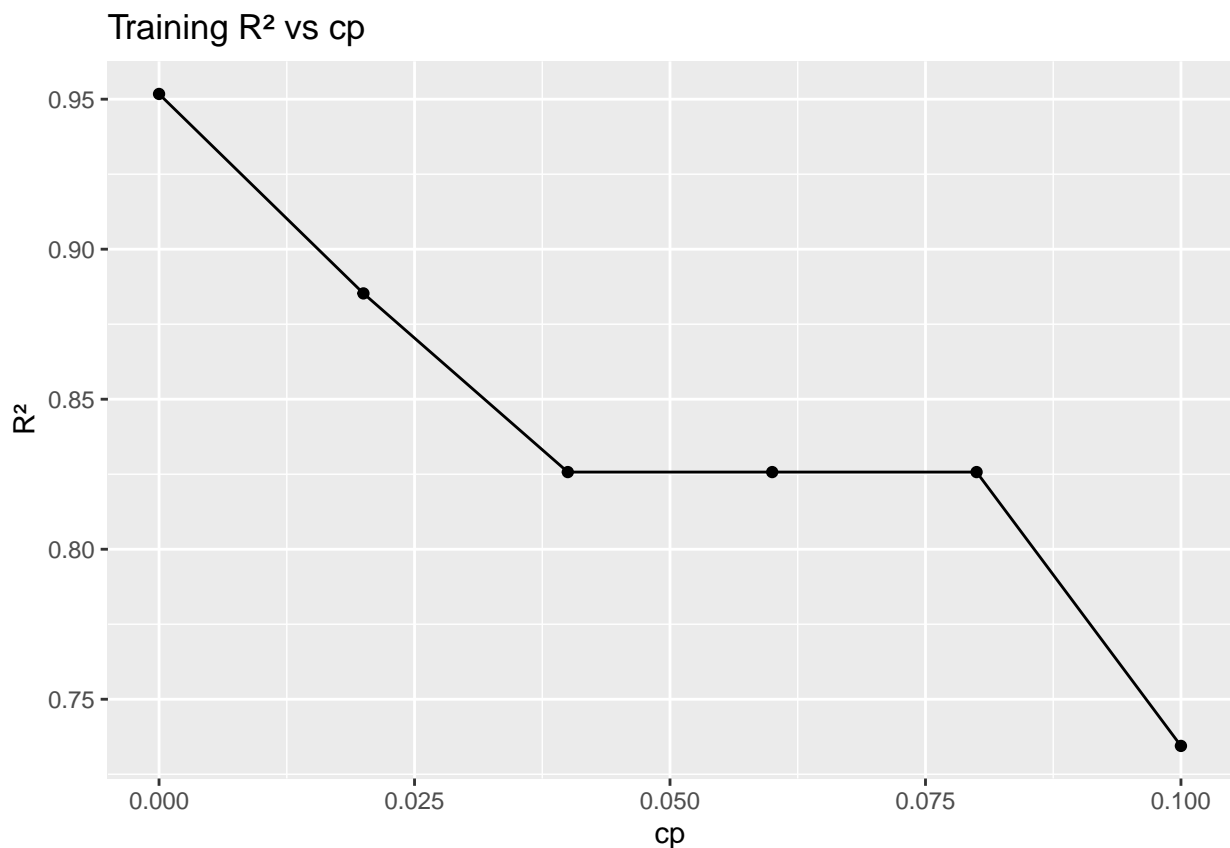
  tibble(cp = cp, R2 = r2, RMSE = rmse, Leaves = leaves, Depth = depth, model =
    ↪ list(fit))
}
```

```
# Run fits
res <- bind_rows(lapply(cps, fit_one))

