

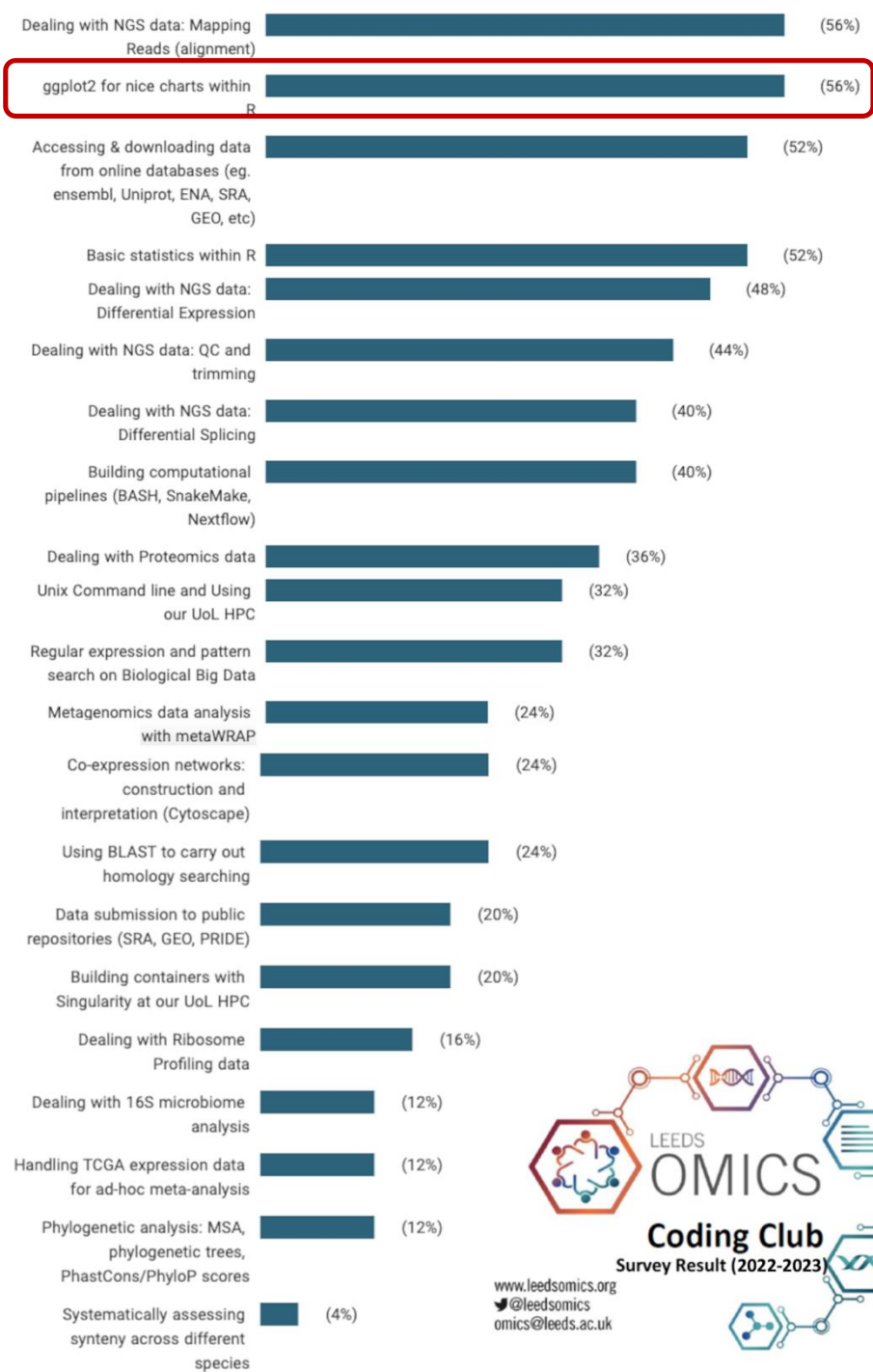
www.leedsomics.org
@leedsomics
omics@leeds.ac.uk

