

# Project 2 Notebook

Introduction (40 points)

- 10 points for background on the protein/gene/species of interest and where the data is sourced from

## Introduction

- 10 points for specific, measurable, and clear scientific question

## Scientific Question

#just add the 4 When examining dog breeds (canis lupus familiaris), will breeds of a similar size (e.g. Cocker Spaniel, English Cocker Spaniel) have more related genes and SNP's surrounding longevity than breeds of a different size (e.g. Doberman Pinscher, Miniature Pinscher)? Will these differences result in expression changes between breeds of different sizes?

Note: I only selected 4 genes most closely associated with life span (IGF1, IGF1R, LCORL, and SMAD2). There are more genes involved in this, but these are the most significant.

Note: Size will be determined by the American Kennel Club (AKC). You can filter by all AKC recognized dog breeds by size. This is categorical data; if it is easier for me to work with numerical instead, I will instead use the ideal height and weight, as outlined in the official standard of each breed.

- 10 points for clear, specific, and measurable scientific hypothesis that is in the form of an if-then statement

## Scientific Hypothesis

If you examine canine breeds, then breeds of a similar size (e.g. Cocker Spaniel, English Cocker Spaniel) they will have more related SNPs and/or fragments of genes surrounding longevity than breeds of a different size (e.g. Doberman Pinscher, Miniature Pinscher). Changes in these SNPs and gene fragments will result in expressional changes between breeds.

- 10 points for description of what analyses were done and how the data was downloaded for the project  
## Analysis Performed:

## SNP's

- Exploratory Data Analysis (EDA): Scatterplots of SNP nucleotide vs size to check for visible trends before analysis
- Multiple Sequence Alignment (of SNP+border sequences), which was then visualized with msaPrettyPrint()
- Clustering of MSA results, which were visualized as dendrograms

## Gene Fragments:

- Multiple Sequence Alignment (of SNP+border sequences), which was then visualized with `msaPrettyPrint()` — Clustering of MSA results, which were visualized as dendrograms

## Expression data:

- EDA: see if expression, not changes in snps is the reason behind differences
- perform clustering of expression data, visualized as dendrograms

## Data Sourcing

- Dog breed information (sizing): American Kennel Club [LINK](#) — Data downloads:
- SNP and gene positionings:
- 25 points for definition of each of the packages loaded
- 5 points for correctly loading all of the packages needed

```
#for reading in fasta files
library("BiocManager")
#for reading in excel files
library("readxl")
#change readxl to this one
library("xlsx")
#for multiple sequence alignment
library("msa")

## Loading required package: Biostrings
## Loading required package: BiocGenerics
##
## Attaching package: 'BiocGenerics'
##
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
##
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min
##
## Loading required package: S4Vectors
## Loading required package: stats4
##
## Attaching package: 'S4Vectors'
##
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
```

```

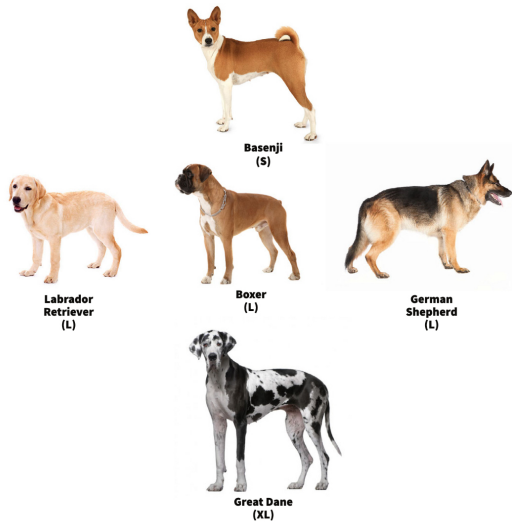
## Loading required package: IRanges
## Loading required package: XVector
## Loading required package: GenomeInfoDb
##
## Attaching package: 'Biostrings'
## The following object is masked from 'package:base':
##
##     strsplit
##
## Attaching package: 'msa'
## The following object is masked from 'package:BiocManager':
##
##     version
#for msa pretty print
library("tinytex")
#visualization of results
library("ggplot2")
#for clustering of DNA seqs
library("DECIPHER")

## Loading required package: RSQLite
## Loading required package: parallel
#for cleaning up dendograms
library('dendextend')

##
## -----
## Welcome to dendextend version 1.15.2
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
## To suppress this message use: suppressPackageStartupMessages(library(dendextend))
## -----
##
## Attaching package: 'dendextend'
## The following object is masked from 'package:Biostrings':
##
##     nnodes
##
## The following object is masked from 'package:stats':
##
##     cutree

```

First we will be performing MSA and clustering with various sequence data (SNPS and gene fragments). Unfortunately, the scope of this data is limited to the 5 breeds shown below