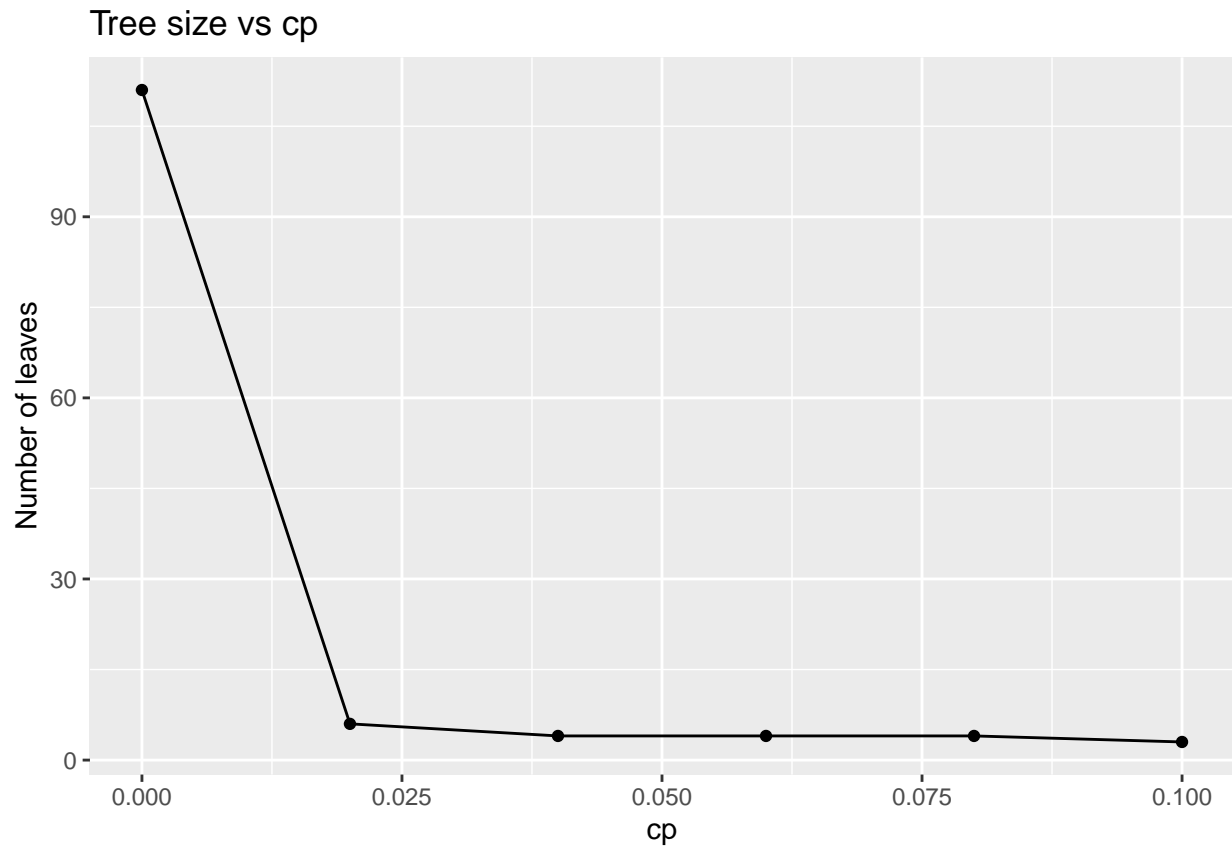
# Show summary table (metrics only)
res %>% select(cp, R2, RMSE, Leaves, Depth)
```

```
## # A tibble: 6 x 5
##   cp    R2  RMSE Leaves Depth
##   <dbl> <dbl> <dbl>   <int> <dbl>
## 1  0    0.952  169.    111    12
## 2  0.02 0.885  261.     6     3
## 3  0.04 0.826  321.     4     2
## 4  0.06 0.826  321.     4     2
## 5  0.08 0.826  321.     4     2
## 6  0.1  0.734  397.     3     2
```

```
# Plots how fit and complexity change with cp
ggplot(res, aes(cp, R2)) + geom_line() + geom_point() +
  labs(title = "Training R2 vs cp", x = "cp", y = "R2")
```



