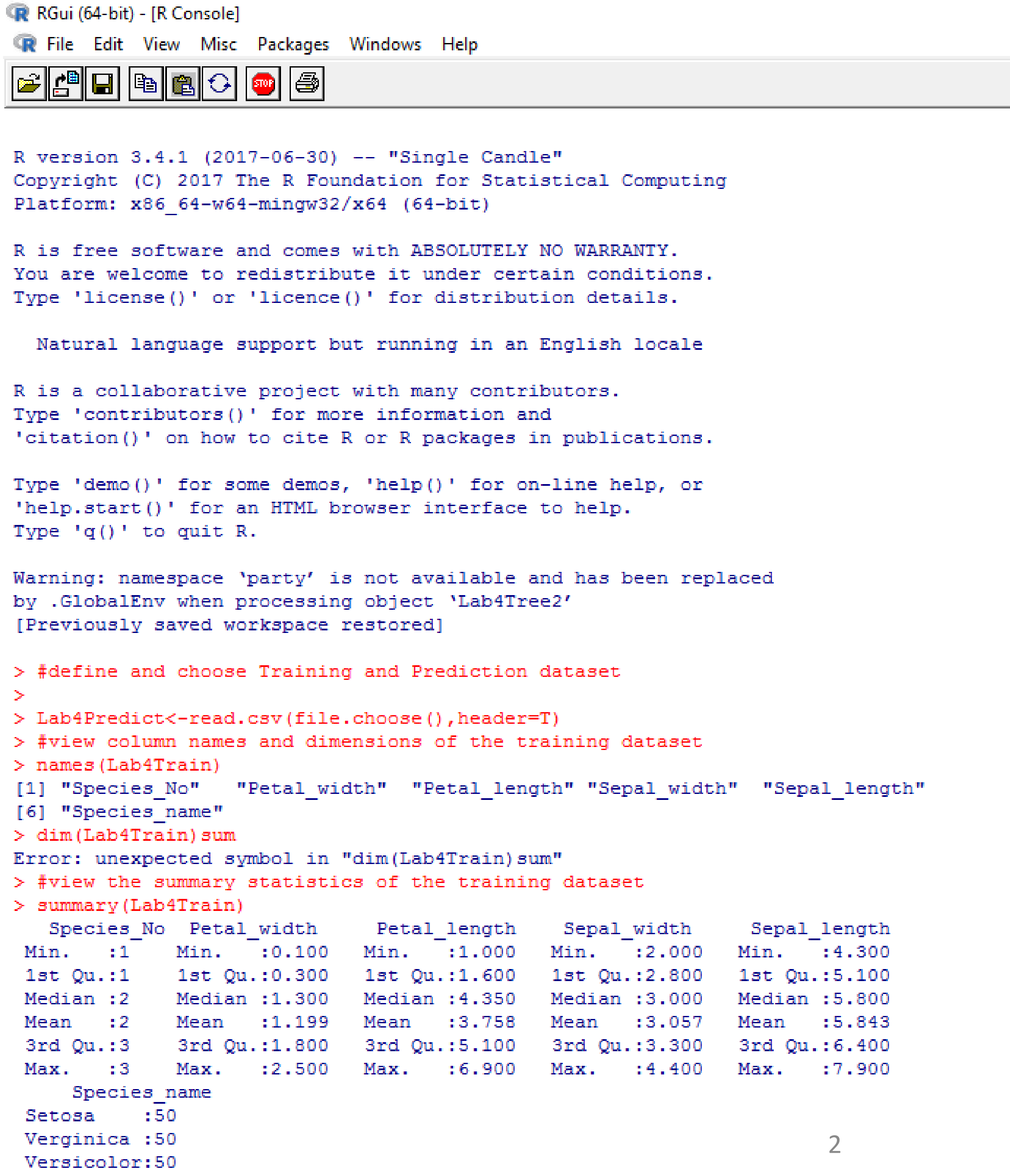
RDataMining

Week 4 Lab in R

Dr. Liang (Leon) Chen

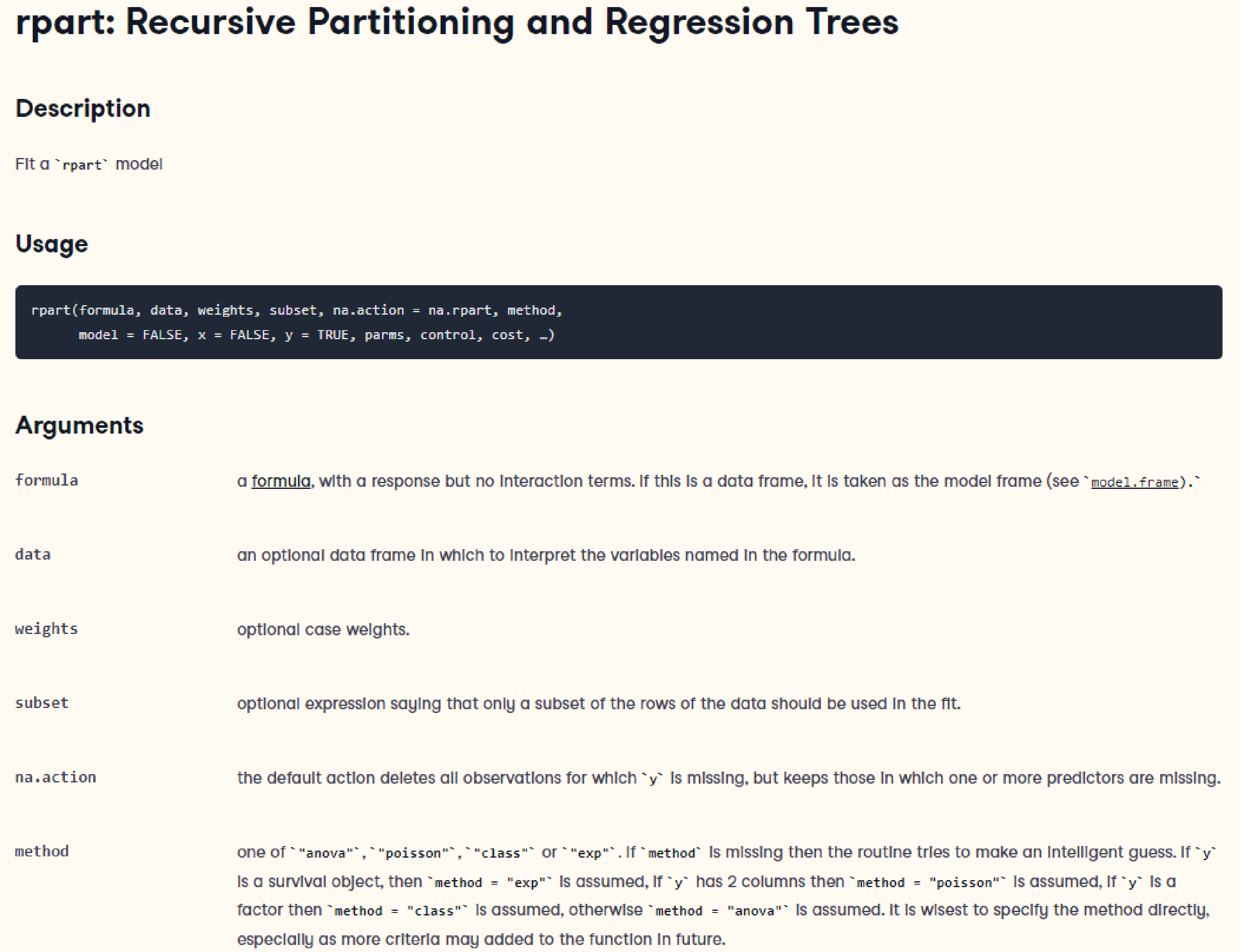
Instructions

- Two methods in R are provided, but you are required to choose just one of them to complete this lab. In your homework, please indicate which method you choose.
- Type and then run the script provided in the one of the next two slides.
- You do not need to type notes (starting at #), but it's a good manner to have them in you code.
- In order to see codes and notes clearly, I show the script in RStudio.



```
R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Warning: namespace ‘party’ is not available and has been replaced
by .GlobalEnv when processing object ‘Lab4Tree2’
[Previously saved workspace restored]

> #define and choose Training and Prediction dataset
>
> Lab4Predict<-read.csv(file.choose(),header=T)
> #view column names and dimensions of the training dataset
> names(Lab4Train)
[1] "Species_No"   "Petal_width"  "Petal_length" "Sepal_width"  "Sepal_length"
