Decision_Tree

Josephine Decker

2023-04-30

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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(MASS )
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(tree)
library(e1071)
```

Let's read in the csv file

```
project <- read_csv("data.csv")
```

```
## Rows: 5726 Columns: 60
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (41): Country, Economy Code, ISO Code, Region, Income Group, Can a woman...
## dbl (19): Year, WBL INDEX, MOBILITY, WORKPLACE, PAY, MARRIAGE, PARENTHOOD, L...
##
## 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.
```

```r
data_18 <- project%>%
  filter(Year == 2018)
```

```r
data_19 <- project%>%
  filter(Year == 2019)
```

```r
data_20 <- project%>%
  filter(Year == 2020)
```

```r
data_21 <- project%>%
  filter(Year == 2021)
```

```r
summary(data_18$`Total(thousands)`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##       9    1806    4393   17118   13879  211998      88
```

Adding in the new Column (target) for 2018

```r
data_18$T_rank <- as.factor(ifelse(data_18$`Total(thousands)` < 1806, 'Low',
                     ifelse(data_18$`Total(thousands)` < 4393, 'LowMedium',
                     ifelse(data_18$`Total(thousands)` < 13879, 'HighMedium', 'High'))))


data_18%>%
  count(T_rank)
```

```
## # A tibble: 5 x 2
##   T_rank         n
##   <fct>      <int>
## 1 High          28
## 2 HighMedium    27
## 3 Low           28
## 4 LowMedium     27
## 5 <NA>          88
```

2019

```r
summary(data_19$`Total(thousands)`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##      12    2027    4931   17980   14797  217877      89
```

```r
data_19$T_rank <- as.factor(ifelse(data_19$`Total(thousands)` < 2027, 'Low',
                     ifelse(data_19$`Total(thousands)` < 4931, 'LowMedium',
                     ifelse(data_19$`Total(thousands)` < 14797, 'HighMedium', 'High'))))


data_19%>%
  count(T_rank)
```

```
## # A tibble: 5 x 2
##   T_rank         n
##   <fct>      <int>
## 1 High          28
## 2 HighMedium    27
## 3 Low           27
## 4 LowMedium     27
## 5 <NA>          89
```

2020

```
summary(data_20$`Total(thousands)`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       2     507    1311    6067    3837  117109      93
```

```
data_20$T_rank <- as.factor(ifelse(data_20$`Total(thousands)` < 507, 'Low',
                     ifelse(data_20$`Total(thousands)` < 1311, 'LowMedium',
                     ifelse(data_20$`Total(thousands)` < 3837, 'HighMedium', 'High'))))


data_20%>%
  count(T_rank)
```

```
## # A tibble: 5 x 2
##   T_rank         n
##   <fct>      <int>
## 1 High          27
## 2 HighMedium    26
## 3 Low           26
## 4 LowMedium     26
## 5 <NA>          93
```

2021

```
summary(data_21$`Total(thousands)`)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##      3.0    255.5    826.0   7056.6   3265.0 141297.0      103
```

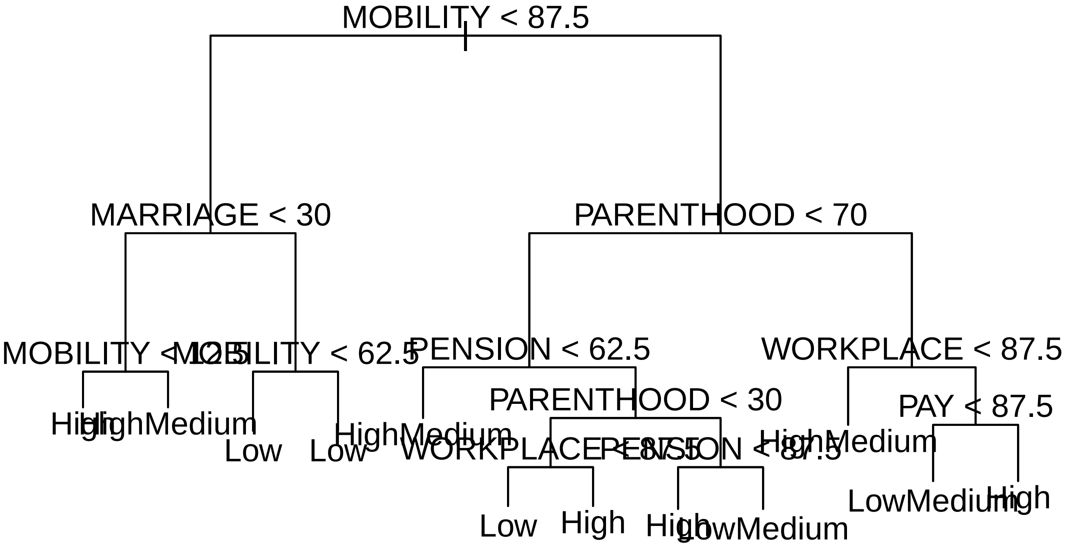
```
data_21$T_rank <- as.factor(ifelse(data_21$`Total(thousands)` < 255.5, 'Low',
                     ifelse(data_21$`Total(thousands)` < 826, 'LowMedium',
                     ifelse(data_21$`Total(thousands)` < 3265, 'HighMedium', 'High'))))


data_21%>%
  count(T_rank)
```

```
## # A tibble: 5 x 2
##   T_rank         n
##   <fct>      <int>
## 1 High          24
## 2 HighMedium    24
## 3 Low           24
## 4 LowMedium     23
## 5 <NA>         103
```

Decision Tree

```
tr1 <- tree(as.factor(T_rank) ~MOBILITY + MARRIAGE + WORKPLACE + PAY + PARENTHOOD + ENTREPRENEURSHIP +
plot(tr1)
text(tr1)
```



2018

```
tr1
```

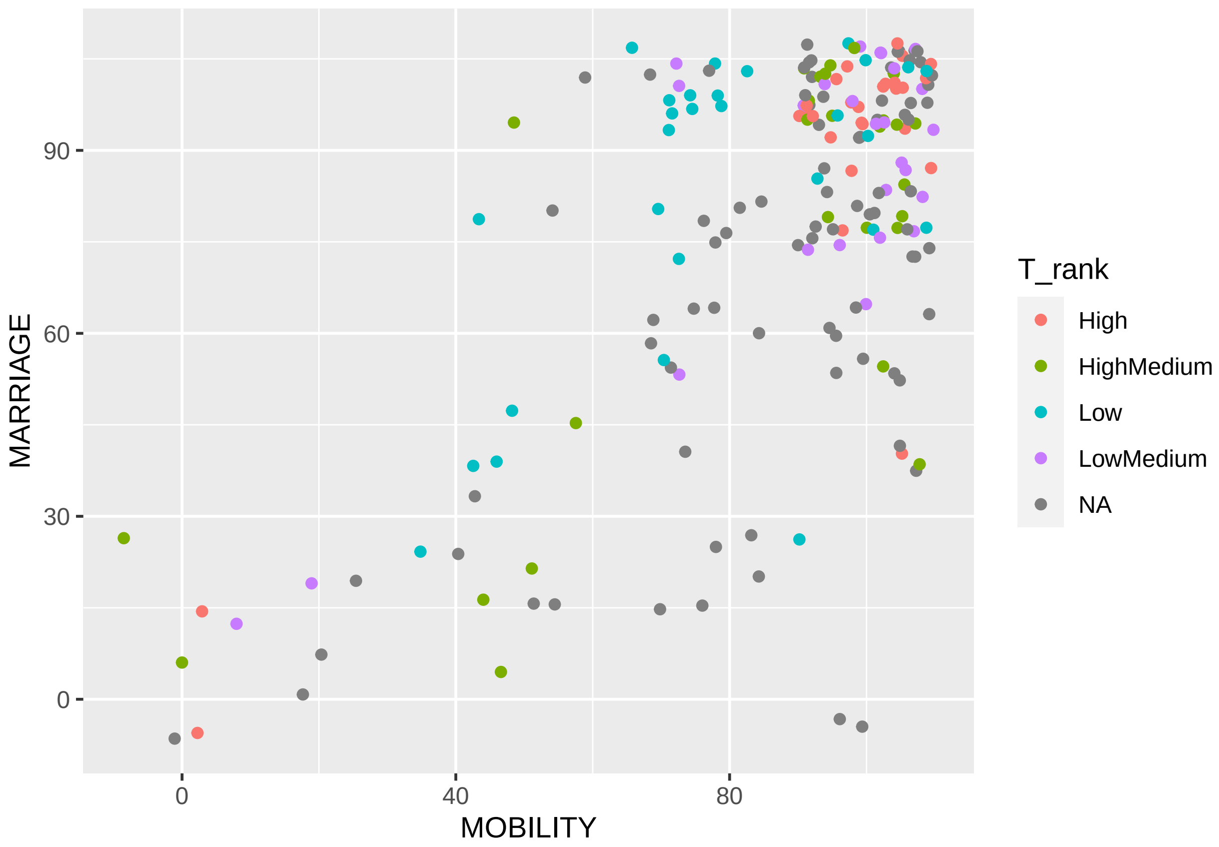
```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 108 299.400 Low ( 0.25000 0.25000 0.25926 0.24074 )
##    2) MOBILITY < 87.5 32  71.640 Low ( 0.06250 0.21875 0.56250 0.15625 )
##      4) MARRIAGE < 30 10  24.410 HighMedium ( 0.20000 0.50000 0.10000 0.20000 )
##        8) MOBILITY < 12.5 5  10.550 High ( 0.40000 0.40000 0.00000 0.20000 ) *
##        9) MOBILITY > 12.5 5   9.503 HighMedium ( 0.00000 0.60000 0.20000 0.20000 ) *
##       5) MARRIAGE > 30 22  30.310 Low ( 0.00000 0.09091 0.77273 0.13636 )
##        10) MOBILITY < 62.5 6   7.638 Low ( 0.00000 0.33333 0.66667 0.00000 ) *
##        11) MOBILITY > 62.5 16  15.440 Low ( 0.00000 0.00000 0.81250 0.18750 ) *
##     3) MOBILITY > 87.5 76 203.600 High ( 0.32895 0.26316 0.13158 0.27632 )
##       6) PARENTHOOD < 70 35  93.890 LowMedium ( 0.20000 0.17143 0.25714 0.37143 )
##        12) PENSION < 62.5 9  21.870 HighMedium ( 0.11111 0.44444 0.11111 0.33333 ) *
##        13) PENSION > 62.5 26  65.820 LowMedium ( 0.23077 0.07692 0.30769 0.38462 )
##          26) PARENTHOOD < 30 11  25.710 Low ( 0.18182 0.09091 0.54545 0.18182 )
##            52) WORKPLACE < 87.5 5   5.004 Low ( 0.00000 0.00000 0.80000 0.20000 ) *
##            53) WORKPLACE > 87.5 6  15.960 High ( 0.33333 0.16667 0.33333 0.16667 ) *
##          27) PARENTHOOD > 30 15  34.110 LowMedium ( 0.26667 0.06667 0.13333 0.53333 )
##            54) PENSION < 87.5 9  23.590 High ( 0.33333 0.11111 0.22222 0.33333 ) *
##            55) PENSION > 87.5 6   5.407 LowMedium ( 0.16667 0.00000 0.00000 0.83333 ) *
##       7) PARENTHOOD > 70 41  93.290 High ( 0.43902 0.34146 0.02439 0.19512 )
##        14) WORKPLACE < 87.5 6  12.140 HighMedium ( 0.33333 0.50000 0.16667 0.00000 ) *
##        15) WORKPLACE > 87.5 35  74.130 High ( 0.45714 0.31429 0.00000 0.22857 )
##          30) PAY < 87.5 13  26.320 LowMedium ( 0.38462 0.15385 0.00000 0.46154 ) *
##          31) PAY > 87.5 22  40.930 High ( 0.50000 0.40909 0.00000 0.09091 ) *
```