```
ggplot(res, aes(cp, Leaves)) + geom_line() + geom_point() +
  labs(title = "Tree size vs cp", x = "cp", y = "Number of leaves")
```



```
# Plot each tree (one after another)
for (i in seq_len(nrow(res))) {
  cat("\n\n## cp =", res$cp[i], "\n")
  rpart.plot(res$model[[i]],
    main = paste0("CART: cardiovascular_care_cost (cp = ", res$cp[i], ")")
  )
}
```

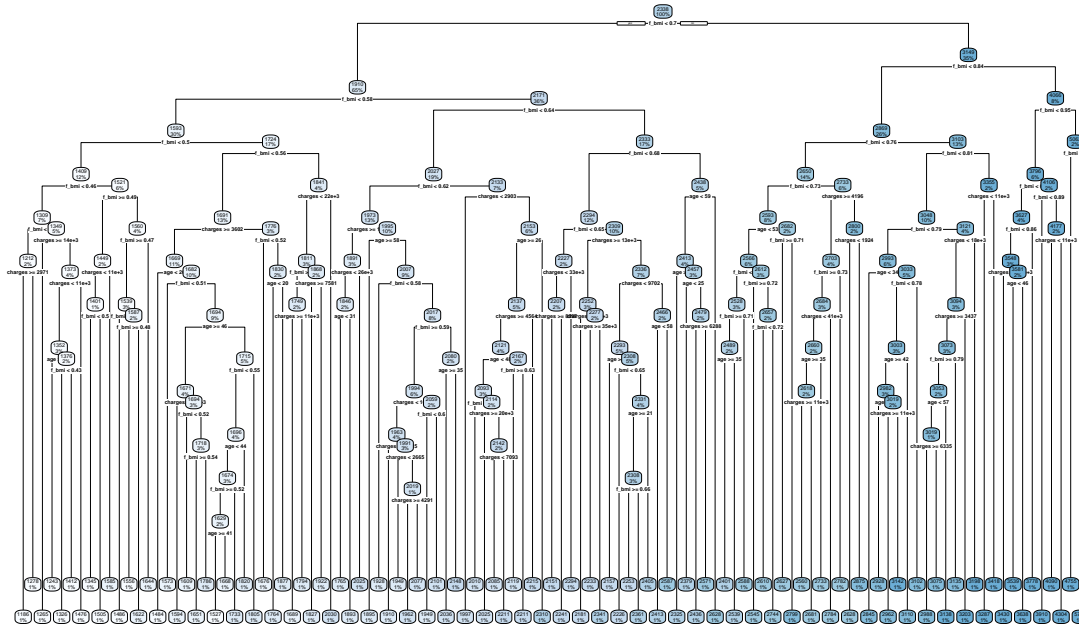
```
##
```

