Class 16: Regression trees

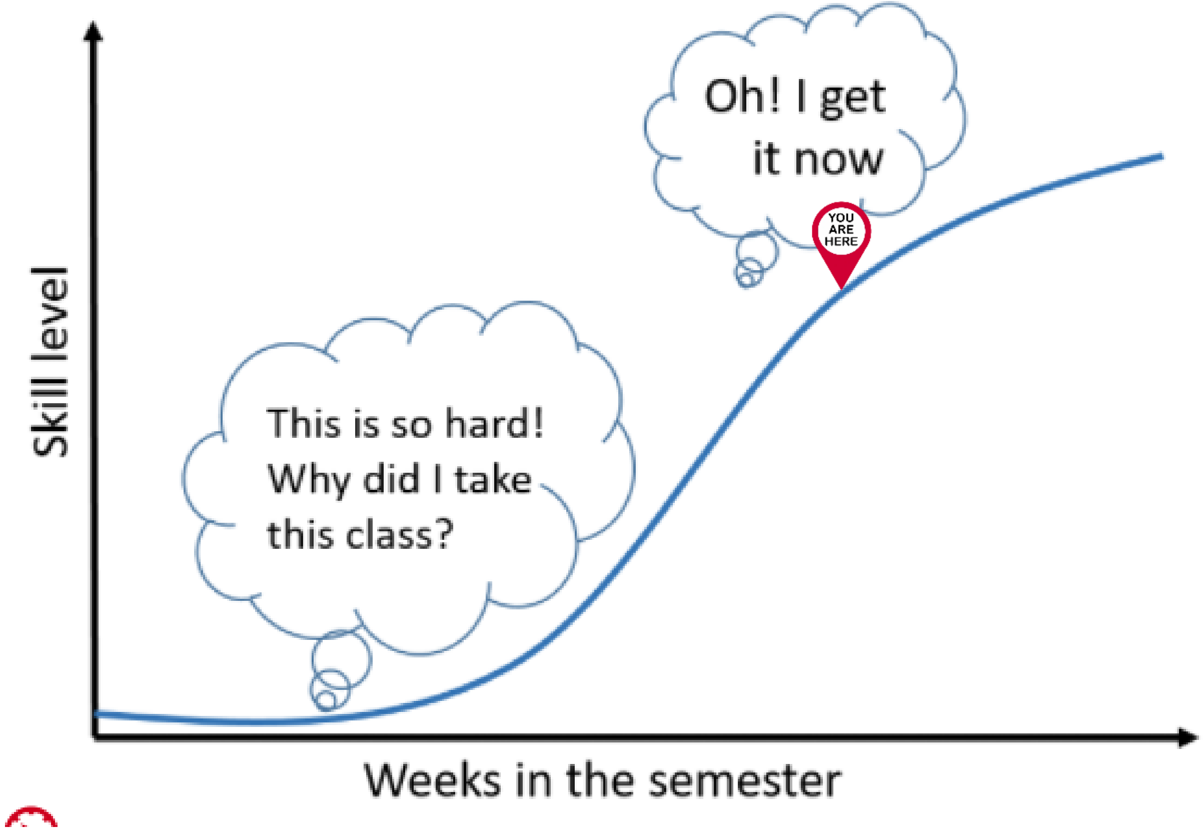
MGSC 310

Prof. Jonathan Hersh

Class 16 Announcements

1. No Quiz This Week
2. Thursday (11/5): Interview with Sumair Shah (Minnesota Twins)
3. Final project one sheet due November 19
4. Pset 5 due Tuesday, November 17 (will post later this week)
5. Midterms will grade by next week

Welcome to the Easier Part of the Course



Thursday: Sumair Shah

R&D / Player Development
Minnesota Twins

Chapman ’18

BA, Business Admin and Finance

Georgia Institute of Tech ‘19

MA, Analytics and Machine Learning

Data Analytics Association Guest Speaker

Annie Wang

JD Candidate, Yale

Formerly:

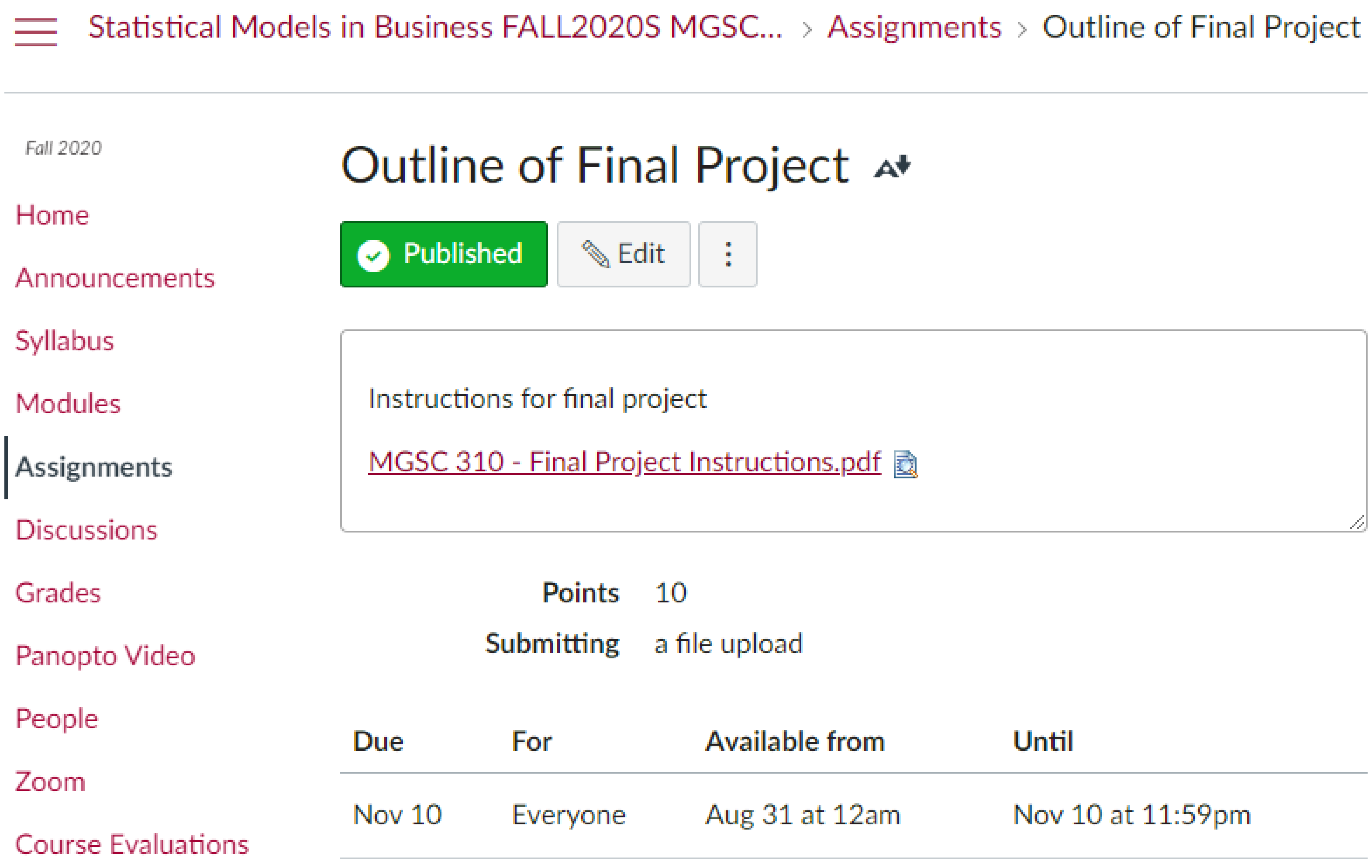
Director of Data Science, Warren for President

Director of Research and Analytics, Analyst Institute

Civis Analytics, Senior Applied Data Scientist

Tuesday, Nov 10 @ 7pm

Final Project: One-Page Outline Due Nov 10



Outline of Final Project

Instructions for final project

MGSC 310 - Final Project Instructions.pdf

Points	10
Submitting	a file upload

Due	For	Available from	Until
Nov 10	Everyone	Aug 31 at 12am	Nov 10 at 11:59pm

Final Project: One-Page Outline Due Nov 10

- **Your final project should take a real-world data set, and estimate a series of predictive models against this dataset.**
- You should identify a business use-case for the prediction, and stimate at least three predictive models we covered in this class against the dataset.

November 10th – Due: students must upload to Canvas a one-page outline of their project. This outline should include:

a) identify a dataset you will use

b) the outcome you are trying to predict, and what variables you will use to predict it

c) motivation to your project -- as in the business or practical management use case of such a prediction

d) three methods you will use to analyze your question of interest

e) the names of the students who will be part of your group (up to four)

f) Rmarkdown document showing summary statistics (mean, std dev, min max) against the baseline dataset.

Where to Find Datasets?

- Kaggle: https://www.kaggle.com/datasets
- Kaggle: https://www.kaggle.com/annavictoria/ml-friendly-public-datasets
- FiveThirtyEight https://data.fivethirtyeight.com/
- TidyTuesday: https://github.com/rfordatascience/tidytuesday
- UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets.php

How to Find Dataset and Join Groups?

- Think about the topic/industry you want to cover in your final project
  - Finance?
  - Sports Analytics?
  - Entertainment?
  - Marketing Analytics?
  - Personnel Analytics?
  - Other?