```
summary(tr1)
```

```
##
## Classification tree:
## tree(formula = as.factor(T_rank) ~ MOBILITY + MARRIAGE + WORKPLACE +
##     PAY + PARENTHOOD + ENTREPRENEURSHIP + ASSETS + PENSION, data = data_18)
## Variables actually used in tree construction:
## [1] "MOBILITY"   "MARRIAGE"   "PARENTHOOD" "PENSION"    "WORKPLACE"
## [6] "PAY"
## Number of terminal nodes:  12
## Residual mean deviance:  2.024 = 194.3 / 96
## Misclassification error rate: 0.4444 = 48 / 108
```

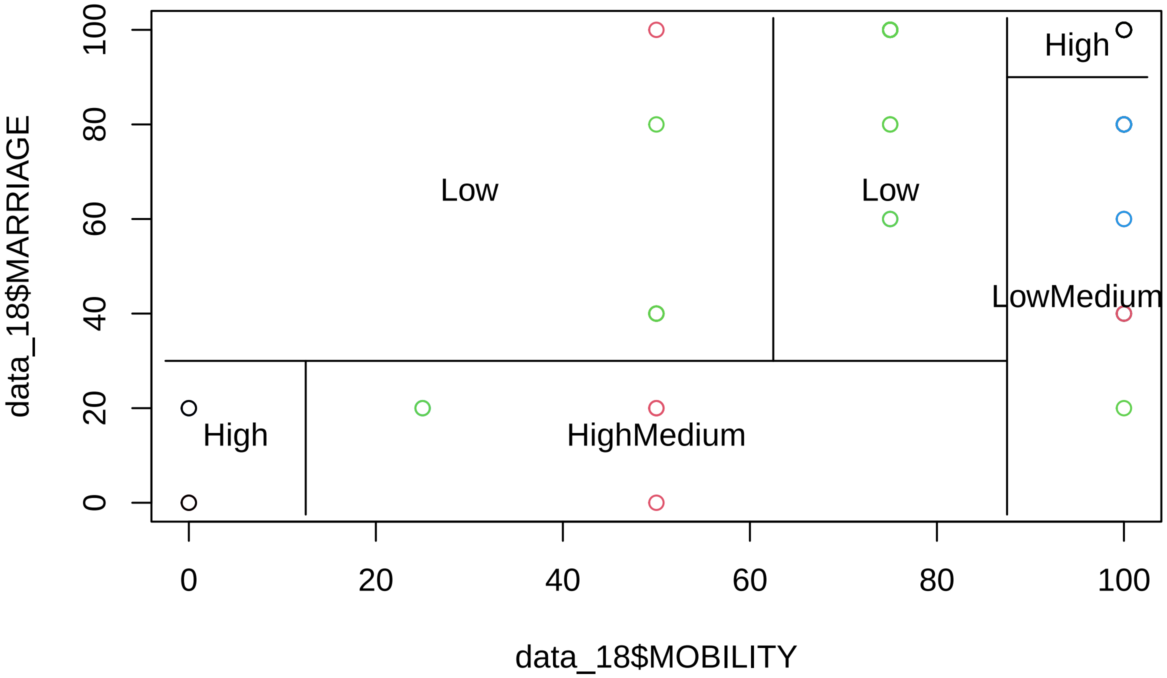
```
ggplot(data_18, aes(MOBILITY, MARRIAGE, color = T_rank)) +
  geom_jitter()
```

```
## Warning: Removed 8 rows containing missing values (‘geom_point()‘).
```

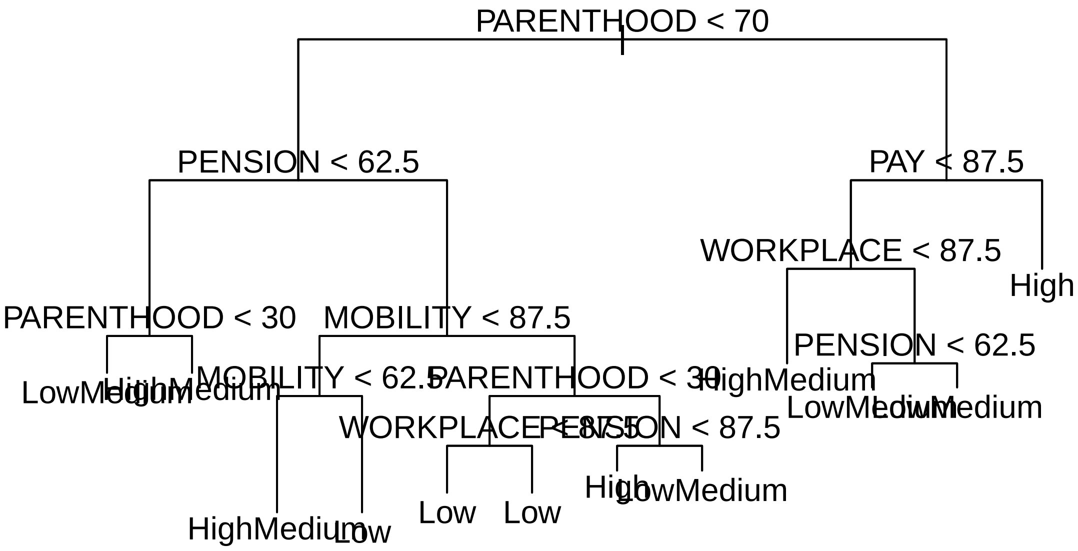


```
tr1a <- tree(as.factor(T_rank) ~MOBILITY + MARRIAGE, data = data_18)
```

```
plot(data_18$MOBILITY, data_18$MARRIAGE, col = as.factor(data_18$T_rank))
partition.tree(tr1a, add = TRUE)
```



```
tr2 <- tree(as.factor(T_rank) ~MOBILITY + MARRIAGE + WORKPLACE + PAY + PARENTHOOD + ENTREPRENEURSHIP +
plot(tr2)
text(tr2)
```



2019

