Edge_HW2_Q1

2025-09-23

Question 1 (A)

```
data <- read.csv("insurance_charges.csv")

colnames(data)
```

```
## [1] "age"                     "bmi"
## [3] "charges"                 "f_bmi"
## [5] "cardiovascular_care_cost"
```

```
library(rpart)
```

```
## 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
```

```
model = rpart(data = data,
              charges ~ bmi + age)
summary(model)
```

```
## Call:
## rpart(formula = charges ~ bmi + age, data = data)
##   n= 1338
##
##           CP nsplit rel error    xerror       xstd
## 1 0.07793137      0 1.0000000 1.0020632 0.05195904
## 2 0.01920458      1 0.9220686 0.9269748 0.04936215
## 3 0.01507422      2 0.9028640 0.9328067 0.04663839
## 4 0.01000000      3 0.8877898 0.9315428 0.04626160
##
## 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
```

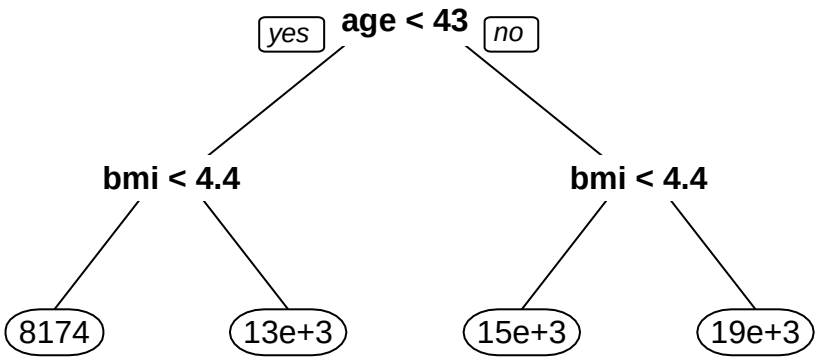
```
r2 <- function(y, yhat) 1 - (sum((y - yhat)^2) / sum((y - mean(y))^2))

pred_a <- predict(model)
r2_a <- r2(data$charges,pred_a)
```

```
cat(r2_a)
```

```
## 0.1122102
```

The R^2 value for this model is 0.1122102

```
prp(model)
```



(B)

```
model2 = rpart(data = data,
               charges ~ f_bmi + age)
summary(model2)
```

```
## Call:
## rpart(formula = charges ~ f_bmi + age, data = data)
##   n= 1338
##
##           CP nsplit rel error    xerror       xstd
## 1 0.07793137      0 1.0000000 1.0012930 0.05192333
## 2 0.01920458      1 0.9220686 0.9255019 0.04932833
## 3 0.01507422      2 0.9028640 0.9257450 0.04651140
## 4 0.01000000      3 0.8877898 0.9231287 0.04491452
##
## 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:
```

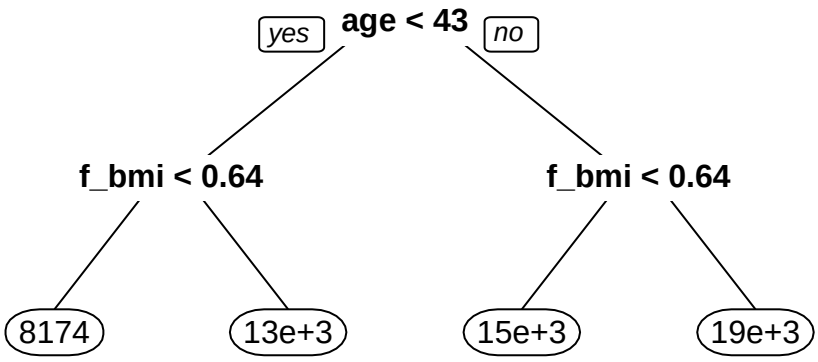
```
##       f_bmi < 0.7580472 to the left,  agree=0.582, adj=0.041, (0 split)
##
## Node number 2: 755 observations,    complexity param=0.01920458
##   mean=10300.81, MSE=1.313497e+08
##   left son=4 (396 obs) right son=5 (359 obs)
##   Primary splits:
##       f_bmi < 0.6392812 to the left,  improve=0.03797077, (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
```

```
pred_b <- predict(model2)
r2_b <- r2(data$charges,pred_b)
cat(r2_b)
```

```
## 0.1122102
```

The R^2 for this model is identical to the previous one = 0.1122102

```
prp(model2)
```