#### Function definitions

```
#global variable
alignment_name<="-"
myColors <- c("antiquewhite4", " coral3", "darkgoldenrod4","darkcyan", "darkorchid4")

#notebook functions

#multi_alignments align fasta depending on inputs, returns msaprettyprint alignment or plaintext alignment
#file_name: fasta file to be read in
#fasta_names: names to display for prettyprint alignment, "names" in file_name files are entire paragraphs
#big_aln: determines if msaprettyprint is run (True=run return pdf name for display by knitr, False=return text)
#dna_set: determines how the fasta data is read in (True=DNA, False=Amino Acid) this speeds up alignment
mult_alignments<-function(file_name,fasta_names,name,big_aln=FALSE,dna_set=TRUE){

  #read in fasta for all dogs
  #use DNA for small files
  if(dna_set=="TRUE"){
    string_set<-readDNAStringSet(file=file_name,use.names=FALSE)
  }
  #AA for large files
  else{
    string_set<-readAAStringSet(file=file_name,use.names=FALSE)
  }

  #read in seq names as list
  table=read.table(fasta_names, header = FALSE, sep = "\n")[[ "V1"]]

  #update names for pretty print
  names(string_set)<-table

  #align unnamed seqs
  alignment<-msa(string_set,order="input")
  #if seq cant be display with msa pretty print, return
  if(big_aln==TRUE){
```

```

    return(alignment)
}
#update global variable so multiple pretty print runs dont overrun eachother
alignment_name<-gsub(" ", "", paste(name,".pdf"), fixed = TRUE)

#return pretty alignment, does not show up on my console
msaPrettyPrint(alignment, file=alignment_name,output="pdf",
               showNames="right",showLogo="top",askForOverwrite=FALSE,
               showNumbering="none",paperWidth=6,paperHeight=3)

#return
return(alignment_name)
}
#have figure with white background, no gridline and only ticks along x and y axis
#fig: ggplot to be altered
tune_figure<-function(fig){
  return(fig+theme_minimal()+theme(
    plot.background = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank()))
}
#create dendrogram based on fasta files, names of items clustered in fasta_names
#fasta_path: path to fasta file
#fasta_names: path to names to display on dendrogram
create_dendrogram<-function(fasta_path, fasta_names, fig_title){
  #grab DNA info from coallated file
  dna <- string_set<-readDNAStrngSet(file=fasta_path,use.names=FALSE)
  #get sequence names
  names(dna)=read.table(fasta_names, header = FALSE, sep = "\n")[[ "V1"]]
  #create distance matrix for clustering
  d1 <- DistanceMatrix(dna, type="dist")
  #form dendrogram
  dendrogram<-IdClusters(d1, method="complete", cutoff=0.05, showPlot=FALSE,
                        type="dendrogram")
  #fix names being cut-off
  nodePar <- list(lab.cex = 0.6, pch = c(NA, 19),
                  cex = 0.7, col = "black")
  #plot results
  plot(as.dendrogram(dendrogram), ylab = "Height", nodePar =nodePar,main=fig_title)
  return(as.dendrogram(dendrogram))
}

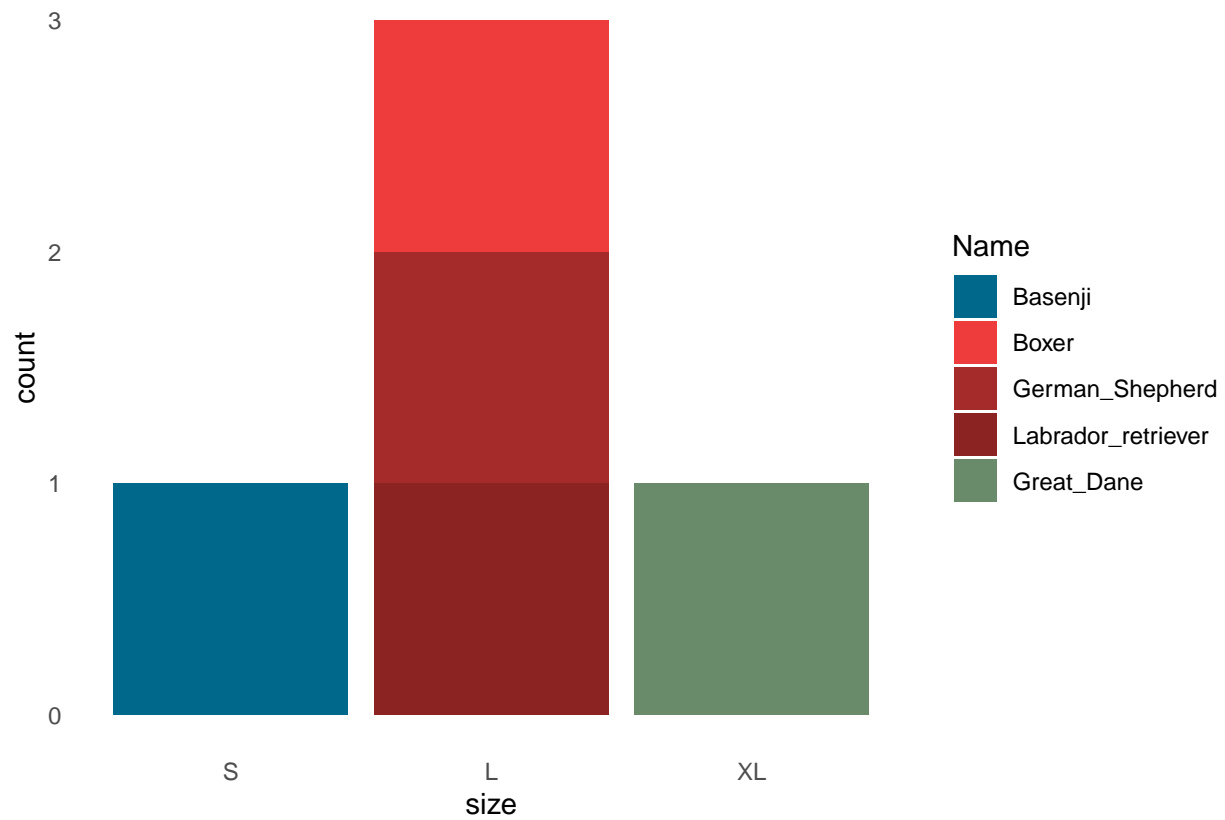
```