```
tr2
```

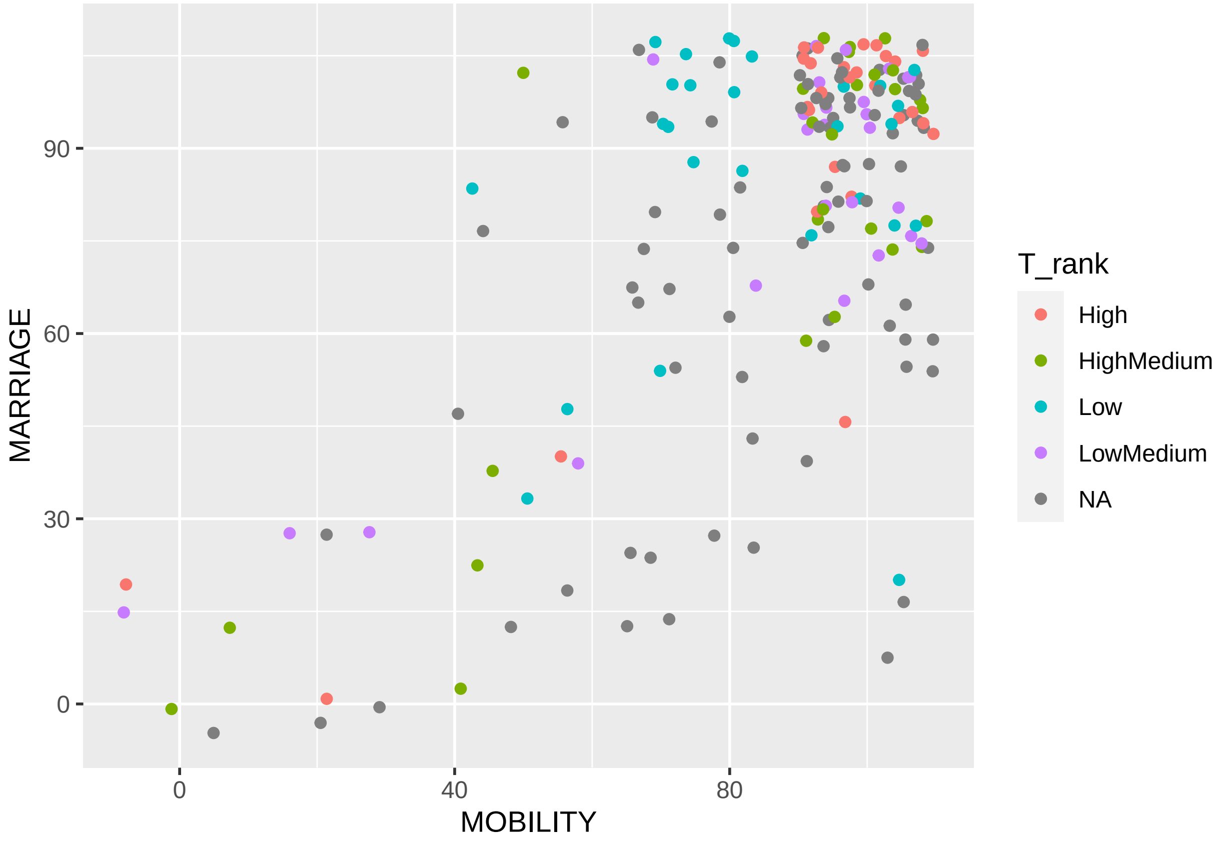
```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 107 296.600 High ( 0.25234 0.25234 0.25234 0.24299 )
##    2) PARENTHOOD < 70 61 161.400 Low ( 0.13115 0.22951 0.37705 0.26230 )
##      4) PENSION < 62.5 15  26.760 LowMedium ( 0.06667 0.46667 0.00000 0.46667 )
##        8) PARENTHOOD < 30 6  10.410 LowMedium ( 0.16667 0.16667 0.00000 0.66667 ) *
##        9) PARENTHOOD > 30 9  11.460 HighMedium ( 0.00000 0.66667 0.00000 0.33333 ) *
##      5) PENSION > 62.5 46 114.000 Low ( 0.15217 0.15217 0.50000 0.19565 )
##       10) MOBILITY < 87.5 20  36.570 Low ( 0.10000 0.15000 0.70000 0.05000 )
##         20) MOBILITY < 62.5 8  21.130 HighMedium ( 0.25000 0.37500 0.25000 0.12500 ) *
##         21) MOBILITY > 62.5 12   0.000 Low ( 0.00000 0.00000 1.00000 0.00000 ) *
##       11) MOBILITY > 87.5 26  69.420 Low ( 0.19231 0.15385 0.34615 0.30769 )
##         22) PARENTHOOD < 30 10  21.780 Low ( 0.10000 0.20000 0.60000 0.10000 )
##           44) WORKPLACE < 87.5 5   5.004 Low ( 0.00000 0.00000 0.80000 0.20000 ) *
##           45) WORKPLACE > 87.5 5  10.550 Low ( 0.20000 0.40000 0.40000 0.00000 ) *
##         23) PARENTHOOD > 30 16  41.030 LowMedium ( 0.25000 0.12500 0.18750 0.43750 )
##           46) PENSION < 87.5 10  27.320 High ( 0.30000 0.20000 0.20000 0.30000 ) *
##           47) PENSION > 87.5 6  10.410 LowMedium ( 0.16667 0.00000 0.16667 0.66667 ) *
##    3) PARENTHOOD > 70 46 116.500 High ( 0.41304 0.28261 0.08696 0.21739 )
##      6) PAY < 87.5 24  63.820 High ( 0.33333 0.16667 0.16667 0.33333 )
##       12) WORKPLACE < 87.5 8  17.320 HighMedium ( 0.25000 0.37500 0.37500 0.00000 ) *
##       13) WORKPLACE > 87.5 16  33.950 LowMedium ( 0.37500 0.06250 0.06250 0.50000 )
##         26) PENSION < 62.5 7  13.380 LowMedium ( 0.28571 0.00000 0.14286 0.57143 ) *
##         27) PENSION > 62.5 9  17.370 LowMedium ( 0.44444 0.11111 0.00000 0.44444 ) *
##      7) PAY > 87.5 22  40.930 High ( 0.50000 0.40909 0.00000 0.09091 ) *
```

```
summary(tr2)
```

```
##
## Classification tree:
## tree(formula = as.factor(T_rank) ~ MOBILITY + MARRIAGE + WORKPLACE +
##     PAY + PARENTHOOD + ENTREPRENEURSHIP + ASSETS + PENSION, data = data_19)
## Variables actually used in tree construction:
## [1] "PARENTHOOD" "PENSION"    "MOBILITY"   "WORKPLACE"  "PAY"
## Number of terminal nodes:  12
## Residual mean deviance:  1.95 = 185.3 / 95
## Misclassification error rate: 0.4393 = 47 / 107
```

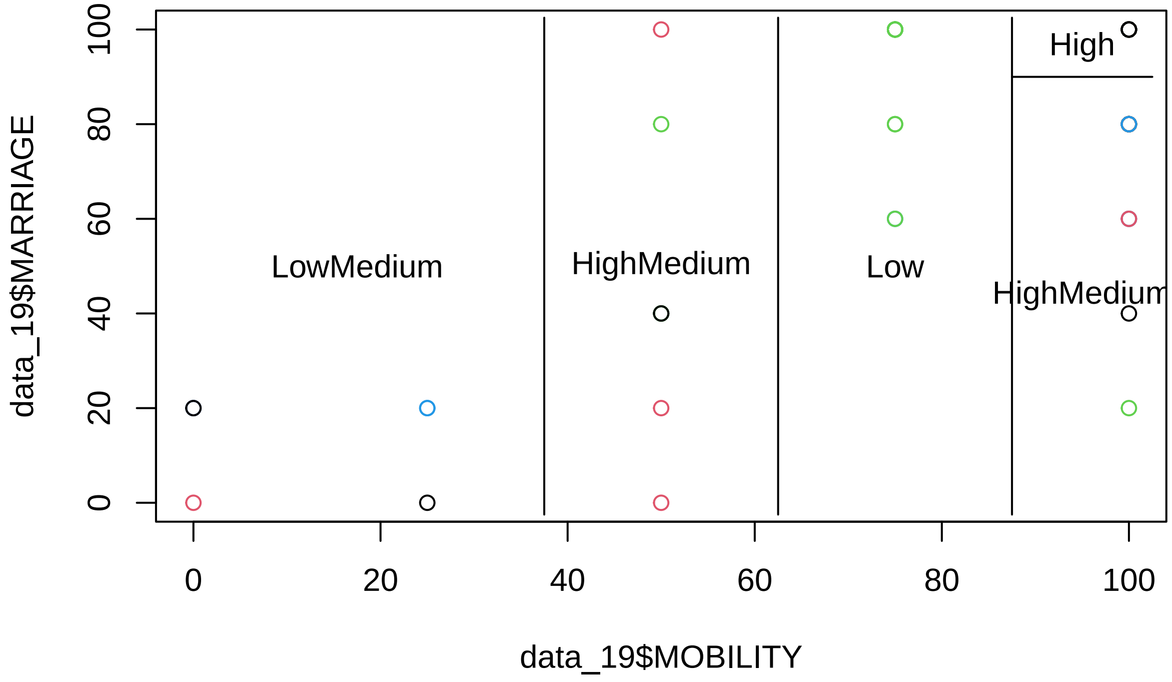
```
ggplot(data_19, aes(MOBILITY, MARRIAGE, color = T_rank)) +
  geom_jitter()
```