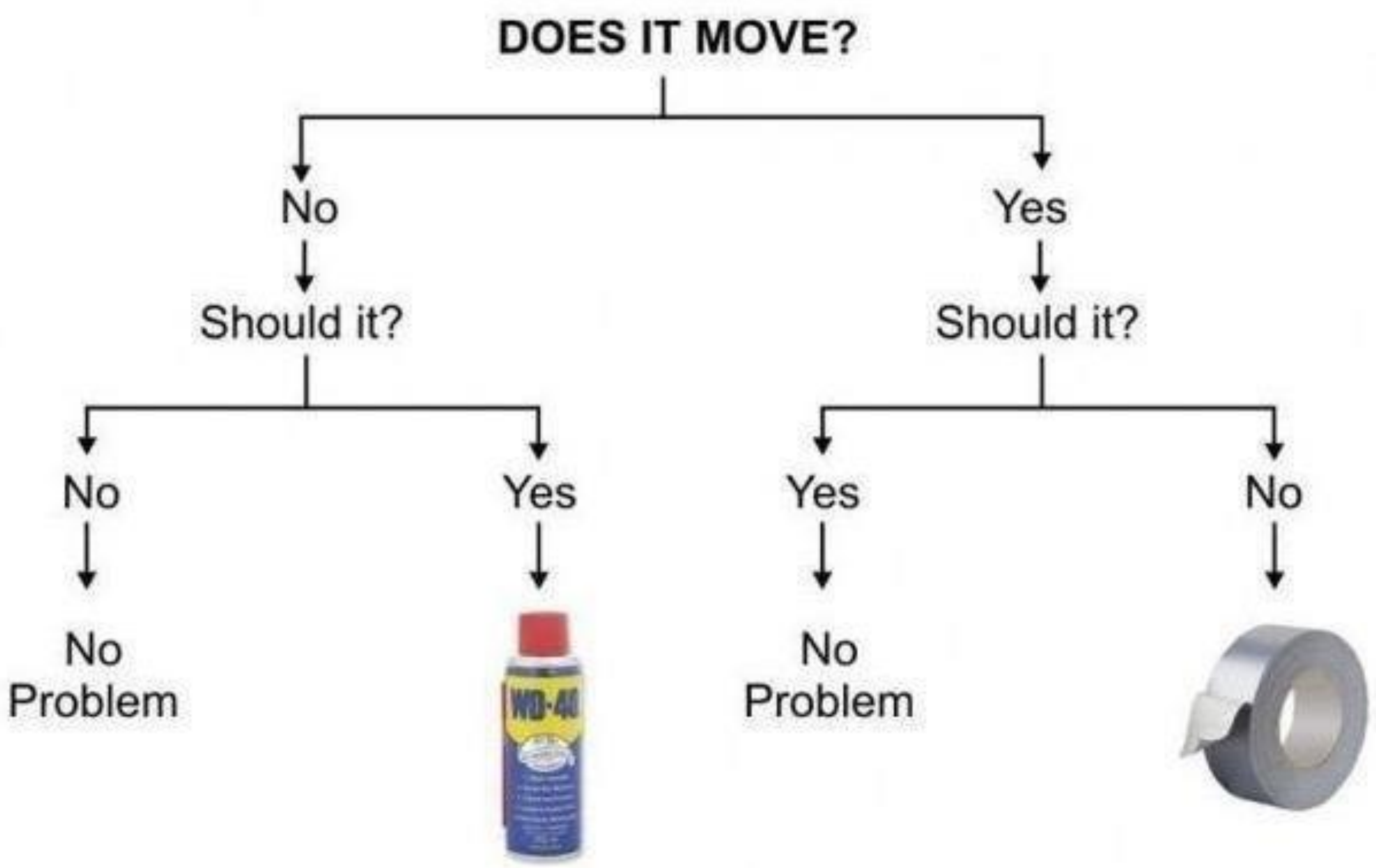
Breakout Rooms!

- Spend 5-10 minutes in a breakout rooms
- (Upgrade to latest zoom and can self-join different rooms based on topics)
- Find a group of up to four students
- Brainstorm ideas for datasets (see links at last slide)

Class 16: Outline

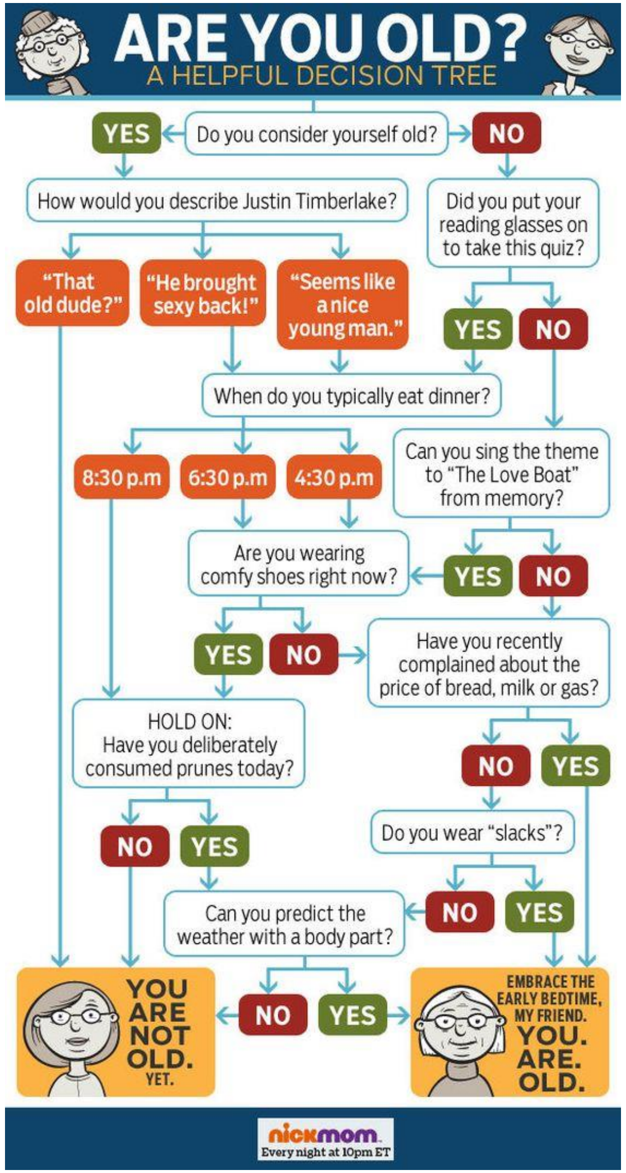
1. Regression Trees
2. Regression Trees in R
3. Regression Tree Lab (Time Permitting

What Are Binary Decision Rules?



- Binary decision rules are any rules with only two options!

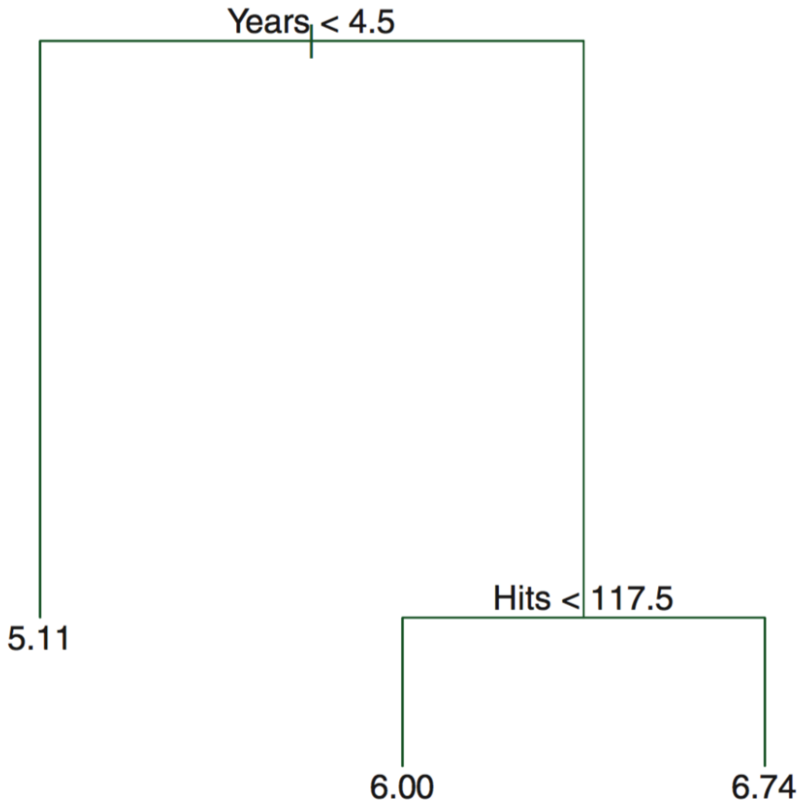
Classification Trees: Series of Binary Decision


ARE YOU OLD?
A HELPFUL DECISION TREE
Do you consider yourself old?
YES
NO
How would you describe Justin Timberlake?
Did you put your reading glasses on to take this quiz?
"That old dude?"
"He brought sexy back!"
"Seems like a nice young man."
YES
NO
When do you typically eat dinner?
8:30 p.m
6:30 p.m
4:30 p.m
Can you sing the theme to "The Love Boat" from memory?
Are you wearing comfy shoes right now?
YES
NO
YES
NO
Have you recently complained about the price of bread, milk or gas?
HOLD ON: Have you deliberately consumed prunes today?
NO
YES
NO
YES
Do you wear "slacks"?
Can you predict the weather with a body part?
NO
YES
NO
YES
YOU ARE NOT OLD. YET.
EMBRACE THE EARLY BEDTIME, MY FRIEND. YOU. ARE. OLD.
nickmom
Every night at 10pm ET

- Combining these binary decisions into a "tree" creates a decision tree
- Either classification ($y \in \{0,1\}$) or regression ($y \in [-\infty, \infty]$) problems can be modeled using a decision tree

Regression Trees

- **Tree based methods *stratify* or *segment* the predictor space into different regions**
- Regions are stratified via simple rules
- The splitting rules can be summarized into a tree that is very intuitive



Pros and Cons of Trees

Pros

- Simple
- Easy to interpret
- Easy to explain
- Can be displayed graphically!
- Bagging, boosting, and random forests very powerful (combining trees)

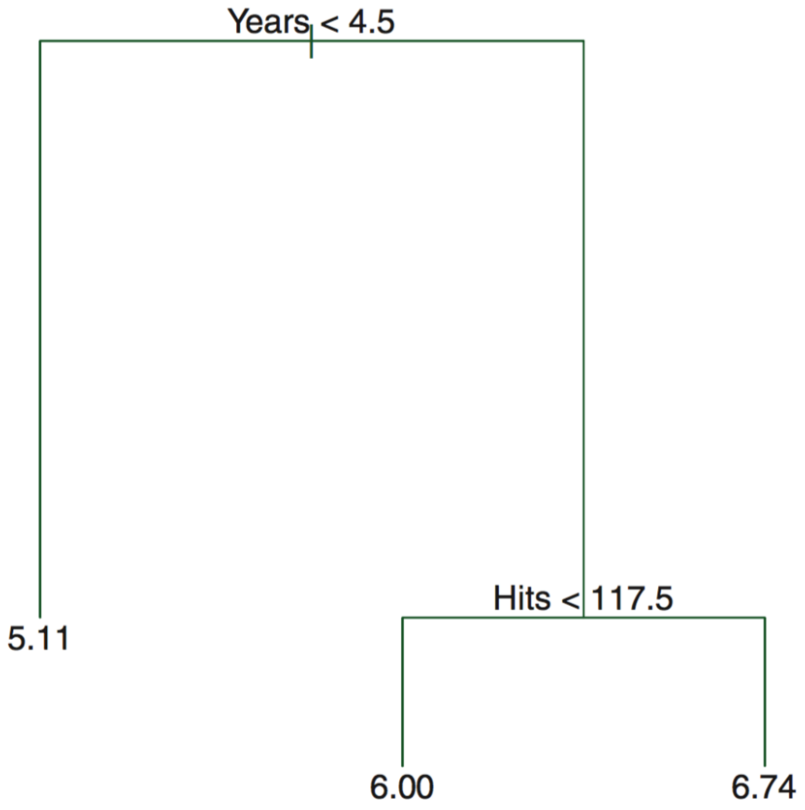
Cons

- Slow with large datasets
- Not easy to use “out of the box”
- Choice of split can be unstable