EDA of Sequence data

```

#visualize size breakdown of dogs
snps<-read_excel("dog snps.xlsx")
#fix ordering of legend
snps$Name <- factor(snps$Name, levels = c("Basenji", "Boxer", "German_Shepherd", "Labrador_retriever", "G
p<-ggplot(data = snps, aes(size))+scale_x_discrete(limits = c("S", "L", "XL"))+geom_bar(aes(fill = Name))
tune_figure(p)

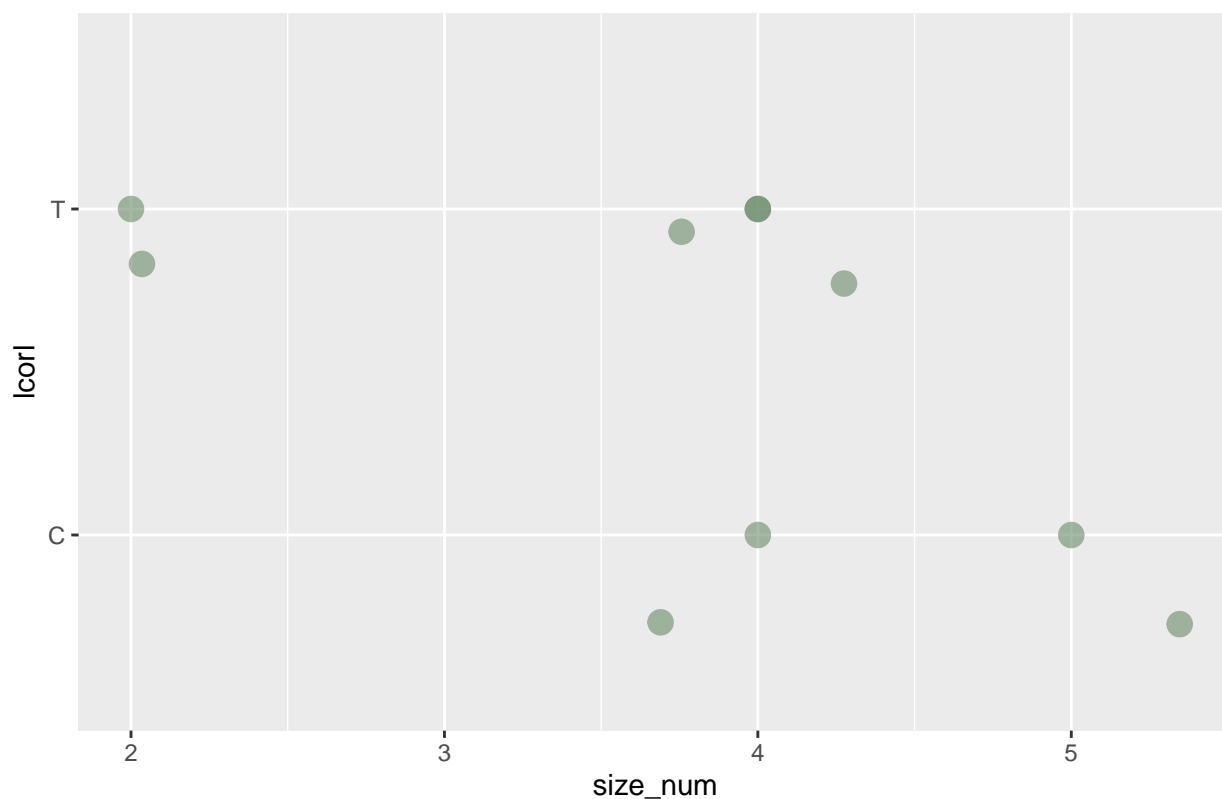
```



LCORL Sequence Analysis SNP scatterplot, see if any obvious trends

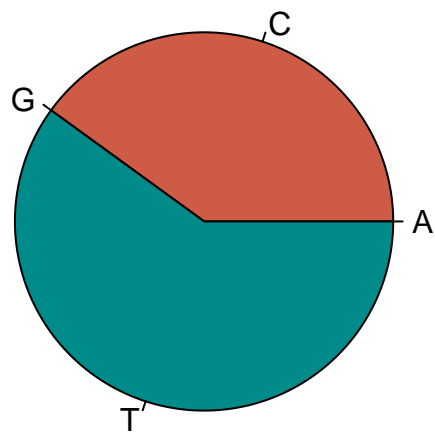
```
#visualize LCORL SNP by size
p<-ggplot(data = snps, mapping = aes(y=lcrl,x=size_num))+geom_point(size=4,
  alpha=0.6,color="darkseagreen4")+
  ggtitle("LCORL SNP Distribution by Dog Size")
#add jitter so points don't sit on top of each other
p+geom_jitter(size=4,alpha=0.6,color="darkseagreen4")
```

## LCORL SNP Distribution by Dog Size



*# pie chart instead of scatter plot for dist*

```
pies<-read_excel("dog snps.xlsx",sheet="pie")
pie(pies$lcorl,labels=pies$Name,col=myColors)
```



LCORL alignment, see if size-grouping is obvious

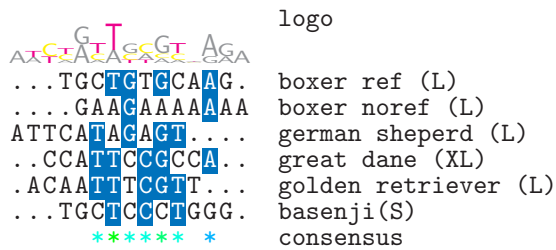
*#LCORL CALL*

```
alignment<-mult_alignments("fasta/LCORL_file.txt","fasta/names.txt","LCORL")
```

```
## use default substitution matrix
```

```
print(alignment_name)
```

```
## [1] "LCORL.pdf"
```



X non-conserved  
X  $\geq 50\%$  conserved

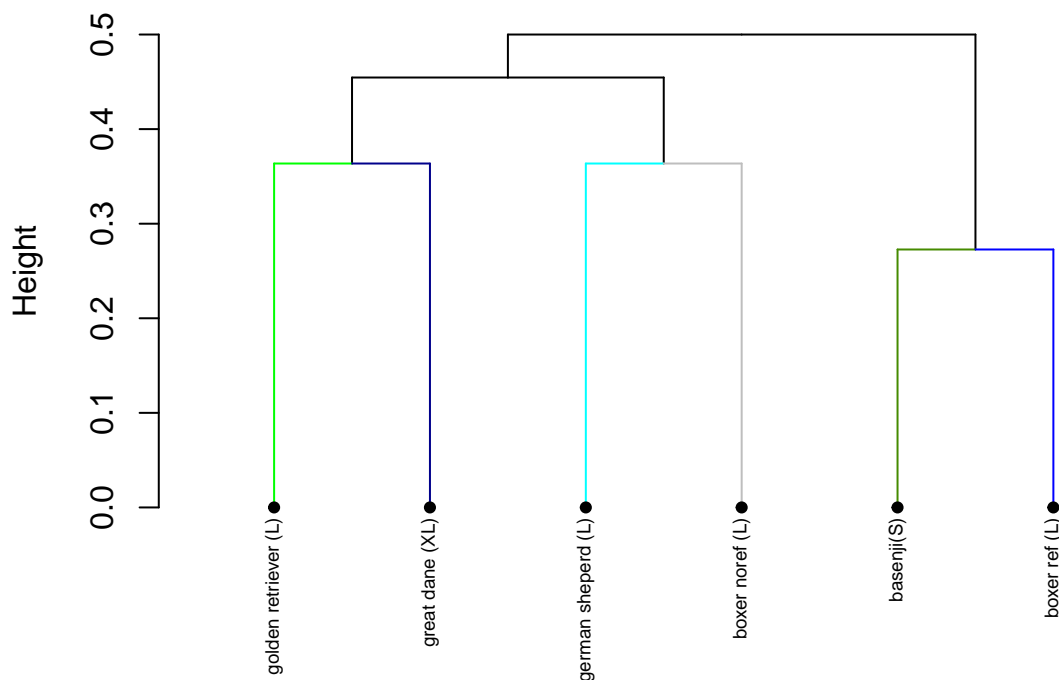
LCORL

clusters, see if group by dog size

```
#Cluster LCORL extended fragment
create_dendrogram("fasta/LCORL_file.txt", "fasta/names.txt", "LCORL Extended Fragment Dendogram")
```

```
## =====
##
## Time difference of 0 secs
##
## =====
##
## Time difference of 0.02 secs
```

