

Introduction to Statistical Learning with R

March 22, 2018

Course Outline

- **What is Statistical Learning**
- **Supervised and unsupervised learning**
- **Regression analysis**
- **Data classification**
 - **Support Vector Machine**
 - **Logistic Regression**
- **Decision trees**
 - **Binary**
 - **Fuzzy logic**
 - **Neural networks**
- **Cluster analysis**
- **Reinforced learning**
- **Informational Entropy**

In Previous Training Sessions ...

We already touched :

- **Exploratory Data Analysis**
- **Regression**
- **Clusters**

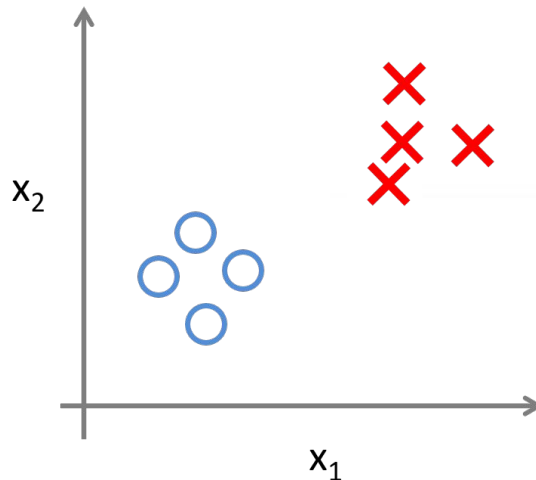
These tools / methods belong to statistical learning

What is Statistical Learning

- **Statistical learning refers to a vast set of tools for understanding data**
- **These tools can be classified as supervised or unsupervised**
- **Supervised statistical learning involves building a statistical model for predicting, or estimating an output based on one or more inputs**
 - **The simplest supervised learning method is a linear regression model which can be used for estimating new input examples**
- **With unsupervised statistical learning, there are inputs but no supervising output; we learn relationships and structure from such data**
 - **The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data**

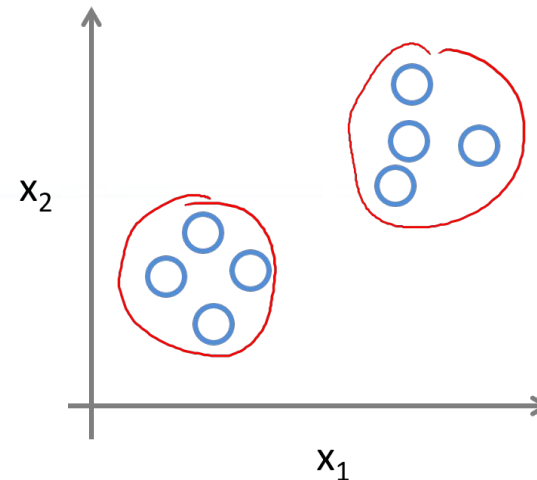
Supervised Vs. Unsupervised Learning

Supervised Learning



Training data is labelled
Categorize new data
based on the training
data

Unsupervised Learning



No training data
Find clusters and look for
other dimensions to infer
their nature

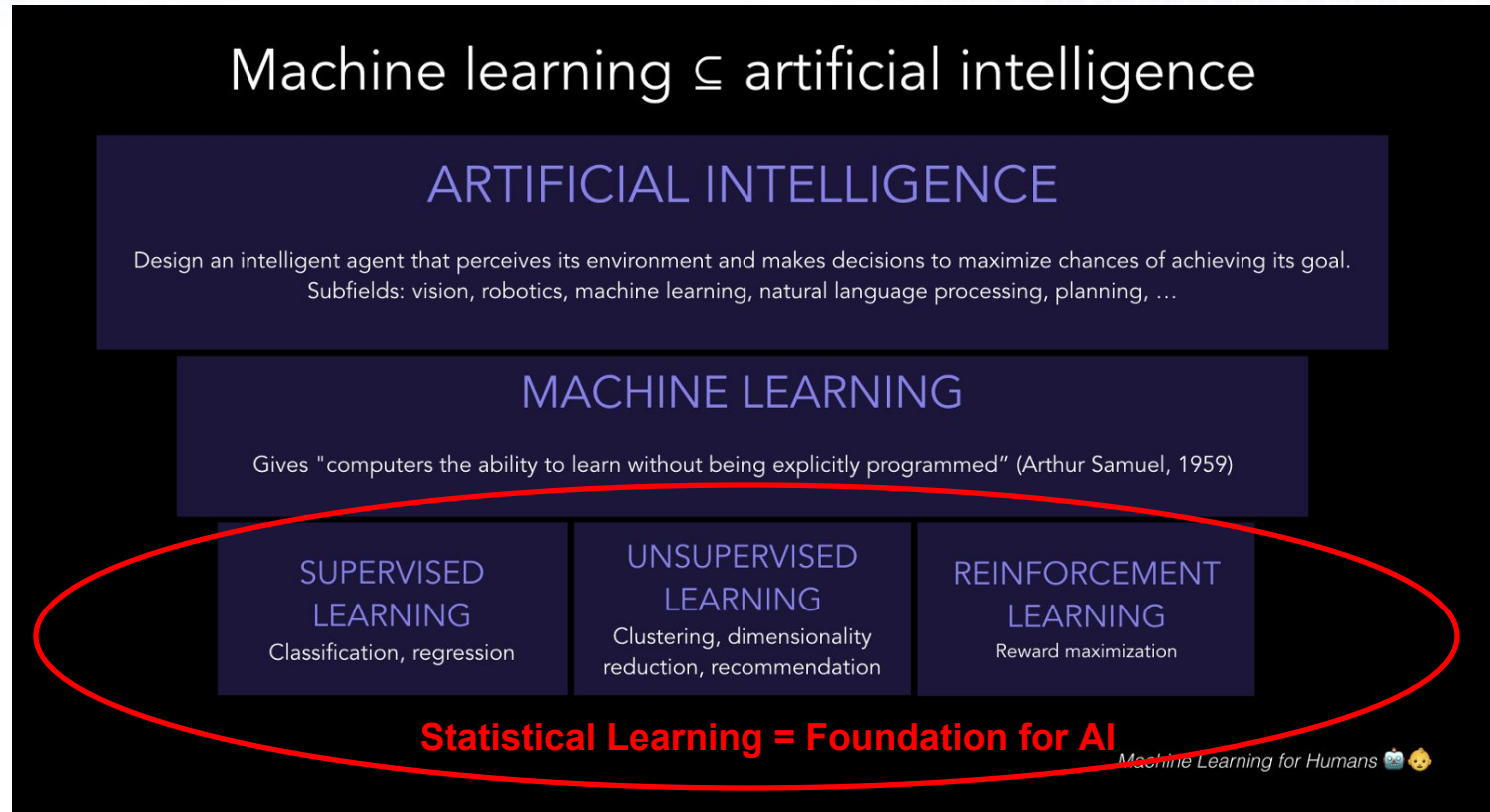
Why Do Statistical Learning

- **Prediction**
 - **Predict how the system works (black box approach)**
- **Inference**
 - **Learn the relations among variables to**
 - **Understand causality**
 - **Optimize the system**

History of Statistical Learning

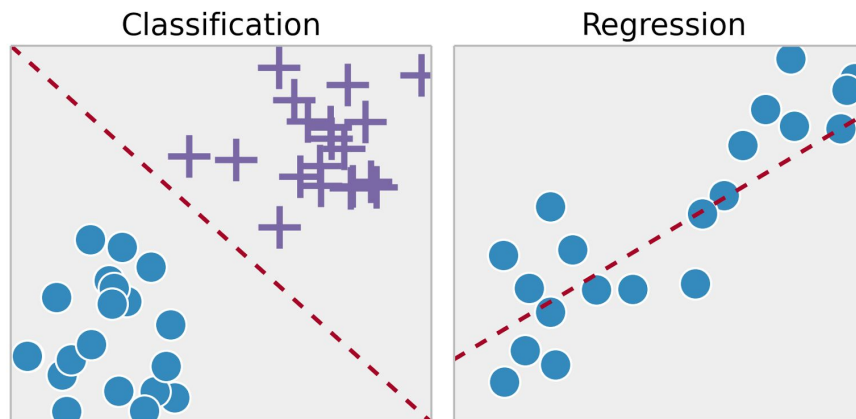
- **Before 1970 – mostly linear methods (not enough computing power)**
 - 1795 – 1805: Least Squares by Gauss
 - 1936: Linear Discriminant by Fisher for qualitative data
 - 1940s: Logistic regression
 - 1970s: Generalized linear model (Non-normal error distribution)
- **1980s: first non-linear methods**
- **2000s: practical machine learning and AI applications**

Statistical Learning, Machine Learning, and AI



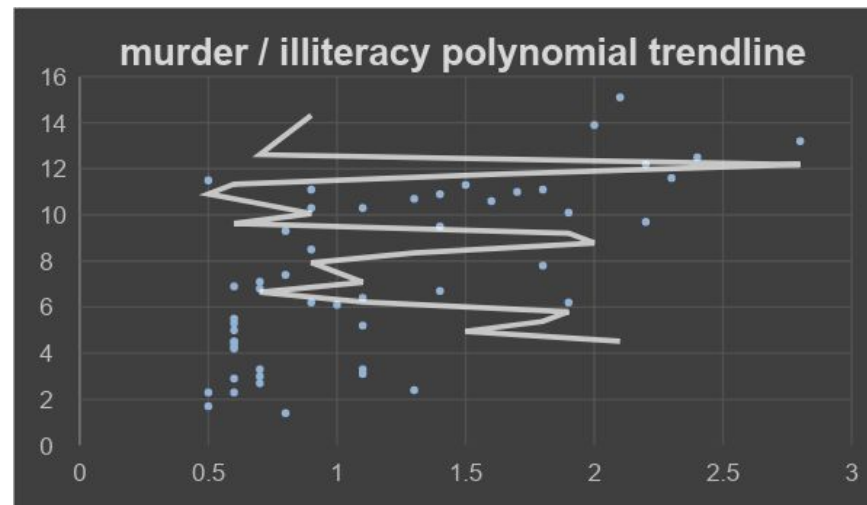
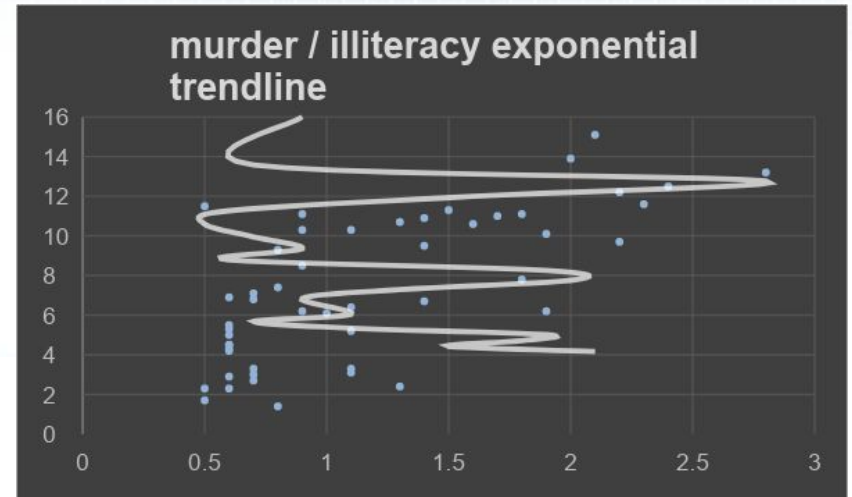
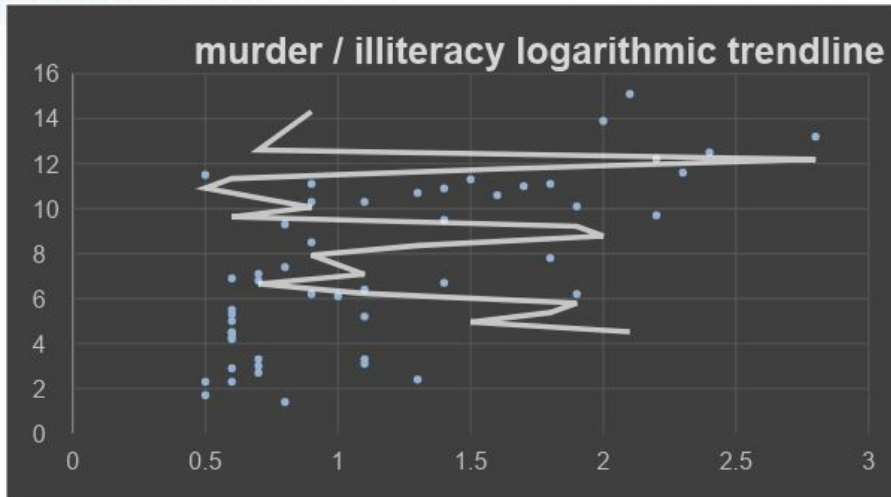
Supervised Learning