[6] "Species_name"
> dim(Lab4Train)sum
Error: unexpected symbol in "dim(Lab4Train)sum"
> #view the summary statistics of the training dataset
> summary(Lab4Train)
   Species_No  Petal_width     Petal_length    Sepal_width     Sepal_length
 Min.   :1    Min.   :0.100   Min.   :1.000   Min.   :2.000   Min.   :4.300
 1st Qu.:1    1st Qu.:0.300   1st Qu.:1.600   1st Qu.:2.800   1st Qu.:5.100
 Median :2    Median :1.300   Median :4.350   Median :3.000   Median :5.800
 Mean   :2    Mean   :1.199   Mean   :3.758   Mean   :3.057   Mean   :5.843
 3rd Qu.:3    3rd Qu.:1.800   3rd Qu.:5.100   3rd Qu.:3.300   3rd Qu.:6.400
 Max.   :3    Max.   :2.500   Max.   :6.900   Max.   :4.400   Max.   :7.900
     Species_name
 Setosa    :50
 Verginica :50
 Versicolor:50
```

Method 1: Use Library rpart



rpart: Recursive Partitioning and Regression Trees

Description

Fit a `rpart` model

Usage

```
rpart(formula, data, weights, subset, na.action = na.rpart, method,
      model = FALSE, x = FALSE, y = TRUE, parms, control, cost, …)
```