### LCORL Extended Fragment Dendogram



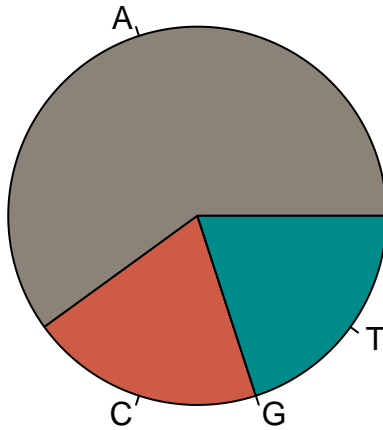


```
## 'dendrogram' with 2 branches and 6 members total, at height 0.5
```

```
#may not be pure breeds  
#try and figure out if regions are correct
```

IGF1 ANALYSIS EDA of IGF1 SNP, see if are any obvious trends

```
pies<-read_excel("dog snps.xlsx",sheet="pie")  
pie(pies$igf1,labels=pies$Name,col=myColors)
```



IGF1 Alignment, see if size-grouping is obvious

```
#IGF1 CALL  
alignment<-mult_alignments("fasta/igf1.fasta","fasta/igf1_names.txt","igf1")
```

```
## use default substitution matrix
```

```

      G T T
      C C C C
      A A T A
TTGCTTTTGTGTA... Boxer no ref (L)
.GGGCCCCGGCTG... Boxer ref (L)
...TCTGAAGAGTA  German shepherd (L)
..GGCAATCCCT... Basenji (S)
AGGTCATGACT.... Great dane (XL)
..AATTCAGTGAA.. Labrador retriever (L)
      *      ****
      consensus

```

```

[X] non-conserved
[X] ≥ 50% conserved

```

<http://www.sthda.com/english/wiki/beautiful-dendrogram-visualizations-in-r-5-must-known-methods-unsupervised-machine-learning#plot.dendrogram-function> for look and non cut off stuff

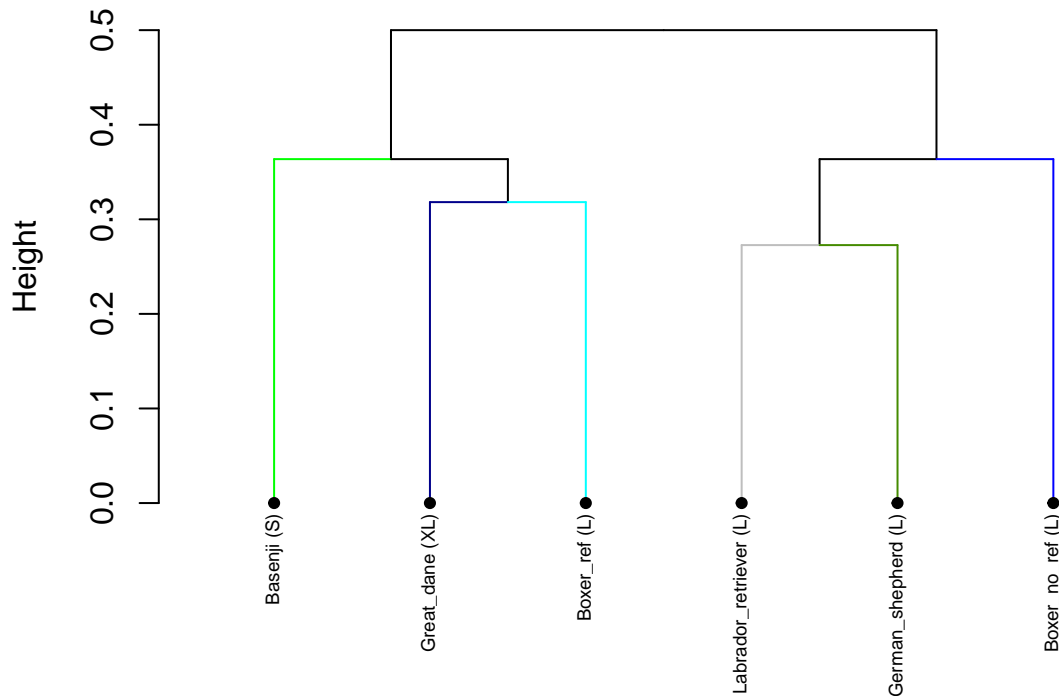
IGF1 Clustering, see if size-grouping is obvious

```
#Cluster IGF1 extended fragment  
create_dendrogram("fasta/igf1.fasta", "fasta/igf1_names.txt", "IGF1 Extended Fragment Dendrogram")
```

```
## =====  
##  
## Time difference of 0 secs
```

```
##
## =====
##
## Time difference of 0.01 secs
```

## IGF1 Extended Fragment Dendrogram



```
## 'dendrogram' with 2 branches and 6 members total, at height 0.5
```

IGSF1 ANALYSIS, add back in for final submit

```
library(seqinr)
```

```
##
## Attaching package: 'seqinr'
## The following object is masked from 'package:Biostrings':
##
##   translate
```