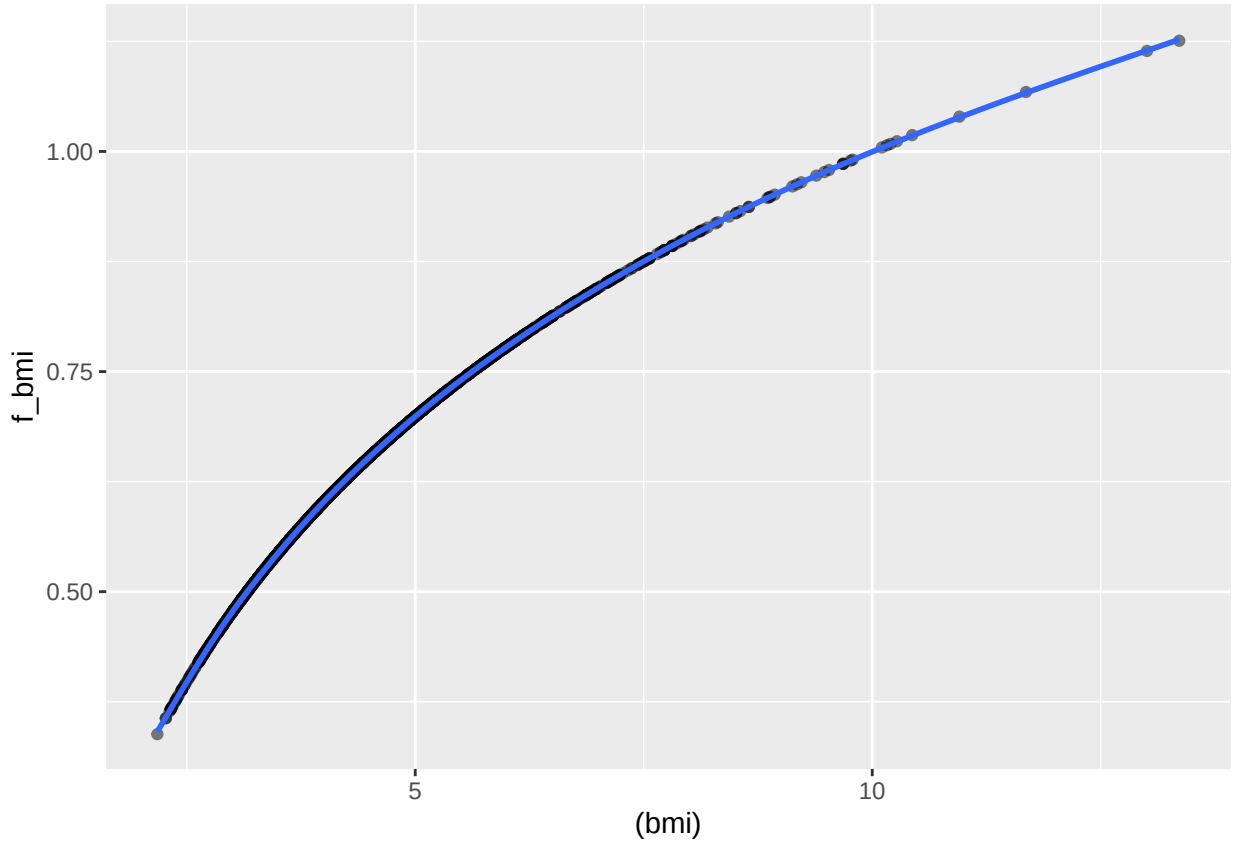
> (C)

```
library(ggplot2)
```

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

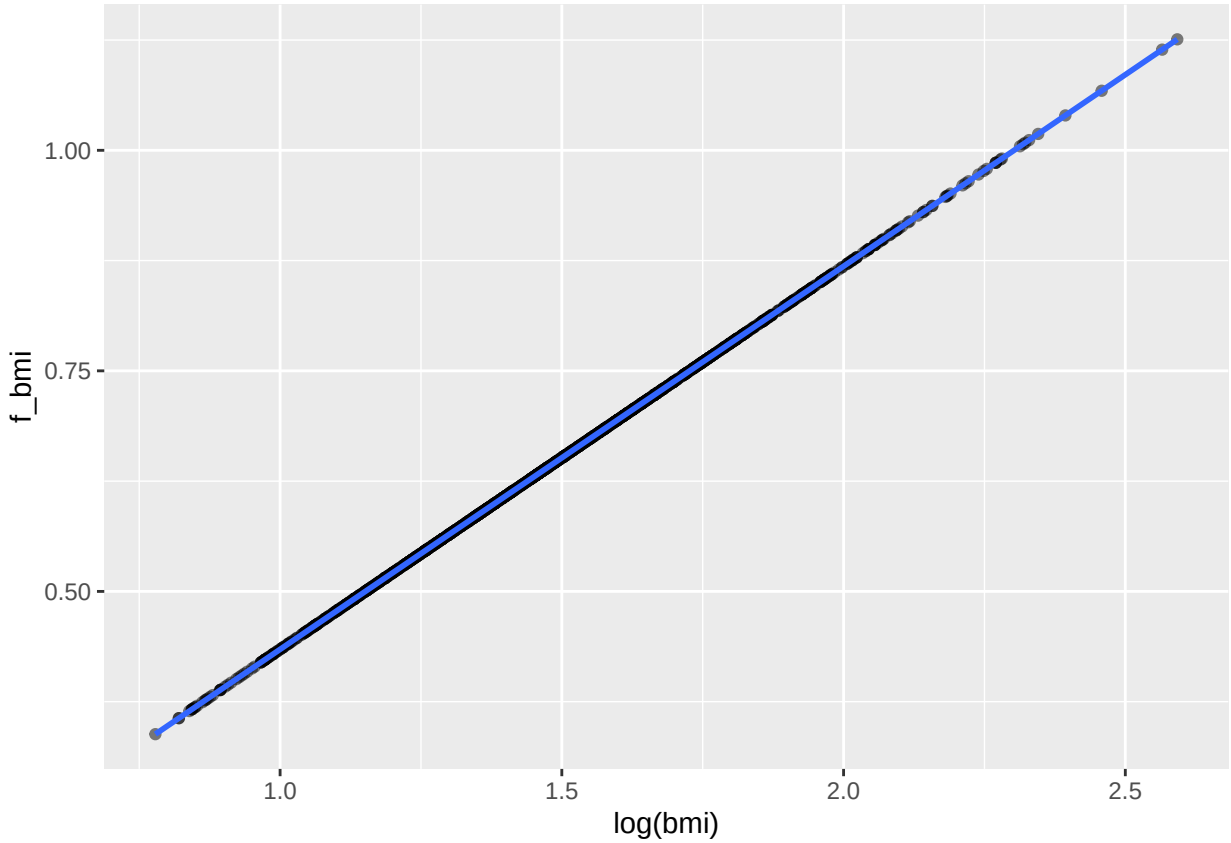
```
ggplot(data, aes((bmi), f_bmi)) + geom_point(alpha=0.5) + geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
ggplot(data, aes(log(bmi), f_bmi)) + geom_point(alpha=0.5) + geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



Here we can see that bmi vs f_bmi is related as f_bmi is a monotonic transform of bmi(scaling, log, square), therefore the trees will have similar R^2 (0.1122102) and comparable split logic although at different threshold values due to the difference in the scaling. Structural differences (number/depth