Arguments

`formula` a formula, with a response but no interaction terms. If this is a data frame, it is taken as the model frame (see `model.frame`).`

`data` an optional data frame in which to interpret the variables named in the formula.

`weights` optional case weights.

`subset` optional expression saying that only a subset of the rows of the data should be used in the fit.

`na.action` the default action deletes all observations for which `y` is missing, but keeps those in which one or more predictors are missing.

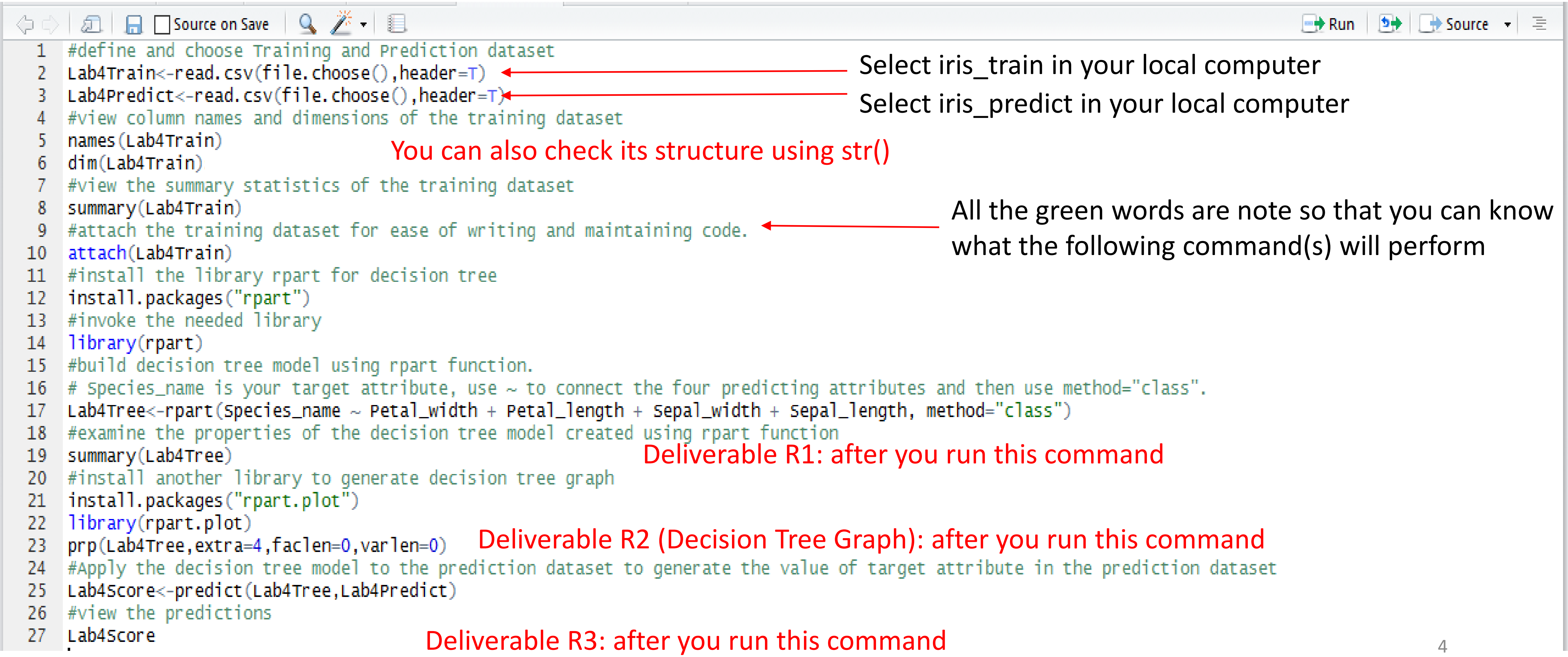
`method` one of `"anova"`, `"poisson"`, `"class"` or `"exp"`. If `method` is missing then the routine tries to make an intelligent guess. If `y` is a survival object, then `method = "exp"` is assumed, if `y` has 2 columns then `method = "poisson"` is assumed, if `y` is a factor then `method = "class"` is assumed, otherwise `method = "anova"` is assumed. It is wisest to specify the method directly, especially as more criteria may added to the function in future.