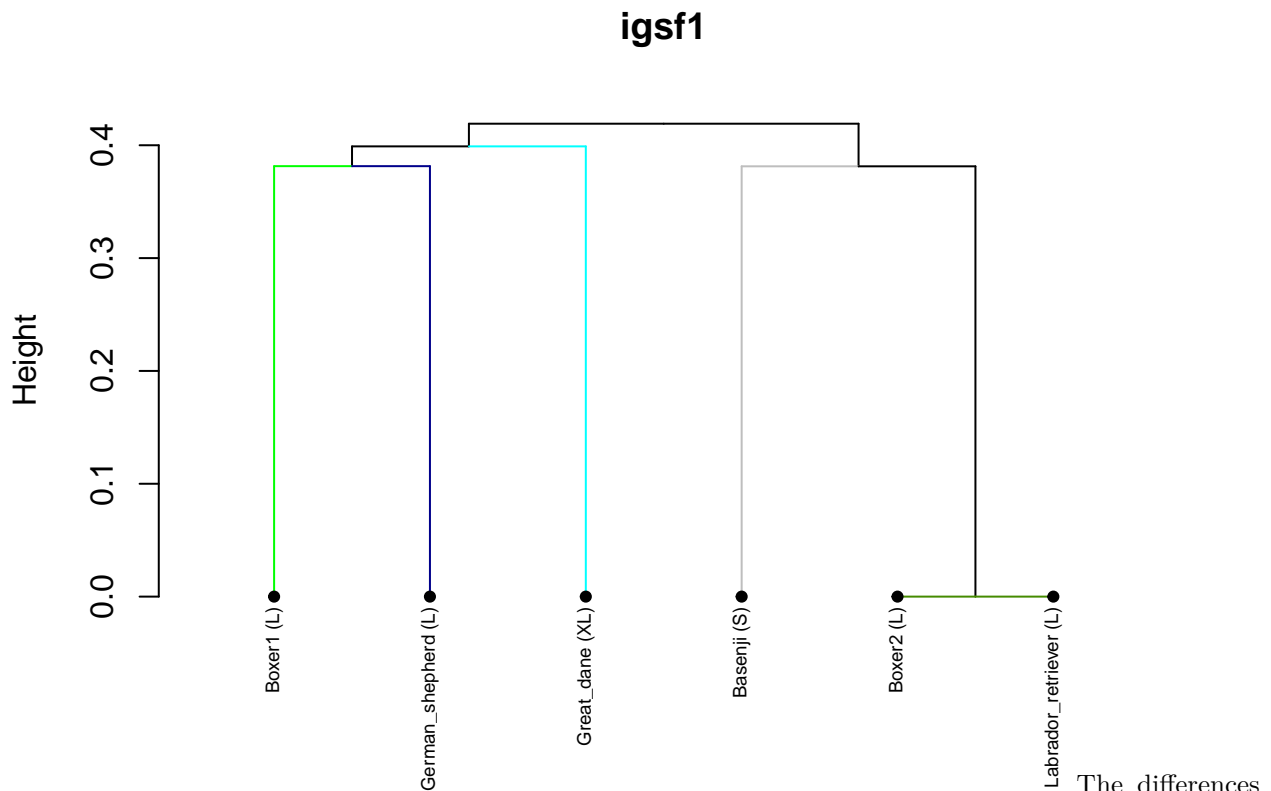
```
library(ape)
```

```
##
## Attaching package: 'ape'
## The following objects are masked from 'package:seqinr':
##
##   as.alignment, consensus
## The following objects are masked from 'package:dendextend':
##
##   ladderize, rotate
## The following object is masked from 'package:Biostrings':
##
```

```
##      complement
#takes too long to run, will add in final submission
igsf1<-mult_alignments("fasta/IGSF1.fasta","fasta/IGSF1_names.txt","IGSF1",TRUE,FALSE)

## use default substitution matrix
igsf1_aln <- msaConvert(igsf1, type="seqinr::alignment")
d <- dist.alignment(igsf1_aln, "identity")
dendogram<-IdClusters(d, method="complete", cutoff=0.05, showPlot=FALSE,type="dendrogram")

## =====
##
## Time difference of 0 secs
#fix names being cut-off
nodePar <- list(lab.cex = 0.6, pch = c(NA, 19),
               cex = 0.7, col = "black")
#plot results
plot(as.dendrogram(dendogram), ylab = "Height", nodePar =nodePar,main="igsf1")
```



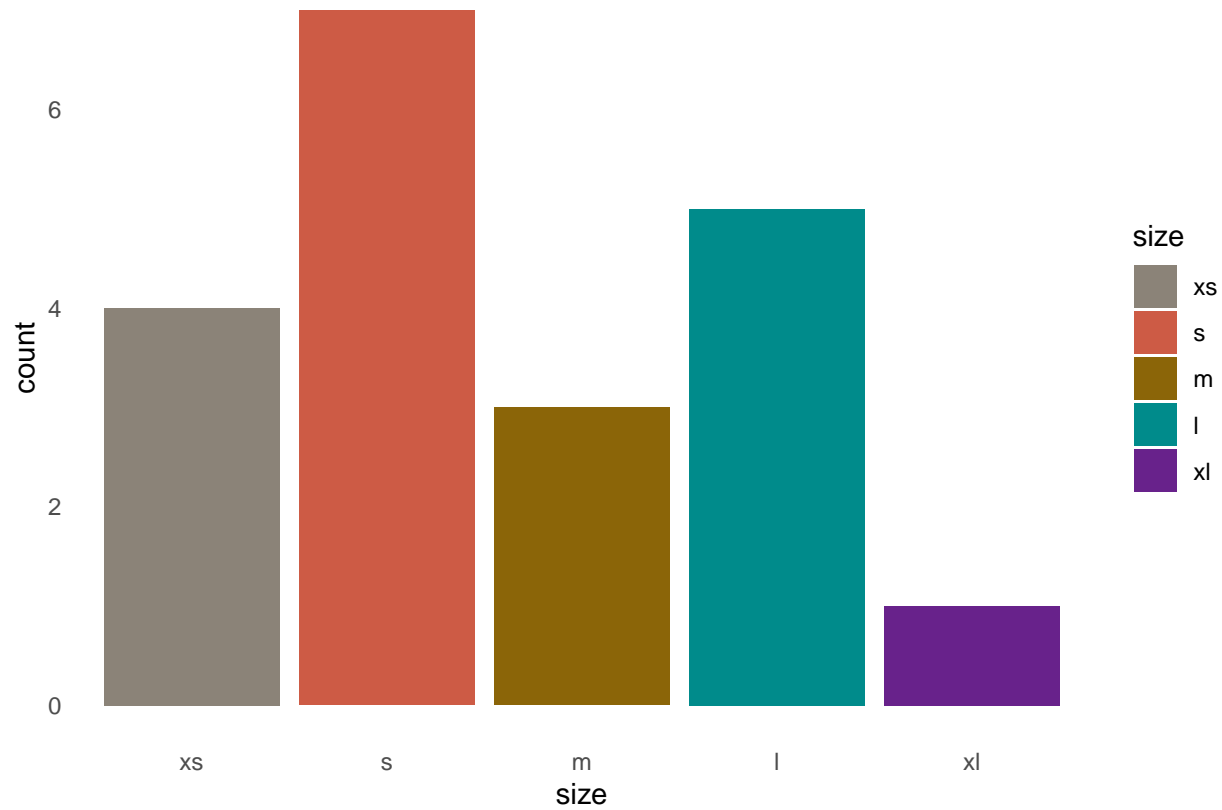
sequence-wise seems to be minor when looking at such a narrow scope and small subsection of dogs. Thankfully there is also expression data for the genes studied which covers more dog-breeds, depicted below