of splits) may be minor; performance tends to stay close because decision trees rely primarily on ordering of values.

(D)

If f_bmi = g(bmi) where g is strictly monotone, then bmi_i < bmi_j <=> g(bmi_i) < g(bmi_j)

CART chooses split thresholds by sorting a feature and scanning cut points; only the order matters. A strictly monotonic transform preserves that order, so the same partitions of the data are available (just at transformed thresholds). Hence training fit and structure remain essentially unchanged (up to ties/rounding). The entropy or impurity of the partitions remain the same as the proportion of points in each bucket doesnt change.

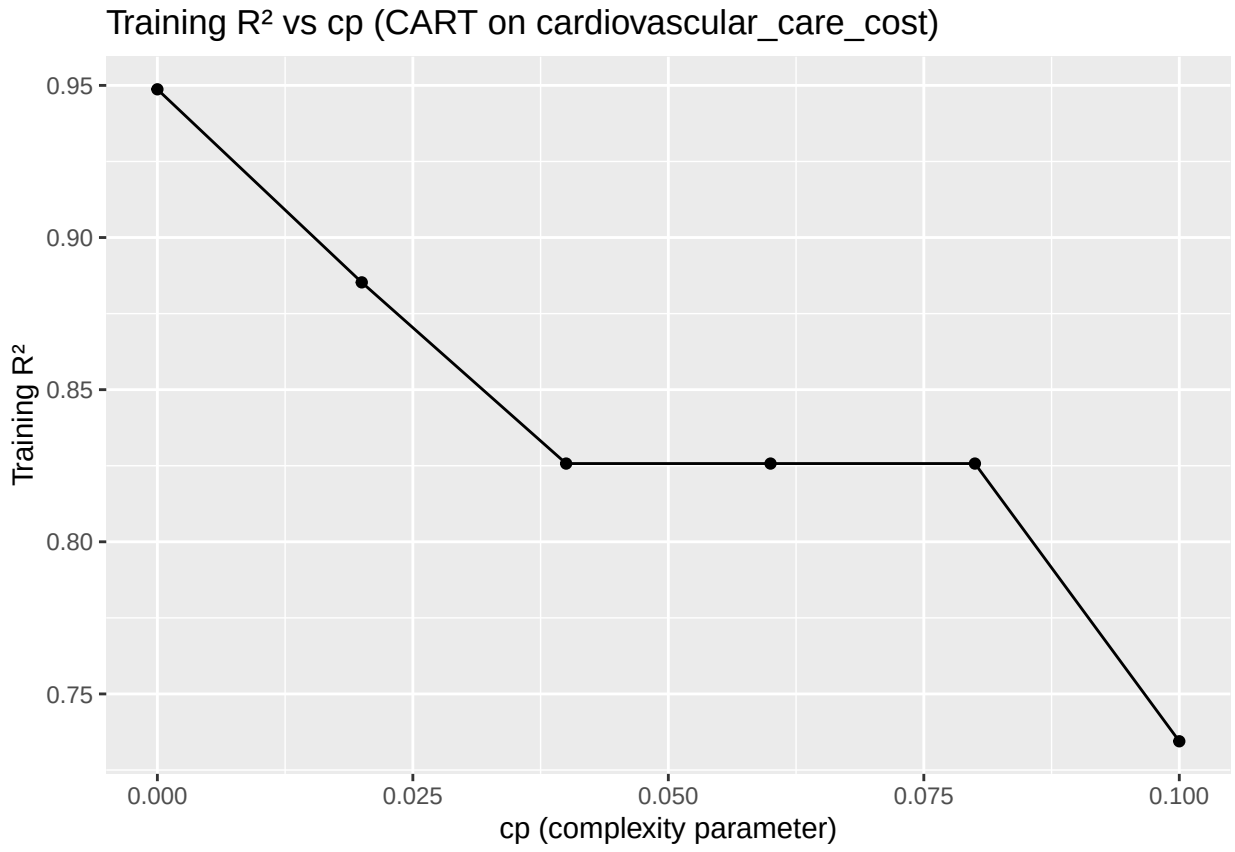
(E)

