Question 10.1

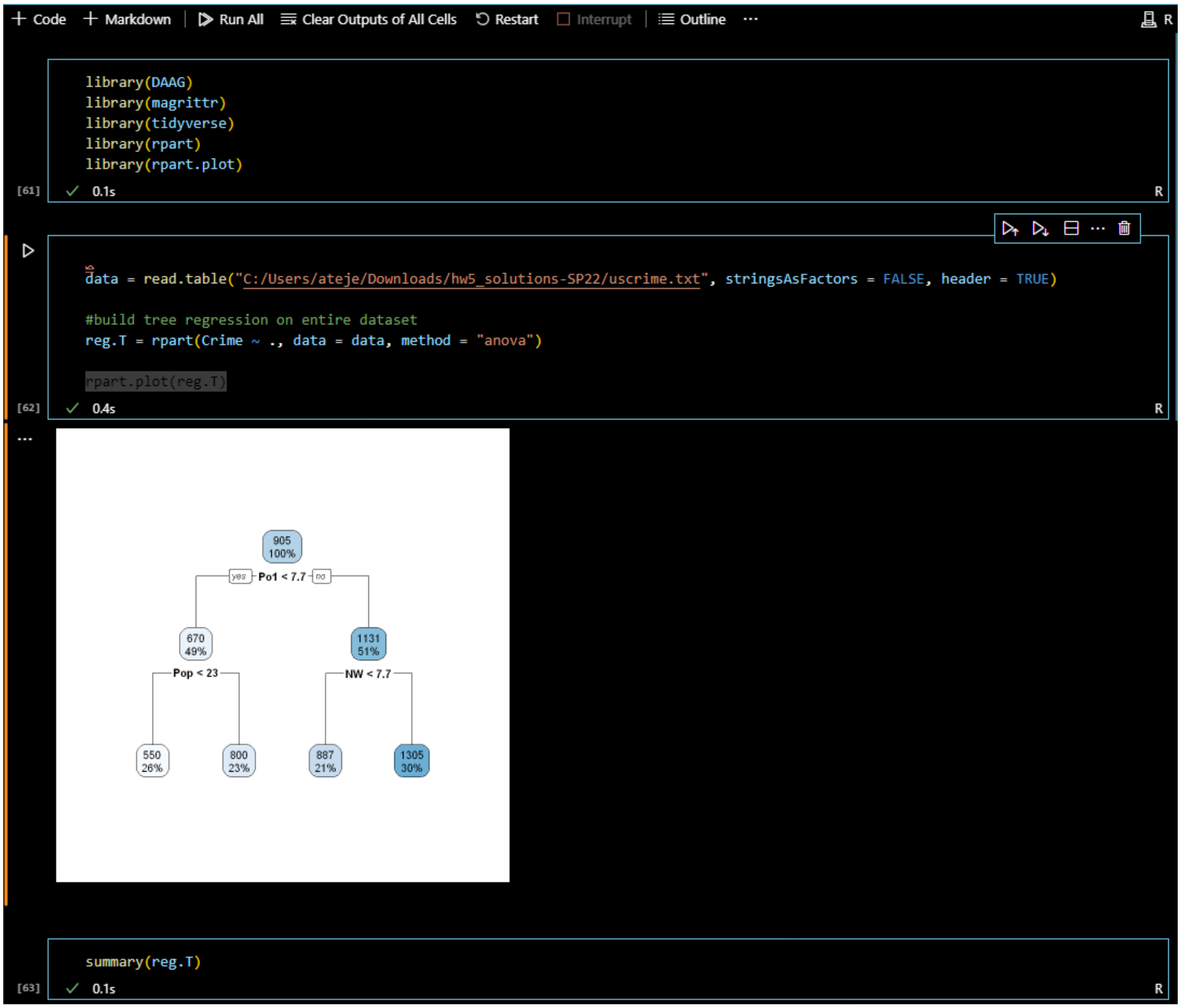
Using the same crime data set `uscrime.txt` as in Questions 8.2 and 9.1, find the best model you can using

(a) a regression tree model, and

(b) a random forest model.