EDA of expression data, see size distribution of dogs

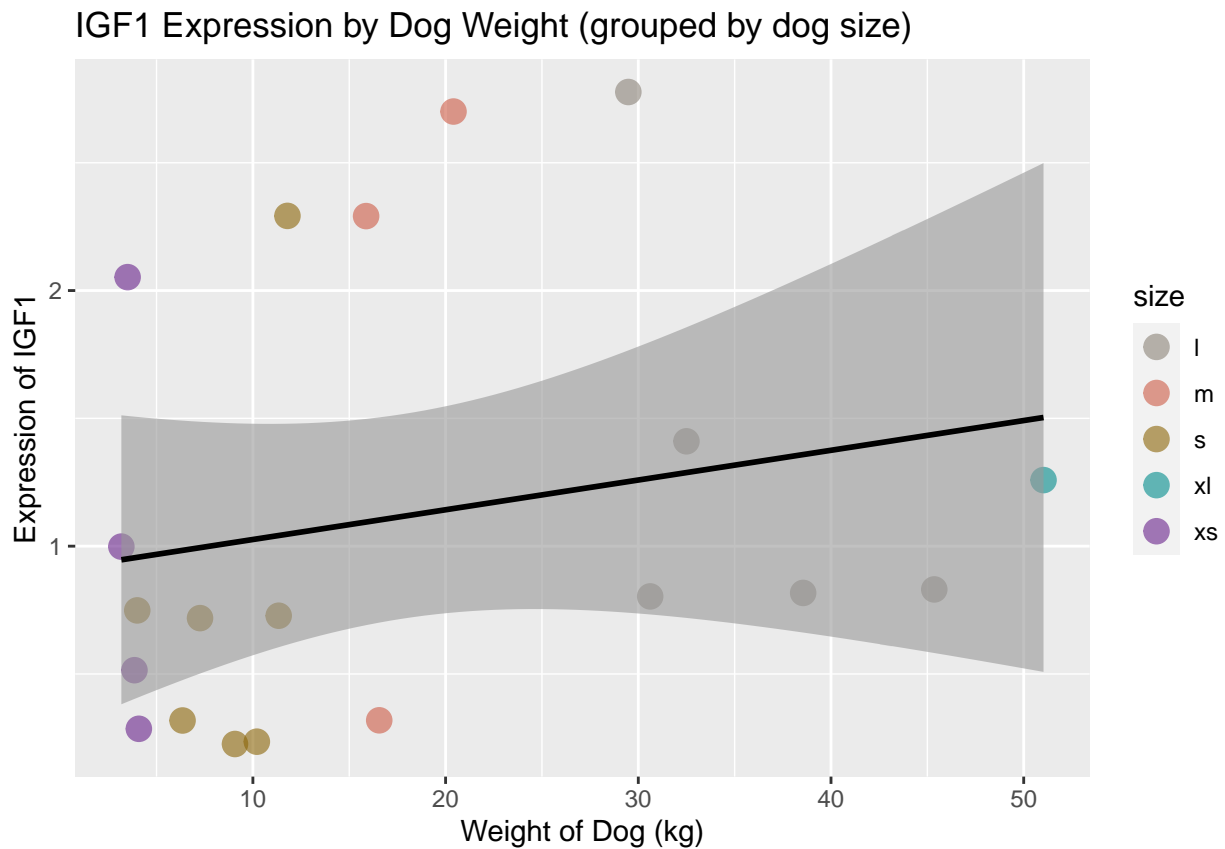
```
#visualize size breakdown of dogs
expression<-read_excel("dog snps.xlsx",sheet="IGF1")
#fix ordering of bars
expression$size <- factor(expression$size, levels = c("xs","s","m","l","xl"))

p<-ggplot(data = expression, aes(size))+geom_bar(aes(fill = size))+scale_fill_manual(values =myColors)
#clean up barplot
tune_figure(p)
```