Decision/Regression Trees

- Decision trees can be applied to both regression problems ($y_i \in R$) and classification problems $y_i \in \{class1, class2, \dots, \}$

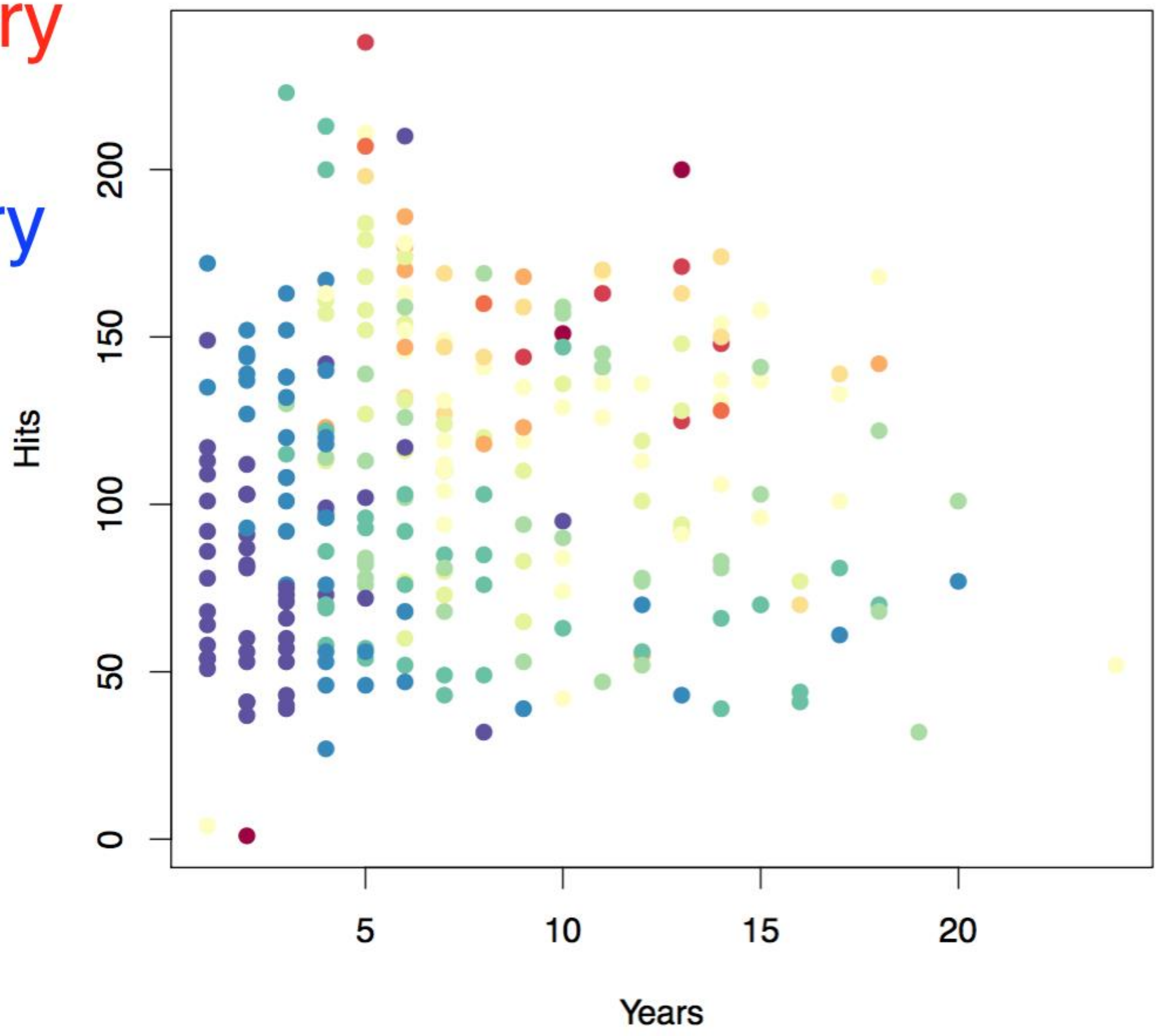
- We'll consider both



Baseball salary data: how to partition/stratify?

High salary red

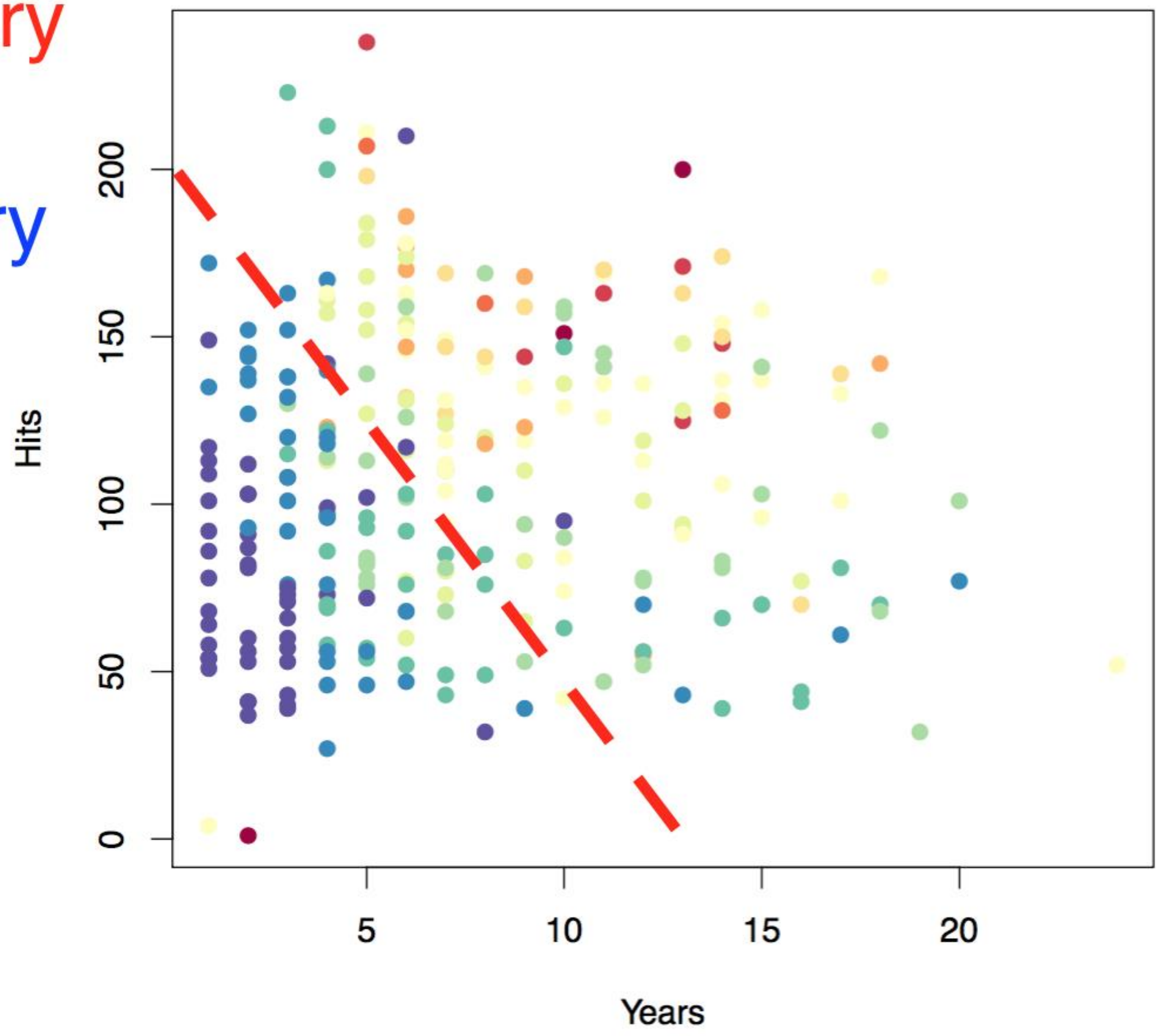
Low salary blue



Baseball salary data: how to partition/stratify?

