DATA SCIENCE

Practical Work 4: Regression

M. Jose Ramirez

MITSS

==========

In this sesion we are going to illustrate how to apply some of the data mining techniques we have seen in the Unit 3 of the course. In particular, we are going to solve regression problems.

1. Plotting a Regression Line: An illustrative example

Firstly, we load the package we are going to use:

```
.lib<-              c("ggplot2","hydroGOF","rpart","rpart.plot","nnet")
.inst         <-          .lib         %in%         installed.packages()
if       (length(.lib[!.inst])>0)       install.packages(.lib[!.inst])
lapply(.lib, require, character.only=TRUE)
```

- Download the file ``cherry.csv'' from poliformat and load the data into R. This data set provides measurements of the girth, height and volume of timber in 31 felled black cherry trees.

```
cherry<-read.csv("./data/cherry.csv",        header=T,        sep=';')
head(cherry)
```

```
##                                 Girth           Height           Volume
##     1               8.3                            70               10.3
##     2               8.6                            65               10.3
##     3               8.8                            63               10.2
##     4           10.5                              72               16.4
##     5           10.7                              81               18.8
## 6  10.8     83   19.7
```

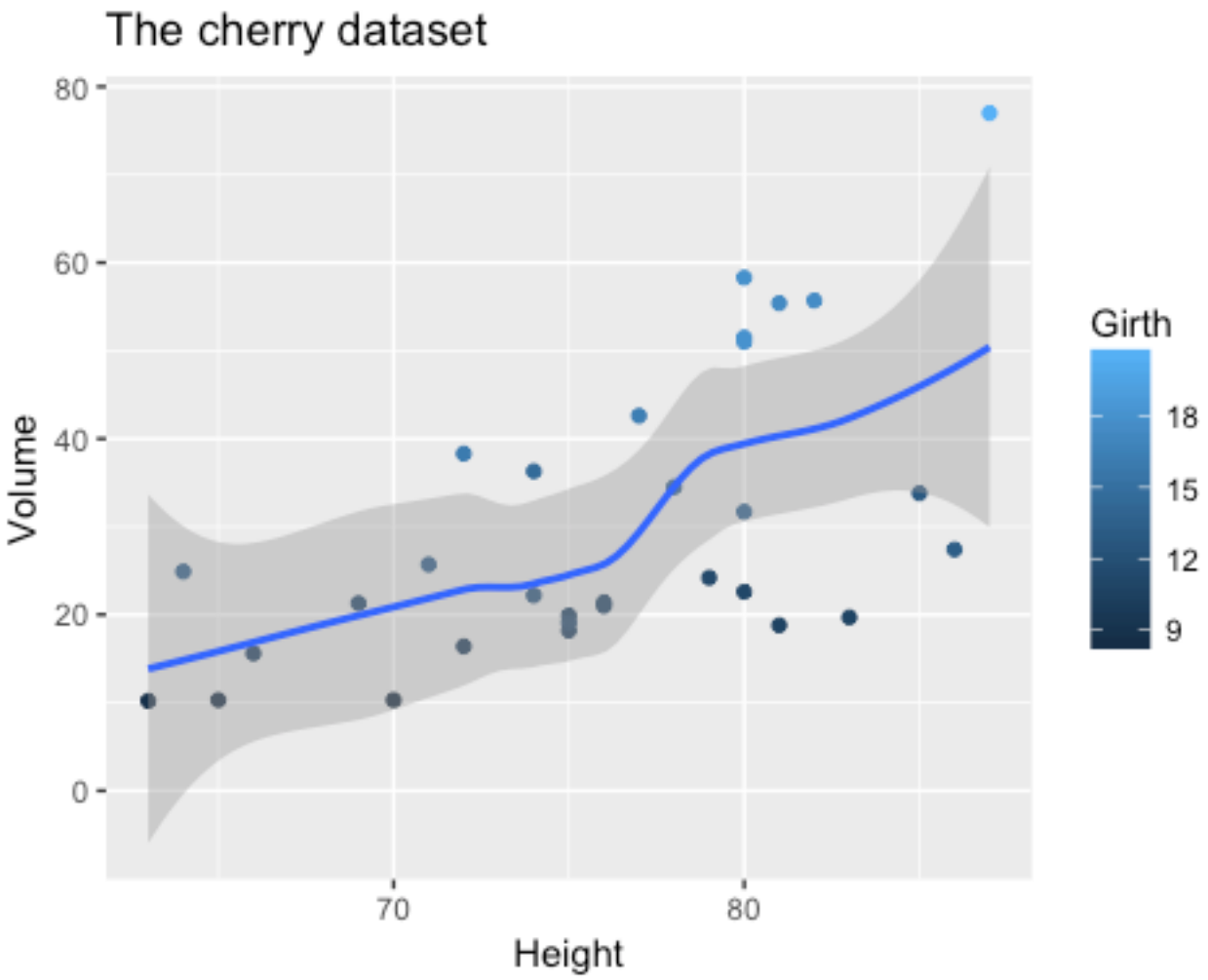
```
summary(cherry)
```

```
##             Girth                          Height               Volume
##     Min.       :  8.30       Min.       :63       Min.       :10.20
##       1st    Qu.:11.05         1st    Qu.:72         1st    Qu.:19.40
##       Median    :12.90         Median    :76         Median    :24.20
##     Mean        :13.25       Mean        :76       Mean        :30.17
##       3rd    Qu.:15.25         3rd    Qu.:80         3rd    Qu.:37.30
##  Max.    :20.60    Max.    :87    Max.    :77.00
```