- Analyze training data with known outcomes
- Create – Train – Test the Model / Algorithm
- Example
 - Predict income based on education
 - $Y = f(X) + e$ (error)
- For qualitative data => classification (label)
- For quantitative data => regression (predict)



Regression Analysis

Review of the example in Intro to Statistics Part 2

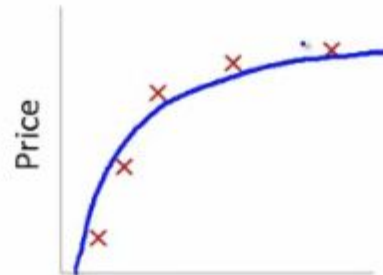


Underfit and Overfit Models



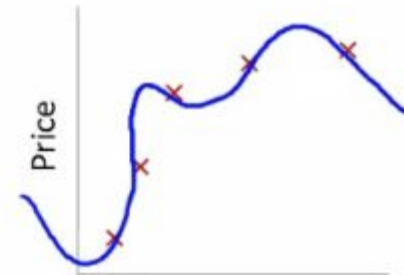
$$\theta_0 + \theta_1 x$$

High bias
(underfit)



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

"Just right"



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

High variance
(overfit)

- Underfit – not accurate enough
- Overfit – not general enough
- Eye test for overfit – too many inflexion points
- Objective test for overfit – high variance

Smoothing Regression

- **Essentially smoothing is a filter (simplest is a low pass like running average)**
- **Typically used when:**
 - **Fitting a line where noisy data values interfere with your ability to see a line of best fit**
 - **Linear regression where least squares fitting doesn't create a line of good fit or is too labor-intensive to use**
 - **Data exploration and analysis in the social sciences, particularly in elections and voting behavior**
- **Benefits of Non-Parametric Smoothing**
 - **Provides a flexible approach to representing data**
 - **Ease of use**
 - **Computations are relatively easy**
- **Disadvantages of Non-Parametric Smoothing**
 - **Can't be used to obtain a simple equation for a set of data**

R Script for Scatter Plot, Linear Regression, and Smoothing

```
## linear regression
# iris data
View(iris)
head(iris, 5)
plot(iris)
plot(iris[1:4])

# linear regression (model)
fit1 = lm(Petal.Length~Petal.Width, data=iris)
summary(fit1)
plot(Petal.Length~Petal.Width, iris)
abline(fit1,col="red")
plot(fit1)

# iris subset setosa
Setosa = subset(iris, Species == "setosa")
plot(Setosa [1:4])
plot(density(Setosa$Petal.Length))
plot(density(Setosa$Sepal.Length))
# non-parametric local regression - filter
scatter.smooth(Petal.Length~Sepal.Length, data = Setosa)
```

Understanding lm() Results

summary(fit1)

Call:
lm(formula = Petal.Length ~ Petal.Width, data = iris)

Residuals:

Min	1Q	Median	3Q	Max
-1.33542	-0.30347	-0.02955	0.25776	1.39453

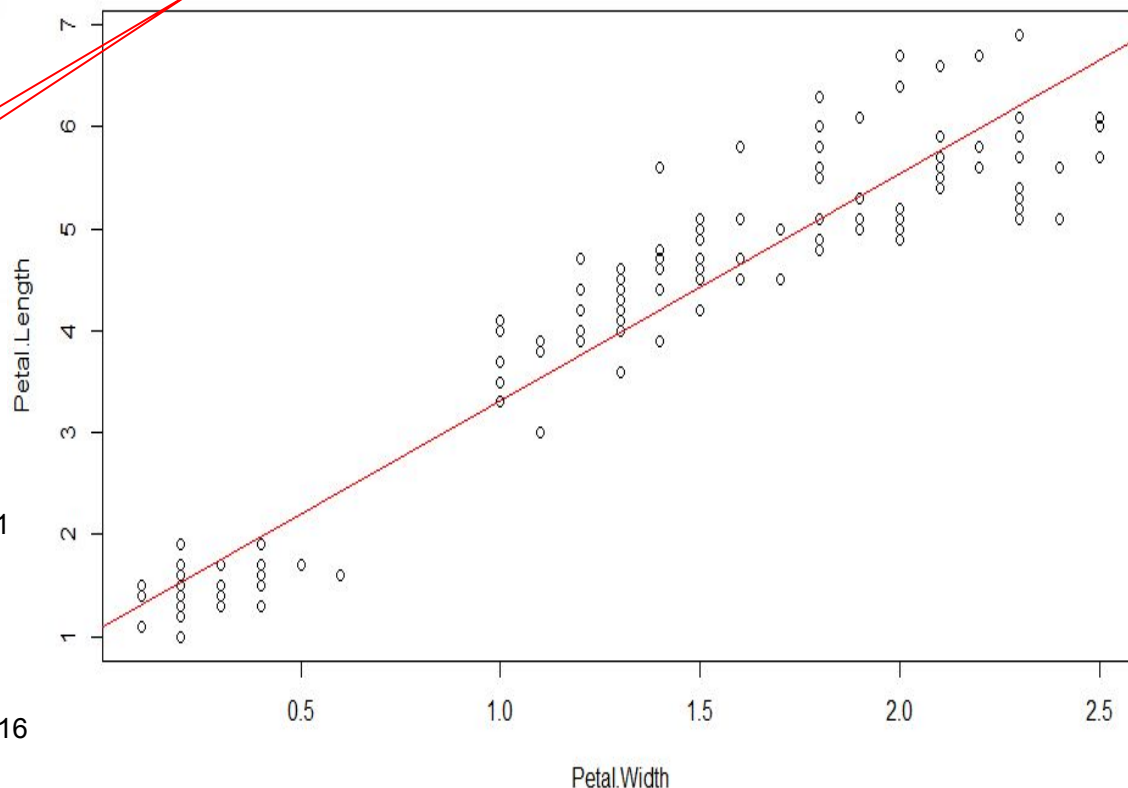
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.08356	0.07297	14.85	<2e-16 ***
Petal.Width	2.22994	0.05140	43.39	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4782 on 148 degrees of freedom
Multiple R-squared: 0.9271,
Adjusted R-squared: 0.9266
F-statistic: 1882 on 1 and 148 DF, p-value: < 2.2e-16

$$\text{Petal.Length} = 2.22994 * \text{Petal.Width} + 1.08356$$



Non-Linear Regression

- Analyst specifies a function with a set of parameters to fit to the data
- The most basic way to estimate such parameters in R is to use a non-linear least squares function
- `nls(formula, data, start, ...)`
 - formula is a nonlinear model formula including variables and parameters.
 - data is a data frame used to evaluate the variables in the formula.
 - start is a named list or named numeric vector of starting estimates

Non-Linear Regression Example

- Data known to be difficult for linear regression
- Michaelis–Menten formula

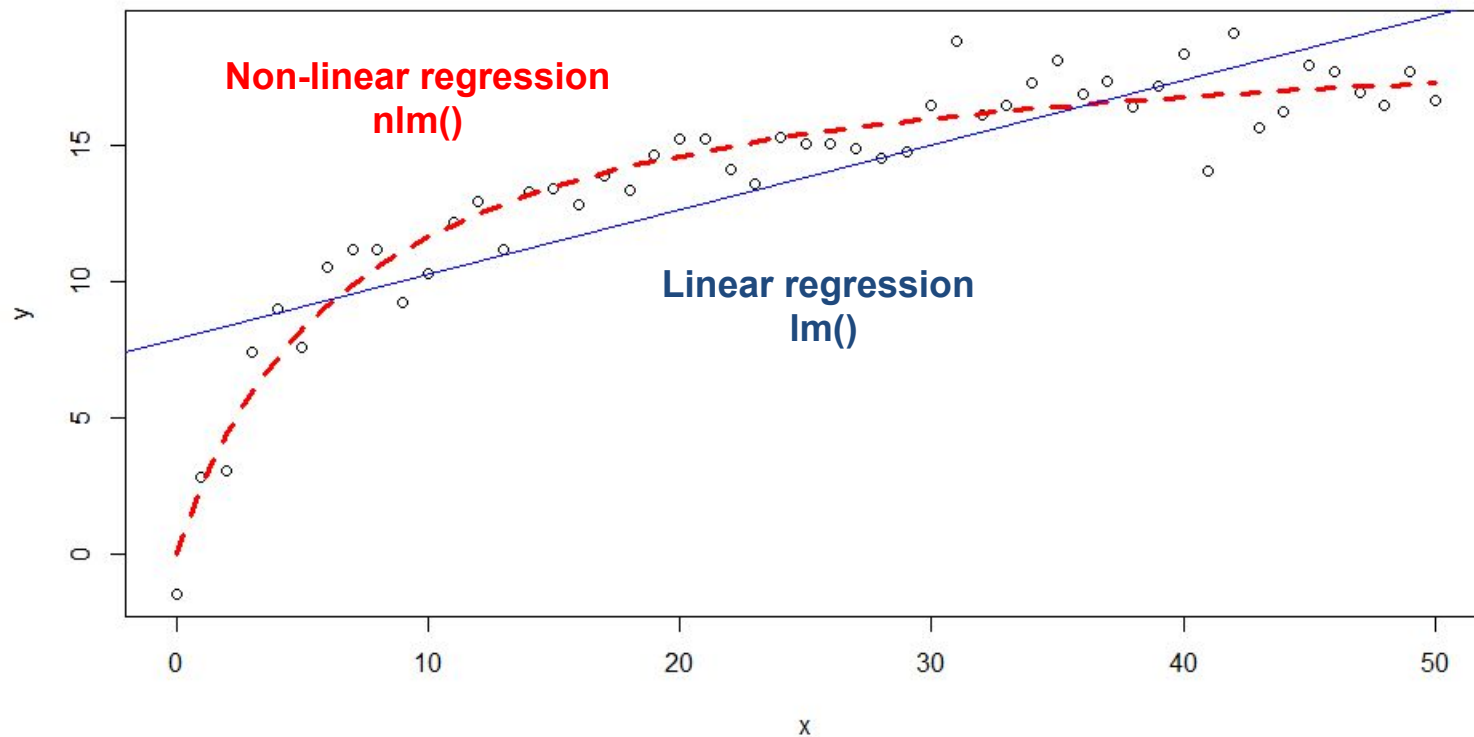
```
## non-linear least square regression
# simulate data

set.seed(20180309)
x<-seq(0,50,1)
y<-((runif(1,10,20)*x)/(runif(1,0,10)+x))+rnorm(51,0,1)
#for simple models nls find good starting values for the parameters even if it throw a warning
m<-nls(y~a*x/(b+x))
#get some estimation of goodness of fit

cor(y,predict(m))

plot(x,y)
lines(x,predict(m),lty=2,col="red",lwd=3)
z = lm(y~x)
abline(z, col = 'blue')
```

lm() and nls() for Michaelis–Menten Data



Linear Regression with ggplot2

- Analyst specifies a function with a set of parameters to fit to the data
- The most basic way to estimate such parameters in R is to use a non-linear least squares approach (nls)
- `nls(formula, data, start, ...)`
 - formula is a nonlinear model formula including variables and parameters.
 - data is a data frame used to evaluate the variables in the formula.
 - start is a named list or named numeric vector of starting estimates