```
cps <- c(0, 0.02, 0.04, 0.06, 0.08, 0.10)

fit_stats <- lapply(cps, function(cp_val) {
  fit <- rpart(
    cardiovascular_care_cost ~ age + bmi + f_bmi,
    data = data,
    control = rpart.control(cp = cp_val)
  )
  yhat <- predict(fit)
  data.frame(
    cp = cp_val,
```

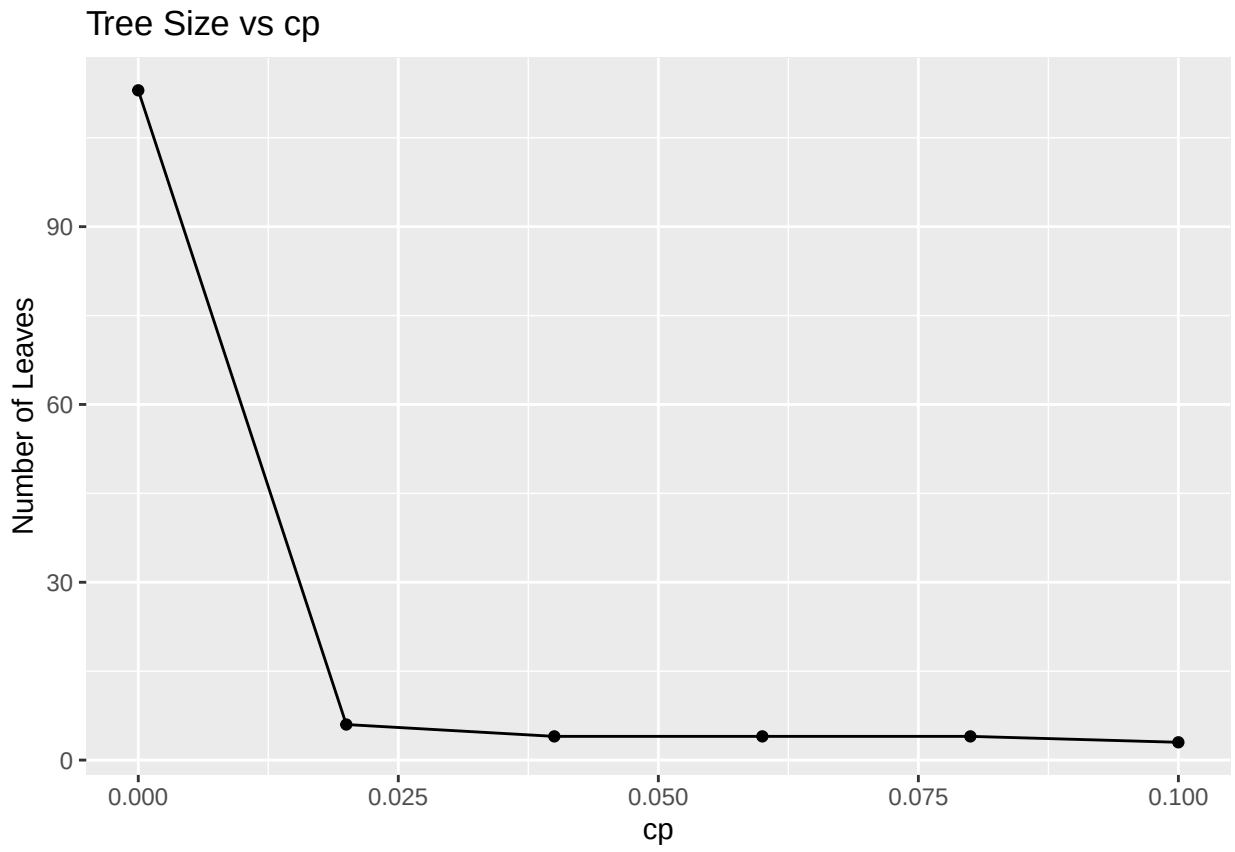
```
    r2 = r2(data$cardiovascular_care_cost, yhat),
    leaves = sum(fit$frame$var == "<leaf>")
  )
})