```
## Warning: Removed 8 rows containing missing values (‘geom_point()‘).
```

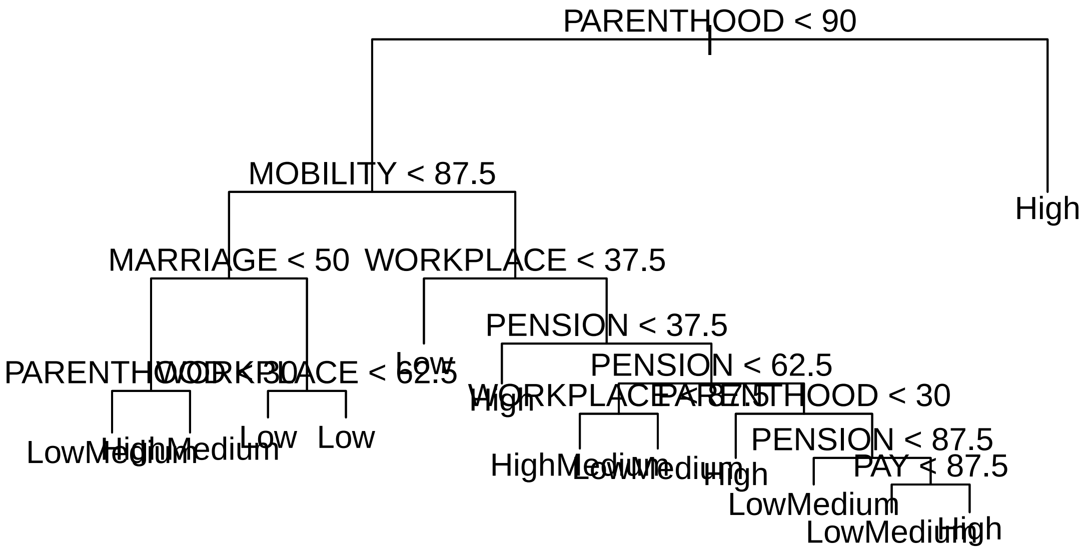


```
tr2a <- tree(as.factor(T_rank) ~MOBILITY + MARRIAGE, data = data_19)
```

```
plot(data_19$MOBILITY, data_19$MARRIAGE, col = as.factor(data_19$T_rank))
partition.tree(tr2a, add = TRUE)
```



```
tr3 <- tree(as.factor(T_rank) ~MOBILITY + MARRIAGE + WORKPLACE + PAY + PARENTHOOD + ENTREPRENEURSHIP + A
plot(tr3)
text(tr3)
```



2020

```
tr3
```

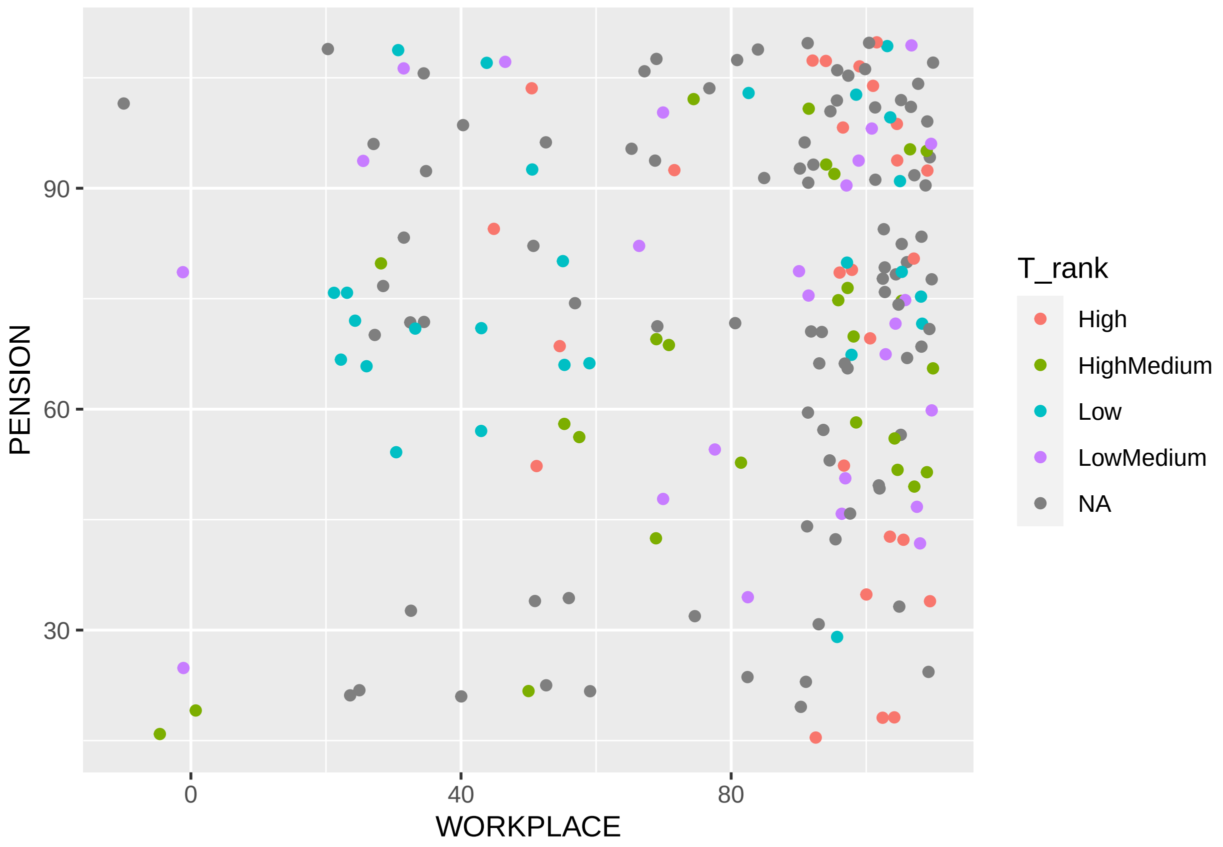
```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##   1) root 103 285.500 High ( 0.25243 0.25243 0.25243 0.24272 )
##     2) PARENTHOOD < 90 86 235.100 Low ( 0.17442 0.24419 0.29070 0.29070 )
##       4) MOBILITY < 87.5 28  64.260 Low ( 0.07143 0.14286 0.53571 0.25000 )
##         8) MARRIAGE < 50 12  31.910 HighMedium ( 0.16667 0.33333 0.16667 0.33333 )
##          16) PARENTHOOD < 30 7  15.110 LowMedium ( 0.28571 0.28571 0.00000 0.42857 ) *
##          17) PARENTHOOD > 30 5  10.550 HighMedium ( 0.00000 0.40000 0.40000 0.20000 ) *
##         9) MARRIAGE > 50 16  15.440 Low ( 0.00000 0.00000 0.81250 0.18750 )
##          18) WORKPLACE < 62.5 7   0.000 Low ( 0.00000 0.00000 1.00000 0.00000 ) *
##          19) WORKPLACE > 62.5 9  11.460 Low ( 0.00000 0.00000 0.66667 0.33333 ) *
##       5) MOBILITY > 87.5 58 157.900 LowMedium ( 0.22414 0.29310 0.17241 0.31034 )
##        10) WORKPLACE < 37.5 6  10.410 Low ( 0.00000 0.16667 0.66667 0.16667 ) *
##        11) WORKPLACE > 37.5 52 137.700 LowMedium ( 0.25000 0.30769 0.11538 0.32692 )
##          22) PENSION < 37.5 5   9.503 High ( 0.60000 0.20000 0.20000 0.00000 ) *
##          23) PENSION > 37.5 47 122.200 LowMedium ( 0.21277 0.31915 0.10638 0.36170 )
##            46) PENSION < 62.5 15  32.500 HighMedium ( 0.06667 0.46667 0.06667 0.40000 )
##              92) WORKPLACE < 87.5 6  14.910 HighMedium ( 0.16667 0.50000 0.16667 0.16667 ) *
##              93) WORKPLACE > 87.5 9  12.370 LowMedium ( 0.00000 0.44444 0.00000 0.55556 ) *
##            47) PENSION > 62.5 32  85.140 LowMedium ( 0.28125 0.25000 0.12500 0.34375 )
##              94) PARENTHOOD < 30 5  10.550 High ( 0.40000 0.20000 0.40000 0.00000 ) *
##              95) PARENTHOOD > 30 27  67.960 LowMedium ( 0.25926 0.25926 0.07407 0.40741 )
##               190) PENSION < 87.5 15  39.690 LowMedium ( 0.20000 0.33333 0.13333 0.33333 ) *
##                 191) PENSION > 87.5 12  24.270 LowMedium ( 0.33333 0.16667 0.00000 0.50000 )
##                   382) PAY < 87.5 7  13.380 LowMedium ( 0.14286 0.28571 0.00000 0.57143 ) *
##                   383) PAY > 87.5 5   6.730 High ( 0.60000 0.00000 0.00000 0.40000 ) *
##     3) PARENTHOOD > 90 17  27.480 High ( 0.64706 0.29412 0.05882 0.00000 ) *
```