Resources:
The library rpart documentation on rdocumentation.org

Package ‘rpart’ on r-project.org, 2022

An Introduction to Recursive Partitioning Using the RPART Routines on r-project.org, 2022

Method 1: Use Library rpart



```r
#define and choose Training and Prediction dataset
Lab4Train<-read.csv(file.choose(),header=T)
Lab4Predict<-read.csv(file.choose(),header=T)
#view column names and dimensions of the training dataset
names(Lab4Train)
dim(Lab4Train)
#view the summary statistics of the training dataset
summary(Lab4Train)
#attach the training dataset for ease of writing and maintaining code.
attach(Lab4Train)
#install the library rpart for decision tree
install.packages("rpart")
#invoke the needed library
library(rpart)
#build decision tree model using rpart function.
# Species_name is your target attribute, use ~ to connect the four predicting attributes and then use method="class".
Lab4Tree<-rpart(Species_name ~ Petal_width + Petal_length + Sepal_width + Sepal_length, method="class")
#examine the properties of the decision tree model created using rpart function
summary(Lab4Tree)
#install another library to generate decision tree graph
install.packages("rpart.plot")
library(rpart.plot)
prp(Lab4Tree,extra=4,faclen=0,varlen=0)
#Apply the decision tree model to the prediction dataset to generate the value of target attribute in the prediction dataset
Lab4Score<-predict(Lab4Tree,Lab4Predict)
#view the predictions
Lab4Score
```

Select iris_train in your local computer

Select iris_predict in your local computer

You can also check its structure using str()