In R, you can use the `tree` package or the `rpart` package, and the `randomForest` package. For each model, describe one or two qualitative takeaways you get from analyzing the results (i.e., don't just stop when you have a good model, but interpret it too).

Solution:



```
library(DAAG)
library(magrittr)
library(tidyverse)
library(rpart)
library(rpart.plot)
```

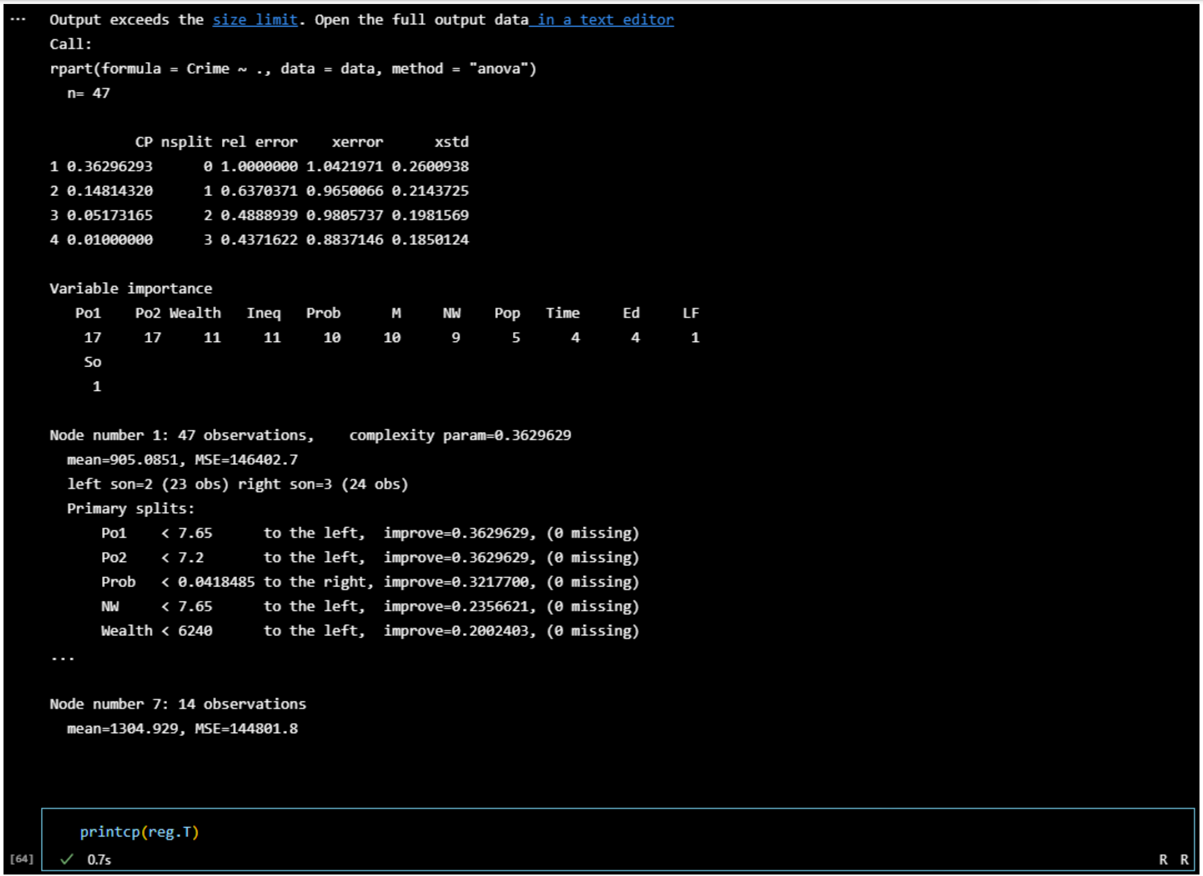
[61] ✓ 0.1s

```
data = read.table("C:/Users/ateje/Downloads/hw5_solutions-SP22/uscrime.txt", stringsAsFactors = FALSE, header = TRUE)

#build tree regression on entire dataset
reg.T = rpart(Crime ~ ., data = data, method = "anova")

rpart.plot(reg.T)
```