fit_stats <- do.call(rbind, fit_stats)
print(fit_stats)
```

```
##     cp        r2 leaves
## 1 0.00 0.9487180    113
## 2 0.02 0.8852636      6
## 3 0.04 0.8257149      4
## 4 0.06 0.8257149      4
## 5 0.08 0.8257149      4
## 6 0.10 0.7344426      3
```

```
# R^2 vs cp
ggplot(fit_stats, aes(cp, r2)) +
  geom_line() + geom_point() +
  labs(title = "Training R² vs cp (CART on cardiovascular_care_cost)",
       x = "cp (complexity parameter)", y = "Training R²")
```



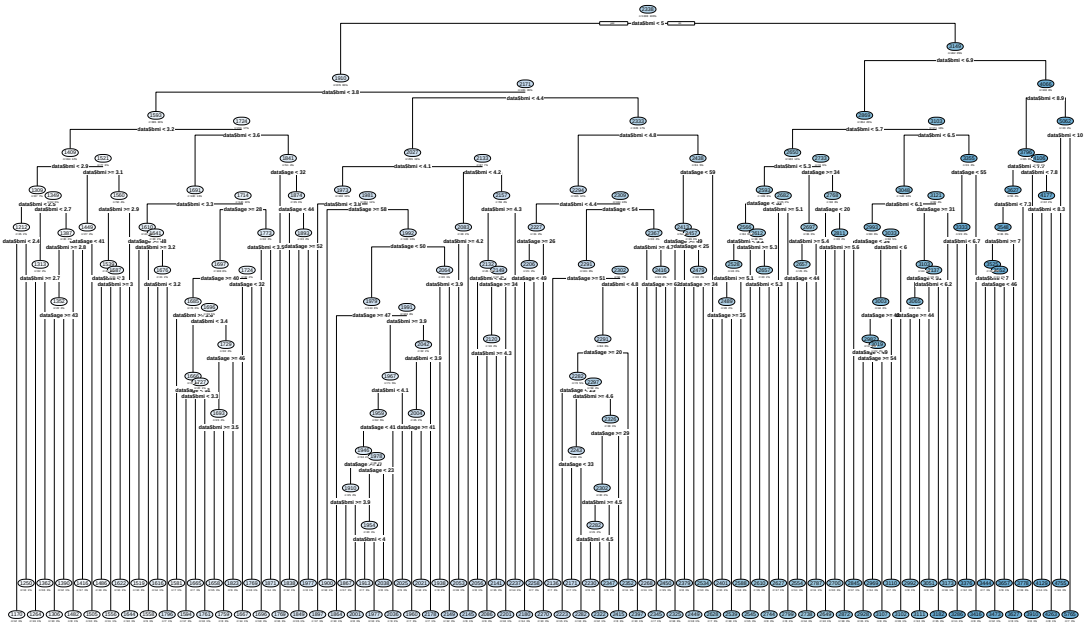