R Script for Regression with ggplot2

```
library(ggplot2)
df = mtcars

# create multiple linear model
lm_fit <- lm(mpg ~ cyl + hp, data=df)
summary(lm_fit)

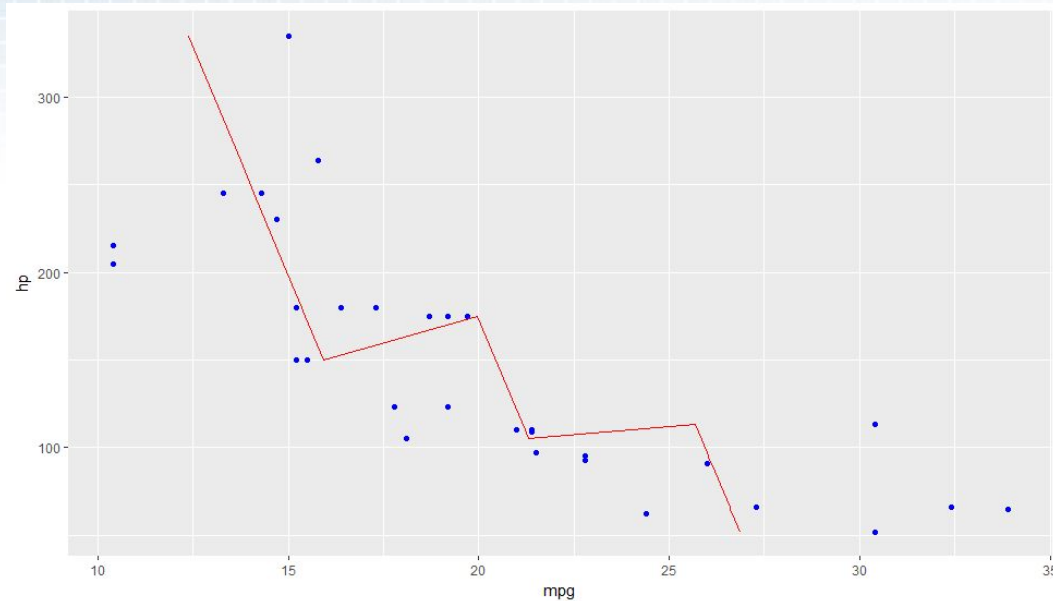
# save predictions of the model in the new data frame
# together with variable you want to plot against
predicted_df <- data.frame(mpg_pred = predict(lm_fit, df), hp=df$hp)

# predicted line of multiple linear regression
ggplot(data = df, aes(x = mpg, y = hp)) +
  geom_point(color='blue') +
  geom_line(color='red', data = predicted_df, aes(x=mpg_pred, y=hp))

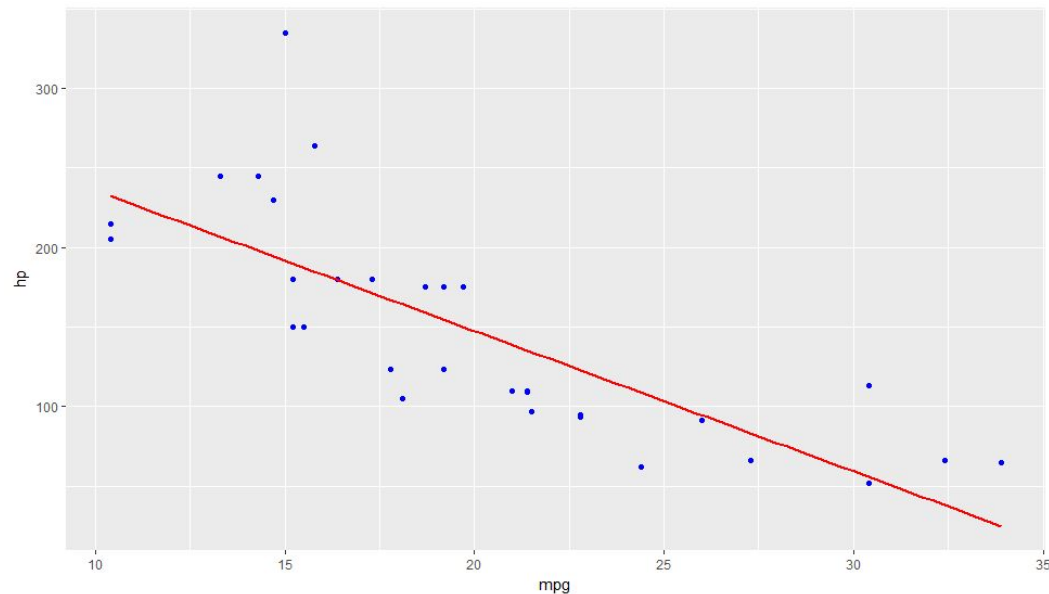
# predicted line comparing only chosen variables
ggplot(data = df, aes(x = mpg, y = hp)) +
  geom_point(color='blue') +
  geom_smooth(method = "lm", col = 'red', se = FALSE)
```

Multiple Linear Regression Vs. Linear Regression

**Multiple Linear
Regression**



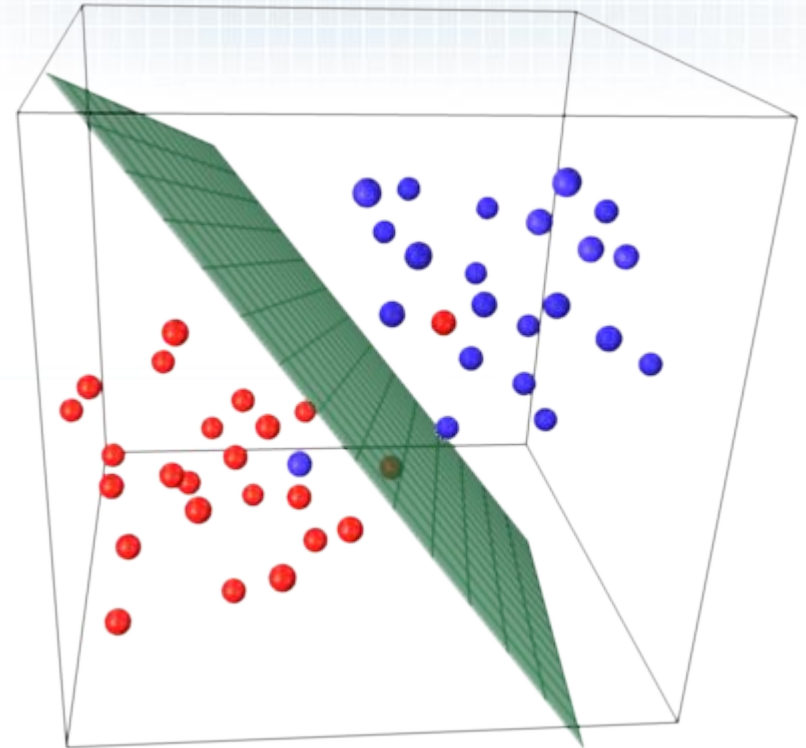
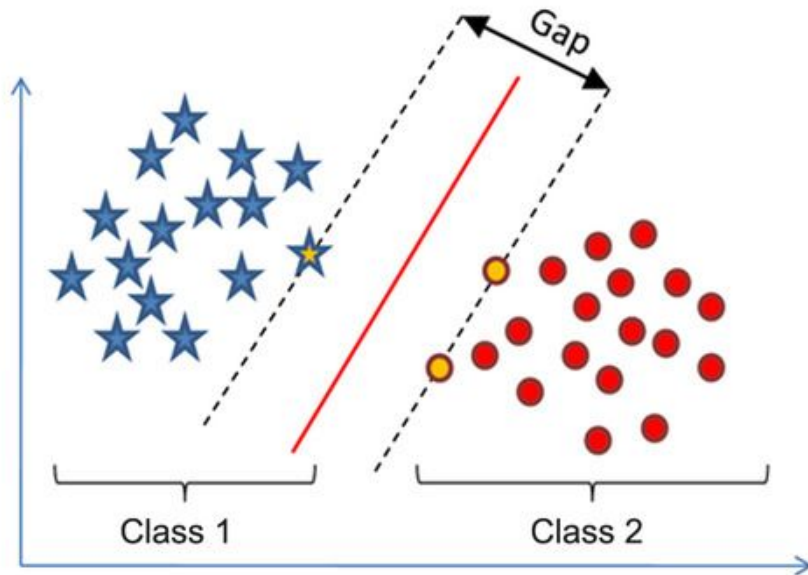
**Linear
Regression**



Question:

**Which one
better shows
the trend?**

Classification with Support Vector Machine (SVM)



- Find a line, curve, plane, surface that separates labelled training data
- Apply to new data to label it

Quiz: What Is a Vector in SVM?

In R

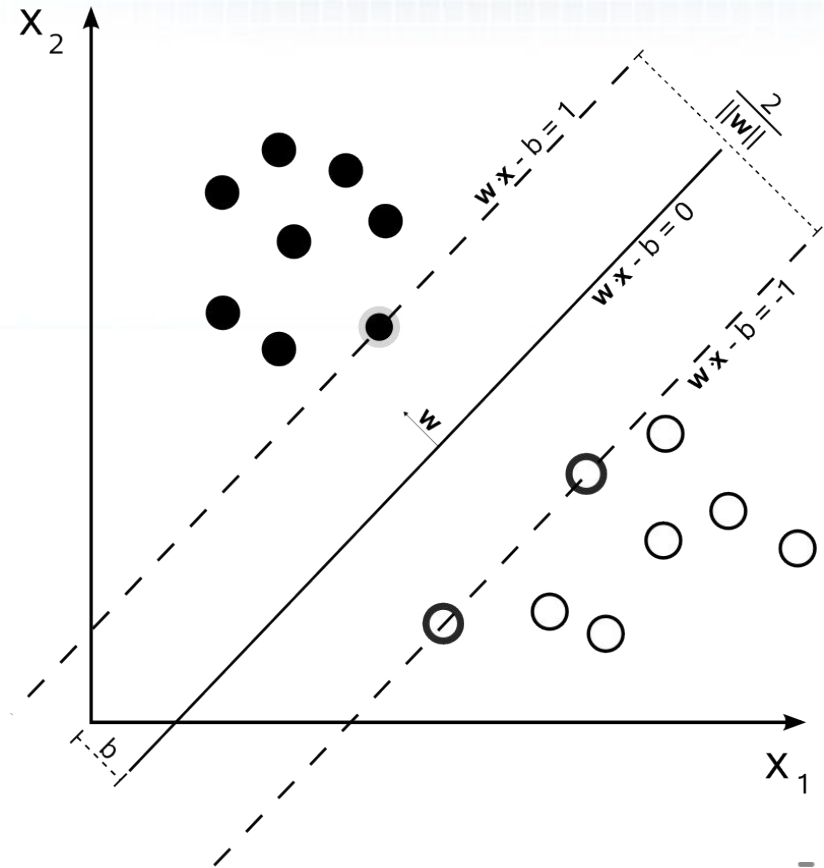
- A vector is a sequence of data elements of the same basic type.

In math:

- A vector is an object that has both a magnitude and a direction: a directed line segment, whose length is the magnitude of the vector and with an arrow indicating the direction.

Question:

- What kind of vector is in SVM?