- We can visualize the data with respect to the variables height and volume

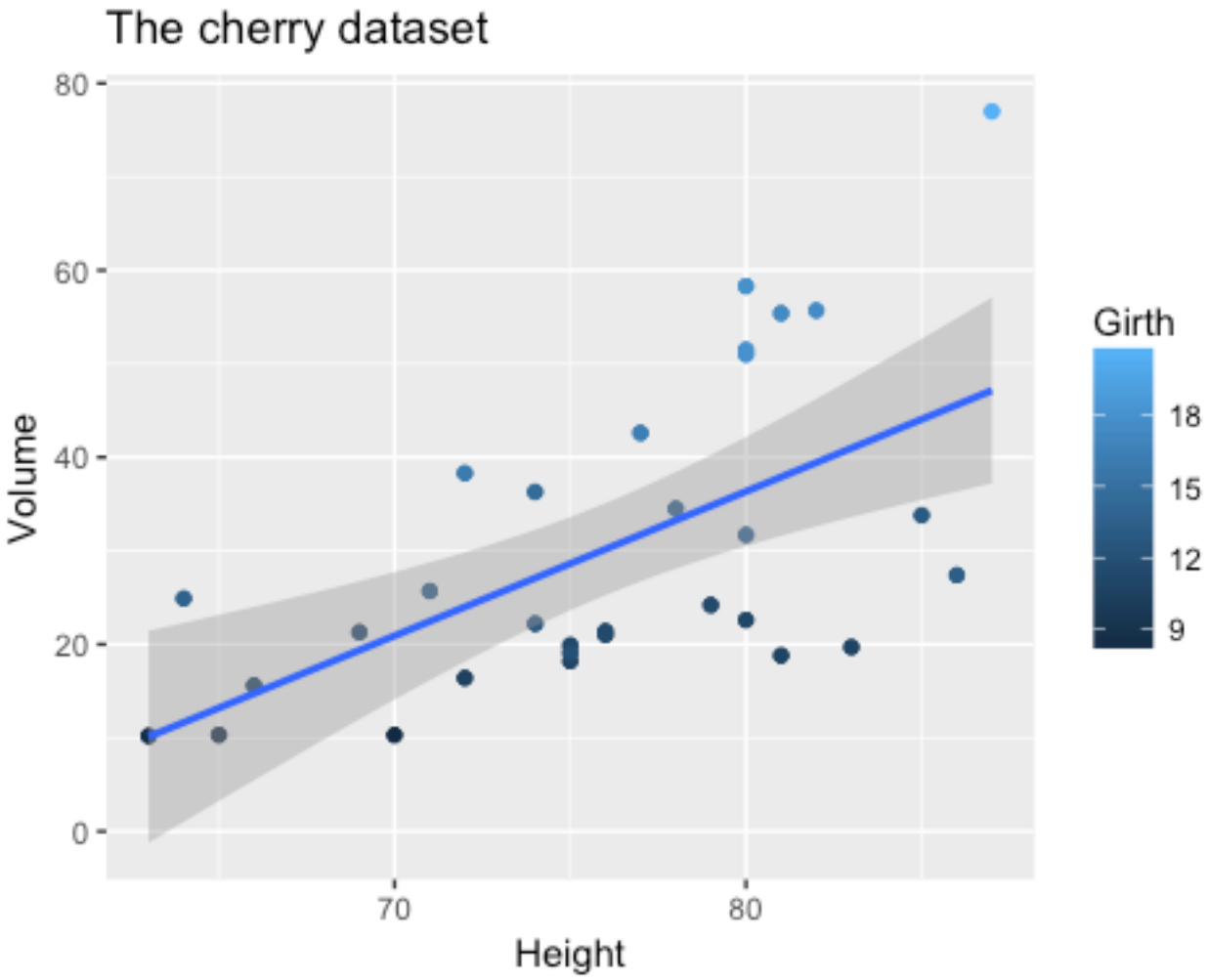
```
qplot(Height,Volume,data=cherry, colour = Girth, main="The cherry data
set", geom = c("point", "smooth"))
```

```
## `geom_smooth()` using method = 'loess'
```