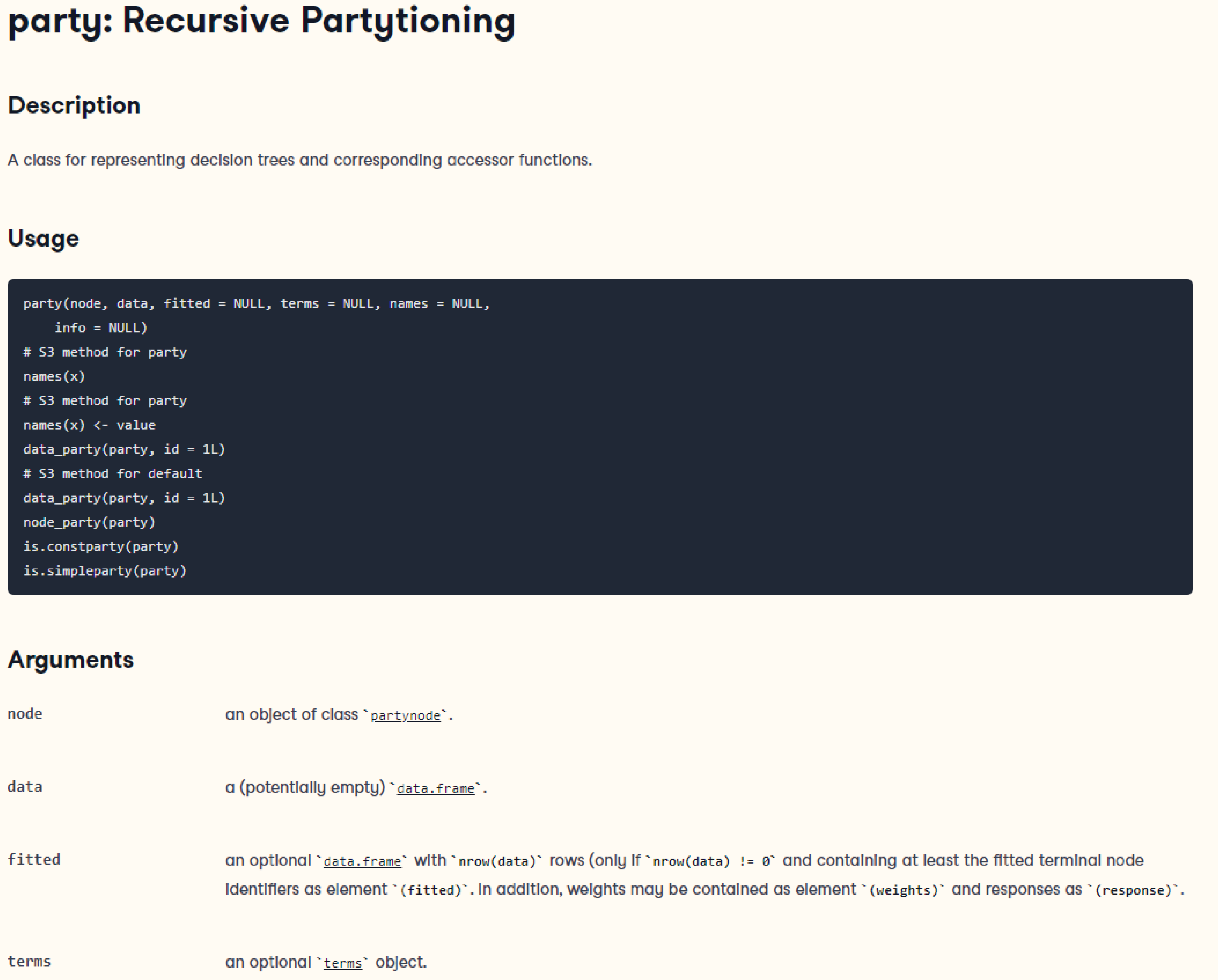
All the green words are note so that you can know what the following command(s) will perform

Deliverable R1: after you run this command

Deliverable R2 (Decision Tree Graph): after you run this command

Deliverable R3: after you run this command

Method 2: Use Library party



party: Recursive Partytioning

Description

A class for representing decision trees and corresponding accessor functions.

Usage

```
party(node, data, fitted = NULL, terms = NULL, names = NULL,
    info = NULL)
# S3 method for party
names(x)
# S3 method for party
names(x) <- value
data_party(party, id = 1L)
# S3 method for default
data_party(party, id = 1L)
node_party(party)
is.constparty(party)
is.simpleparty(party)
```

Arguments

node	an object of class `partynode`.
data	a (potentially empty) `data.frame`.
fitted	an optional `data.frame` with `nrow(data)` rows (only if `nrow(data) != 0` and containing at least the fitted terminal node identifiers as element `(fitted)`. In addition, weights may be contained as element `(weights)` and responses as `(response)`.
terms	an optional `terms` object.

