Lab 6 Solutions

1) `survived` is a numeric value. We need to first transform it to a categorical value. Use `titanic3\$survived = as.factor(titanic3\$survived)` to do so.

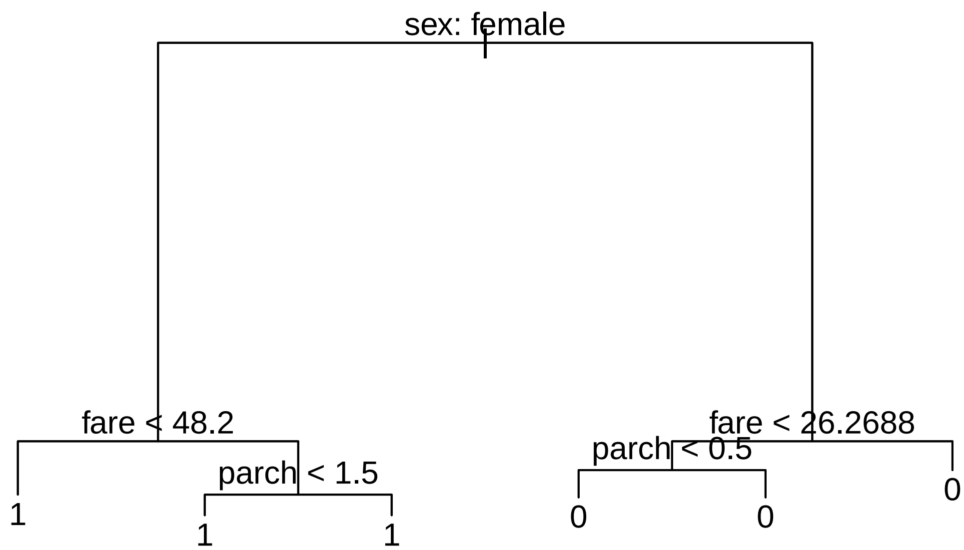
```
library(readr)
library(dplyr)
library(tree)

titanic3 <- "https://goo.gl/At238b" %>%
  read_csv %>% # read in the data
  select(survived, embarked, sex,
         sibsp, parch, fare) %>%
  mutate(embarked = factor(embarked),
         sex = factor(sex))

titanic3$survived <- as.factor(titanic3$survived)
```

2) Fit a classification tree using all the observations. Find out which variables actually contribute to building this tree. Plot the tree.

```
tree.titanic3 <- tree(survived ~ embarked+sex+sibsp+parch+fare, titanic3)
summary(tree.titanic3)
```

```
##
## Classification tree:
## tree(formula = survived ~ embarked + sex + sibsp + parch + fare,
##     data = titanic3)
## Variables actually used in tree construction:
## [1] "sex"   "fare"  "parch"
## Number of terminal nodes:  6
## Residual mean deviance:  0.9582 = 1246 / 1300
## Misclassification error rate: 0.2205 = 288 / 1306
```

```
plot(tree.titanic3)
text(tree.titanic3,pretty=0)
```



Variables actually used in tree construction: `sex`, `fare` and `parch`.

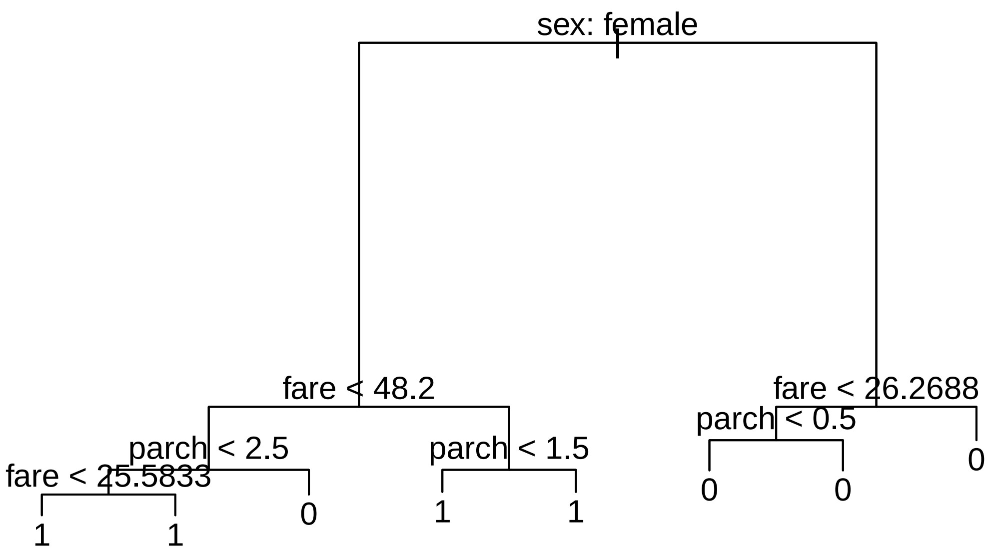
3) Now we are going to estimate the test error:

- a. Split the observations into a training set and a test set

```
set.seed(2)
train <- sample(1:nrow(titanic3), nrow(titanic3)/2)
titanic3.test <- titanic3[-train,]
survived.test <- titanic3$survived[-train]
```

- b. Build the tree using the training set, and plot the tree

```
tree.titanic3.train <- tree(survived ~ embarked+sex+sibsp+parch+fare,
                            titanic3,subset=train)

plot(tree.titanic3.train)
text(tree.titanic3.train,pretty=0)
```



- c. Evaluate its performance on the test data

```
tree.titanic3.pred <- predict(tree.titanic3.train,titanic3.test,type="class")

mean(tree.titanic3.pred!= survived.test)
```

```
## [1] 0.2229008
```

```
#Error rate is
```

Alternatively, use

```
table(tree.titanic3.pred, survived.test)
```

```
##                   survived.test
## tree.titanic3.pred   0   1
##                  0 347  85
##                  1  61 162
```

```
#Error rate is
(85+61)/(85+61+347+162)
```

```
## [1] 0.2229008
```

4) Next, let's find out whether pruning the tree might lead to improved results

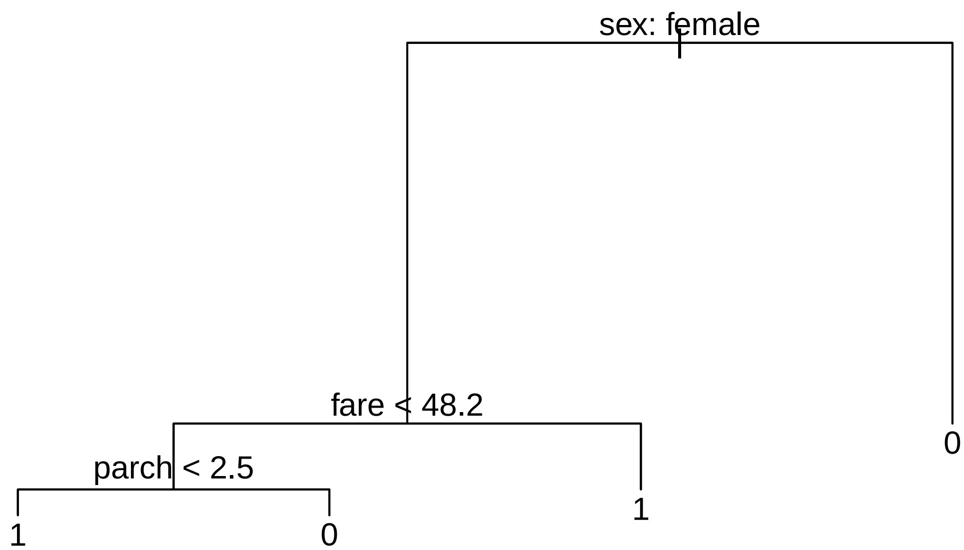
- a. Use `cv.tree()` to determine the optimal level of tree complexity

```r
set.seed(3)
cv.titanic3 <- cv.tree(tree.titanic3.train,FUN=prune.misclass)
print(cv.titanic3)
```

```
## $size
## [1] 8 4 2 1
##
## $dev
## [1] 144 144 146 251
##
## $k
## [1] -Inf    0    3  106
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

- b. According to the result, do you think pruning is necessary? Why or why not? The results show that the best tree has 8 or 4 leaves. There is no need to prune. But we can try to prune the tree to 4 leaves.
- c. If you think it is necessary, or would like to give it a try, use `prune.misclass()` to prune the tree and evaluate the performance of the pruned tree.

```r
prune.titanic3 <- prune.misclass(tree.titanic3.train,best=4)
plot(prune.titanic3)
text(prune.titanic3,pretty=0)
```




```r
tree.prune.titanic3.pred <- predict(prune.titanic3,titanic3.test,type="class")

mean(tree.prune.titanic3.pred!= survived.test)
```

```
## [1] 0.2229008
```

This error rate is the same as the tree with 8 leaves (in my case, the tree is `tree.titanic3.train`). However, considering the interpretability, the tree with 4 leaves is better.

You might have different results as mine if you set different seeds. Any reasonable answers are acceptable.