We can also plot the regression line (instead of the tendency)

```
qplot(Height,Volume,data=cherry, colour = Girth, main="The cherry data
set")+ geom_point() + geom_smooth(method="lm")
```



2. A regression case study

For this case study we will use the Auto MPG Data Set from the UCI repository (https://archive.ics.uci.edu/ml/). This data set concerns city-cycle fuel consumption in miles per gallon, to be predicted in terms of 3 multivalued discrete and 5 continuous attributes: Attribute Information:

```
1.     mpg:                                                continuous
2.      cylinders:                             multi-valued     discrete
3.                displacement:                                continuous
4.          horsepower:                                      continuous
5.      weight:                                              continuous
6.                acceleration:                              continuous
7.      model      year:                       multi-valued     discrete
8.     origin:                                  multi-valued    discrete
9. car name:       string (almost unique for each instance)
```

The main goal of this case study is to obtain predictions of the fuel consumption in miles per gallon (the numerical mpg variable) relating to the set of the other explanatory variables.

Exercises

1. Load the dataset into R (download the file ``auto.txt'' from poliformat) and add the following titles for columns: "mpg", "cyl", "disp", "hp", "wt", "acc", "myear", "orig", "carname". Explore the data and be sure that the type of the attributes are correct according to the above list (otherwise transform them). Check whether the car name attribute is almost unique for each instance, and if so, remove it.

2. Given that the linear regression we will use is not able to deal with unknown values, remove the rows with NA values.

3. Set the seed to a constant value and randomly split the dataset into 75% train and 25% test.

4. Fit several regression models to the training data but only using the numerical attributes: a linear model (using the lm function, which fits a linear model using ordinary least squares), a regression tree (CART) from the rpart package (set the parameter method to anova in order to produce a CART tree), and a neural network from the nnet package (set the parameters skip and linout-numerical output- to TRUE and size-hidden units- to 12).

5. The rpart tree can be graphically visualised using the function rpart.plot().

6. Prune the regression tree using the prune function and setting the cp parameter to the value you consider is a good balance between complexity and performance (the plotcp() function plots tree sizes and relative errors for different values of the complexity parameter). Visualise the new tree.

7. Use the "predict" function to apply the model for predicting the training and the test sets.