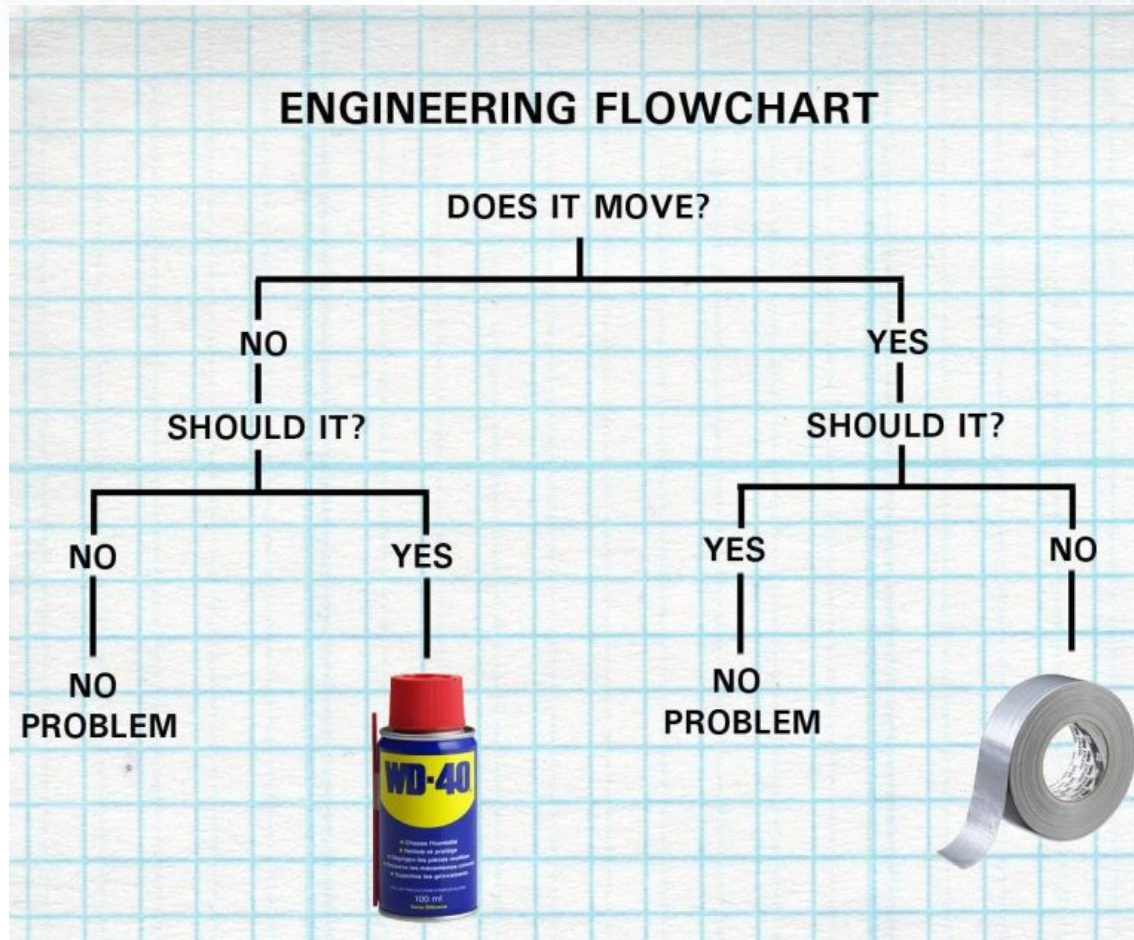
R Script for SVM Example

```
## svm example with iris
library("e1071")
head(iris,5)
attach(iris)
x <- subset(iris, select=-Species)
y <- Species
svm_model <- svm(Species ~ ., data=iris)
summary(svm_model)
# you can do this alternatively
svm_model1 <- svm(x,y)
summary(svm_model1)
# run prediction (and measure execution time)
pred <- predict(svm_model1,x)
system.time(pred <- predict(svm_model1,x))
# see the confusion matrix
table(pred,y)
# tune svm parameters cost and gamma
svm_tune <- tune(svm, train.x=x, train.y=y,
                 kernel="radial", ranges=list(cost=10^(-1:2), gamma=c(.5,1,2)))

print(svm_tune)
# try tuned svm model
svm_model_after_tune <- svm(Species ~ ., data=iris, kernel="radial", cost=1, gamma=0.5)
summary(svm_model_after_tune)
# run prediction with the new model
pred <- predict(svm_model_after_tune,x)
system.time(predict(svm_model_after_tune,x))
# run confusion matrix for the new model
table(pred,y)


# continue perfecting...
```

Logistic Regression with Binary Decision Tree



Logistic Regression

- Regression model for categorical variable
 - Example: binary classification, e.g. Yes/No
- Compare probabilities for Yes and No – odds ratio
- Use log-odds function (or sigmoid to place the value between “0” and “1”)
- Compare result to a threshold



log-odds:
 $\ln[p/(1-p)] = \beta_0 + \beta_1x + \epsilon$

<— IS HE GONNA DIE?

$p = P(\text{Tyrion dies}) = 2/3$

$1-p = P(\text{Tyrion doesn't die}) = 1/3$

odds ratio: $p/(1-p) = 2.0$
"He's gonna die. 2-to-1 odds"

log-odds ratio: $\ln[p/(1-p)] = 0.693$
"He's gonna die. .693 log-odds"

Machine Learning for Humans 🧠🤖

Quiz: Do You Remember Probability

Patient:

Doctor, I just learned that the surgery you are recommending has 10% survival rate!

Doctor:

Do not worry, Sir! The ninth patient I have operated on has just died.

Question 1:

Is it safe now to do the surgery?

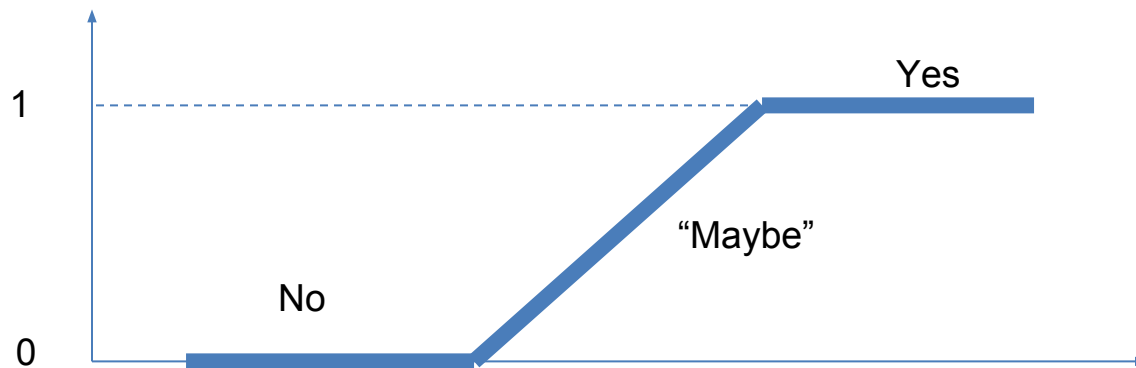
Question 2:

How coin toss is different from taking out (and not putting back) dark and milk chocolate cookie from a jar?

- **Ten times coin toss – does probability of head go up if you have nine tails in a row?**
- **Nine dark cookies and a milk one in a jar – what is a probability of getting the milk cookie after you pulled out and ate nine dark ones?**

Other Types of Decision Engines

- What if you cannot cleanly split your data with a yes/no question?
- What if the answer is maybe?
- For these cases we need human-like reasoning
 - Fuzzy logic
 - Neural networks
- Both turn “maybe” into a probability of Yes or No (“level of truth”)

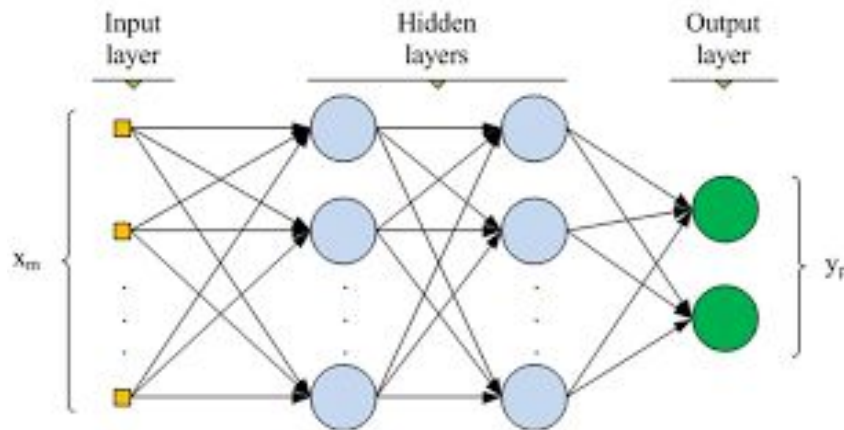
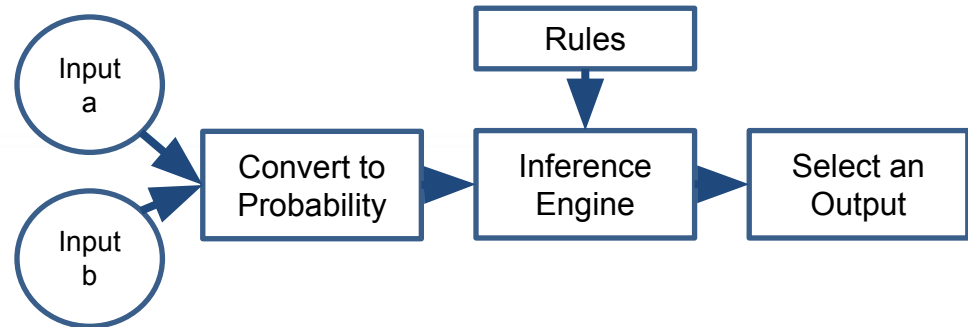


Fuzzy Logic Vs. Neural Network

Both allow to “weigh” your “maybe” based on how much it leans toward yes (or no) If

e.g.: IF $a > b$ AND $b > c$, THEN d

- Fuzzy logic has an inference engine controlled by a set of rules



- Neural network logic is distributed over hidden layer(s) nodes/perceptrons
- Emulates the human neurons

Neural Network Example

- **Algorithm which can guess if a college is private or public based on admission and graduation data**
- **Dataset College from ISLR package**
 - All numeric-value columns
 - Except for one categorical column: Private Yes/No
- **Steps**
 - **Pre-process the data:**
 - Normalize
 - Convert Yes/No to 1/0
 - **Split the data into two sets:**
 - Training
 - Test
 - **Apply neural network package “neuralnet”**
 - **Train the algorithm**
 - **Assess results (confusion matrix)**