```
# leaves vs cp
ggplot(fit_stats, aes(cp, leaves)) +
  geom_line() + geom_point() +
  labs(title = "Tree Size vs cp", x = "cp", y = "Number of Leaves")
```

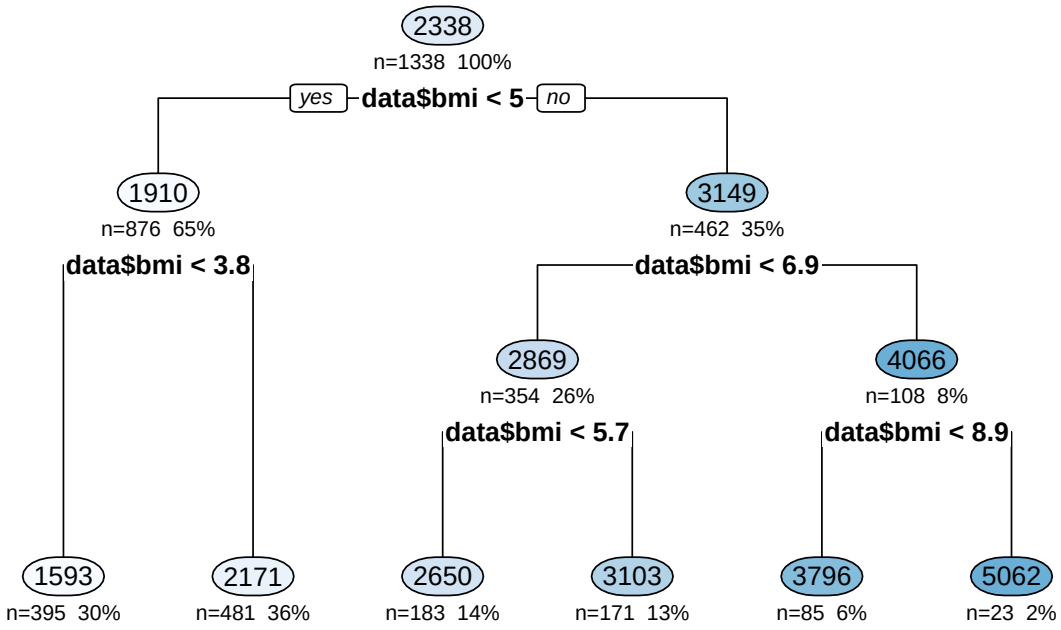


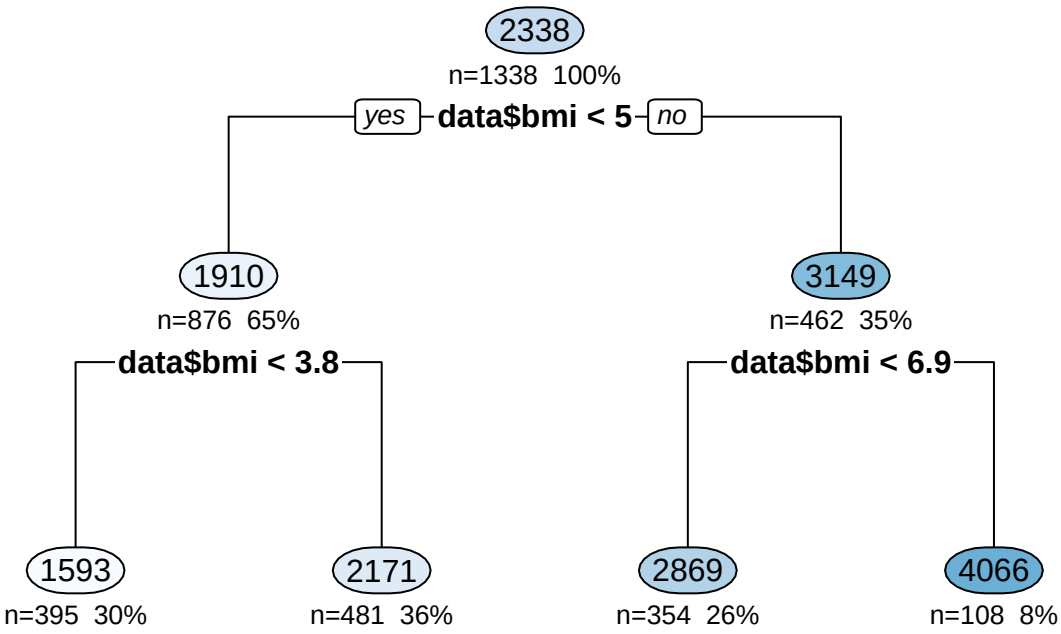