High salary red

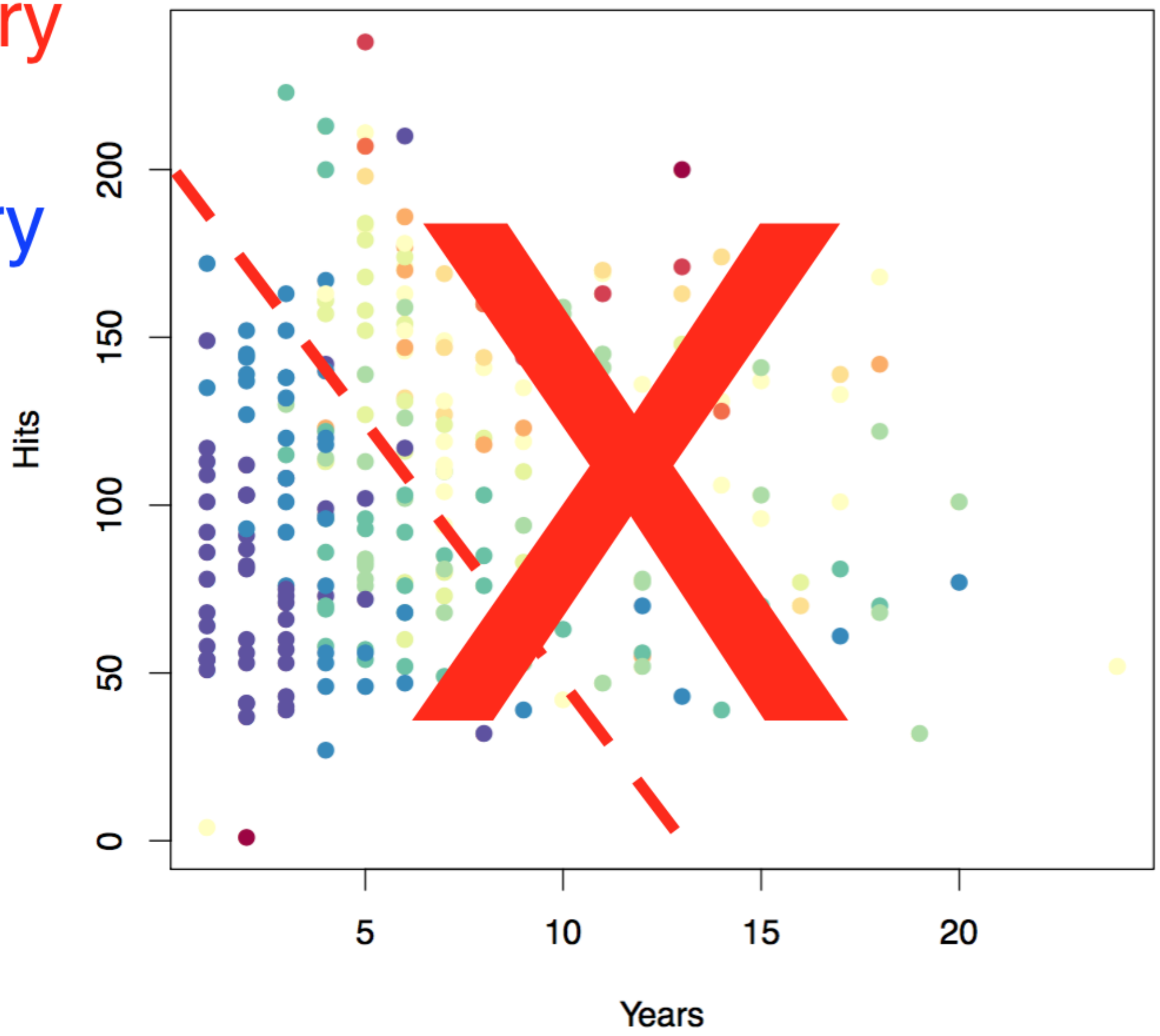
Low salary blue



Baseball salary data: how to partition/stratify?

High salary red

Low salary blue

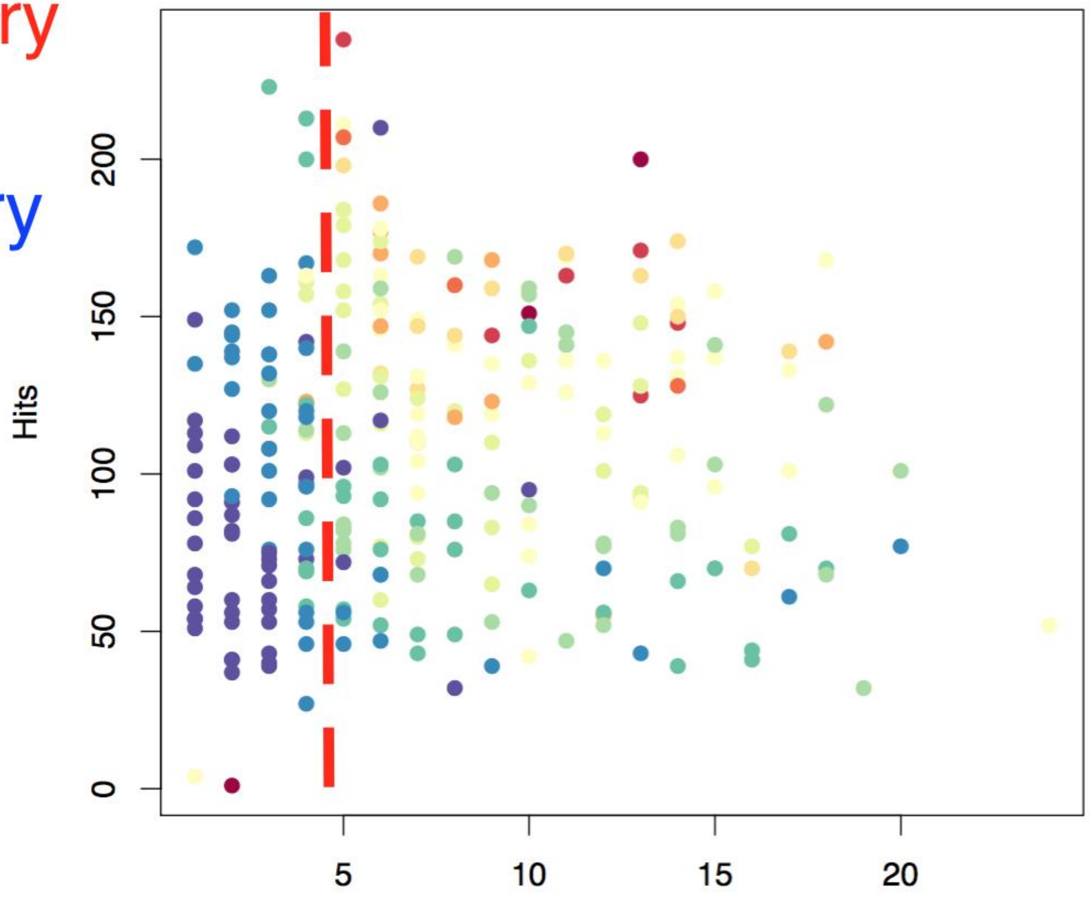


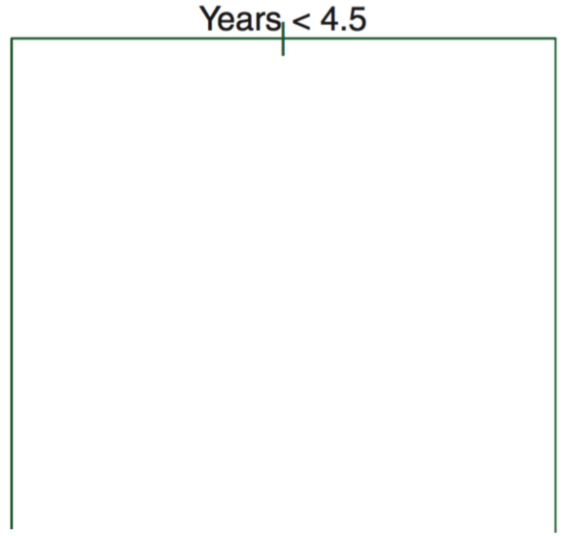
- Only linear classification rules are allowed, e.g. year > 10

How Regression Trees are Constructed

High salary
red

Low salary
blue



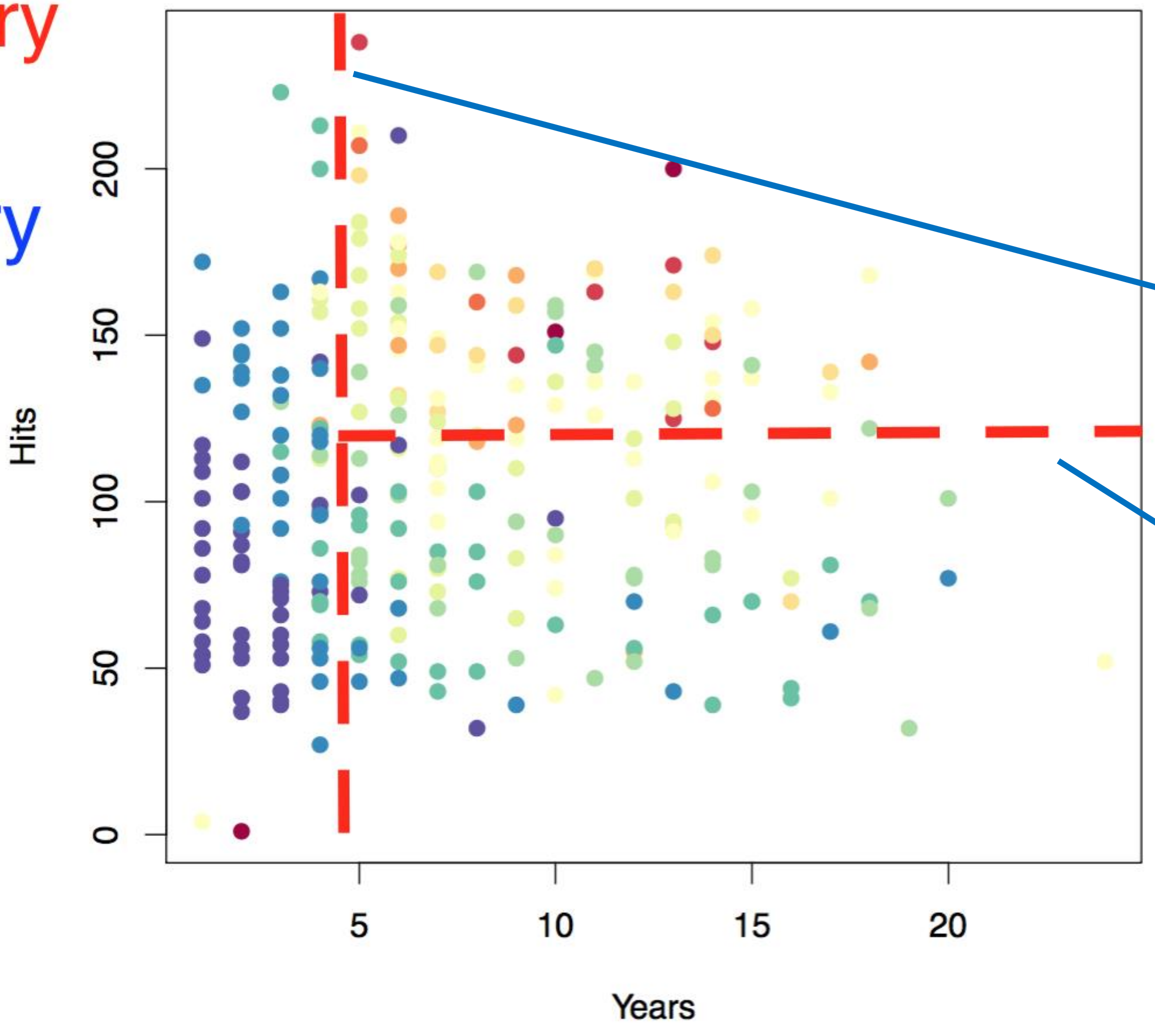


- Every variable and every split is consider.
- Chosen split is one which maximizes separation of high and low salaries:
- Split 1: years > 4.5