Distribution much more balanced than sequence data, but favors small dogs

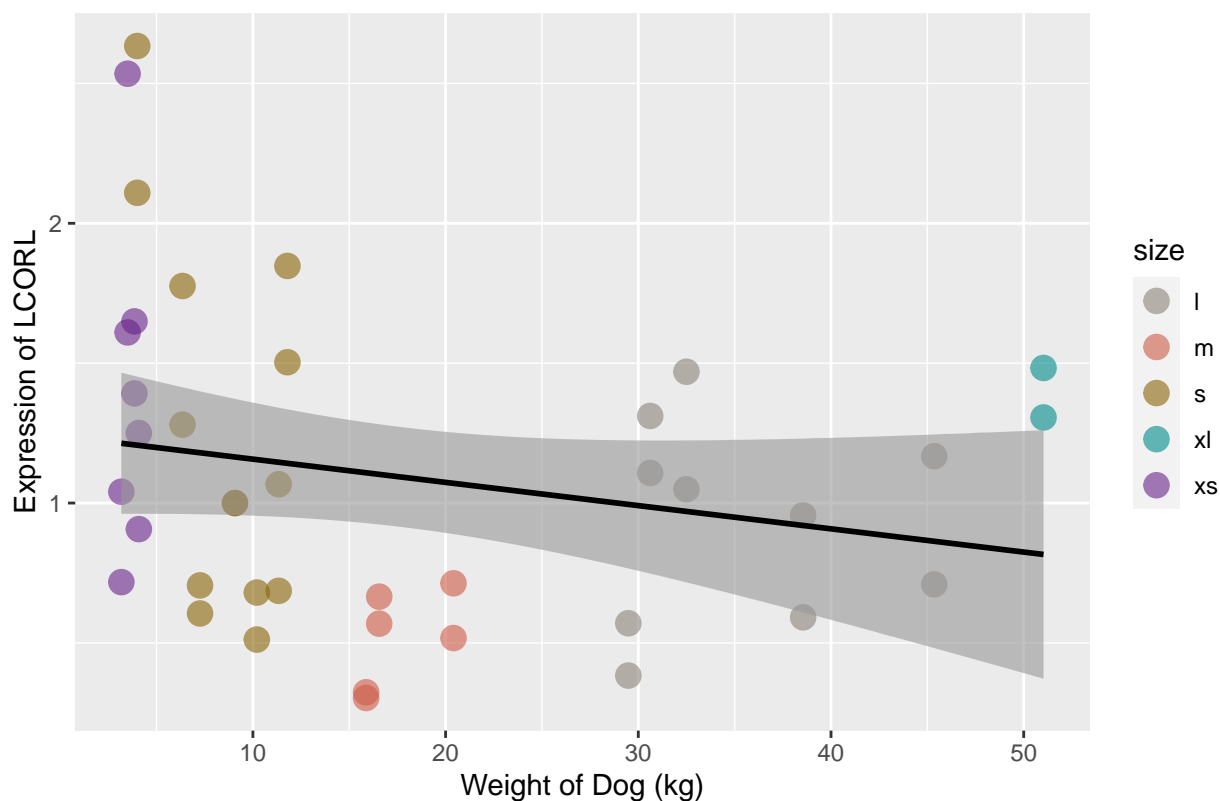
```
#expression data
expression<-read_excel("dog_snps.xlsx",sheet="IGF1")
myColors <- c("antiquewhite4", " coral3", "darkgoldenrod4","darkcyan", "darkorchid4")
p<-ggplot(data = expression, mapping = aes_string(y="norm_exp",x="weight_kg" ,col= "size"))+geom_point()
p+scale_color_manual(values=myColors)+geom_smooth(method = "lm" ,aes(group=1),color="black",alpha=0.6,s
## `geom_smooth()` using formula 'y ~ x'
```



LCORL had two expression sets for each dog, so i included both

```
expression<-read_excel("dog snps.xlsx",sheet="LCORL")
p<-ggplot(data = expression, mapping = aes_string(y="norm_exp",x="weight_kg" ,col= "size"))+geom_point(
p+scale_color_manual(values=myColors) +geom_smooth(method = "lm" ,aes(group=1),color="black",alpha=0.6,
## `geom_smooth()` using formula 'y ~ x'
```

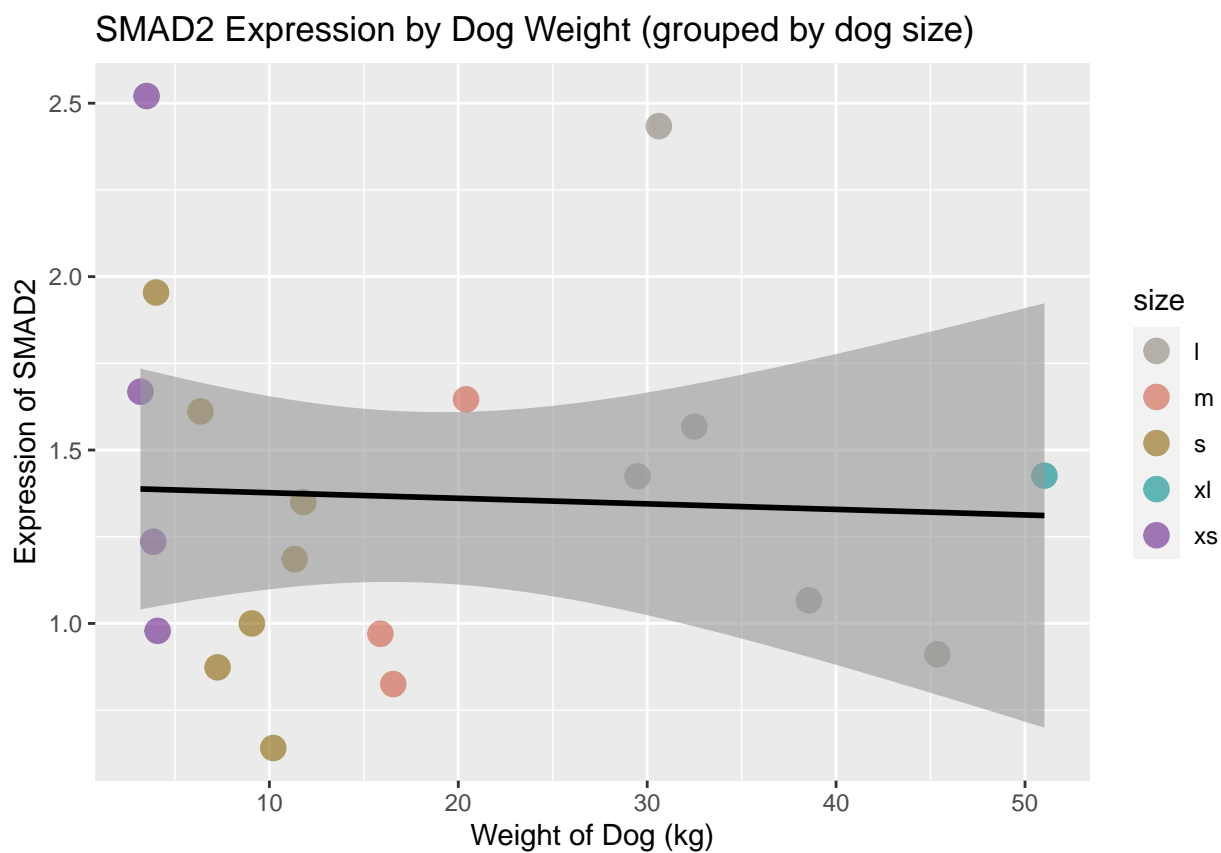
LCORL Expression by Dog Weight (grouped by dog size)



