# R – advanced vizualization and simple data mining algorithms

# More plots in the one

- ggplot lets you combine more plots in one
- adding them in a „sentence“
- e.g. geom_point()+geom_boxplot()

- combining plots:

grid.extra(p1, p2, p3, p4, nrow=2, ncol=2)

- demo

# Saving plots to file with grDevices

- format(path, width, height, units, res)

  ..

  ..

  dev.off()
- format: bmp, png, jpeg, tiff

  jpeg("iris.jpg",units="in",width=5, height=5, res=600)

  grid.arrange(p1, p2, p3, p4, ncol=2, nrow=2)

  dev.off()

# Saving plots to file with ggsave

- ggsave expects ggplot2 object
- g <- arrangeGrob(p1, p2, p3, p4, ncol=2, nrow=2) #generates g
  ggsave(file="iris.pdf", g)

- demo

# Lab

- take dataset cars
- create 2 seperate plots (scatter and boxplot)
- join them by rows
- save it to jpeg

# Splitting data to train and test set

- many functions, e.g.:

  dplyr::sample_n()

  dplyr::sample_frac()

  caTools::sample.split()

demo

# Lab

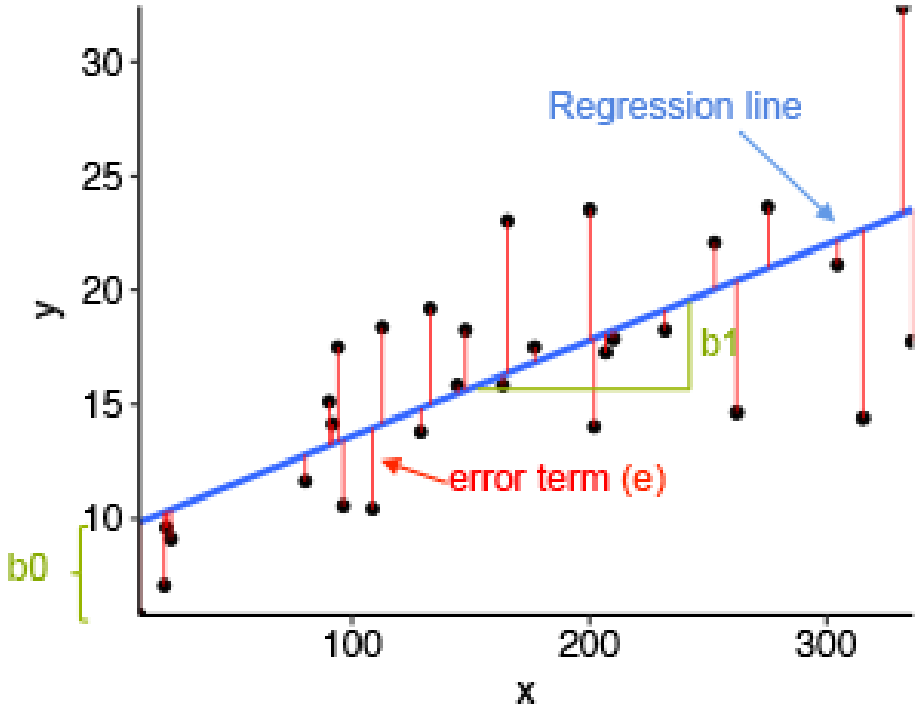
- Split dataset iris to train and test set (0.7/0.3)
- Display number of rows in both sets

# Linear regression

- Linear regression is to predict response with a linear function of predictors as follows: $y = c0 + c1x1 + c2x2 + \cdots + ckxk$,

  where $x1, x2, \cdots, xk$ are predictors and y is the response to predict.

- lm function



# Linear regression

- model<-lm(y~x1+x2+x3, data=dataset
- model summary: summary(model)
- predictions: model %>% predict(test.data)

- model performance(caret package):

  RMSE(predictions, test.data$sales)

demo

# Lab

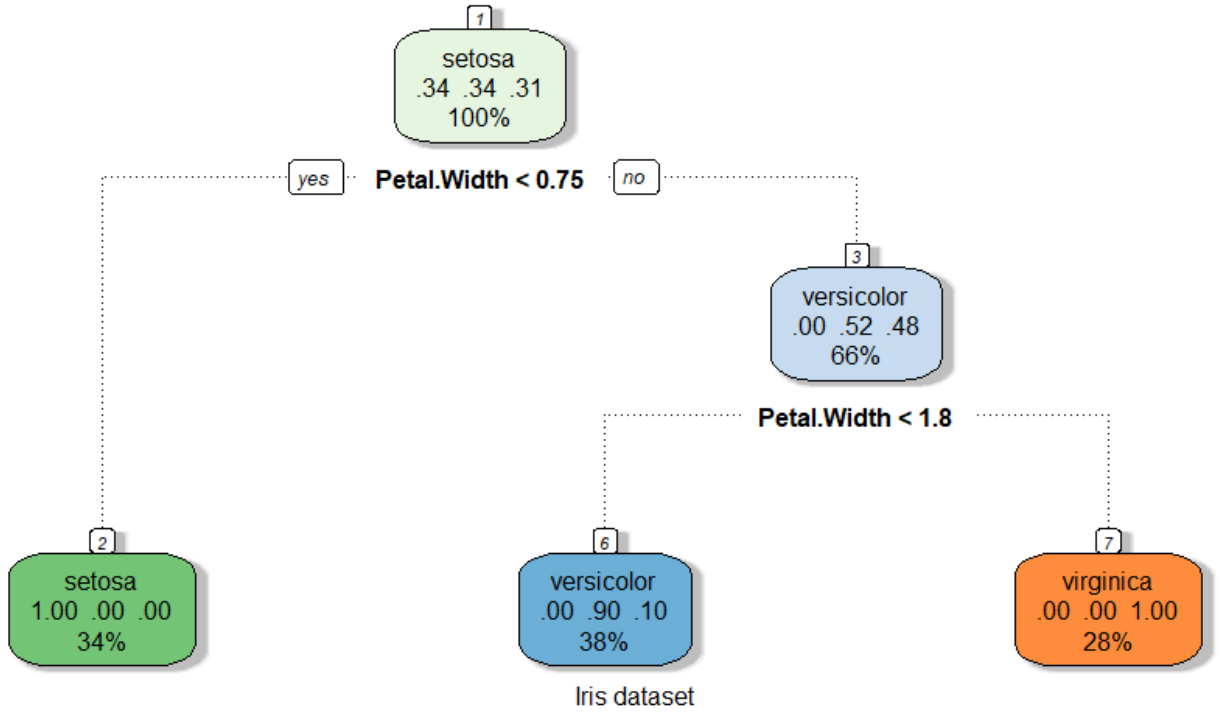
- Create linear regression data mining model for dataset airquality
- Ozone is the dependent variable