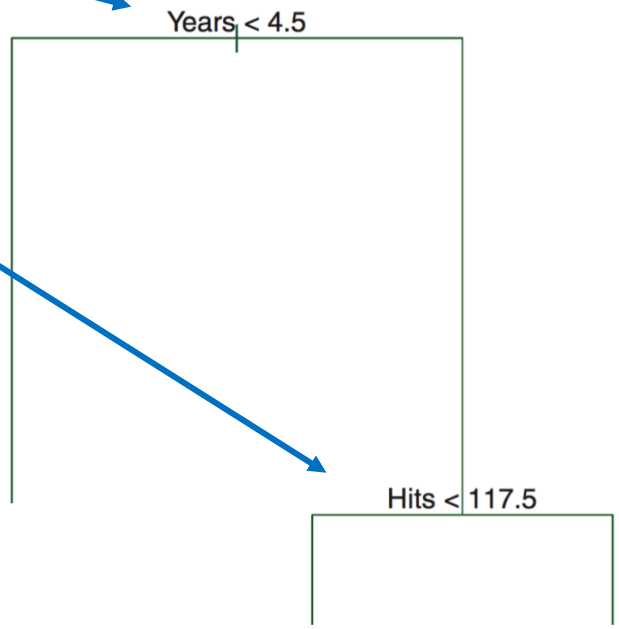
Baseball salary data: split 2

High salary red

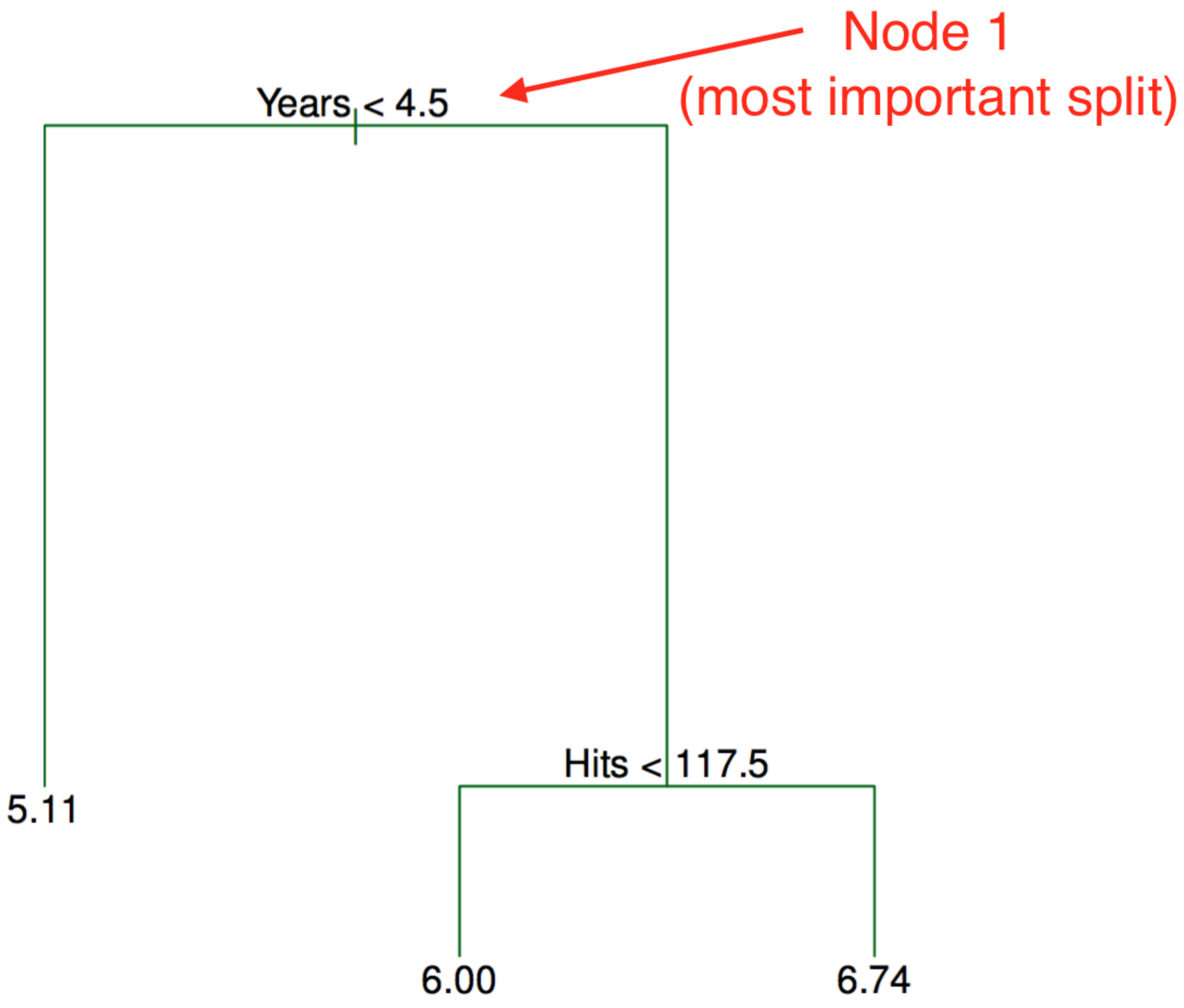
Low salary blue



- Split 1: years > 4.5
- Split 2: hits > 117.5

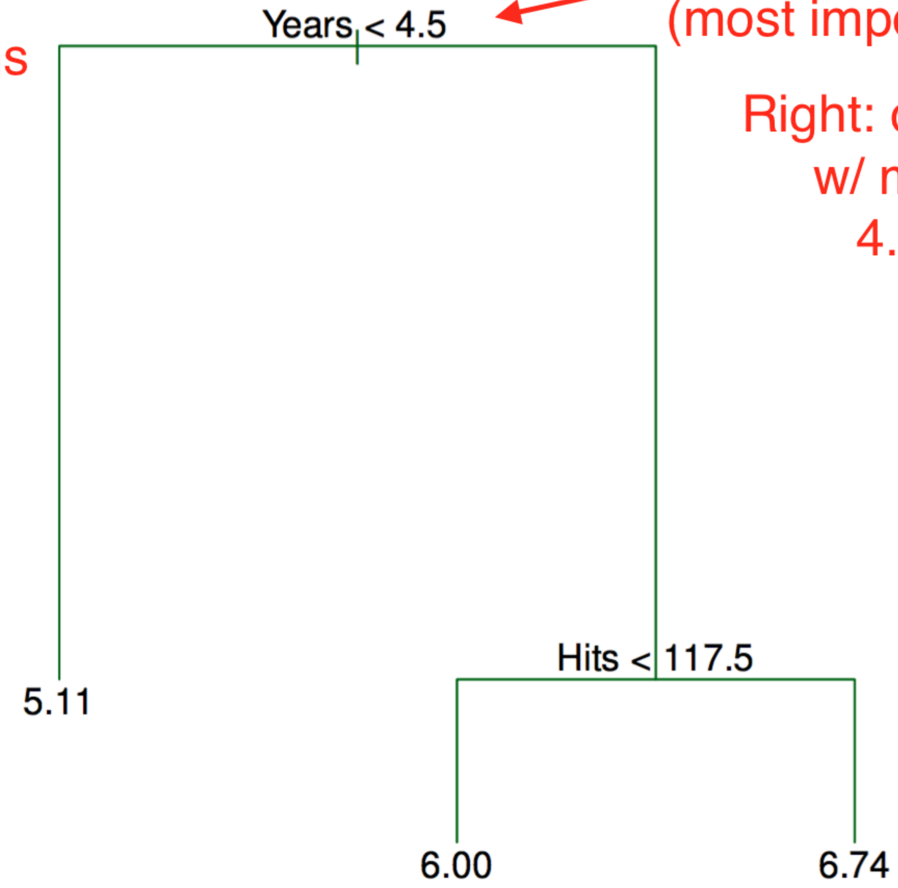


Tree Representation



- Trees are read top-down
- Most important split is at top
- Length represents how much within-cluster variance decreases from split
- So Years explains more variance than Hits for this tree

Tree Representation

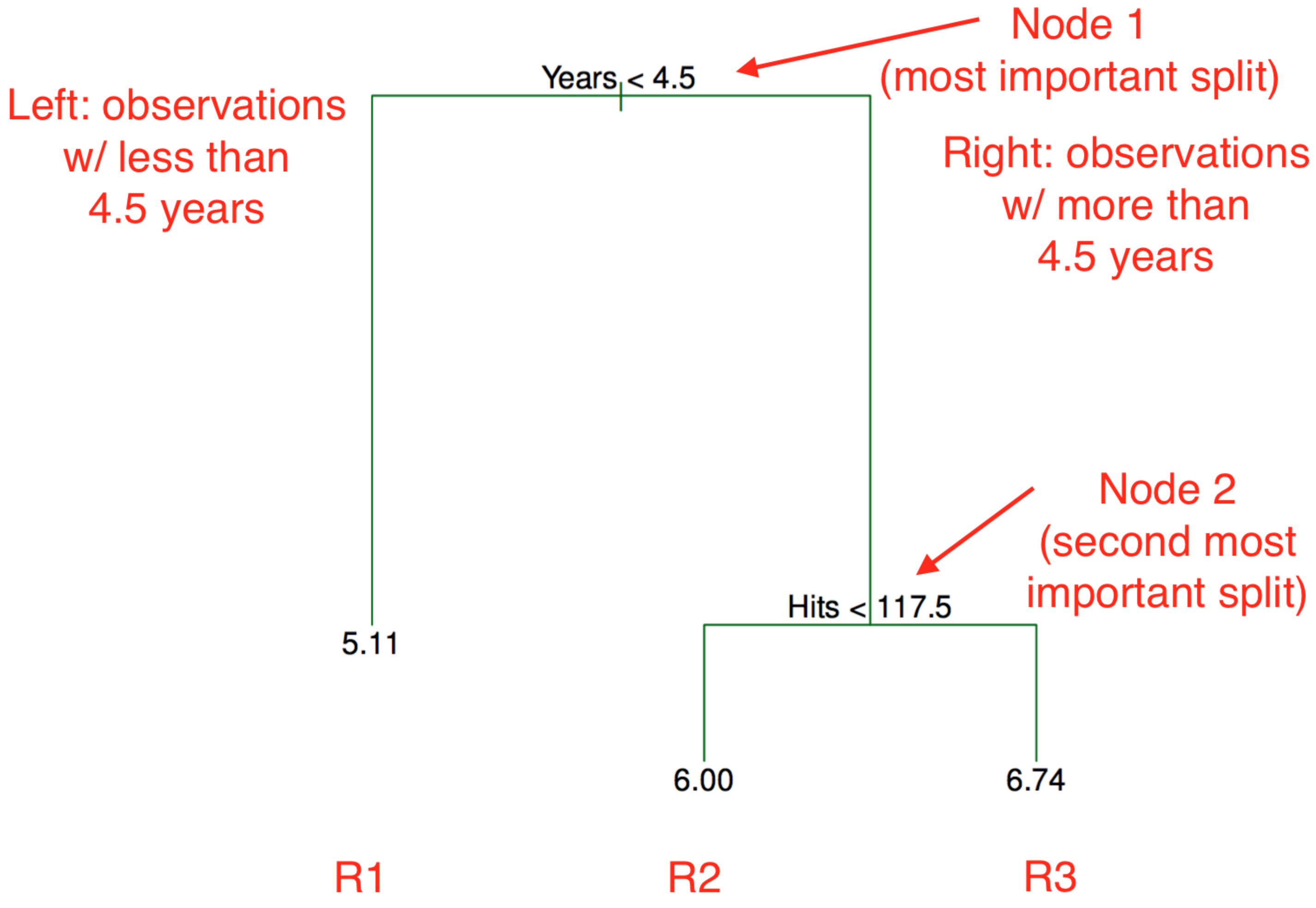


Node 1
(most important split)