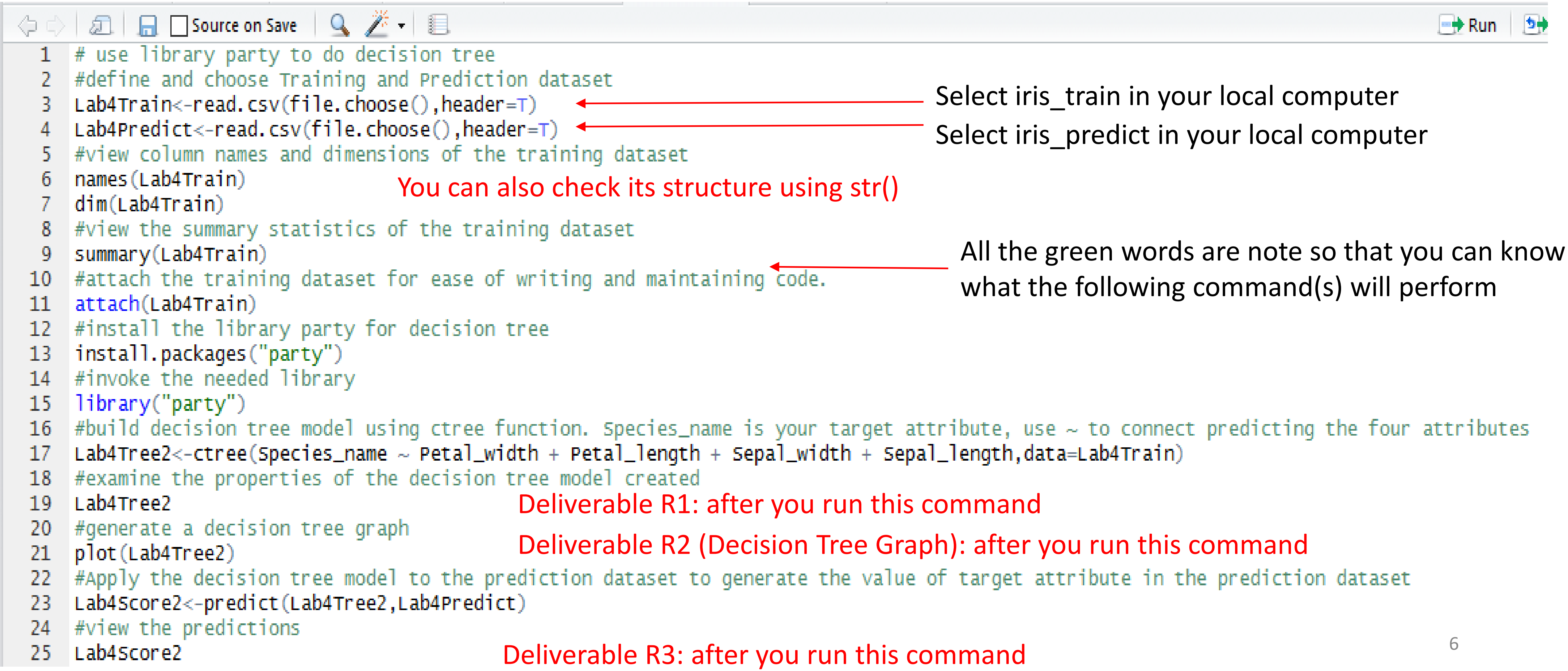
Resources:

Decision Tree Classification Example With ctree in R, 2017

Decision Tree using R Package: 'party'

Package 'party' on r-project.org, 2022

Method 2: Use Library party



```r
# use library party to do decision tree
#define and choose Training and Prediction dataset
Lab4Train<-read.csv(file.choose(),header=T)
Lab4Predict<-read.csv(file.choose(),header=T)
#view column names and dimensions of the training dataset
names(Lab4Train)
dim(Lab4Train)
#view the summary statistics of the training dataset
summary(Lab4Train)
#attach the training dataset for ease of writing and maintaining code.
attach(Lab4Train)
#install the library party for decision tree
install.packages("party")
#invoke the needed library
library("party")
#build decision tree model using ctree function. Species_name is your target attribute, use ~ to connect predicting the four attributes
Lab4Tree2<-ctree(Species_name ~ Petal_width + Petal_length + Sepal_width + Sepal_length,data=Lab4Train)
#examine the properties of the decision tree model created
Lab4Tree2
#generate a decision tree graph
plot(Lab4Tree2)
#Apply the decision tree model to the prediction dataset to generate the value of target attribute in the prediction dataset
Lab4Score2<-predict(Lab4Tree2,Lab4Predict)
#view the predictions
Lab4Score2
```

Select iris_train in your local computer

Select iris_predict in your local computer

You can also check its structure using str()

All the green words are note so that you can know what the following command(s) will perform