R Script for Data Classification Example with Neural Network

```
## create neural network example
library(ISLR)
print(head(College,2))

# Create Vector of Column Max and Min Values
maxs <- apply(College[,2:18], 2, max)
mins <- apply(College[,2:18], 2, min)

# Use scale() and convert the resulting matrix to a data frame
scaled.data <-
as.data.frame(scale(College[,2:18],center =
mins, scale = maxs - mins))

# Check out results
print(head(scaled.data,2))

# split the data into training and test datasets
# Convert Private column from Yes/No to 1/0
Private = as.numeric(College$Private)-1
data = cbind(Private,scaled.data)

library(caTools)
set.seed(101)

# Create Split (any column is fine)
split = sample.split(data$Private, SplitRatio =
0.70)

# Split based off of split Boolean Vector
train = subset(data, split == TRUE)
test = subset(data, split == FALSE)
```

```
# create a formula to insert into the machine learning model
# simple script to create the expanded formula and save us some typing

feats <- names(scaled.data)

# Concatenate strings
f <- paste(feats,collapse=' + ')
f <- paste('Private ~',f)

# Convert to formula
f <- as.formula(f)

f

# install package neuralnet
library(neuralnet)
nn <- neuralnet(f,train,hidden=c(10,10,10),linear.output=FALSE)

# Compute Predictions off Test Set
predicted.nn.values <- compute(nn,test[2:18])

# Check out net.result
print(head(predicted.nn.values$net.result))

predicted.nn.values$net.result <-
sapply(predicted.nn.values$net.result,round,digits=0)

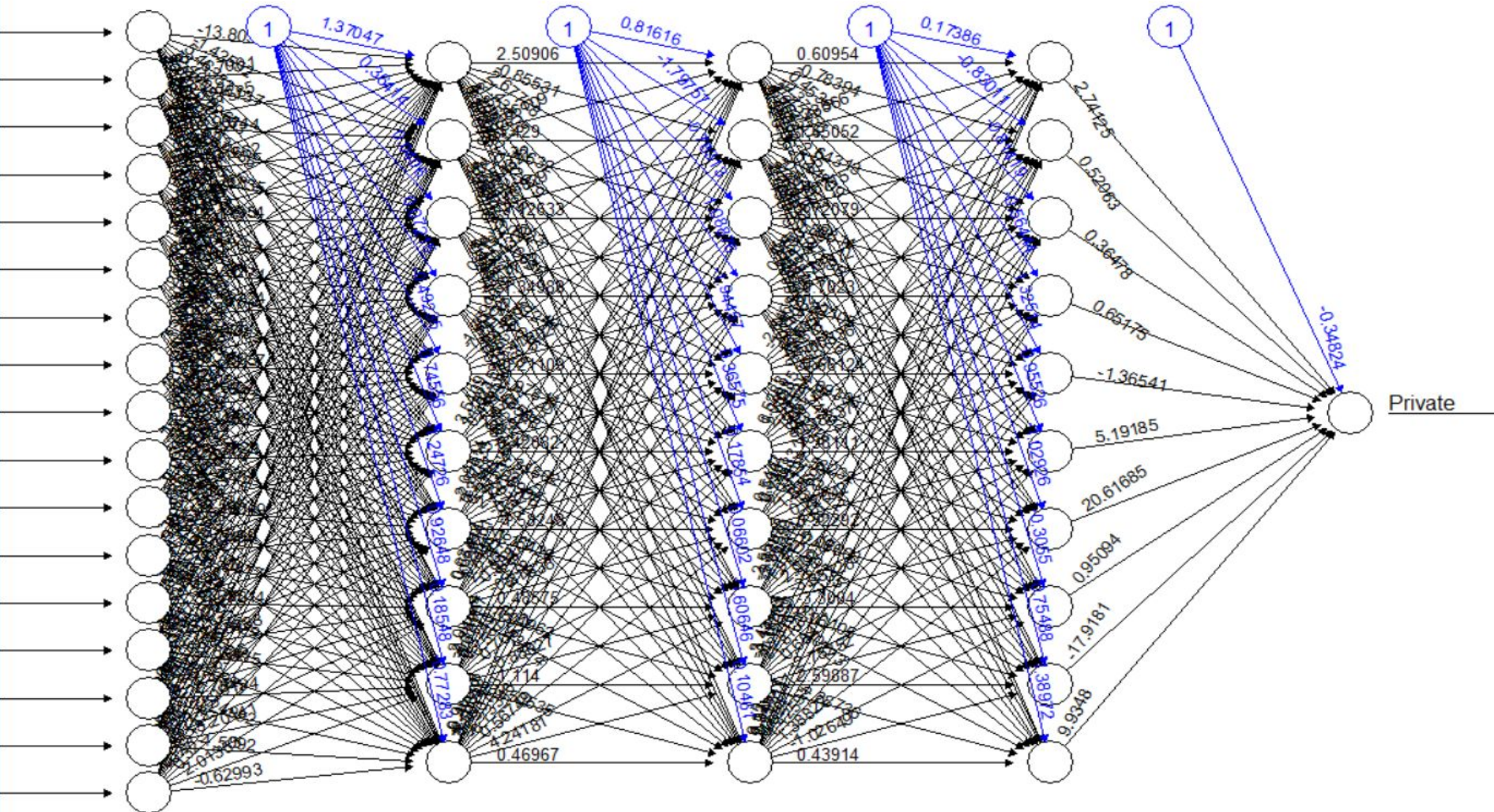
table(test$Private,predicted.nn.values$net.result)

plot(nn)
```


Neural Network to Categorize College as Private or Public

R Graphics: Device 4 (ACTIVE)

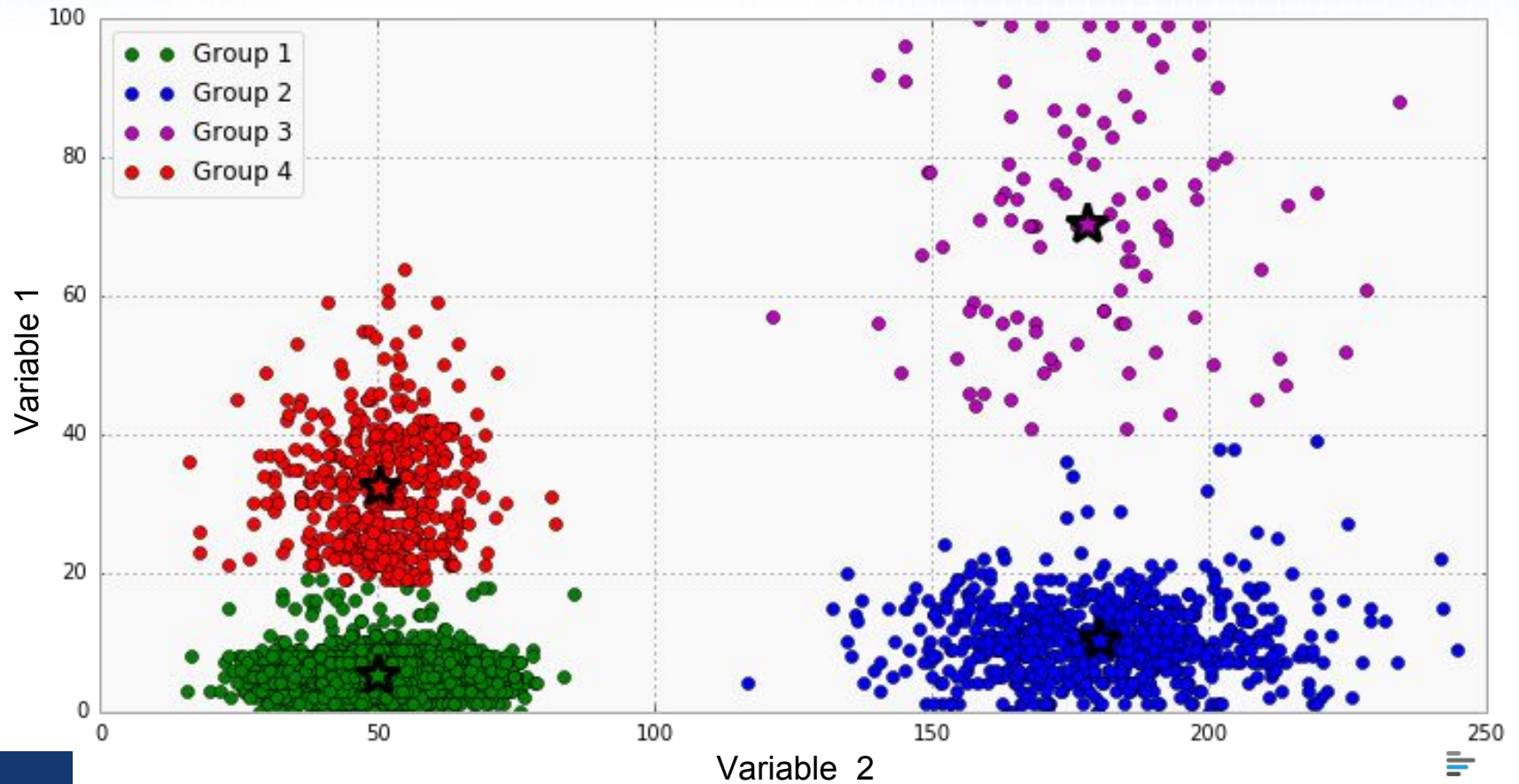
File History Resize



Cluster Analysis

Grouping of similar objects in multivariate data set

Example: grouping by distance from center



R Script for k-means

(Review “From Intermediate R Training Part 2”)

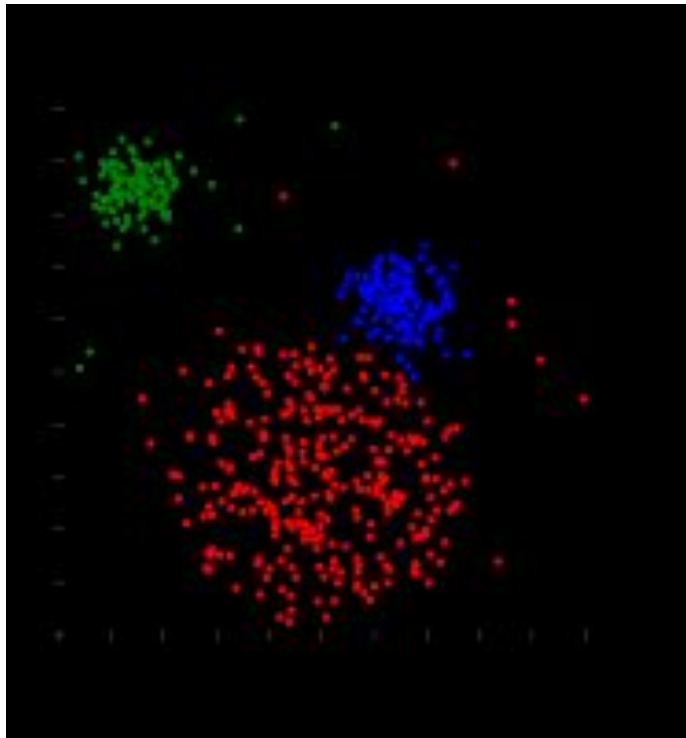
```
library(ggplot2)
ggplot(iris, aes(Petal.Length, Petal.Width, color = Species)) + geom_point()
set.seed(20)
irisCluster <- kmeans(iris[, 3:4], 3, nstart = 20)
irisCluster
table(irisCluster$cluster, iris$Species)
irisCluster$cluster <- as.factor(irisCluster$cluster)
ggplot(iris, aes(Petal.Length, Petal.Width, color = irisCluster$cluster)) + geom_point()
```

Or:

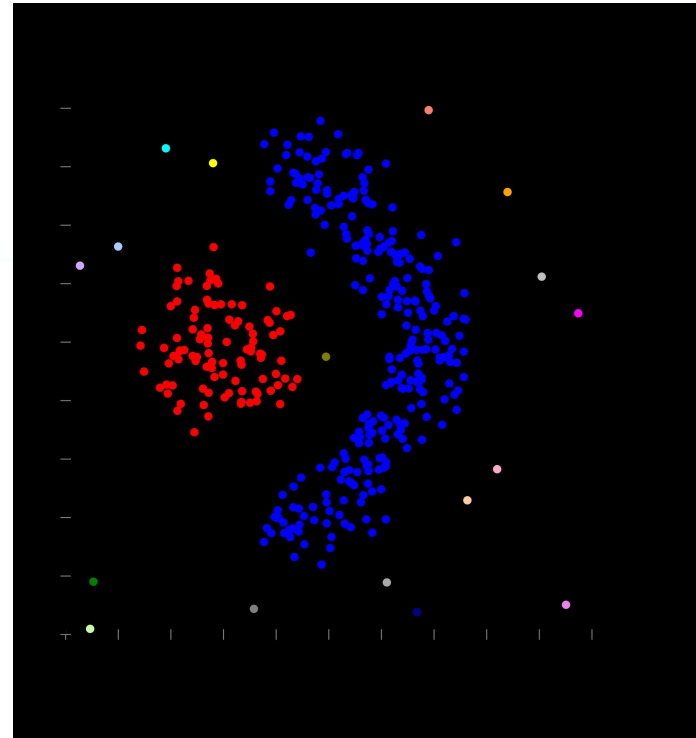
```
ggplot(iris, aes(y = Petal.Length, x = seq(1, length(iris$Sepal.Length)), color =
irisCluster$cluster)) + geom_point()
irisCluster1 = kmeans(iris[,3], 3, nstart = 20)
irisCluster1
table(irisCluster1$cluster, iris$Species)
```