[62] ✓ 0.4s

```
summary(reg.T)
```

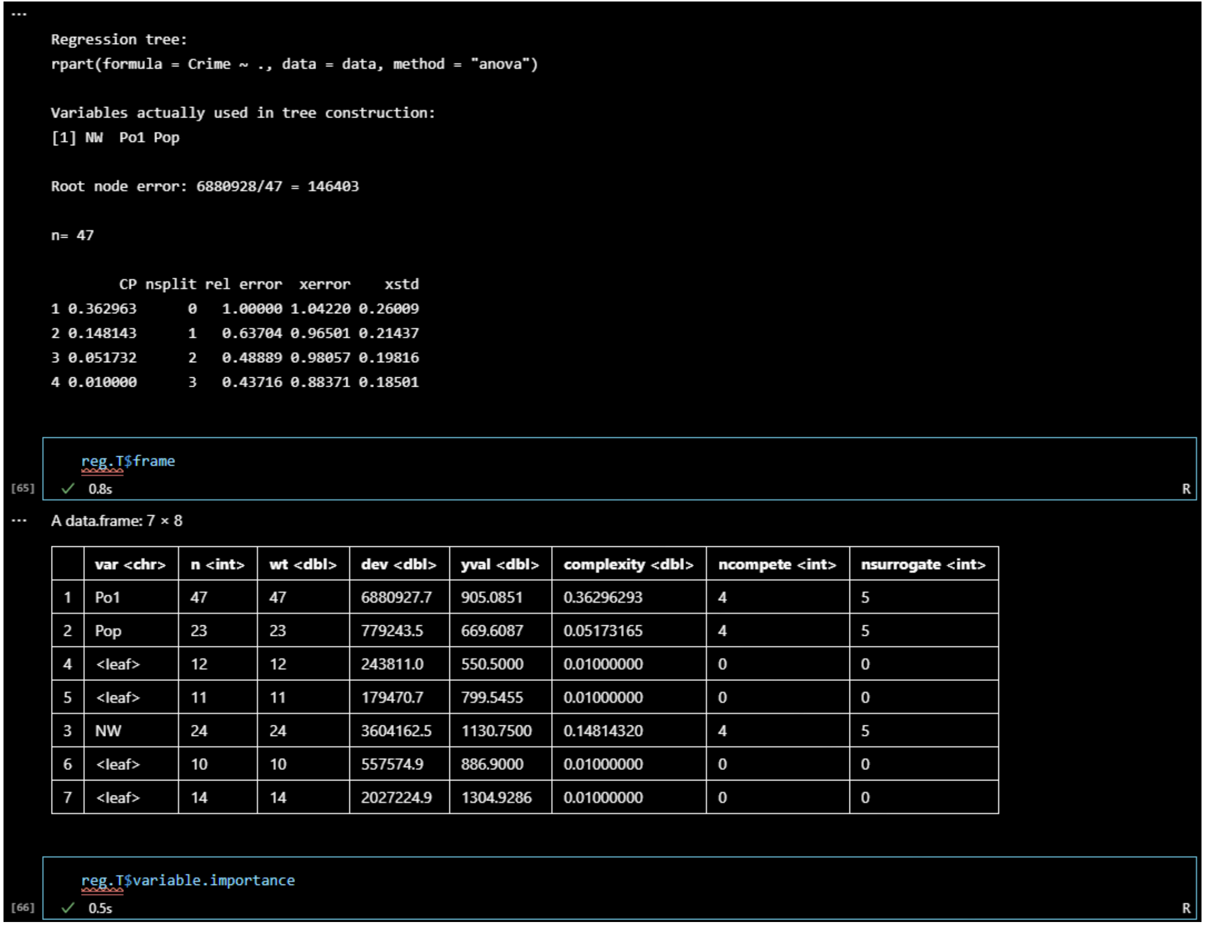
[63] ✓ 0.1s



```
Output exceeds the size limit. Open the full output data in a text editor
Call:
rpart(formula = Crime ~ ., data = data, method = "anova")
  n= 47

          CP nsplit rel error    xerror      xstd
1 0.36296293      0 1.0000000 1.0421971 0.2600938
2 0.14814320      1 0.6370371 0.9650066 0.2143725
3 0.05173165      2 0.4888939 0.9805737 0.1981569
4 0.01000000      3 0.4371622 0.8837146 0.1850124

Variable importance
   Po1    Po2 Wealth   Ineq   Prob      M     NW    Pop   Time     Ed     LF 
    17     17     11     11     10     10      9      5      4      4      1 
    So 
     1 

Node number 1: 47 observations,    complexity param=0.3629629
  mean=905.0851, MSE=146402.7 
  left son=2 (23 obs) right son=3 (24 obs)
  Primary splits:
      Po1    < 7.65      to the left,  improve=0.3629629, (0 missing)
      Po2    < 7.2       to the left,  improve=0.3629629, (0 missing)
      Prob   < 0.0418485 to the right, improve=0.3217700, (0 missing)
      NW     < 7.65      to the left,  improve=0.2356621, (0 missing)
      Wealth < 6240      to the left,  improve=0.2002403, (0 missing)
...

Node number 7: 14 observations
  mean=1304.929, MSE=144801.8 
```

```
printcp(reg.T)
```