```
summary(tr3)
```

```
##
## Classification tree:
## tree(formula = as.factor(T_rank) ~ MOBILITY + MARRIAGE + WORKPLACE +
##     PAY + PARENTHOOD + ENTREPRENEURSHIP + ASSETS + PENSION, data = data_20)
## Variables actually used in tree construction:
## [1] "PARENTHOOD" "MOBILITY"   "MARRIAGE"   "WORKPLACE"  "PENSION"
## [6] "PAY"
## Number of terminal nodes:  13
## Residual mean deviance:  2.024 = 182.1 / 90
## Misclassification error rate: 0.4369 = 45 / 103
```

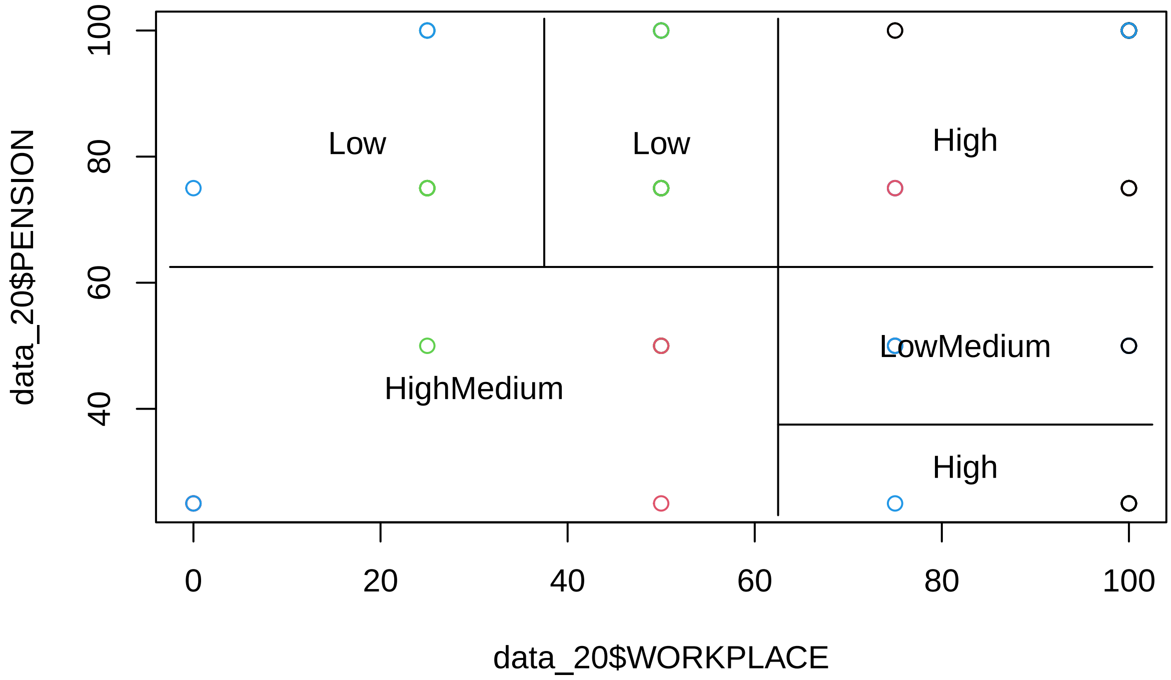
```
ggplot(data_20, aes(WORKPLACE, PENSION, color = T_rank)) +
  geom_jitter()
```

```
## Warning: Removed 8 rows containing missing values (‘geom_point()‘).
```

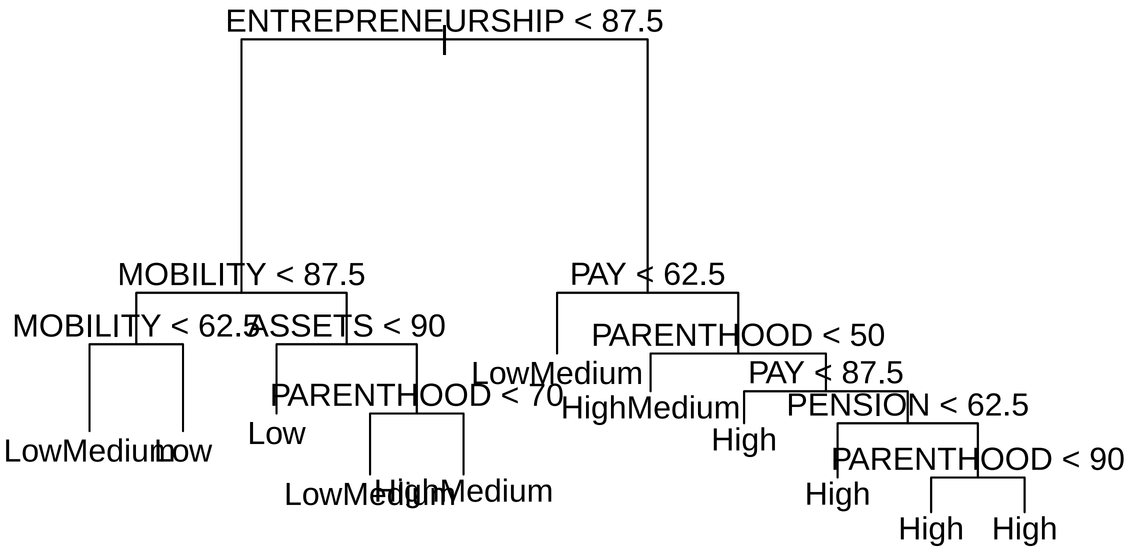


```
tr3a <- tree(as.factor(T_rank) ~WORKPLACE + PENSION, data = data_20)
```

```
plot(data_20$WORKPLACE, data_20$PENSION, col = as.factor(data_20$T_rank))
partition.tree(tr3a, add = TRUE)
```



```
tr4 <- tree(as.factor(T_rank) ~MOBILITY + MARRIAGE + WORKPLACE + PAY + PARENTHOOD + ENTREPRENEURSHIP +
plot(tr4)
text(tr4)
```



2021

```
tr4
```

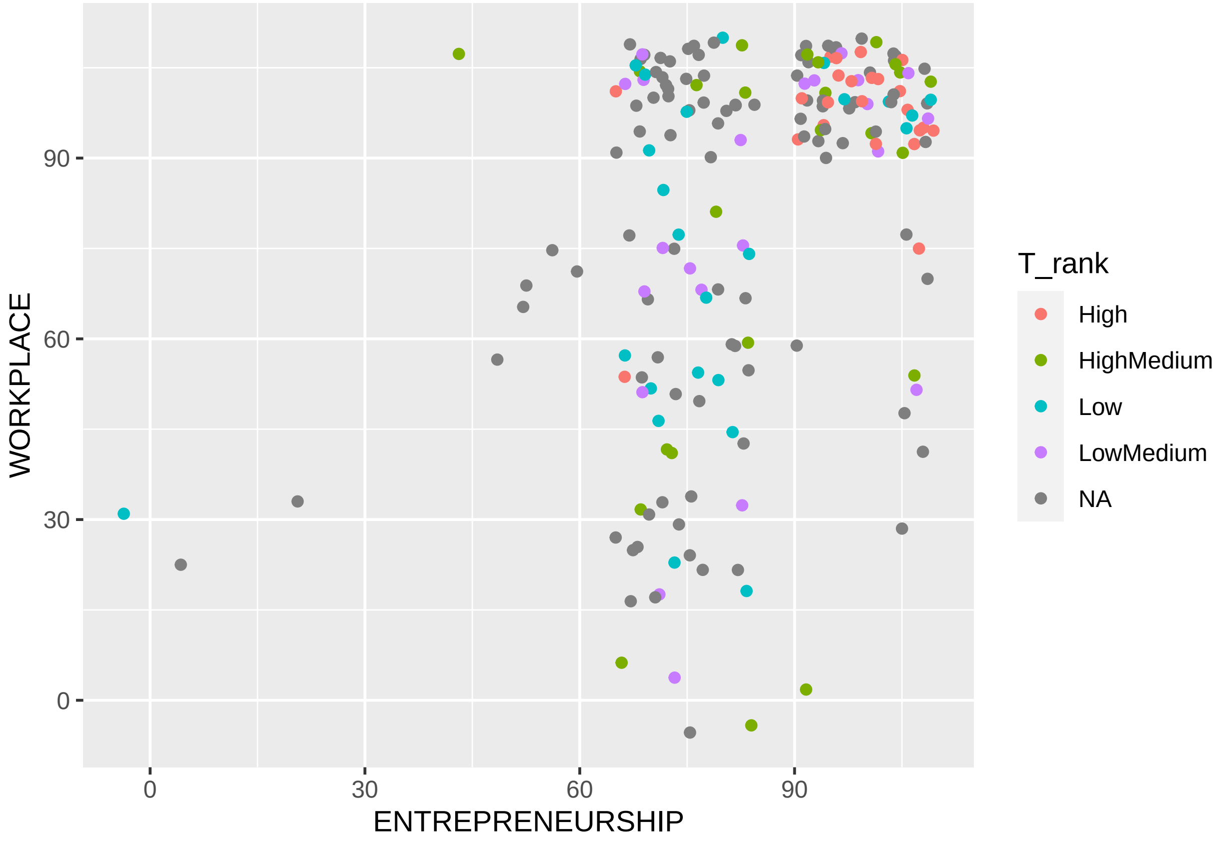
```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##   1) root 93 257.700 HighMedium ( 0.24731 0.25806 0.25806 0.23656 )
##     2) ENTREPRENEURSHIP < 87.5 45 109.400 Low ( 0.04444 0.26667 0.40000 0.28889 )
##       4) MOBILITY < 87.5 19  37.740 Low ( 0.00000 0.15789 0.52632 0.31579 )
##         8) MOBILITY < 62.5 9  19.100 LowMedium ( 0.00000 0.33333 0.22222 0.44444 ) *
##         9) MOBILITY > 62.5 10  10.010 Low ( 0.00000 0.00000 0.80000 0.20000 ) *
##       5) MOBILITY > 87.5 26  66.580 HighMedium ( 0.07692 0.34615 0.30769 0.26923 )
##        10) ASSETS < 90 7  13.380 Low ( 0.14286 0.28571 0.57143 0.00000 ) *
##        11) ASSETS > 90 19  46.310 LowMedium ( 0.05263 0.36842 0.21053 0.36842 )
##          22) PARENTHOOD < 70 11  20.160 LowMedium ( 0.00000 0.36364 0.09091 0.54545 ) *
##          23) PARENTHOOD > 70 8  20.090 HighMedium ( 0.12500 0.37500 0.37500 0.12500 ) *
##     3) ENTREPRENEURSHIP > 87.5 48 123.100 High ( 0.43750 0.25000 0.12500 0.18750 )
##       6) PAY < 62.5 10  21.780 LowMedium ( 0.30000 0.30000 0.00000 0.40000 ) *
##       7) PAY > 62.5 38  95.260 High ( 0.47368 0.23684 0.15789 0.13158 )
##        14) PARENTHOOD < 50 6  15.960 HighMedium ( 0.16667 0.33333 0.16667 0.33333 ) *
##        15) PARENTHOOD > 50 32  75.550 High ( 0.53125 0.21875 0.15625 0.09375 )
##          30) PAY < 87.5 9  19.100 High ( 0.44444 0.33333 0.22222 0.00000 ) *
##          31) PAY > 87.5 23  53.270 High ( 0.56522 0.17391 0.13043 0.13043 )
##            62) PENSION < 62.5 7   8.376 High ( 0.71429 0.28571 0.00000 0.00000 ) *
##            63) PENSION > 62.5 16  39.500 High ( 0.50000 0.12500 0.18750 0.18750 )
##             126) PARENTHOOD < 90 8  16.640 High ( 0.50000 0.00000 0.25000 0.25000 ) *
##             127) PARENTHOOD > 90 8  19.410 High ( 0.50000 0.25000 0.12500 0.12500 ) *
```