Distance-based Clusters

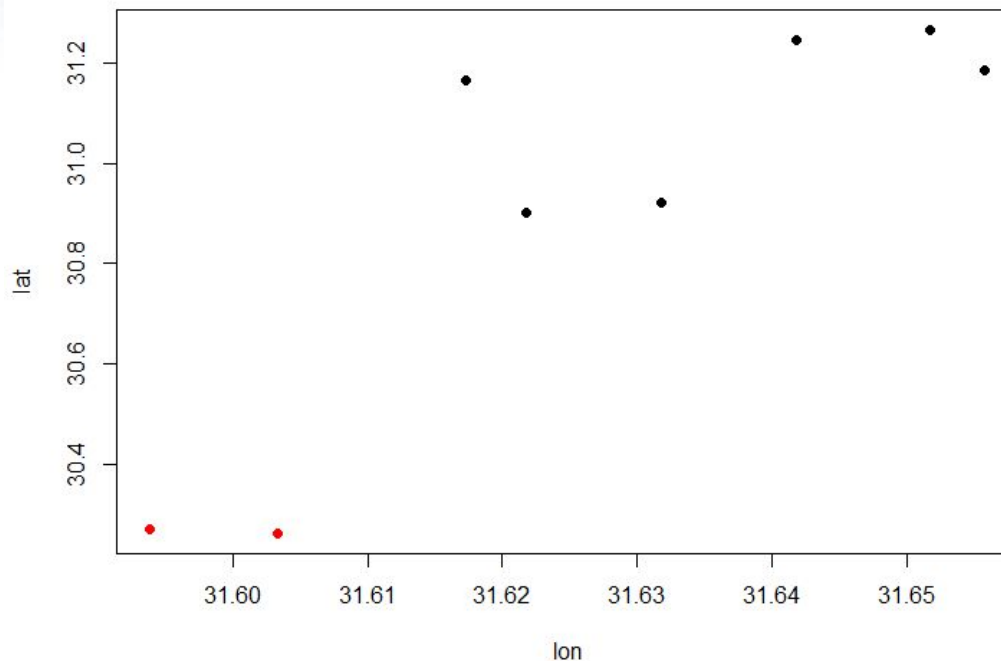
Centroid Clusters
(distance to center)



Curved Line Clusters
(distance to line)



Spatial Clustering Example w/ fields Package



```
library(fields)
lon = c(31.621785, 31.631785, 31.641773, 31.651785, 31.617269, 31.655785, 31.593895, 31.603284)
lat = c(30.901118, 30.921118, 31.245008, 31.265008, 31.163886, 31.183886, 30.27058, 30.262378)
threshold.in.km <- 40
coors <- data.frame(lon,lat)
```

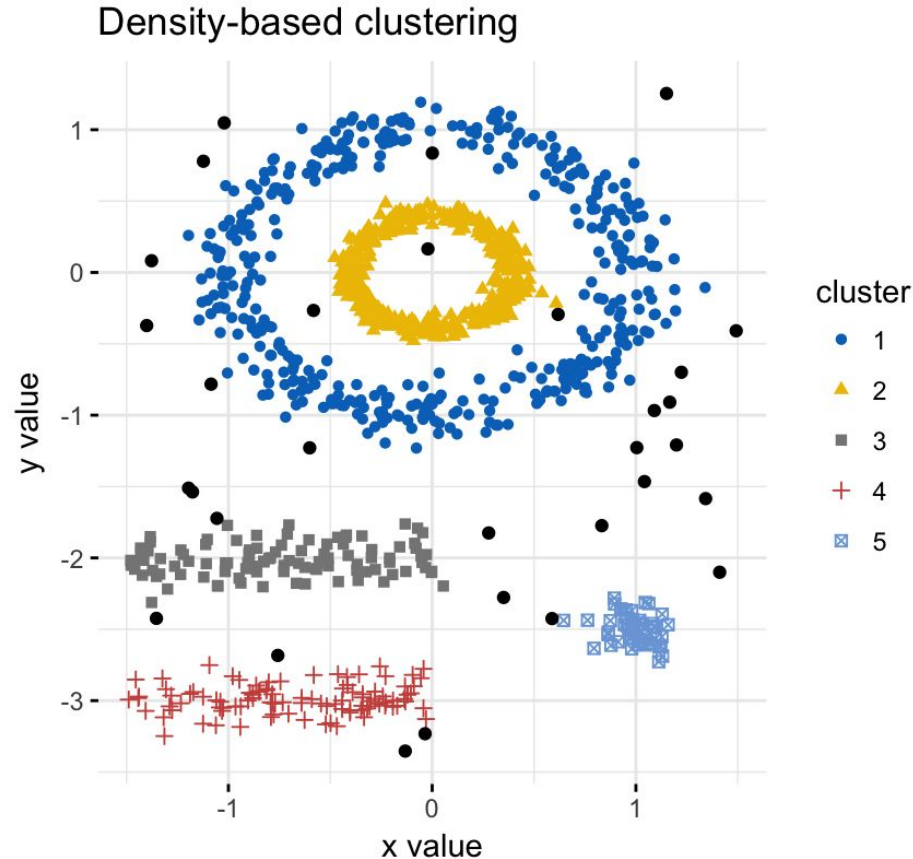
```
#distance matrix
dist.in.km.matrix <- rdist.earth(coors,miles = F,R=6371)
```

```
#clustering
fit <- hclust(as.dist(dist.in.km.matrix), method = "single")
clusters <- cutree(fit,h = threshold.in.km)
```

```
plot(lon, lat, col = clusters, pch = 19)
```

Density-based Clusters

- Distance to adjacent data points



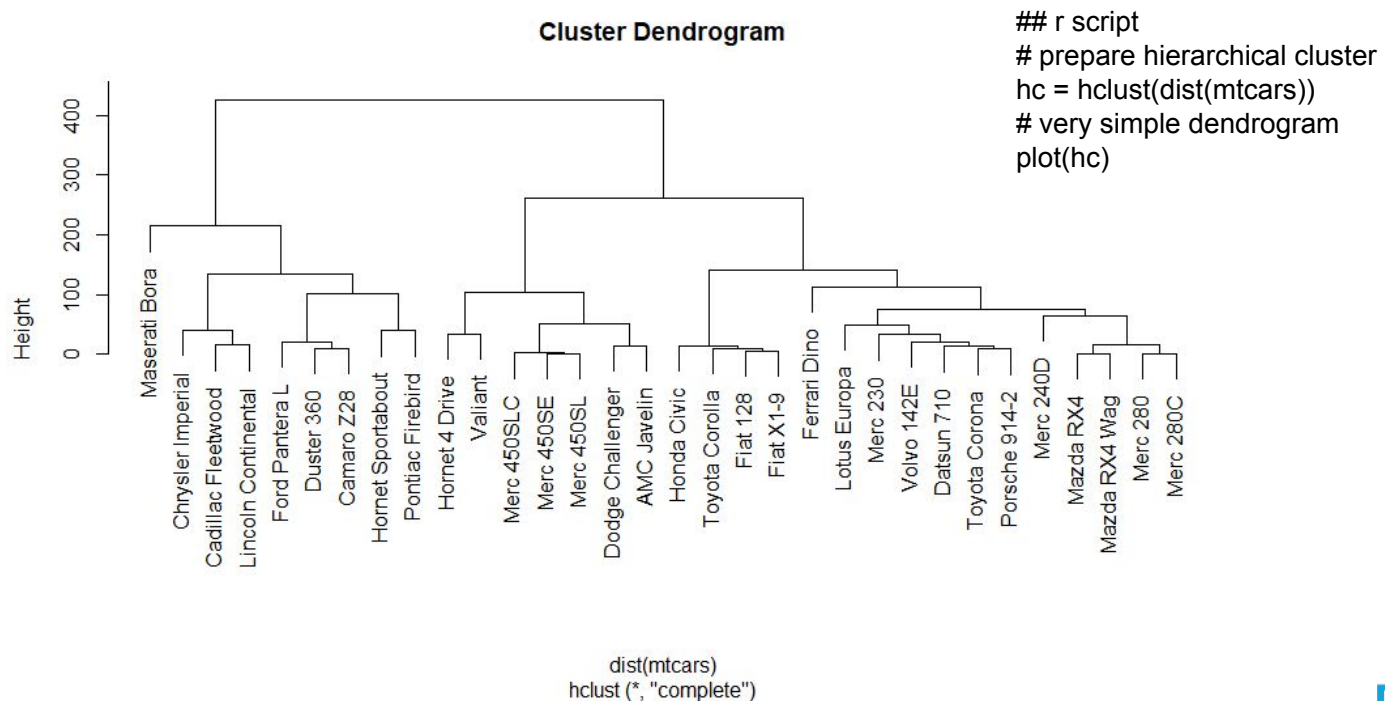
```
# rscript
```

```
install.packages("fpc")  
install.packages("dbscan")  
install.packages("factoextra")  
install.packages("modeltools")
```

```
# Load the data  
data("multishapes", package = "factoextra")  
df <- multishapes[, 1:2]  
# Compute DBSCAN using fpc package  
library("fpc")  
set.seed(123)  
db <- fpc::dbscan(df, eps = 0.15, MinPts = 5)  
# Plot DBSCAN results  
library("factoextra")  
fviz_cluster(db, data = df, stand = FALSE,  
              ellipse = FALSE, show.clust.cent = FALSE,  
              geom = "point", palette = "jco", ggtheme =  
              theme_classic())
```

Hierarchical Clustering: Dendrogram

- Dendrogram is a visual representation of compound-correlation data
- The vertical axis Height refers to a distance measure between compounds or compound clusters
- The height of the node can be thought of as the distance value between the right and left sub-branch clusters



Clustering Tendency

- Clustering algorithms, including partitioning methods (K-means, PAM, CLARA and FANNY) and hierarchical clustering, are used to split the dataset into groups or clusters of similar objects
- Before applying any clustering method on the dataset, a natural question is:

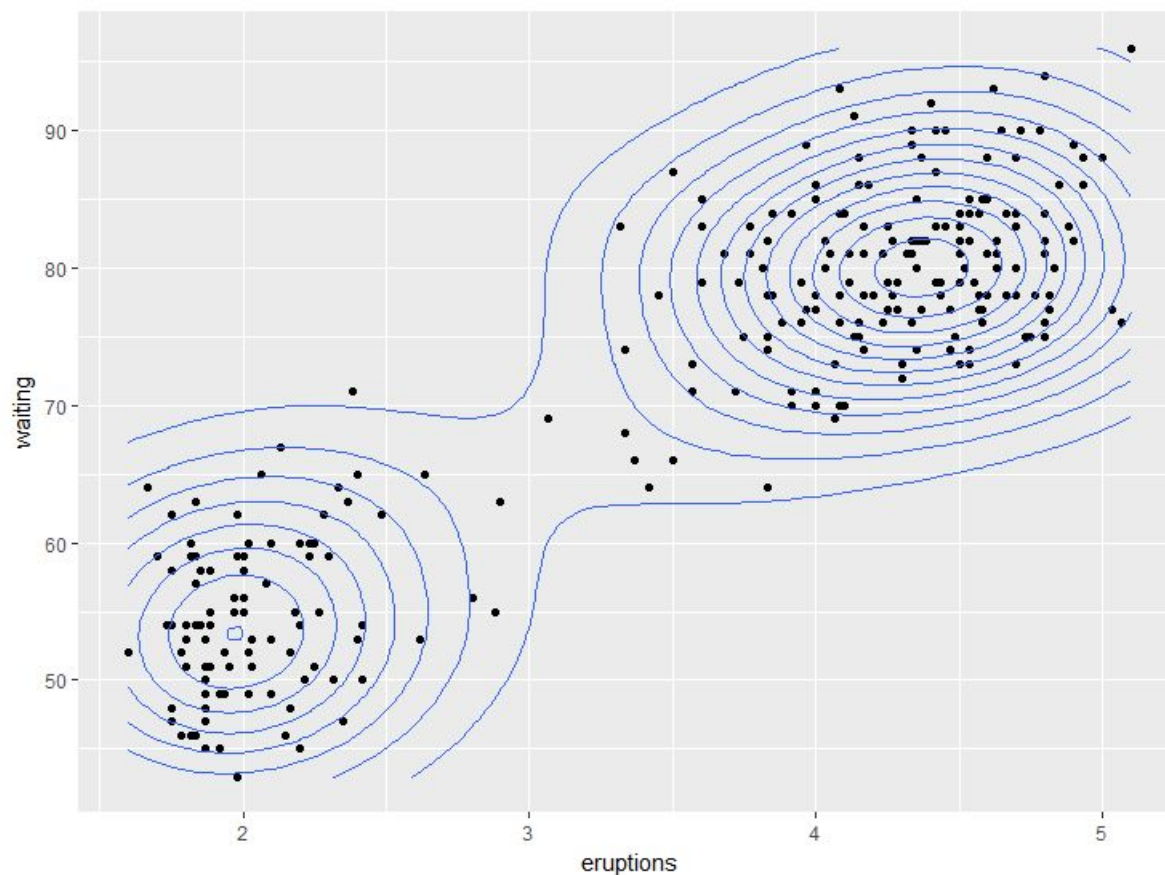
Does the dataset contain any inherent clusters?