Left: observations
w/ less than
4.5 years

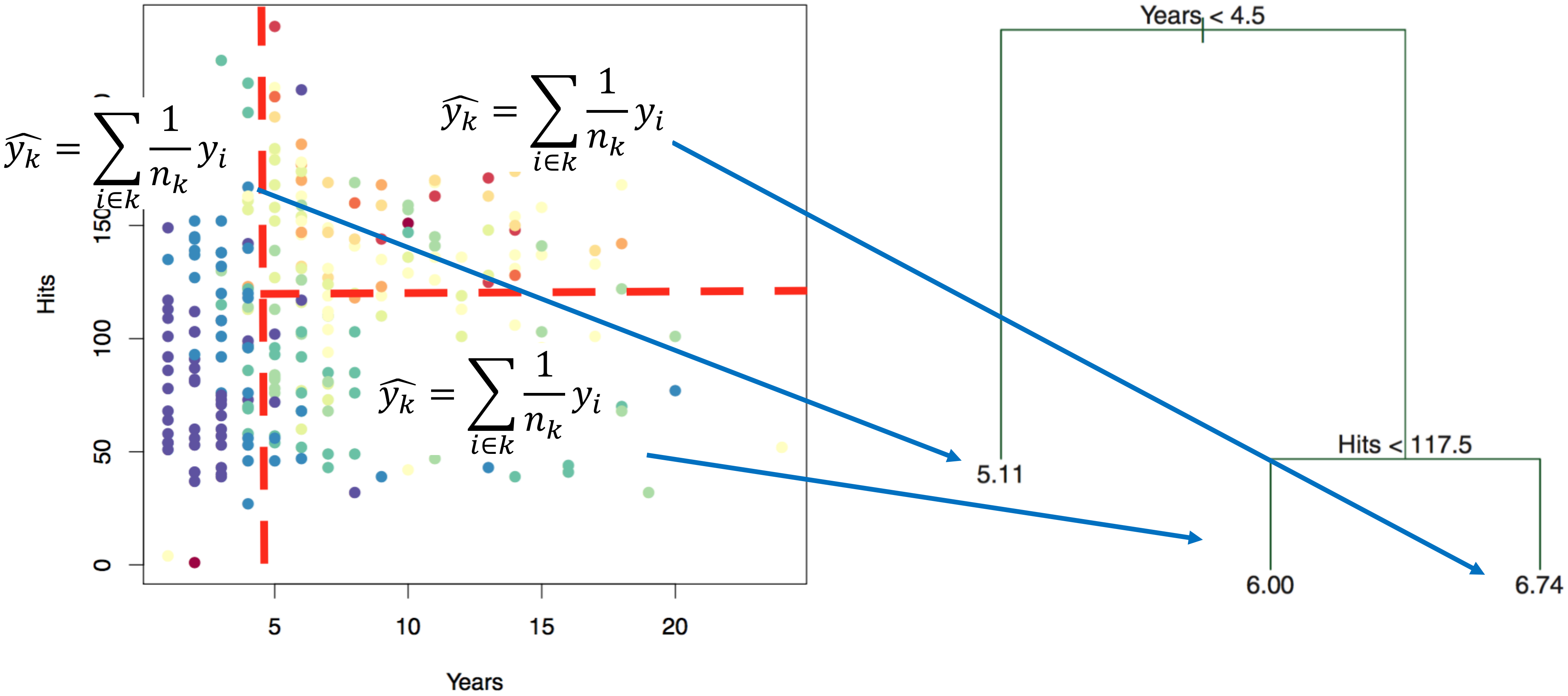
Right: observations
w/ more than
4.5 years

Predictions for Trees? “Leaf” Values


Node 1
(most important split)
Years < 4.5
Left: observations w/ less than 4.5 years
Right: observations w/ more than 4.5 years
Node 2
(second most important split)
Hits < 117.5
5.11
6.00
6.74
R1
R2
R3

- At the end of the tree are “leafs” (R1, R2, R3)
- How do we predict using a tree? Using the training data we find the average y for all the observations in the leaf $\bar{y}_{leaf} = \frac{1}{n_{leaf}} \sum_{i \in leaf} y_i$
- Any new Xs gets sorted into leafs and assigned the y average for that leaf: $\hat{y}_{i \in leaf} = \bar{y}_{leaf}$

Predictions for Trees? “Leaf” Values



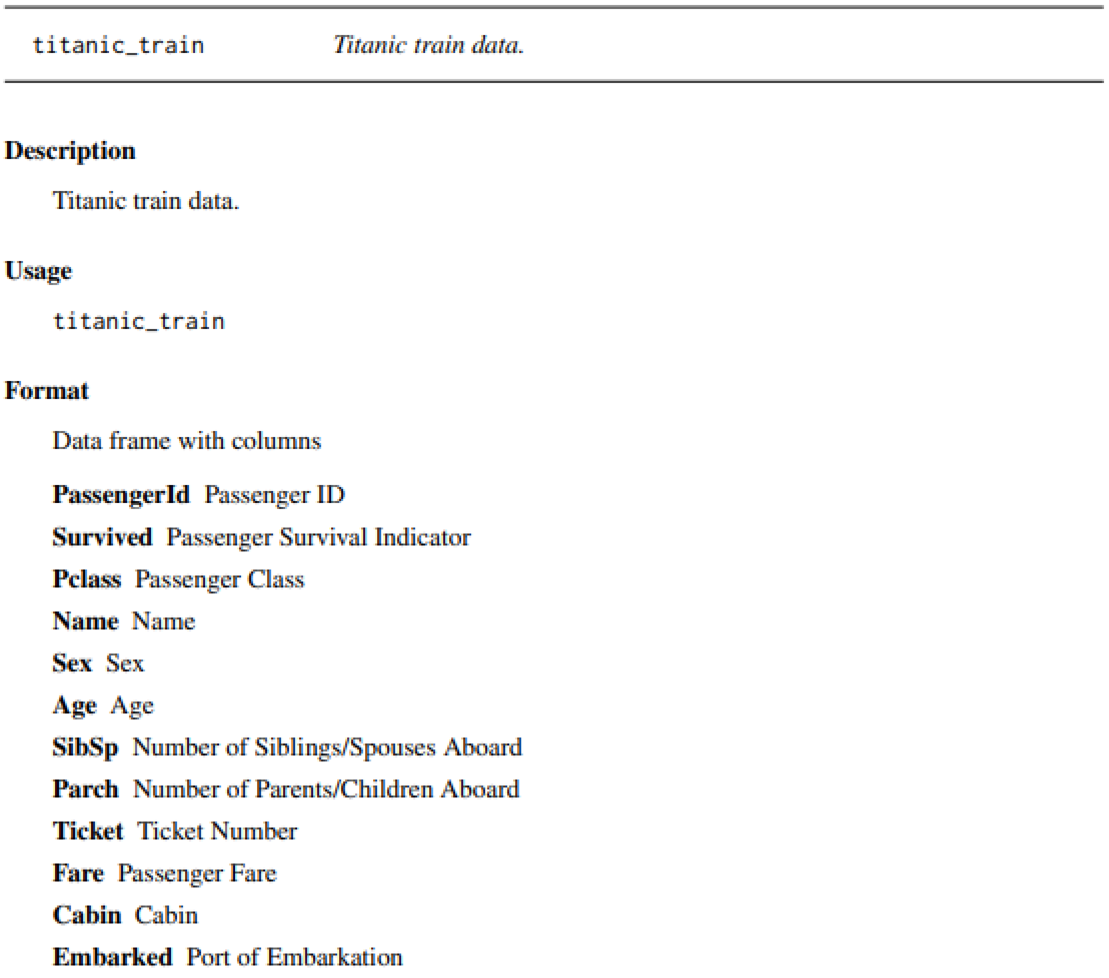
$$\widehat{y_k} = \sum_{i \in k} \frac{1}{n_k} y_i$$

$$\widehat{y_k} = \sum_{i \in k} \frac{1}{n_k} y_i$$

$$\widehat{y_k} = \sum_{i \in k} \frac{1}{n_k} y_i$$

- Leaf predictions are average values in each partition, k

Titanic dataset in ‘titanic’ package



`titanic_train` *Titanic train data.*

Description

Titanic train data.