```
summary(tr4)
```

```
##
## Classification tree:
## tree(formula = as.factor(T_rank) ~ MOBILITY + MARRIAGE + WORKPLACE +
##     PAY + PARENTHOOD + ENTREPRENEURSHIP + ASSETS + PENSION, data = data_21)
## Variables actually used in tree construction:
## [1] "ENTREPRENEURSHIP" "MOBILITY"         "ASSETS"           "PARENTHOOD"
## [5] "PAY"              "PENSION"
## Number of terminal nodes:  11
## Residual mean deviance:  2.244 = 184 / 82
## Misclassification error rate: 0.4839 = 45 / 93
```

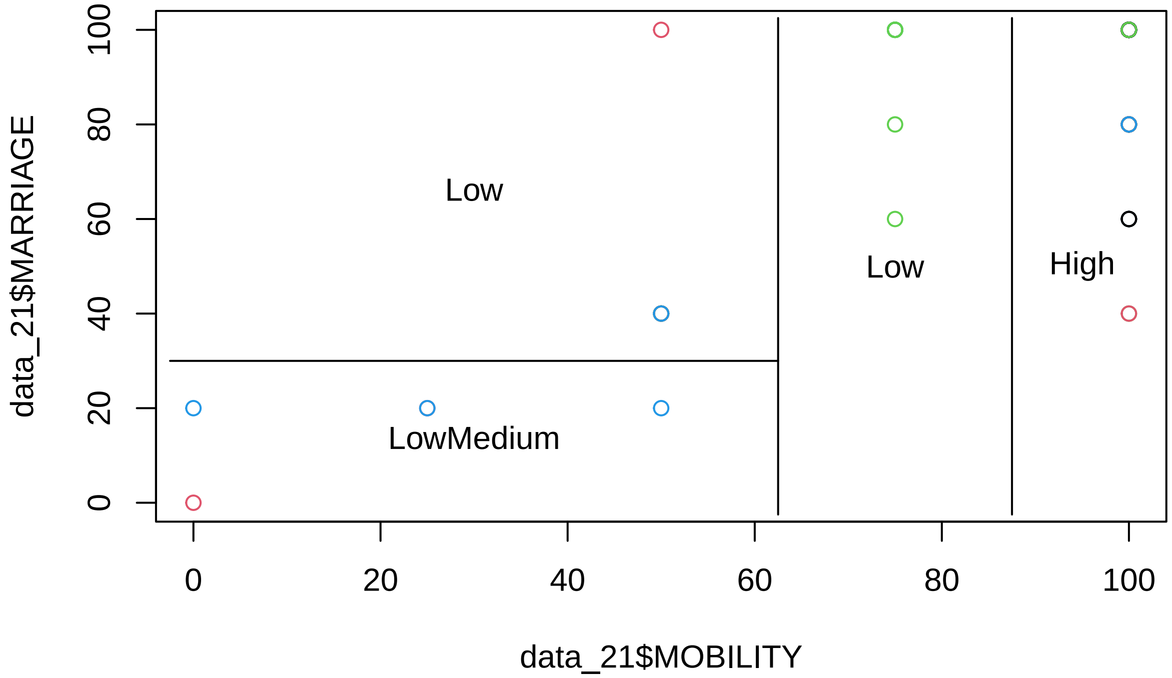
```
ggplot(data_21, aes(ENTREPRENEURSHIP, WORKPLACE, color = T_rank)) +
  geom_jitter()
```

```
## Warning: Removed 8 rows containing missing values (‘geom_point()‘).
```



```
tr4a <- tree(as.factor(T_rank) ~MOBILITY + MARRIAGE, data = data_21)
```

```
plot(data_21$MOBILITY, data_21$MARRIAGE, col = as.factor(data_21$T_rank))
partition.tree(tr4a, add = TRUE)
```



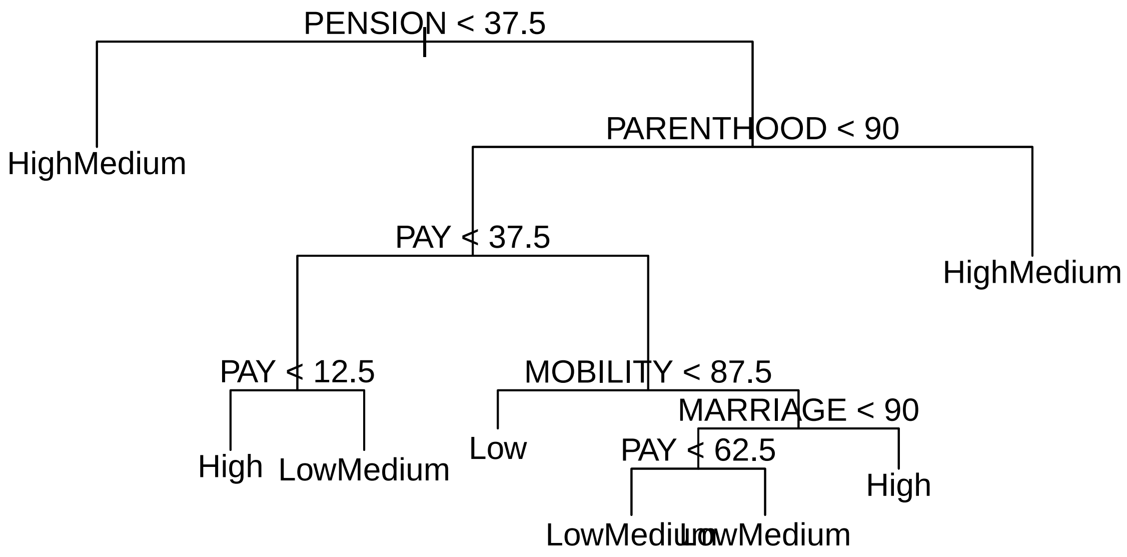
Cross- Validation of Decisison Tree

```
set.seed(123)
Z <-  sample(nrow(data_18), nrow(data_18)/2)
tr <- tree(as.factor(T_rank) ~ MOBILITY + MARRIAGE + WORKPLACE + PAY + PARENTHOOD + ENTREPRENEURSHIP +
tr
```

2018

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##   1) root 52 142.400 LowMedium ( 0.26923 0.26923 0.17308 0.28846 )
##     2) PENSION < 37.5 8  10.590 HighMedium ( 0.37500 0.62500 0.00000 0.00000 ) *
##     3) PENSION > 37.5 44 119.900 LowMedium ( 0.25000 0.20455 0.20455 0.34091 )
##       6) PARENTHOOD < 90 38 100.000 LowMedium ( 0.23684 0.13158 0.23684 0.39474 )
##        12) PAY < 37.5 13  33.960 HighMedium ( 0.30769 0.38462 0.15385 0.15385 )
##          24) PAY < 12.5 6   8.318 High ( 0.50000 0.50000 0.00000 0.00000 ) *
##          25) PAY > 12.5 7  18.920 LowMedium ( 0.14286 0.28571 0.28571 0.28571 ) *
##        13) PAY > 37.5 25  50.920 LowMedium ( 0.20000 0.00000 0.28000 0.52000 )
##          26) MOBILITY < 87.5 5   6.730 Low ( 0.00000 0.00000 0.60000 0.40000 ) *
##          27) MOBILITY > 87.5 20  39.890 LowMedium ( 0.25000 0.00000 0.20000 0.55000 )
##            54) MARRIAGE < 90 12  19.780 LowMedium ( 0.08333 0.00000 0.25000 0.66667 )
##              108) PAY < 62.5 5   5.004 LowMedium ( 0.20000 0.00000 0.00000 0.80000 ) *
##              109) PAY > 62.5 7   9.561 LowMedium ( 0.00000 0.00000 0.42857 0.57143 ) *
##            55) MARRIAGE > 90 8  15.590 High ( 0.50000 0.00000 0.12500 0.37500 ) *
##        7) PARENTHOOD > 90 6   7.638 HighMedium ( 0.33333 0.66667 0.00000 0.00000 ) *
```

```
plot(tr)
text(tr)
```



```
Yhat = predict(tr, newdata = data_18[-Z,])
summary(Yhat)
```

```
##       High          HighMedium          Low           LowMedium
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.1429
##  Median :0.3333   Median :0.0000   Median :0.1250   Median :0.3750
##  Mean   :0.2672   Mean   :0.1886   Mean   :0.2300   Mean   :0.3142
##  3rd Qu.:0.5000   3rd Qu.:0.3929   3rd Qu.:0.4286   3rd Qu.:0.4000
##  Max.   :0.5000   Max.   :0.6667   Max.   :0.6000   Max.   :0.8000
```

```
Yhat = predict(tr, newdata = data_18[-Z,], type = "class")
summary(Yhat)
```

```
##       High HighMedium        Low  LowMedium
##         31         25         25         18
```

```
table(Yhat, data_18$T_rank[-Z])
```

```
##
## Yhat         High HighMedium Low LowMedium
##   High          6          5   5         5
##   HighMedium    5          2   1         4
##   Low           1          2  11         1
##   LowMedium     1          4   2         1
```

```
(table(Yhat, data_18$T_rank[-Z])[1, 2] +
    table(Yhat, data_18$T_rank[-Z])[2, 1]+
   table(Yhat, data_18$T_rank[-Z])[3, 4]+
   table(Yhat, data_18$T_rank[-Z])[4, 3]) /
  sum(table(Yhat, data_18$T_rank[-Z]))
```

```
## [1] 0.2321429
```

```
mean(Yhat != data_18$T_rank[-Z])
```

```
## [1] NA
```