- A big issue is that clustering methods will return clusters even if the data does not contain any clusters
- if you blindly apply a clustering analysis on a dataset, it will divide the data into clusters because that is what it supposed to do
- Therefore, before choosing a clustering approach, an analyst has to decide whether the dataset contains meaningful clusters (i.e. nonrandom structures) or not, and, if yes, then how many clusters are there
- This process is defined as the assessing of clustering tendency or the feasibility of the clustering analysis

<http://www.sthda.com/english/wiki/print.php?id=238>

Clustering Tendency with ggplot2

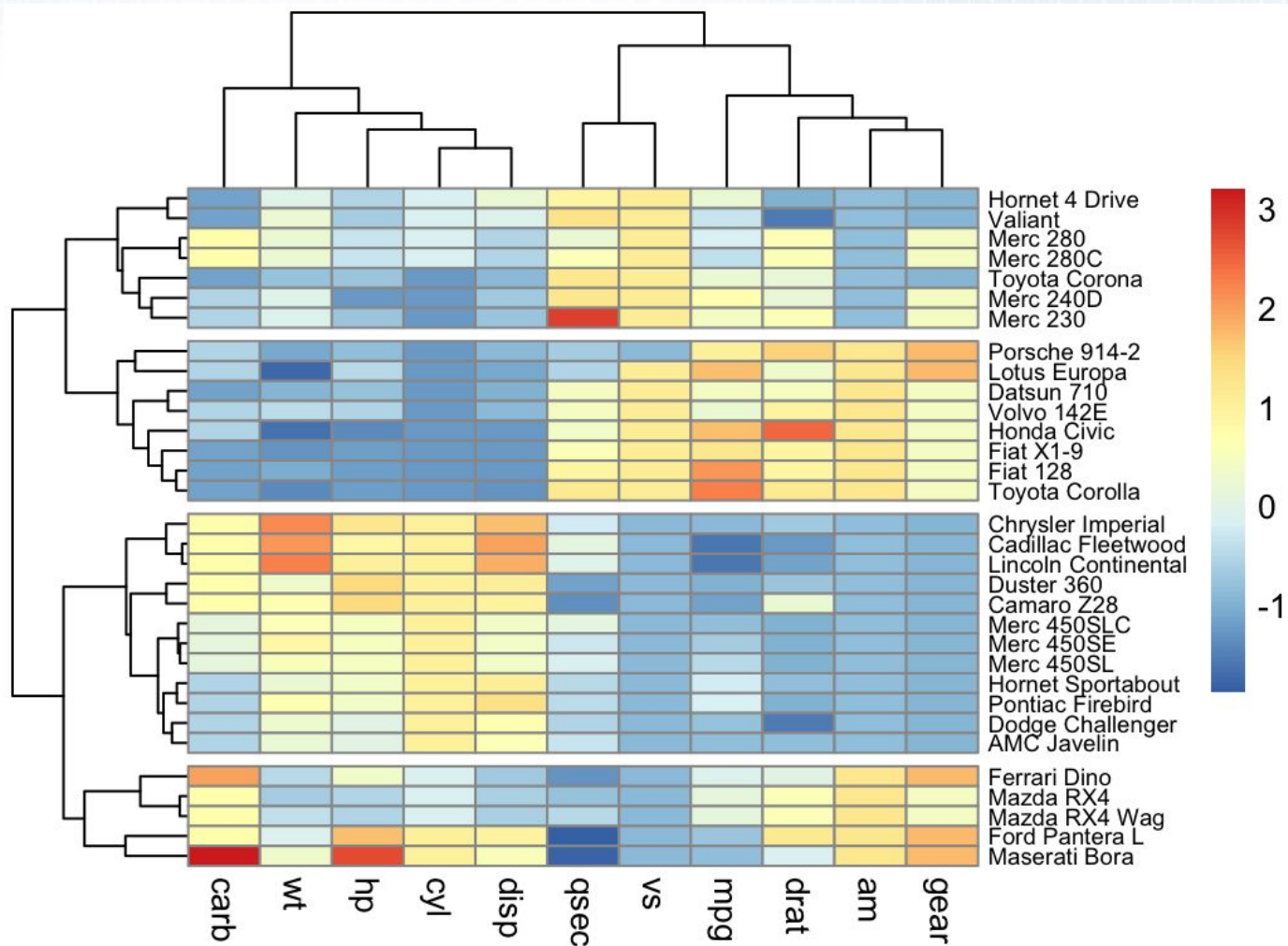
- **geom_density** parameter – a smoothed version of histogram
- **Creates density isolines**
- **Example uses “faithful” dataset in R built in**



```
## r script
## Clustering tendency with ggplot
data("faithful")
df <- faithful

library("ggplot2")
ggplot(df, aes(x=eruptions,
y=waiting)) +
  geom_point() + # Scatter plot
  geom_density_2d() # Add 2d density
estimation
```

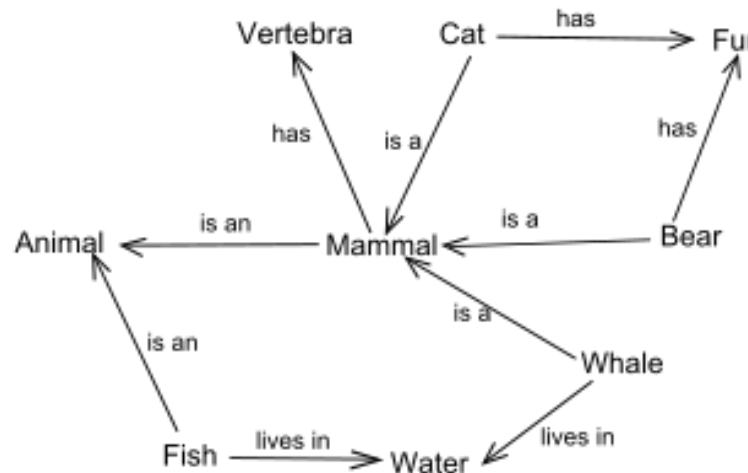
Clustering Tendency with Heatmap and Dendrogram



Word Clusters

by Semantic Similarity, Relatedness, and Distance

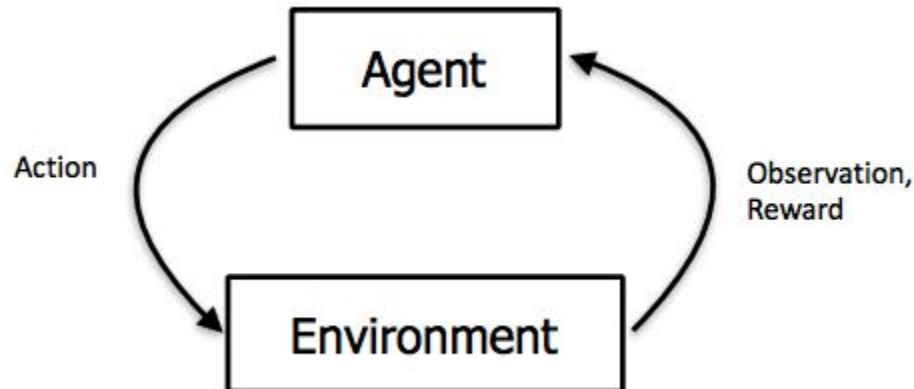
- **Similarity (likeness, synonymy)**
 - Bank – trust
- **Relatedness**
 - Car – wheel
 - Pencil – paper
- **Distance: inverse of Similarity or Relatedness**
- **Distance can be computed using topology tree**



- **Once we have semantic distances, we can create semantic clusters**
- **R packages NLTK, wordnet**

Reinforced Learning

- **Extension of supervised learning**
- **Based on Markov Decision Process**
- **An outcome value for an action (input value) is not explicitly provided (therefore, an error is not computed)**
- **Instead, the system (“environment”) “rewards” “agent’s” actions towards a desirable outcome and “punishes” actions taking an away from a goal**



Information Content of a Variable

- Regression allow to explore relationships among variable
- What about info contained in a variable?
- We can assess a variable
 - Histogram
 - Factor variable (e.g. list all values a variable can take)
<https://stats.idre.ucla.edu/r/modules/factor-variables/>
- Question: is there a way to measure amount of information in a variable?

Amount of Information

- We can count number of bytes, but this would be rather size than amount of information
- Informational Entropy offers a way to measure the amount of info contained in a variable
- Entropy depends on probability of a variable to take certain value
- For a discrete variable, entropy is a function of frequency with which the variable is taking certain values
- The more values the variable can take, the high is the entropy of the variable

Informational Entropy

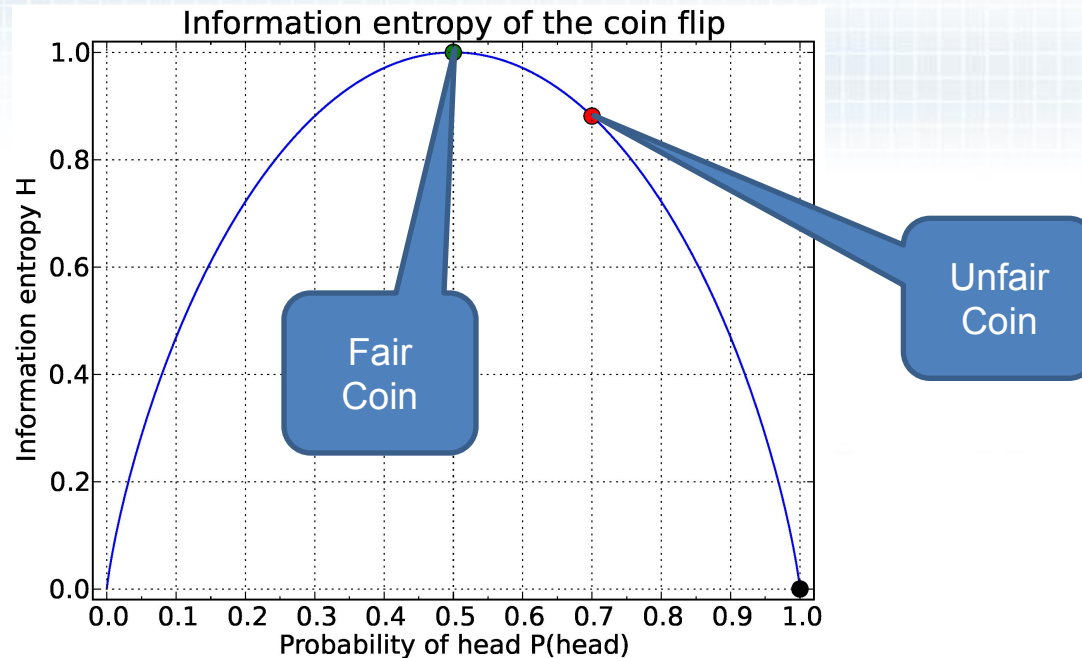
- Came into statistics from information science
- Is a measure of unpredictability (uncertainty) of a variable

$$H(X) = - \sum_{i=1}^n P(x_i) \log_b P(x_i)$$

- $P(x_i)$ – probability of a value of a discrete variable
- E.g.: (Frequency of value)/(Number of values)
- Entropy has a range from 0 to 1
- If the data is completely homogeneous (all samples are same), the entropy is zero
- If the data is totally random, the entropy is 1

Entropy of a Coin Toss

(Binary Function)



Claude Shannon definition of entropy:

$$H(X) = -Q(Y) \log_2 Q(X) - P(X) \log_2 P(X)$$

Think of a coin toss as a variable with two outcomes: heads $Q(X)$ or tails $P(X)$

\log_2 – defines entropy units - bits (can also be in Ln - *nats* and Lg - *bans*)

Probability 0 or 1: Entropy = 0, homogeneous data (“data purity”)

Probability 0.5: Entropy = 1, equally divided data

Why Shannon Used Logarithmic Function in Entropy?