[64] ✓ 0.7s



```
Regression tree:
rpart(formula = Crime ~ ., data = data, method = "anova")

Variables actually used in tree construction:
[1] NW  Po1 Pop

Root node error: 6880928/47 = 146403

n= 47 

        CP nsplit rel error  xerror    xstd
1 0.362963      0   1.00000 1.04220 0.26009
2 0.148143      1   0.63704 0.96501 0.21437
3 0.051732      2   0.48889 0.98057 0.19816
4 0.010000      3   0.43716 0.88371 0.18501
```

```
reg.T$frame
```

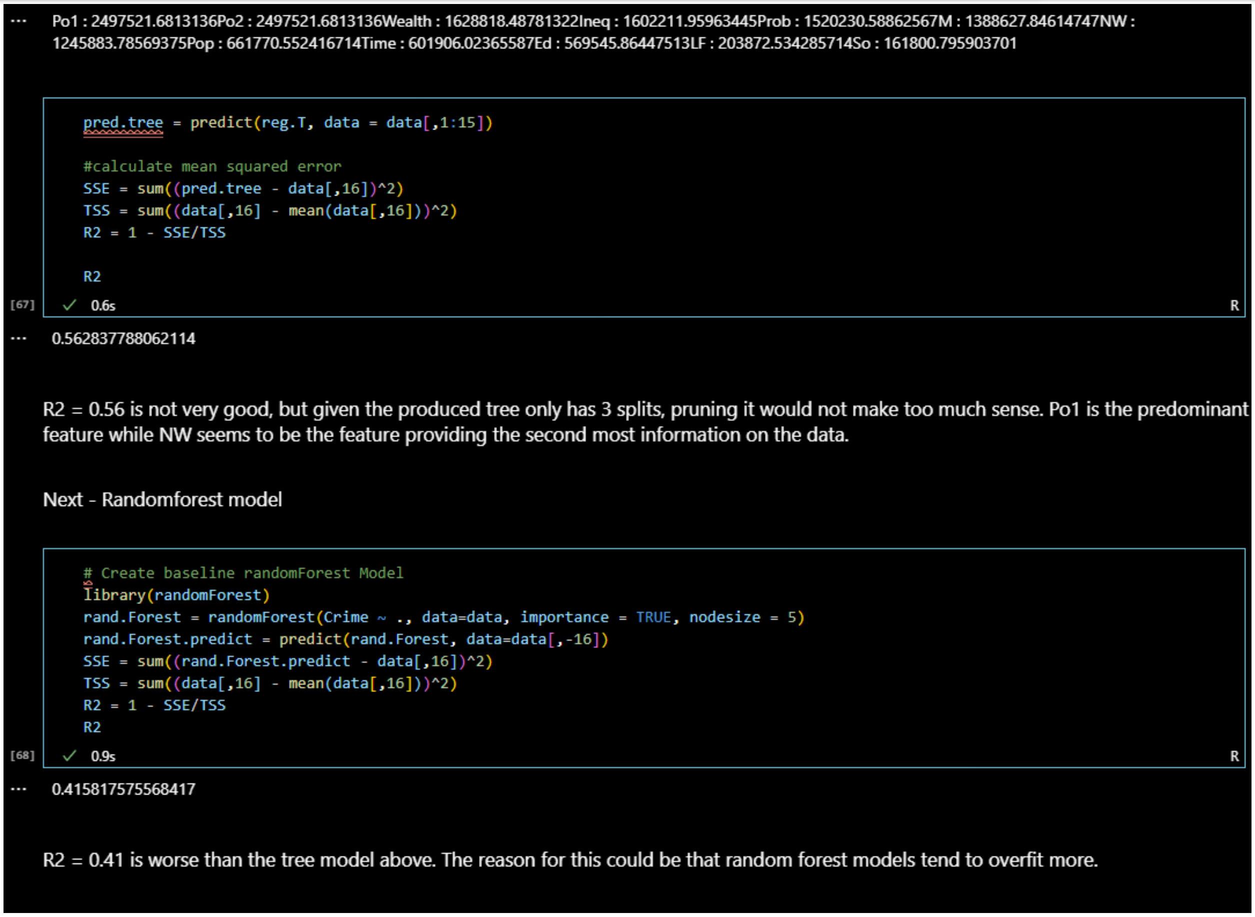
[65] ✓ 0.8s

A data.frame: 7 × 8

	var <chr>	n <int>	wt <dbl>	dev <dbl>	yval <dbl>	complexity <dbl>	ncompete <int>	nsurrogate <int>
1	Po1	47	47	6880927.7	905.0851	0.36296293	4	5
2	Pop	23	23	779243.5	669.6087	0.05173165	4	5
4	<leaf>	12	12	243811.0	550.5000	0.01000000	0	0
5	<leaf>	11	11	179470.7	799.5455	0.01000000	0	0
3	NW	24	24	3604162.5	1130.7500	0.14814320	4	5
6	<leaf>	10	10	557574.9	886.9000	0.01000000	0	0
7	<leaf>	14	14	2027224.9	1304.9286	0.01000000	0	0

```
reg.T$variable.importance
```

[66] ✓ 0.5s



```
Po1 : 2497521.6813136Po2 : 2497521.6813136Wealth : 1628818.48781322Ineq : 1602211.95963445Prob : 1520230.58862567M : 1388627.84614747NW :
1245883.78569375Pop : 661770.552416714Time : 601906.02365587Ed : 569545.86447513LF : 203872.534285714So : 161800.795903701
```

```
pred.tree = predict(reg.T, data = data[,1:15])

#calculate mean squared error
SSE = sum((pred.tree - data[,16])^2)
TSS = sum((data[,16] - mean(data[,16]))^2)
R2 = 1 - SSE/TSS

R2
```

[67] ✓ 0.6s R

```
0.562837788062114
```

R2 = 0.56 is not very good, but given the produced tree only has 3 splits, pruning it would not make too much sense. Po1 is the predominant feature while NW seems to be the feature providing the second most information on the data.

Next - Randomforest model

```
# Create baseline randomForest Model
library(randomForest)
rand.Forest = randomForest(Crime ~ ., data=data, importance = TRUE, nodesize = 5)
rand.Forest.predict = predict(rand.Forest, data=data[,-16])
SSE = sum((rand.Forest.predict - data[,16])^2)
TSS = sum((data[,16] - mean(data[,16]))^2)
R2 = 1 - SSE/TSS
R2
```

[68] ✓ 0.9s R

```
0.415817575568417
```

R2 = 0.41 is worse than the tree model above. The reason for this could be that random forest models tend to overfit more.