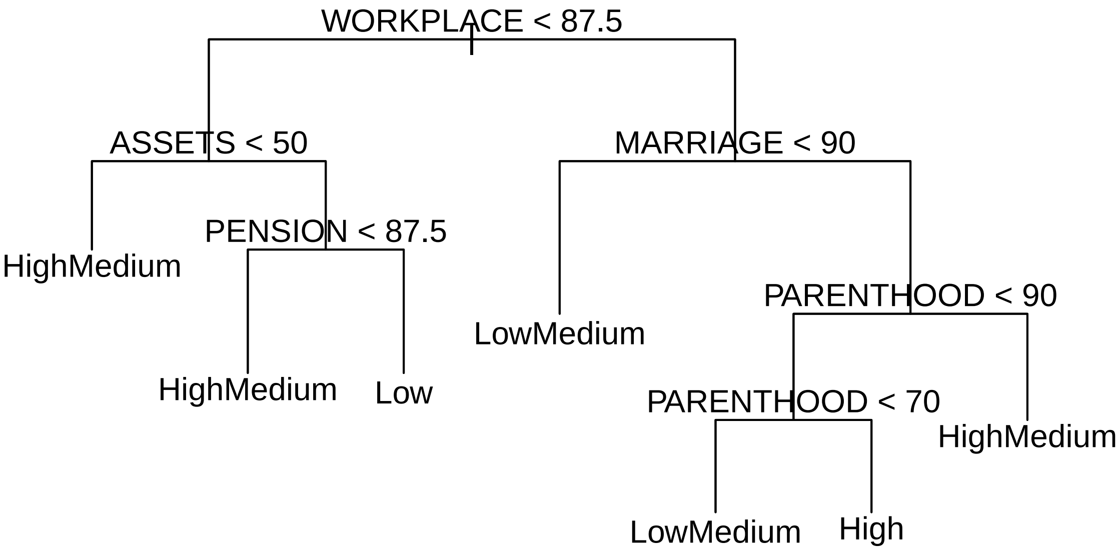
```


set.seed(123)
Z <-  sample(nrow(data_19), nrow(data_19)/2)
tr <- tree(as.factor(T_rank) ~ MOBILITY + MARRIAGE + WORKPLACE + PAY + PARENTHOOD + ENTREPRENEURSHIP +
tr
```

2019

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 51 139.200 HighMedium ( 0.27451 0.31373 0.17647 0.23529 )
##    2) WORKPLACE < 87.5 20  49.410 HighMedium ( 0.15000 0.45000 0.30000 0.10000 )
##      4) ASSETS < 50 7  11.150 HighMedium ( 0.14286 0.71429 0.00000 0.14286 ) *
##      5) ASSETS > 50 13  31.320 Low ( 0.15385 0.30769 0.46154 0.07692 )
##       10) PENSION < 87.5 8  16.640 HighMedium ( 0.25000 0.50000 0.25000 0.00000 ) *
##       11) PENSION > 87.5 5   5.004 Low ( 0.00000 0.00000 0.80000 0.20000 ) *
##    3) WORKPLACE > 87.5 31  80.270 High ( 0.35484 0.22581 0.09677 0.32258 )
##      6) MARRIAGE < 90 8  14.400 LowMedium ( 0.00000 0.12500 0.25000 0.62500 ) *
##      7) MARRIAGE > 90 23  53.880 High ( 0.47826 0.26087 0.04348 0.21739 )
##       14) PARENTHOOD < 90 17  37.910 High ( 0.52941 0.11765 0.05882 0.29412 )
##         28) PARENTHOOD < 70 8  21.130 LowMedium ( 0.25000 0.25000 0.12500 0.37500 ) *
##         29) PARENTHOOD > 70 9   9.535 High ( 0.77778 0.00000 0.00000 0.22222 ) *
##       15) PARENTHOOD > 90 6   7.638 HighMedium ( 0.33333 0.66667 0.00000 0.00000 ) *
```

```
plot(tr)
text(tr)
```



```
Yhat = predict(tr, newdata = data_19[-Z,])
summary(Yhat)
```

```
##       High          HighMedium          Low           LowMedium
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.1250   1st Qu.:0.0000   1st Qu.:0.1429
##  Median :0.2500   Median :0.2500   Median :0.1250   Median :0.2000
##  Mean   :0.2426   Mean   :0.3237   Mean   :0.1793   Mean   :0.2544
##  3rd Qu.:0.2623   3rd Qu.:0.6667   3rd Qu.:0.2500   3rd Qu.:0.3750
##  Max.   :0.7778   Max.   :0.7143   Max.   :0.8000   Max.   :0.6250
```

```
Yhat = predict(tr, newdata = data_19[-Z,], type = "class")
summary(Yhat)
```

```
##       High HighMedium        Low  LowMedium
##         15         43          9         32
```

```
table(Yhat, data_19$T_rank[-Z])
```

```
##
## Yhat         High HighMedium Low LowMedium
##   High          3          2   1         4
##   HighMedium    7          4   8         6
##   Low           1          0   3         1
##   LowMedium     2          5   6         3
```

```
(table(Yhat, data_19$T_rank[-Z])[1, 2] +
    table(Yhat, data_19$T_rank[-Z])[2, 1]+
   table(Yhat, data_19$T_rank[-Z])[3, 4]+
   table(Yhat, data_19$T_rank[-Z])[4, 3]) /
  sum(table(Yhat, data_19$T_rank[-Z]))
```