```
##
```

```
## ## cp = 0
```

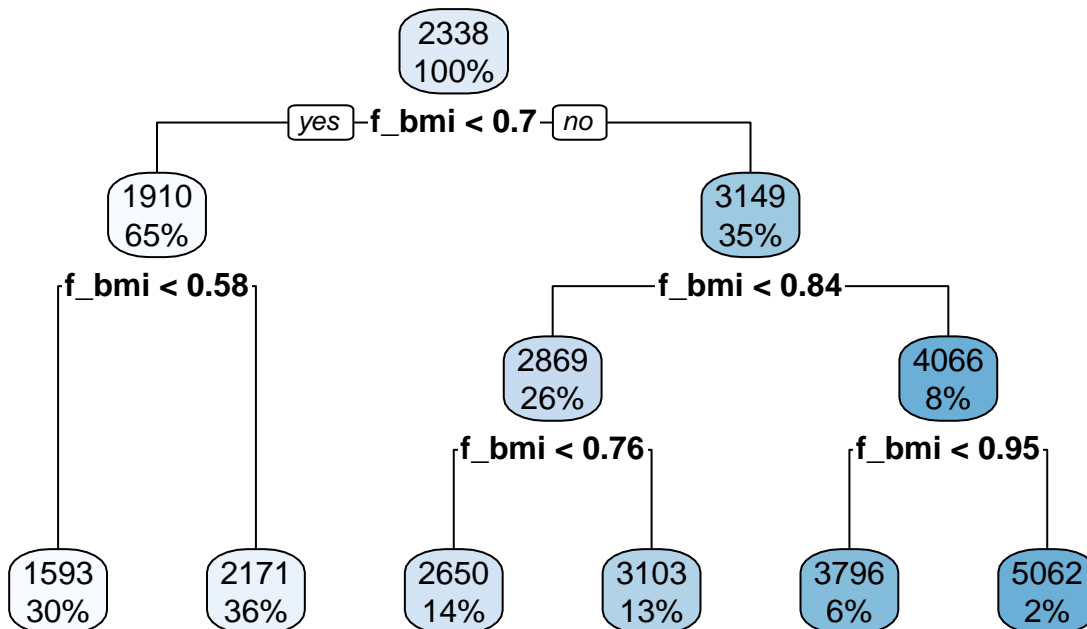
```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```

CART: cardiovascular_care_cost (cp = 0)



cp = 0.02

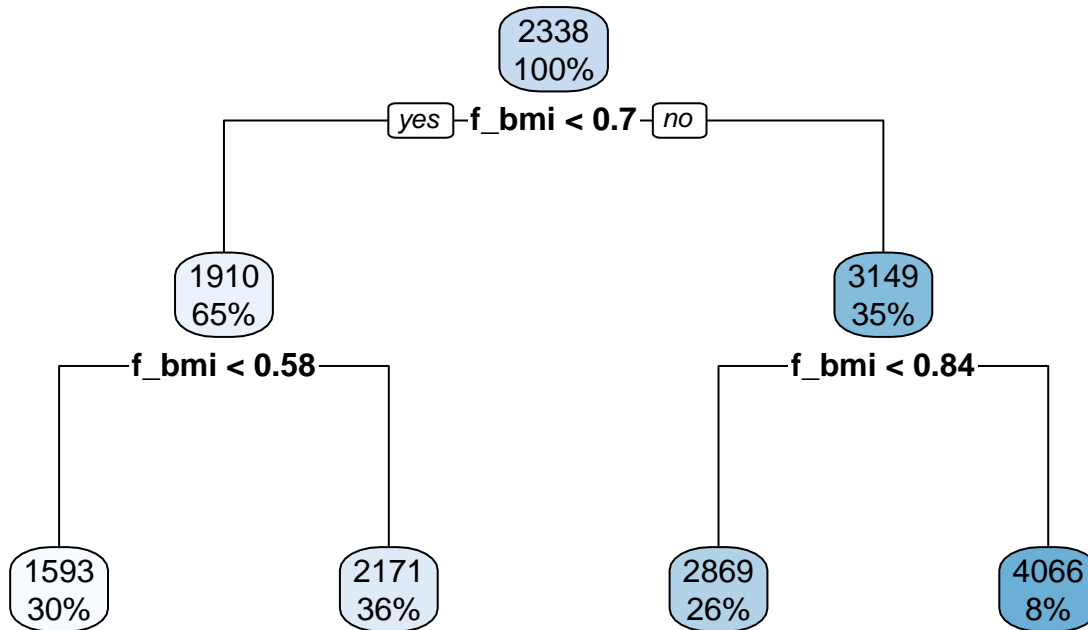
CART: cardiovascular_care_cost (cp = 0.02)



##

cp = 0.04

CART: cardiovascular_care_cost (cp = 0.04)