igsf1 has no gene exp data so i swapped to smad2

```
expression<-read_excel("dog snps.xlsx",sheet="SMAD2")
p<-ggplot(data = expression, mapping = aes_string(y="norm_exp",x="weight_kg" ,col= "size"))+geom_point()
p+scale_color_manual(values=myColors)+ geom_smooth(method = "lm" ,aes(group=1),color="black",alpha=0.4)

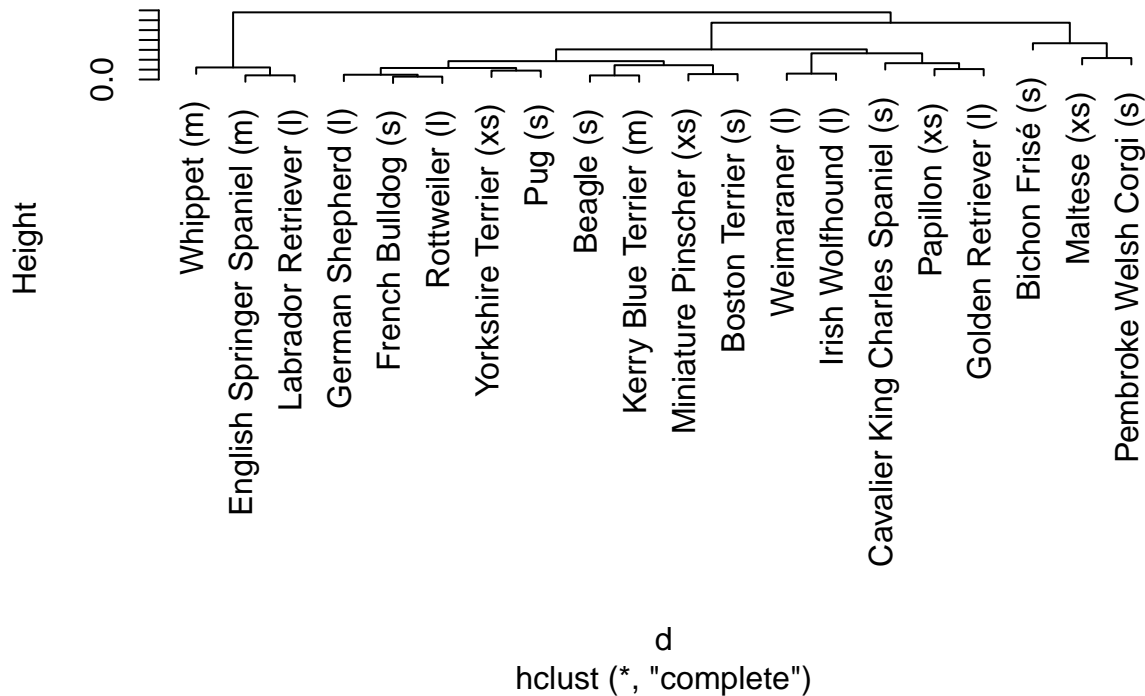
## `geom_smooth()` using formula 'y ~ x'
```



```
clusters<-read.xlsx2("dog snps.xlsx",row.names=1,sheetName="clustering")
d <- dist(clusters, method = "euclidean") # distance matrix
fit <- hclust(d, method="complete")
plot(fit) # display dendrogram
```



## Cluster Dendrogram



```
groups <- cutree(fit, k=4) # cut tree into 5 clusters
#fix names being cut-off
nodePar <- list(lab.cex = 0.6, pch = c(NA, 19),
               cex = 0.7, col = "black")
plot(fit, cex = 0.6, hang = -1, nodePar = nodePar)
```

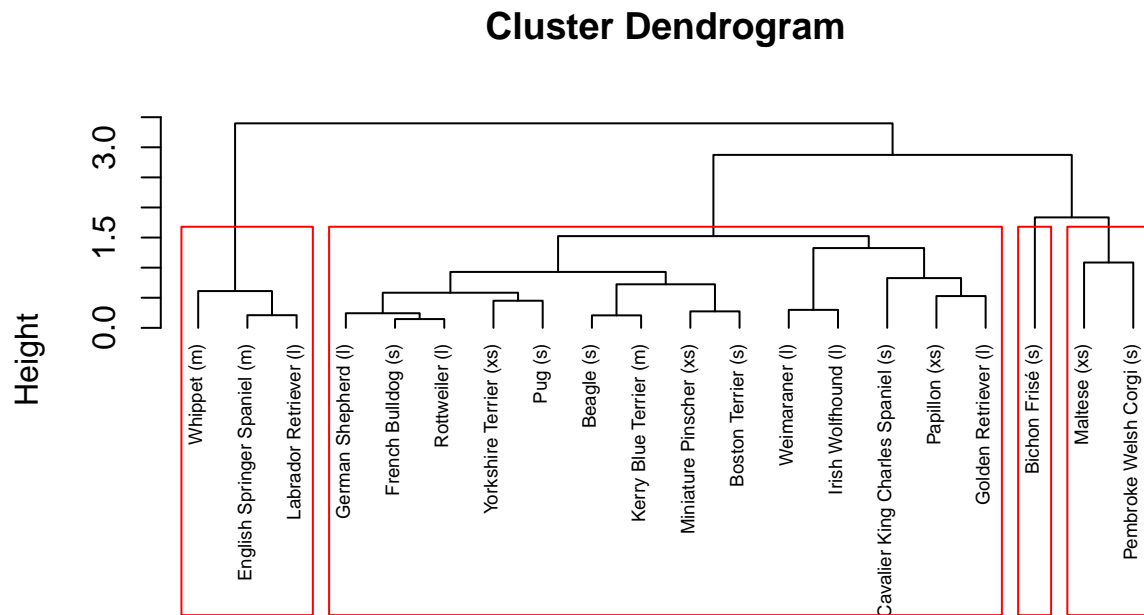
```
## Warning in graphics::plotHclust(n1, merge, height, order(x$order), hang, :
## "nodePar" is not a graphical parameter
```

```
## Warning in graphics::plotHclust(n1, merge, height, order(x$order), hang, :
## "nodePar" is not a graphical parameter
```

```
## Warning in axis(2, at = pretty(range(height)), ...): "nodePar" is not a
## graphical parameter
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## "nodePar" is not a graphical parameter
```

```
rect.hclust(fit, k=4, border="red")
```



d  
hclust (\*, "complete")

```
expression<-read_excel("dog snps.xlsx",sheet="scatter")
p<-ggplot(data = expression, mapping = aes_string(y="norm_exp",x="weight_kg" ,col= "size"))+geom_point(
p+scale_color_manual(values=myColors)+geom_smooth(method = "lm" ,aes(group=1),color="black",alpha=0.6,s
## `geom_smooth()` using formula 'y ~ x'
```

SMAD2, IGF1,LCORL Expression by Dog Weight (grouped by dog size)