```
## [1] 0.2857143
```

```
set.seed(123)
Z <-  sample(nrow(data_20), nrow(data_20)/2)
tr <- tree(as.factor(T_rank) ~ MOBILITY + MARRIAGE + WORKPLACE + PAY + PARENTHOOD + ENTREPRENEURSHIP +
tr
```

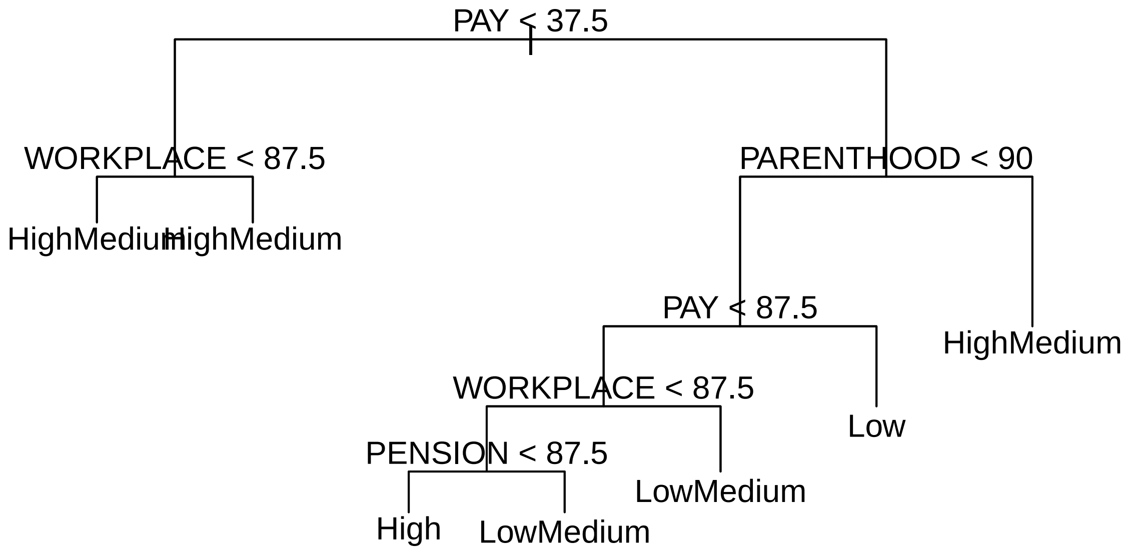
2020

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 48 128.300 High ( 0.31250 0.29167 0.12500 0.27083 )
##    2) PAY < 37.5 11  16.710 HighMedium ( 0.18182 0.72727 0.00000 0.09091 )
##      4) WORKPLACE < 87.5 6   5.407 HighMedium ( 0.00000 0.83333 0.00000 0.16667 ) *
##      5) WORKPLACE > 87.5 5   6.730 HighMedium ( 0.40000 0.60000 0.00000 0.00000 ) *
##    3) PAY > 37.5 37  97.880 High ( 0.35135 0.16216 0.16216 0.32432 )
##      6) PARENTHOOD < 90 29  71.840 LowMedium ( 0.31034 0.06897 0.20690 0.41379 )
##       12) PAY < 87.5 22  50.460 LowMedium ( 0.31818 0.09091 0.09091 0.50000 )
##         24) WORKPLACE < 87.5 11  29.530 LowMedium ( 0.27273 0.18182 0.18182 0.36364 )
##           48) PENSION < 87.5 6  15.960 High ( 0.33333 0.33333 0.16667 0.16667 ) *
##           49) PENSION > 87.5 5   9.503 LowMedium ( 0.20000 0.00000 0.20000 0.60000 ) *
##         25) WORKPLACE > 87.5 11  14.420 LowMedium ( 0.36364 0.00000 0.00000 0.63636 ) *
##       13) PAY > 87.5 7  13.380 Low ( 0.28571 0.00000 0.57143 0.14286 ) *
##      7) PARENTHOOD > 90 8  11.090 HighMedium ( 0.50000 0.50000 0.00000 0.00000 ) *
```

```
summary(tr)
```

```
##
## Classification tree:
## tree(formula = as.factor(T_rank) ~ MOBILITY + MARRIAGE + WORKPLACE +
##     PAY + PARENTHOOD + ENTREPRENEURSHIP + ASSETS + PENSION, data = data_20,
##     subset = Z)
## Variables actually used in tree construction:
## [1] "PAY"        "WORKPLACE"  "PARENTHOOD" "PENSION"
## Number of terminal nodes:  7
## Residual mean deviance:  1.866 = 76.49 / 41
## Misclassification error rate: 0.4167 = 20 / 48
```

```
plot(tr)
text(tr)
```



```
Yhat = predict(tr, newdata = data_20[-Z,])
summary(Yhat)
```

```
##       High          HighMedium          Low            LowMedium
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.2857   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.1429
##  Median :0.3333   Median :0.0000   Median :0.1250   Median :0.1667
##  Mean   :0.3133   Mean   :0.2380   Mean   :0.1798   Mean   :0.2690
##  3rd Qu.:0.3636   3rd Qu.:0.5000   3rd Qu.:0.2000   3rd Qu.:0.6000
##  Max.   :0.5000   Max.   :0.8333   Max.   :0.5714   Max.   :0.6364
```

```
Yhat = predict(tr, newdata = data_20[-Z,], type = "class")
summary(Yhat)
```

```
##       High HighMedium        Low  LowMedium
##         16         31         23         29
```

```
table(Yhat, data_20$T_rank[-Z])
```

```
##
## Yhat         High HighMedium Low LowMedium
##   High          2          1   4         2
##   HighMedium    7          3   8         3
##   Low           2          2   1         3
##   LowMedium     0          6   7         4
```

```
(table(Yhat, data_20$T_rank[-Z])[1, 2] +
    table(Yhat, data_20$T_rank[-Z])[2, 1]+
   table(Yhat, data_20$T_rank[-Z])[3, 4]+
   table(Yhat, data_20$T_rank[-Z])[4, 3]) /
  sum(table(Yhat, data_20$T_rank[-Z]))
```

```
## [1] 0.3272727
```

```


set.seed(123)
Z <-  sample(nrow(data_21), nrow(data_21)/2)
tr <- tree(as.factor(T_rank) ~ MOBILITY + MARRIAGE + WORKPLACE + PAY + PARENTHOOD + ENTREPRENEURSHIP +
tr
```

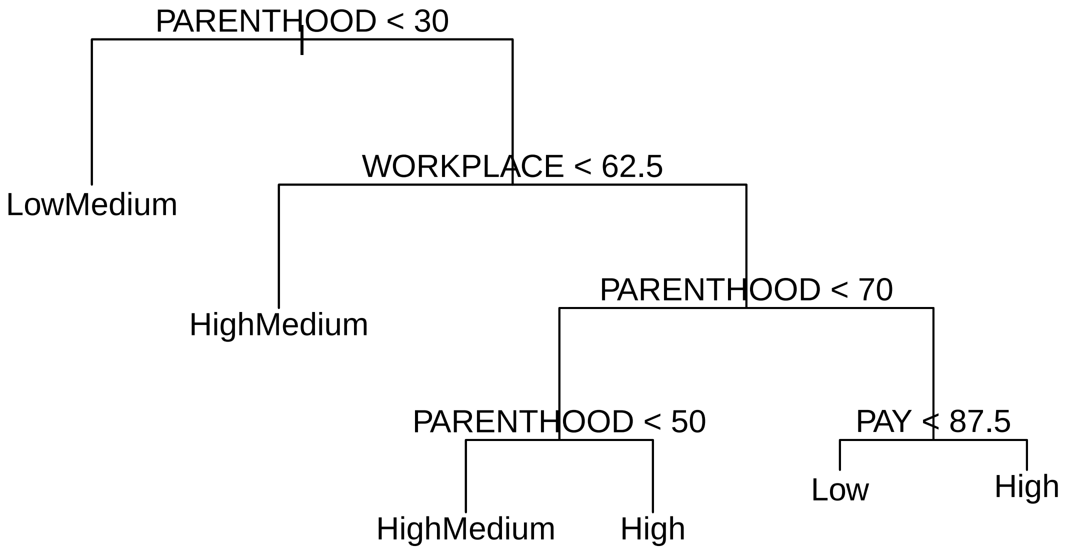
2021

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 45 121.100 HighMedium ( 0.2444 0.3556 0.1556 0.2444 )
##    2) PARENTHOOD < 30 7   9.561 LowMedium ( 0.0000 0.4286 0.0000 0.5714 ) *
##    3) PARENTHOOD > 30 38 102.500 HighMedium ( 0.2895 0.3421 0.1842 0.1842 )
##      6) WORKPLACE < 62.5 8  14.400 HighMedium ( 0.0000 0.6250 0.1250 0.2500 ) *
##      7) WORKPLACE > 62.5 30  80.450 High ( 0.3667 0.2667 0.2000 0.1667 )
##       14) PARENTHOOD < 70 15  39.690 High ( 0.3333 0.2000 0.1333 0.3333 )
##         28) PARENTHOOD < 50 8  21.130 HighMedium ( 0.2500 0.3750 0.1250 0.2500 ) *
##         29) PARENTHOOD > 50 7  14.060 High ( 0.4286 0.0000 0.1429 0.4286 ) *
##       15) PARENTHOOD > 70 15  32.560 High ( 0.4000 0.3333 0.2667 0.0000 )
##         30) PAY < 87.5 7  15.110 Low ( 0.2857 0.2857 0.4286 0.0000 ) *
##         31) PAY > 87.5 8  15.590 High ( 0.5000 0.3750 0.1250 0.0000 ) *
```

```
summary(tr)
```

```
##
## Classification tree:
## tree(formula = as.factor(T_rank) ~ MOBILITY + MARRIAGE + WORKPLACE +
##     PAY + PARENTHOOD + ENTREPRENEURSHIP + ASSETS + PENSION, data = data_21,
##     subset = Z)
## Variables actually used in tree construction:
## [1] "PARENTHOOD" "WORKPLACE"  "PAY"
## Number of terminal nodes:  6
## Residual mean deviance:  2.304 = 89.85 / 39
## Misclassification error rate: 0.5111 = 23 / 45
```

```
plot(tr)
text(tr)
```



```
Yhat = predict(tr, newdata = data_21[-Z,])
summary(Yhat)
```

```
##       High          HighMedium          Low           LowMedium
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.2857   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.2500   Median :0.3750   Median :0.1250   Median :0.2500
##  Mean   :0.2514   Mean   :0.3277   Mean   :0.1237   Mean   :0.2972
##  3rd Qu.:0.4286   3rd Qu.:0.4286   3rd Qu.:0.1429   3rd Qu.:0.5714
##  Max.   :0.5000   Max.   :0.6250   Max.   :0.4286   Max.   :0.5714
```

```
Yhat = predict(tr, newdata = data_21[-Z,], type = "class")
summary(Yhat)
```

```
##       High HighMedium        Low  LowMedium
##         28         24          9         38
```

```
table(Yhat, data_21$T_rank[-Z])
```

```
##
## Yhat         High HighMedium Low LowMedium
##   High          9          2   2         4
##   HighMedium    2          1   5         3
##   Low           1          1   1         2
##   LowMedium     0          4   9         2
```

```
(table(Yhat, data_21$T_rank[-Z])[1, 2] +
    table(Yhat, data_21$T_rank[-Z])[2, 1]+
   table(Yhat, data_21$T_rank[-Z])[3, 4]+
   table(Yhat, data_21$T_rank[-Z])[4, 3]) /
  sum(table(Yhat, data_21$T_rank[-Z]))
```

```
## [1] 0.3125
```