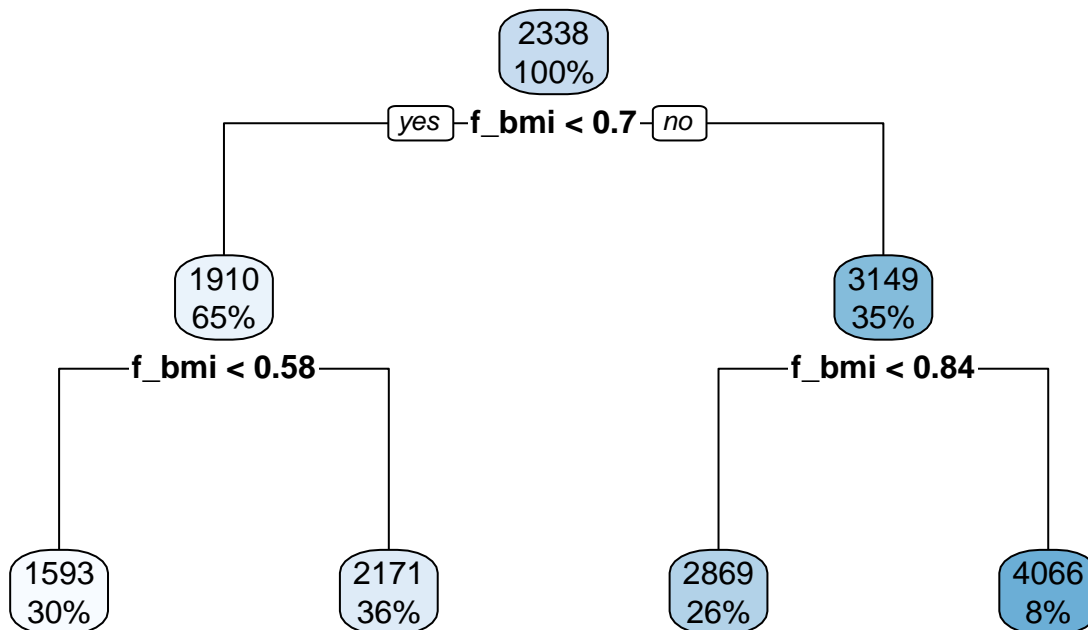


##

##

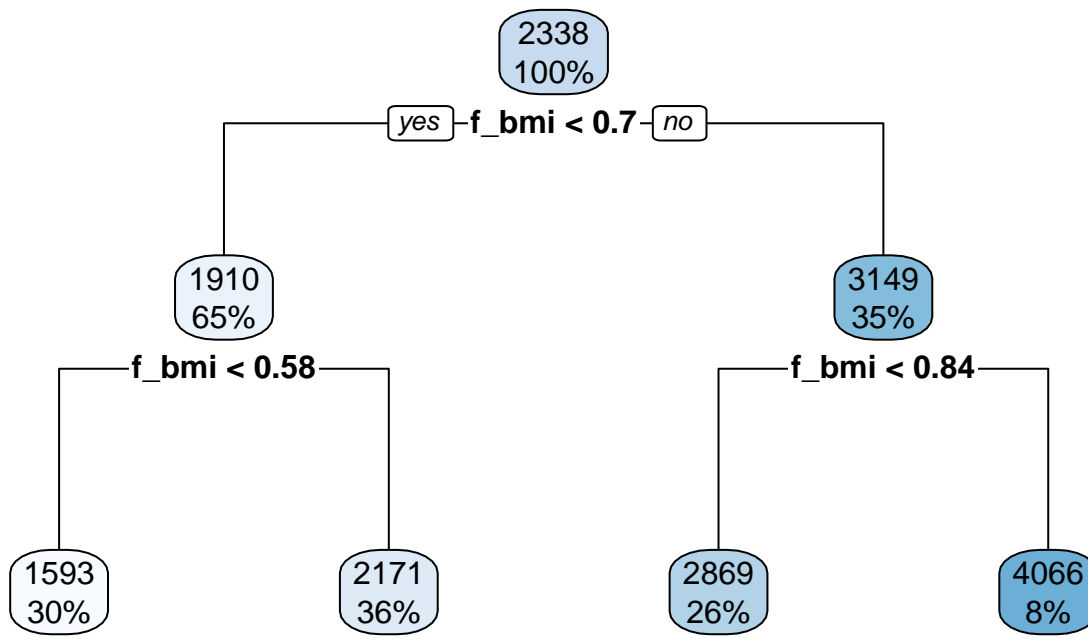
cp = 0.06

CART: cardiovascular_care_cost (cp = 0.06)