Usage

```
titanic_train
```

Format

Data frame with columns

PassengerId Passenger ID

Survived Passenger Survival Indicator

Pclass Passenger Class

Name Name

Sex Sex

Age Age

SibSp Number of Siblings/Spouses Aboard

Parch Number of Parents/Children Aboard

Ticket Ticket Number

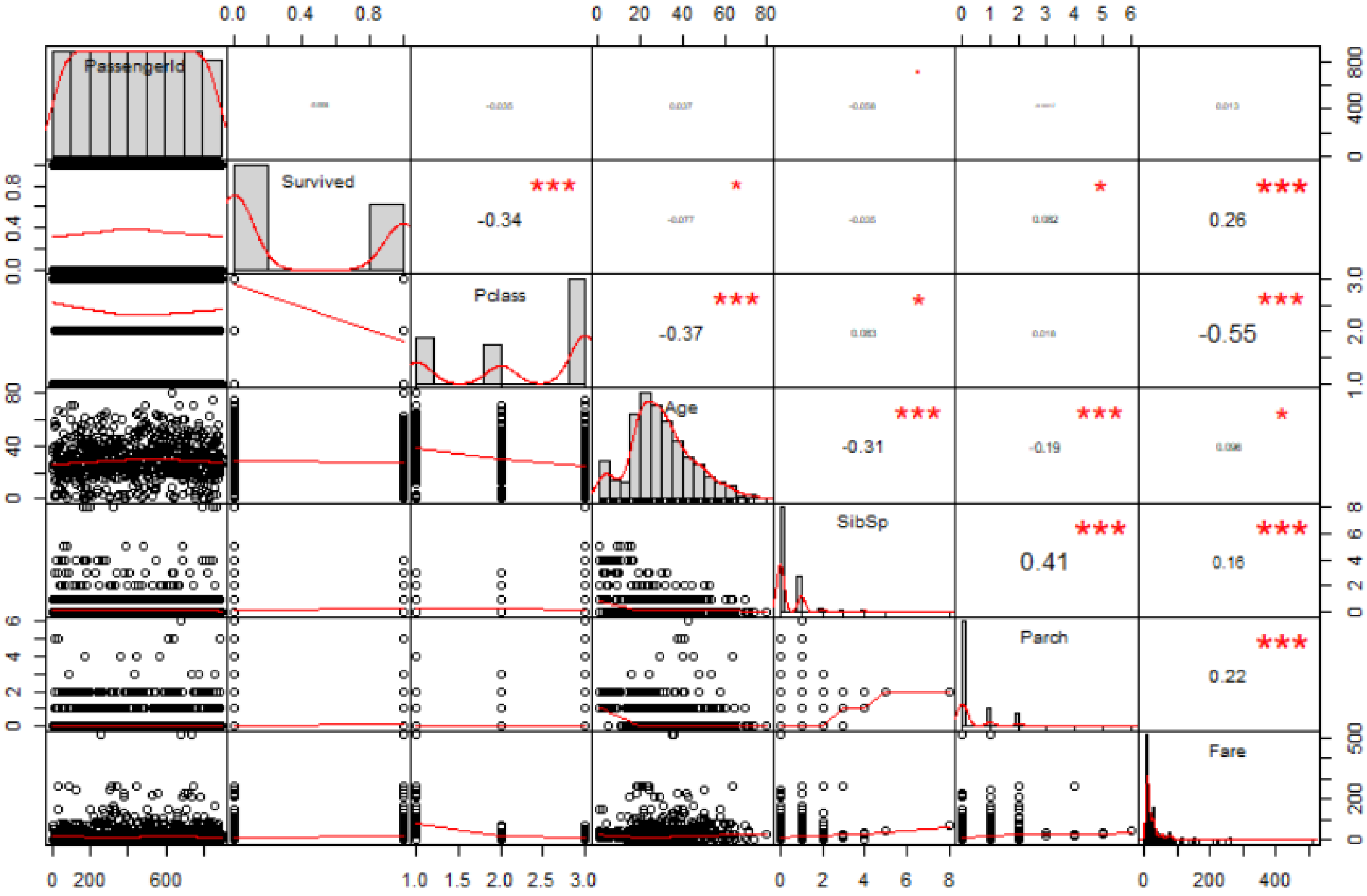
Fare Passenger Fare

Cabin Cabin

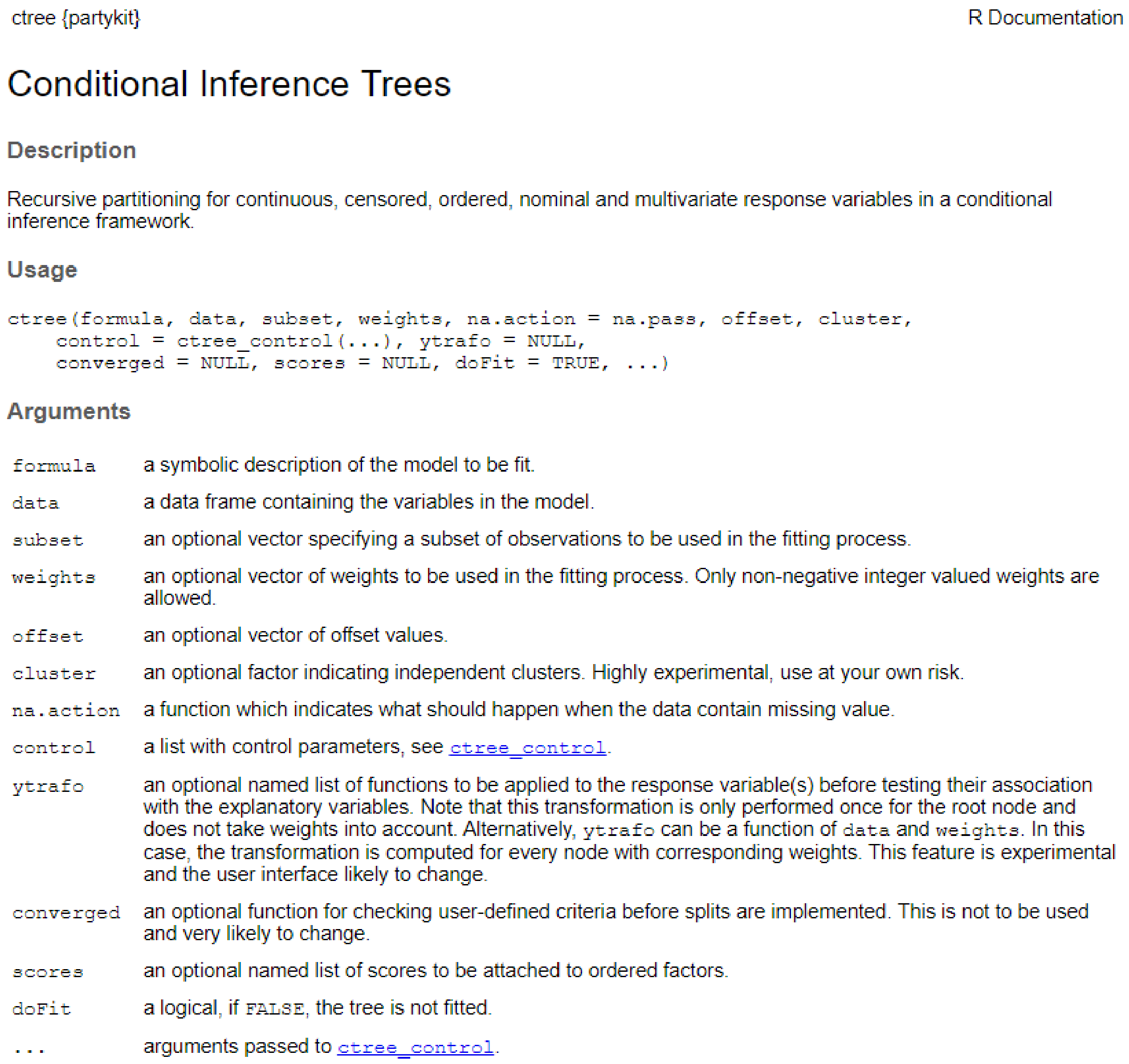
Embarked Port of Embarkation

- To estimate regression trees in R we’re going to use the ‘titanic’ dataset as an example.

Titanic dataset in ‘titanic’ package



ctree() function in package “partykit” to build regression tree



ctree {partykit} R Documentation

Conditional Inference Trees

Description

Recursive partitioning for continuous, censored, ordered, nominal and multivariate response variables in a conditional inference framework.

Usage

```
ctree(formula, data, subset, weights, na.action = na.pass, offset, cluster,
    control = ctree_control(...), ytrafo = NULL,
    converged = NULL, scores = NULL, doFit = TRUE, ...)
```

Arguments

`formula` a symbolic description of the model to be fit.

`data` a data frame containing the variables in the model.

`subset` an optional vector specifying a subset of observations to be used in the fitting process.

`weights` an optional vector of weights to be used in the fitting process. Only non-negative integer valued weights are allowed.

`offset` an optional vector of offset values.

`cluster` an optional factor indicating independent clusters. Highly experimental, use at your own risk.

`na.action` a function which indicates what should happen when the data contain missing value.

`control` a list with control parameters, see `ctree_control`.

`ytrafo` an optional named list of functions to be applied to the response variable(s) before testing their association with the explanatory variables. Note that this transformation is only performed once for the root node and does not take weights into account. Alternatively, `ytrafo` can be a function of `data` and `weights`. In this case, the transformation is computed for every node with corresponding weights. This feature is experimental and the user interface likely to change.

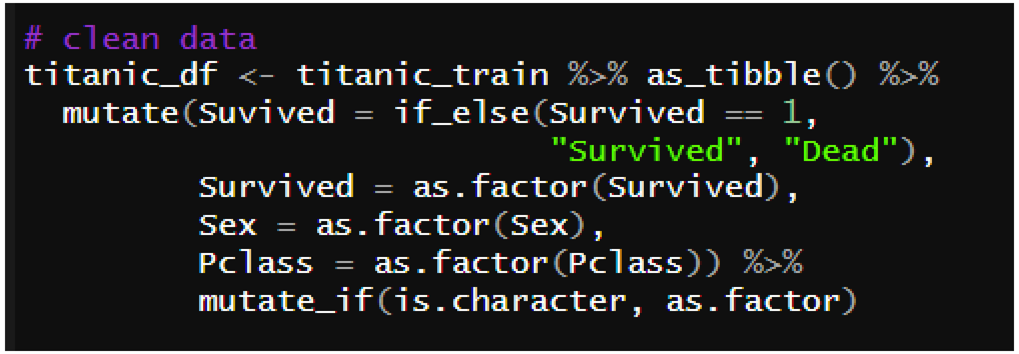
`converged` an optional function for checking user-defined criteria before splits are implemented. This is not to be used and very likely to change.

`scores` an optional named list of scores to be attached to ordered factors.

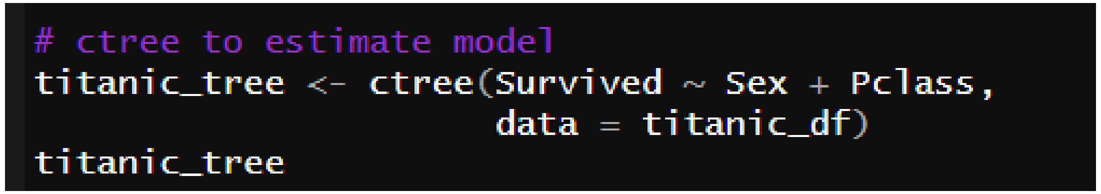
`doFit` a logical, if `FALSE`, the tree is not fitted.

`...` arguments passed to `ctree_control`.