```
for (cp_val in cps) {
  fit <- rpart(
    data$cardiovascular_care_cost ~ data$age + data$bmi + data$f_bmi,
    data = data,
    control = rpart.control(cp = cp_val)
  )
  rpart.plot(fit, type = 2, extra = 101, under = TRUE,
             main = paste0("CART tree (cp = ", cp_val, ")"))
}
```

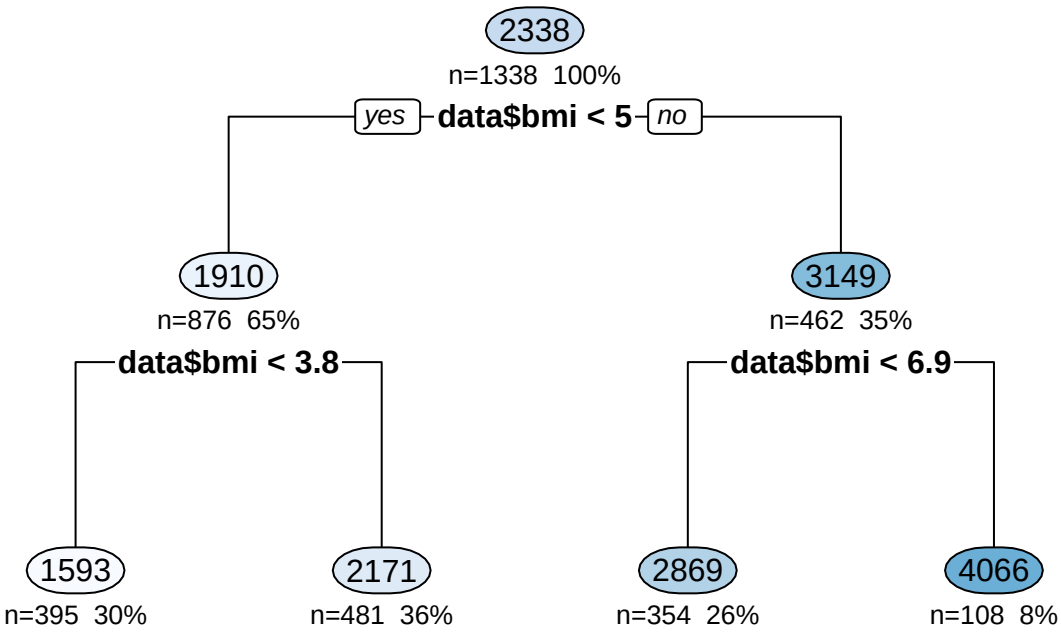
```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```

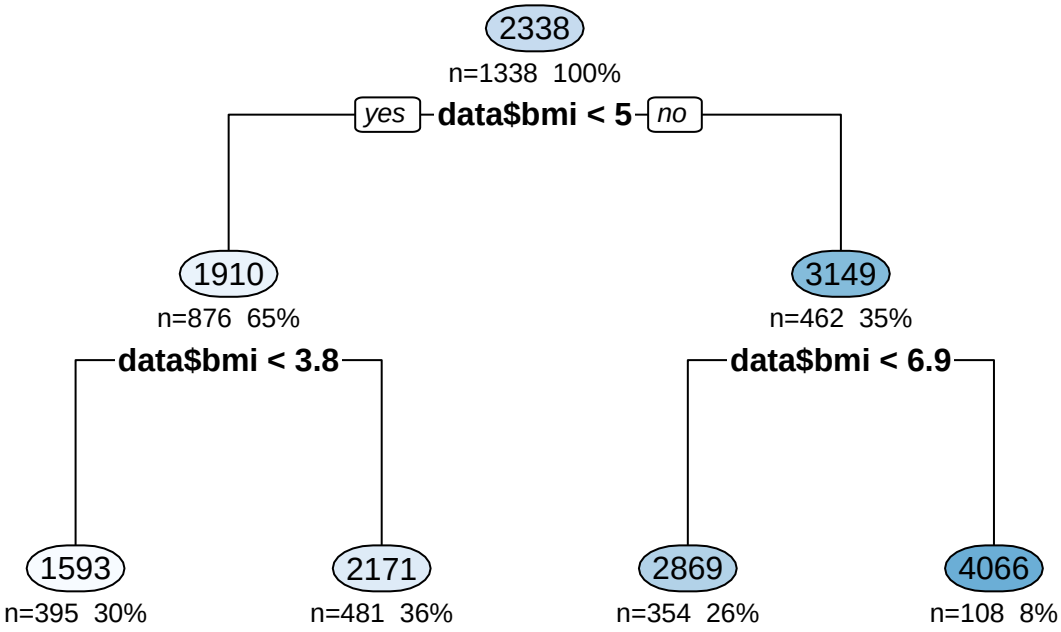
CART tree (cp = 0)