- Why is $\log()$ often used in theory of probability?
 - Short answer: because it works
 - Why does log work well with probabilities?
- Conditional probability (probability of A if B already happen)
 - $P(A|B) = P(A) * P(B)$
 - $P(A|B|C) = P(A) * P(B) * P(C)$
- (Computing permutations also require multiplication)
- Basic property of logarithms
 - $\text{Log}(P(A) * P(B)) = \text{Log}(P(A)) + \text{Log}(P(B))$
- Addition is much easier to work with than multiplication

Information Gain

- The information gain is based on the decrease in entropy after a dataset is split on an attribute

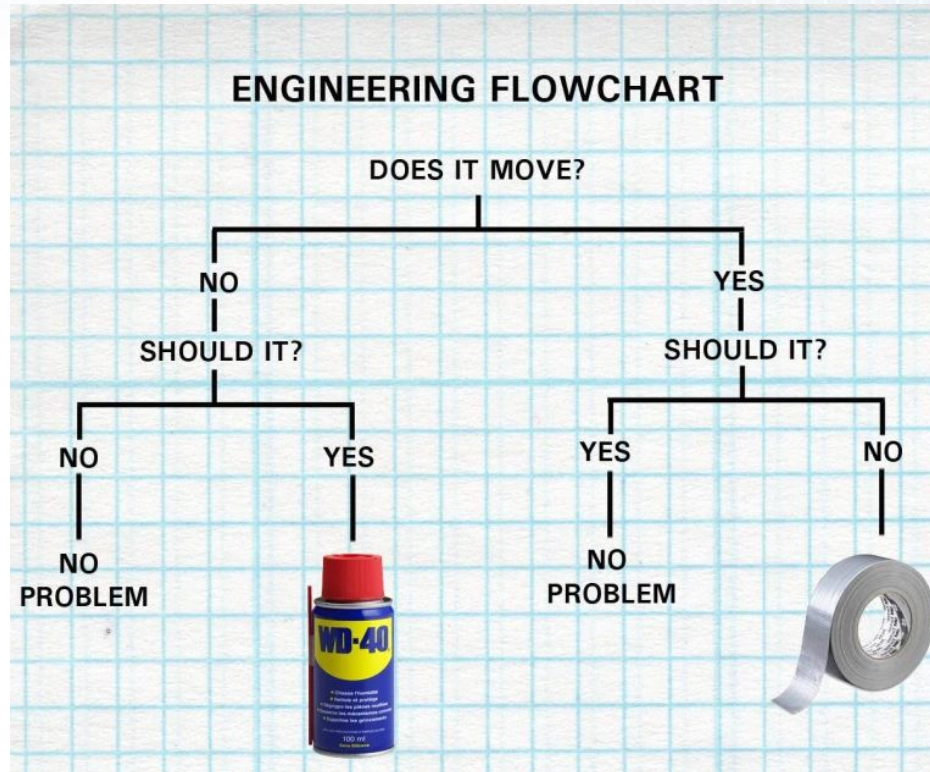
$$\text{Information Gain} = \text{Entropy}(\text{parent}) - \text{Entropy}(\text{children})$$

- Find an attribute that returns the highest information gain for an attribute you are interested in

Split on attribute A, so you get max gain on attribute

B

Entropy Consideration for Binary Decision Tree



- Split on attribute “Does it move?”
- Analyze gain on attribute “Should it move?”

Information Gain Example

Using Frequencies of Attribute Values

	Object Name	Does it Move (DIM)	Should it Move (SIM)	
Subset ABC DIM?=Y	A	Y	N	1 out of 3
	B	Y	Y	2 out of 3
	C	Y	Y	
Subset DEFG DIM?=N	D	N	Y	3 out of 4
	E	N	Y	
	F	N	Y	
	G	N	N	1 out of 4

Summary of frequencies from the table:

- For DIM=Y (Subset ABC): 3 out of 7 objects (A, B, C) have DIM=Y. Of these, 2 out of 3 (B, C) should move (SIM=Y).
- For DIM=N (Subset DEFG): 4 out of 7 objects (D, E, F, G) have DIM=N. Of these, 3 out of 4 (D, E, F) should move (SIM=Y), and 1 out of 4 (G) should not move (SIM=N).

- Entropy of the full set ABCDEF on DIM attribute =
 $-(3/7)\log_2(3/7) - (4/7)\log_2(4/7) = 0.5 + 0.5 = 0.99$
- Entropy of the child set ABC =
 $= -(2/3)\log_2(2/3) - (1/3)\log_2(1/3) = 0.39 + 0.53 = 0.92$
- Entropy of the child set DEFG =
 $= -(3/4)\log_2(3/4) - (1/4)\log_2(1/4) = 0.31 + 0.5 = 0.81$
- Information gain =
 $= \text{Entropy}(ABCDEF) - (3/7) * \text{Entropy}(ABC) - (4/7) * \text{Entropy}(DEFG) = 0.25$

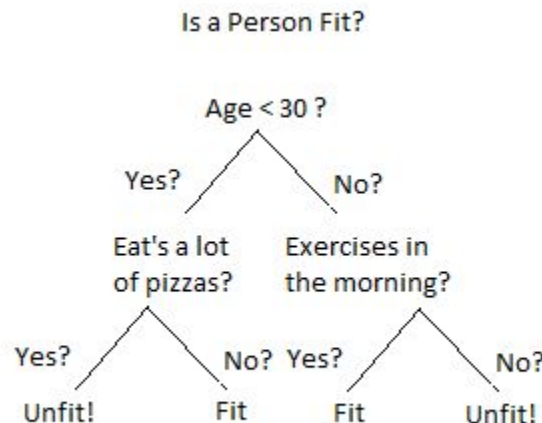
Decision Tree

Information Gain ID3 Algorithm

(Iterative Dichotomiser 3)

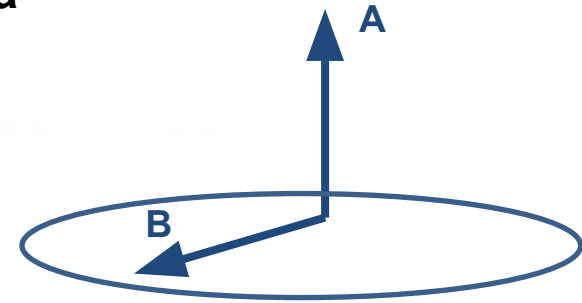
Data mining algorithm invented by J.Ross Quinlan in 1975

- Compute entropy of the total data set
- Split the dataset on different attributes
- Compute entropy for each branch
- Define the attribute giving the biggest information gain (decrease of entropy)
- Use this attribute for decision node



Orthogonality in Statistics

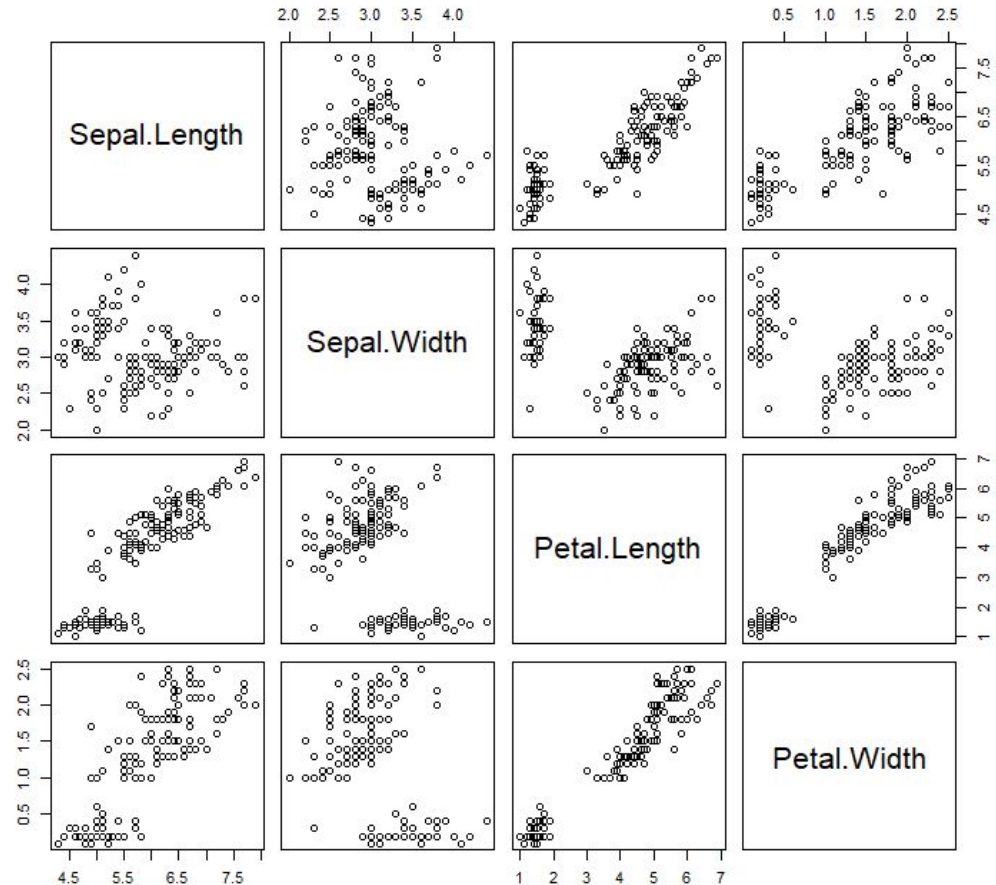
- **Term comes from vector math: orthogonal means lies in perpendicular plane**
 - Projection of A on B is 0
- **Made into statistics through linear algebra**
 - For matrices: $[X]^T * [X] = I$
 - For vectors: $A.B = 0$
- **Meaning for statistical variables in data frame: uncorrelated - not truly orthogonal**
- **Advantages of datasets with truly independent (“orthogonal”) variables**
 - Easy to analyze
 - Better support for decision trees
 - Can be converted into matrix and allow vector math (linear algebra)



Quiz: Orthogonality of Dimensions

- In math dimensions are orthogonal: length, width, height
- Analytical tools use “generalized” definition of dimension, dimension is an attribute, a column name in a table
- Show pairs of truly orthogonal dimensions on the graph
- What math functions can be performed on these pairs?

Setosa = subset(iris, Species == "setosa")
plot(Setosa [1:4])



References

General

- <http://www-bcf.usc.edu/~gareth/ISL/ISLR%20First%20Printing.pdf>
- <https://www.r-bloggers.com/in-depth-introduction-to-machine-learning-in-15-hours-of-experiment-videos/>
- <https://web.stanford.edu/~hastie/Papers/ESLII.pdf>
- <https://medium.com/machine-learning-for-humans/why-machine-learning-matters-6164faf1df12>

Linear Regression

- <http://r-statistics.co/Linear-Regression.html>

SVM

- http://math.stanford.edu/~yuany/course/2015.fall/SVM_in_R.pdf
- <https://rischanlab.github.io/SVM.html>

Neural networks

- <https://www.kdnuggets.com/2016/08/beginners-guide-neural-networks-r.html>

Entropy

- <https://stackoverflow.com/questions/1859554/what-is-entropy-and-information-gain>
- <https://www.miniwebtool.com/log-base-2-calculator/>

Cluster Analysis

- <http://www.sthda.com/english/articles/30-advanced-clustering/105-dbscan-density-based-clustering-essentials/>

Q & A