Deliverable R1: after you run this command

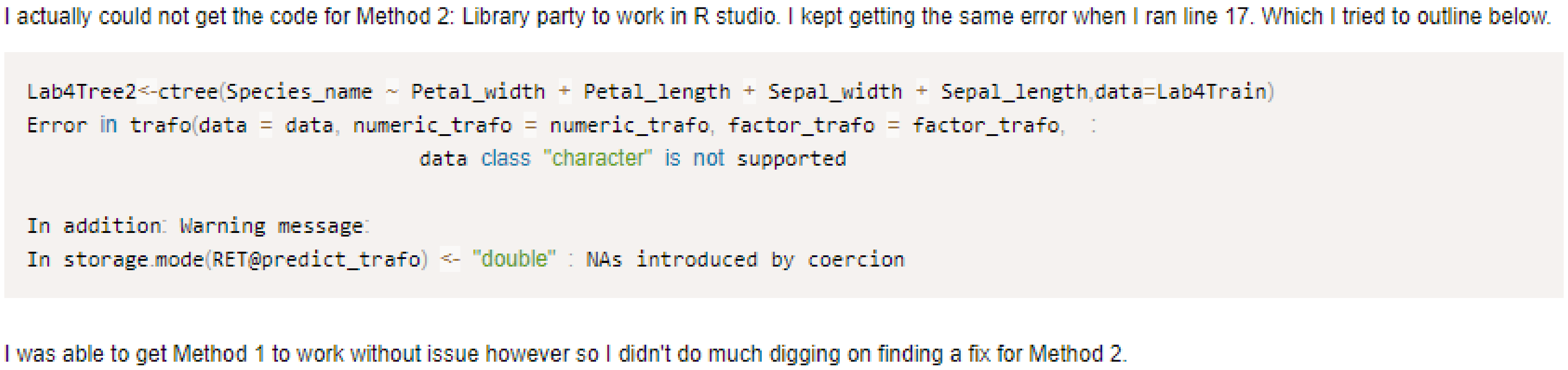
Deliverable R2 (Decision Tree Graph): after you run this command

Deliverable R3: after you run this command

Deliverables

- Deliverable R1: take a screenshot of your decision tree model. Try to use the resources provided to understand its output
- Deliverable R2: take a screenshot of your decision tree graph and briefly describe it. Your description must include the root node, split nodes, and leaf nodes.
- Deliverable R3: after you apply the decision tree model to your prediction dataset, take a screenshot of the prediction result and briefly describe how the result help you determine the predicted class of each case.
- Deliverable R4: Compare the decision tree models generated in RapidMiner and R, and then point out at least two differences that you observe.

FAQs



I actually could not get the code for Method 2: Library party to work in R studio. I kept getting the same error when I ran line 17. Which I tried to outline below.

```
Lab4Tree2<-ctree(Species_name ~ Petal_width + Petal_length + Sepal_width + Sepal_length,data=Lab4Train)
Error in trafo(data = data, numeric_trafo = numeric_trafo, factor_trafo = factor_trafo,  :
                          data class "character" is not supported

In addition: Warning message:
In storage.mode(RET@predict_trafo) <- "double" : NAs introduced by coercion
```

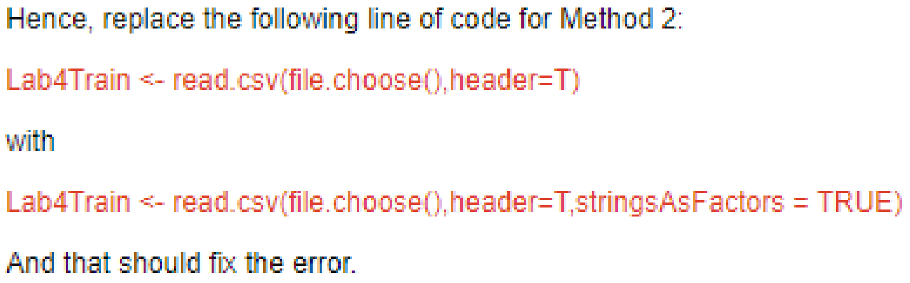
I was able to get Method 1 to work without issue however so I didn't do much digging on finding a fix for Method 2.

Answer: Please use str() to check the structure of all the variables in your dataset. For some reason, when you import the csv file to R, Species_name may be recognized as a class, instead of a factor. If so, you need to convert it to a factor using as.factor() function via one of them below:

- Lab4Train[,6]<-as.factor(Lab4Train[,6])
- Lab4Train$Species_name <-as.factor(Lab4Train$Species_name)

I hope this solves your problem.

An alternative method is to add a parameter when importing your dataset



Hence, replace the following line of code for Method 2:

```
Lab4Train <- read.csv(file.choose(),header=T)
```

with

```
Lab4Train <- read.csv(file.choose(),header=T,stringsAsFactors = TRUE)
```

And that should fix the error.