- Evaluate the model

# Decision tree

- rpart package, rattle for fancyRpartPlot
- method: class, anova, poisson, exp

- fancyRpartPlot



# Decision tree

- accuracy: TP+TN/TP+FP+FN+TN
- precision: TP/TP+FP
- recall: TP/TP+FN
- F1 score: 2*(Recall * Precision) / (Recall + Precision)

| | | Predicted class | |
|---|---|---|---|
| Actual Class | | Class = Yes | Class = No |
| | Class = Yes | True Positive | False Negative |
| | Class = No | False Positive | True Negative |

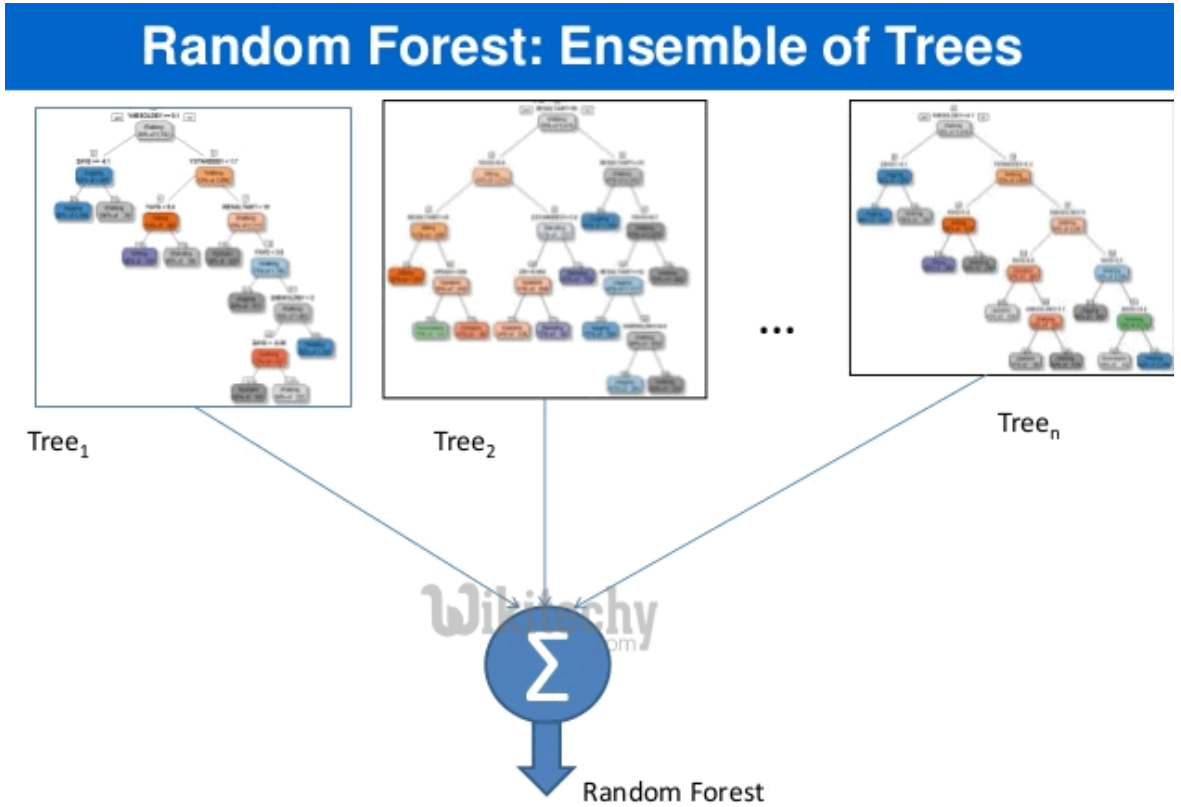
- demo

# Lab

- Import the titanic.csv dataset
- Create decision tree model of survival based on sex, age and pclass.
- Evaluate the model

# Random forest

- randomForest package
- number of decision trees




- varImpPlot: importance of variables
- demo

# Lab

- Import the titanic.csv dataset
- Create random forest model of survival based on sex, age and pclass.
- Evaluate the model