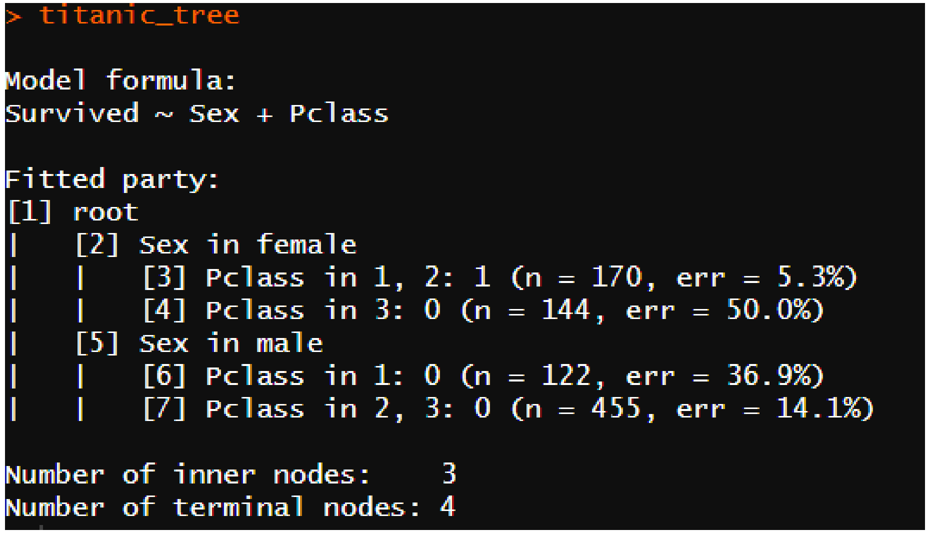
Estimate a tree model to predict titanic survival



```
# clean data
titanic_df <- titanic_train %>% as_tibble() %>%
  mutate(Suvived = if_else(Survived == 1,
                           "Survived", "Dead"),
         Survived = as.factor(Survived),
         Sex = as.factor(Sex),
         Pclass = as.factor(Pclass)) %>%
         mutate_if(is.character, as.factor)
```



```
# ctree to estimate model
titanic_tree <- ctree(Survived ~ Sex + Pclass,
                      data = titanic_df)
titanic_tree
```



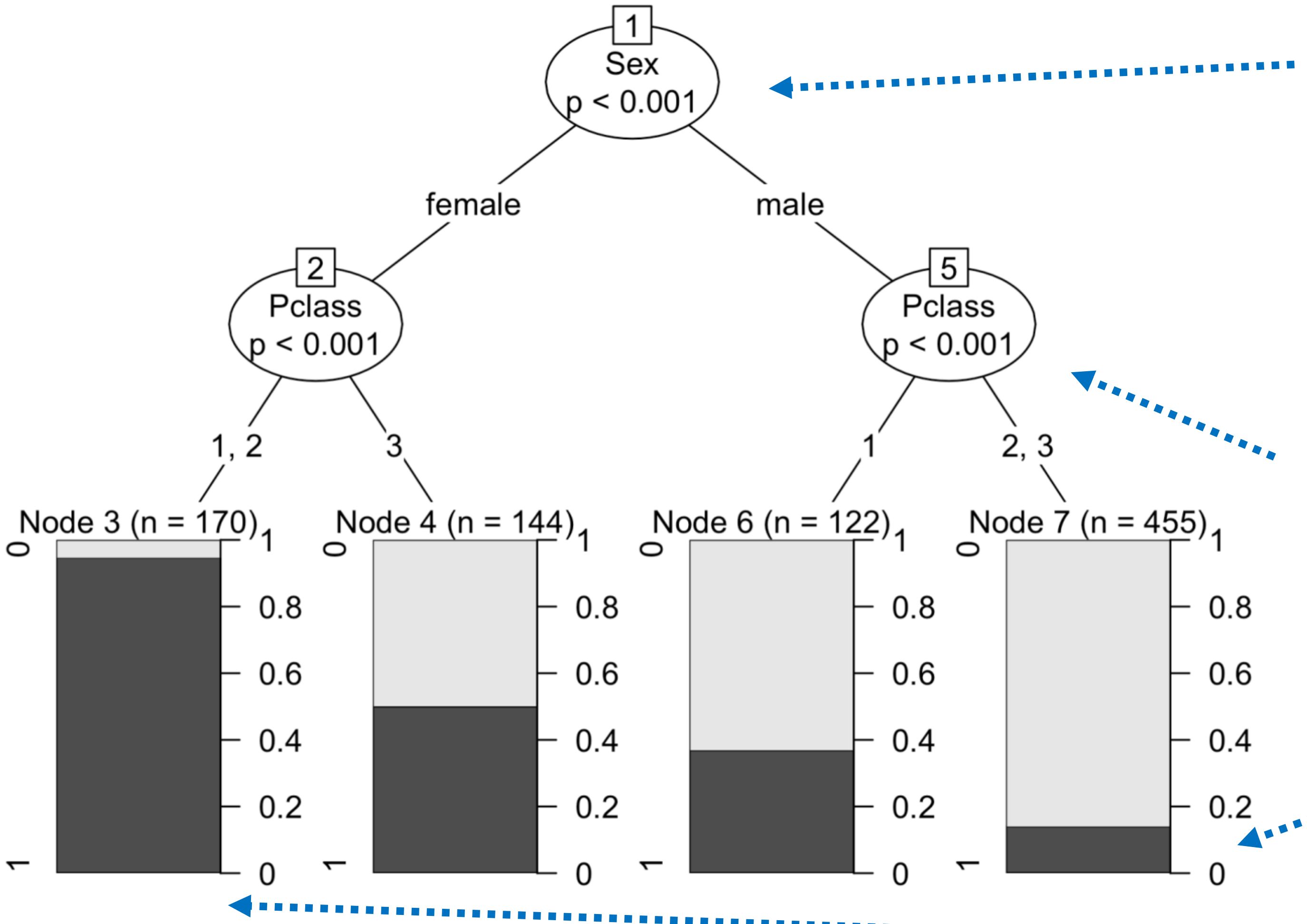
```
> titanic_tree

Model formula:
Survived ~ Sex + Pclass

Fitted party:
[1] root
|   [2] Sex in female
|   |   [3] Pclass in 1, 2: 1 (n = 170, err = 5.3%)
|   |   [4] Pclass in 3: 0 (n = 144, err = 50.0%)
|   [5] Sex in male
|   |   [6] Pclass in 1: 0 (n = 122, err = 36.9%)
|   |   [7] Pclass in 2, 3: 0 (n = 455, err = 14.1%)

Number of inner nodes:    3
Number of terminal nodes: 4
```

- First a bit of data cleaning against the titanic data frame.
- Note we change the characters to factors using mutate_if()

- ctree() function estimates a regression tree
- We need: formula (Y and Xs)
- Data set to estimate (cleaned training)

- Raw model output isn't the prettiest to read (but does show fitted model.)

Plot Fitted Regression Tree

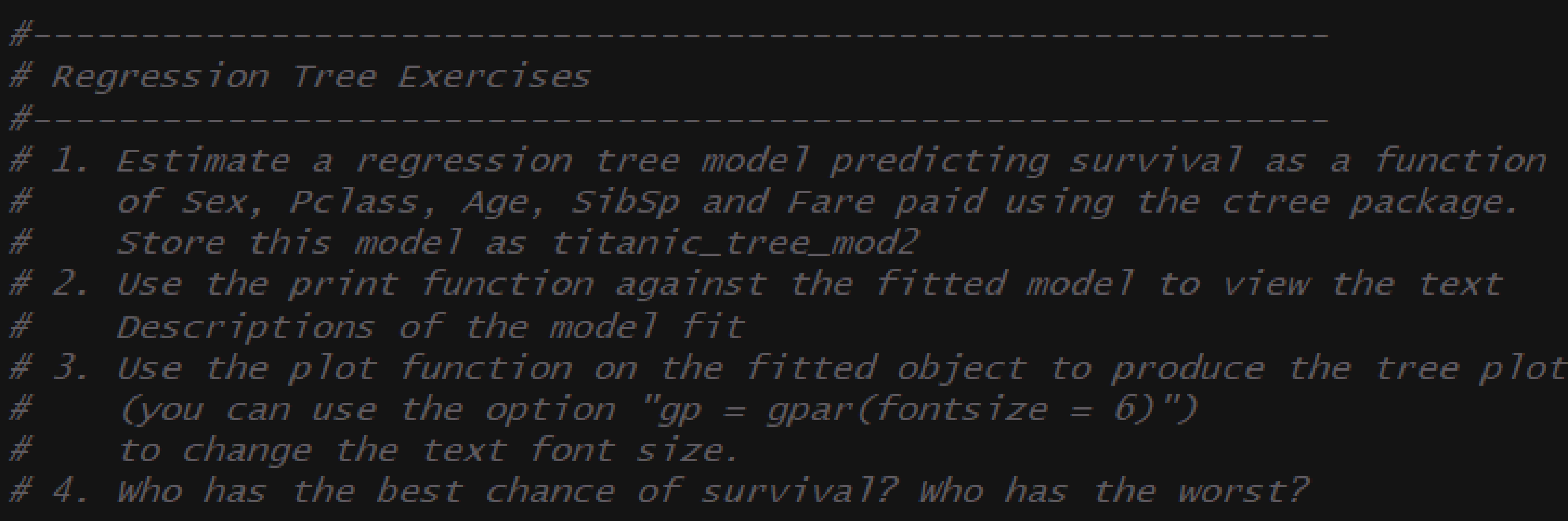


- First split is shown at top with p-value with null hypothesis that split doesn’t increase class “separation”. Sex of passenger is the most important variable, and p-value shows it improves model fit

- Second split is given by successive node. For males, passenger class is second most important variable

- Survival rates are shown in the leaves summaries at the bottom. There are 455 males with passenger class 2 or 3, and about 15% of them survived.

- For females in classes 1 or 2, nearly all survive. Class 3 females have a survival rate of 50%.



```
#------------------------------------------------------------
# Regression Tree Exercises
#------------------------------------------------------------
# 1. Estimate a regression tree model predicting survival as a function
#    of Sex, Pclass, Age, SibSp and Fare paid using the ctree package.
#    Store this model as titanic_tree_mod2
# 2. Use the print function against the fitted model to view the text
#    Descriptions of the model fit
# 3. Use the plot function on the fitted object to produce the tree plot
#    (you can use the option "gp = gpar(fontsize = 6)")
#    to change the text font size.
# 4. Who has the best chance of survival? Who has the worst?
```

Lab Time!

Class 16 Summary

- Regression trees use binary decisions of the X variables to predict the outcome
- Regression trees can be used for either regression or classification problems
- We must control the depth and/or complexity of a tree or else it is prone to overfitting