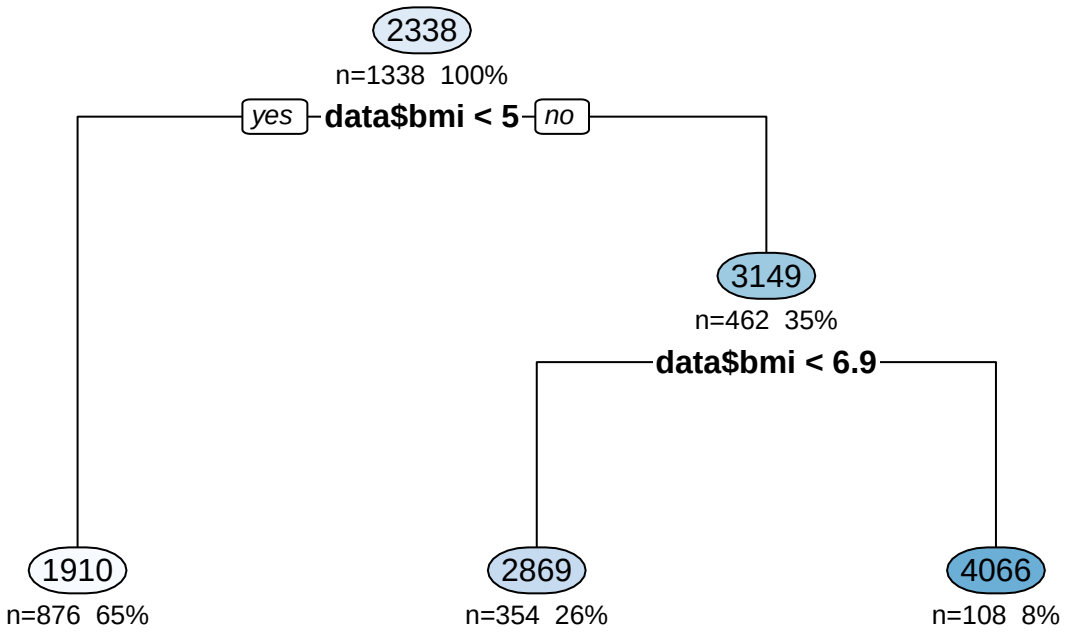



CART tree (cp = 0.02)
2338
n=1338 100%
yes
data$bmi < 5
no
1910
n=876 65%
3149
n=462 35%
data$bmi < 3.8
data$bmi < 6.9
2869
n=354 26%
4066
n=108 8%
data$bmi < 5.7
data$bmi < 8.9
1593
n=395 30%
2171
n=481 36%
2650
n=183 14%
3103
n=171 13%
3796
n=85 6%
5062
n=23 2%


CART tree (cp = 0.04)
2338
n=1338 100%
yes
data$bmi < 5
no
1910
n=876 65%
3149
n=462 35%
data$bmi < 3.8
data$bmi < 6.9
1593
n=395 30%
2171
n=481 36%
2869
n=354 26%
4066
n=108 8%


CART tree (cp = 0.06)
2338
n=1338 100%
yes
data$bmi < 5
no
1910
n=876 65%
3149
n=462 35%
data$bmi < 3.8
data$bmi < 6.9
1593
n=395 30%
2171
n=481 36%
2869
n=354 26%
4066
n=108 8%


CART tree (cp = 0.08)
2338
n=1338 100%
yes
data$bmi < 5
no
1910
n=876 65%
3149
n=462 35%
data$bmi < 3.8
data$bmi < 6.9
1593
n=395 30%
2171
n=481 36%
2869
n=354 26%
4066
n=108 8%



Low cp (0.00): The tree is essentially unpruned. It grows very deep (113 leaves), fitting the training data extremely well ($R^2 \sim 0.95$). This is classic overfitting — it memorizes fine details and noise.

Moderate cp (0.02–0.04): A small increase in cp dramatically reduces complexity. The tree drops to 6 leaves (cp=0.02) and then stabilizes at 4 leaves for cp between 0.04 and 0.08. Training R^2 decreases to ~0.82–0.89, but the model is much simpler and likely generalizes better.

High cp (0.10): The tree prunes too aggressively, leaving only 3 leaves. Training R^2 falls to ~0.73, suggesting underfitting (too simple to capture real structure).

The CART model's fit decreases as cp increases: R^2 falls from 0.95 (unpruned) to 0.73 (heavily pruned), while the number of leaves drops from 113 to just 3. At low cp, the model overfits; at high cp, it underfits. Intermediate cp values (0.02–0.04) provide the best balance between complexity and fit, yielding a smaller tree that still explains most of the variance.