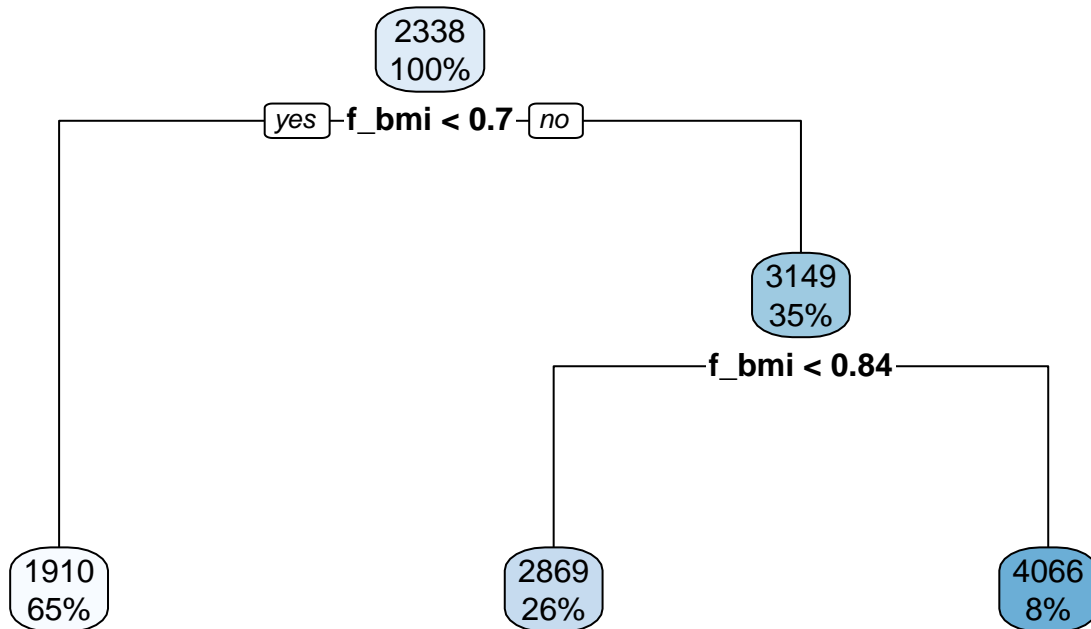
```
##
##
## ## cp = 0.08
```

CART: cardiovascular_care_cost (cp = 0.08)



```
##
##
## ## cp = 0.1
```

CART: cardiovascular_care_cost (cp = 0.1)



When $cp = 0$, the tree grows very deep (12 levels, 111 leaves), leading to extremely high training R^2 (0.95) but clear overfitting, as seen in the highly complex structure. Increasing cp prunes the tree aggressively: at $cp = 0.02$, the tree shrinks to just 6 leaves and R^2 drops to 0.89. For $cp = 0.04$ – 0.08 , the tree stabilizes with 4 leaves and depth 2, achieving R^2 around 0.83, which indicates a simpler but still reasonable fit. Finally, at $cp = 0.10$, the tree becomes even smaller (3 leaves, depth 2) and the R^2 falls further to 0.73, showing underfitting.

In summary, smaller cp values allow deeper trees with very high training R^2 (0.95) but clear overfitting, while larger cp values prune the tree aggressively, reducing variance but increasing bias. The appropriate cp should be chosen by cross-validated error using the 1-SE rule. In this case, cp around 0.04–0.06 yields a shallow tree (4 leaves, depth 2) with R^2 0.83 — a balance between interpretability and reasonable fit.