Plotting charts with ggplot2 in R

Club Moderators: Elton Vasconcelos, Euan McDonnell

Topics to be addressed on the 2022-23 season - Survey Result

2nd session



LEEDS
OMICS

Coding Club
Survey Result (2022-2023)

www.leedsomics.org
@leedsomics
omics@leeds.ac.uk

R has its own command line environment

Table of Useful R commands

https://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf

Command	Purpose	Command	Purpose
<code>help()</code>	Obtain documentation for a given R command	<code>plot()</code>	Produces a scatterplot
<code>example()</code>	View some examples on the use of a command	<code>xyplot()</code>	Lattice command for producing a scatterplot
<code>c()</code> , <code>scan()</code>	Enter data manually to a vector in R	<code>lm()</code>	Determine the least-squares regression line
<code>seq()</code>	Make arithmetic progression vector	<code>anova()</code>	Analysis of variance (can use on results of <code>lm()</code>)
<code>rep()</code>	Make vector of repeated values	<code>predict()</code>	Obtain predicted values from linear model
<code>data()</code>	Load (often into a data.frame) built-in dataset	<code>nls()</code>	estimate parameters of a nonlinear model
<code>View()</code>	View dataset in a spreadsheet-type format	<code>residuals()</code>	gives (observed - predicted) for a model fit to data
<code>str()</code>	Display internal structure of an R object	<code>sample()</code>	take a sample from a vector of data
<code>read.csv()</code> , <code>read.table()</code>	Load into a data.frame an existing data file	<code>replicate()</code>	repeat some process a set number of times
<code>library()</code> , <code>require()</code>	Make available an R add-on package	<code>cumsum()</code>	produce running total of values for input vector
<code>dim()</code>	See dimensions (# of rows/cols) of data.frame	<code>ecdf()</code>	builds empirical cumulative distribution function
<code>length()</code>	Give length of a vector	<code>dbinom()</code> , etc.	tools for binomial distributions
<code>ls()</code>	Lists memory contents	<code>dpois()</code> , etc.	tools for Poisson distributions
<code>rm()</code>	Removes an item from memory	<code>pnorm()</code> , etc.	tools for normal distributions
<code>names()</code>	Lists names of variables in a data.frame	<code>qt()</code> , etc.	tools for student <i>t</i> distributions
<code>hist()</code>	Command for producing a histogram	<code>pchisq()</code> , etc.	tools for chi-square distributions
<code>histogram()</code>	Lattice command for producing a histogram	<code>binom.test()</code>	hypothesis test and confidence interval for 1 proportion
<code>stem()</code>	Make a stem plot	<code>prop.test()</code>	inference for 1 proportion using normal approx.
<code>table()</code>	List all values of a variable with frequencies	<code>chisq.test()</code>	carries out a chi-square test
<code>xtabs()</code>	Cross-tabulation tables using formulas	<code>fisher.test()</code>	Fisher test for contingency table
<code>mosaicplot()</code>	Make a mosaic plot	<code>t.test()</code>	student <i>t</i> test for inference on population mean
<code>cut()</code>	Groups values of a variable into larger bins	<code>qqnorm()</code> , <code>qqline()</code>	tools for checking normality
<code>mean()</code> , <code>median()</code>	Identify “center” of distribution	<code>addmargins()</code>	adds marginal sums to an existing table
<code>by()</code>	apply function to a column split by factors	<code>prop.table()</code>	compute proportions from a contingency table
<code>summary()</code>	Display 5-number summary and mean	<code>par()</code>	query and edit graphical settings
<code>var()</code> , <code>sd()</code>	Find variance, sd of values in vector	<code>power.t.test()</code>	power calculations for 1- and 2-sample <i>t</i>
<code>sum()</code>	Add up all values in a vector	<code>anova()</code>	compute analysis of variance table for fitted model
<code>quantile()</code>	Find the position of a quantile in a dataset		
<code>barplot()</code>	Produces a bar graph		
<code>barchart()</code>	Lattice command for producing bar graphs		
<code>boxplot()</code>	Produces a boxplot		
<code>bwplot()</code>	Lattice command for producing boxplots		

https://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf

http://www.math.umt.edu/olear/stat458/Rseminar_2.pdf



Other useful R material for beginners

Brief explanations on:

Regular **plot** function → <https://www.datamentor.io/r-programming/plot-function/>

The powerful **ggplot2** package → <http://r-statistics.co/ggplot2-Tutorial-With-R.html>

- `install.packages("ggplot2")`
- `library(ggplot2)`

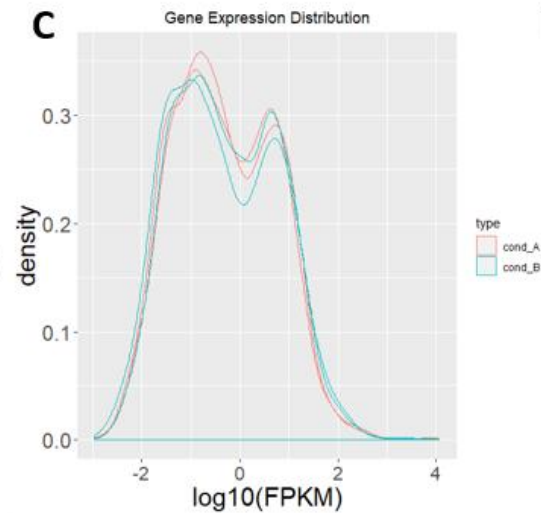
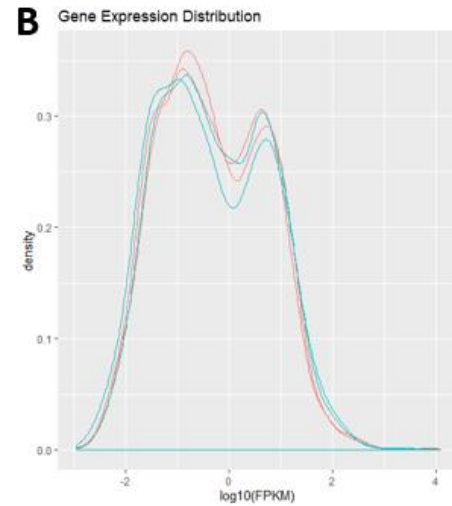
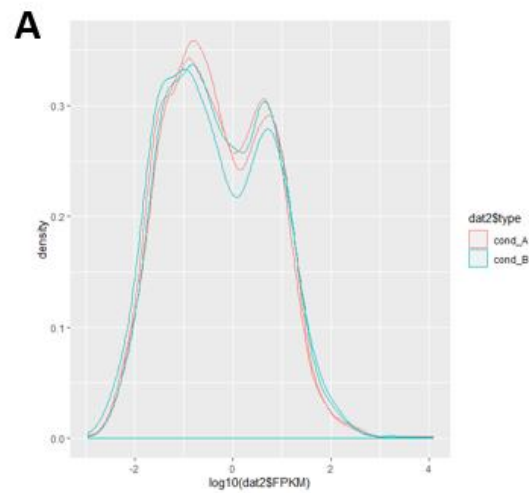
Some important ggplot functions and plot types

Function	Description
<code>ggplot()</code>	Create a new ggplot using a data frame as input
<code>aes()</code>	Construct aesthetic mappings (goes within ggplot brackets)
<code>+</code>	Add components to a plot
<code>geom_</code>	Geometric object (the actual chart type)
<code>theme()</code>	Set x and y axes parameters
<code>ggtitle()</code>	Set a title
<code>ggsave()</code>	Save a ggplot (or other grid object) with sensible defaults

Sub-function	Chart type
<code>geom_density</code>	Histogram
<code>geom_boxplot</code>	Boxplot
<code>geom_jitter</code>	Stripchart
<code>geom_violin</code>	Violin plot
<code>geom_point</code>	Scatter plot
<code>geom_bar</code>	Bar and pie charts
<code>geom_tile</code>	Heatmaps

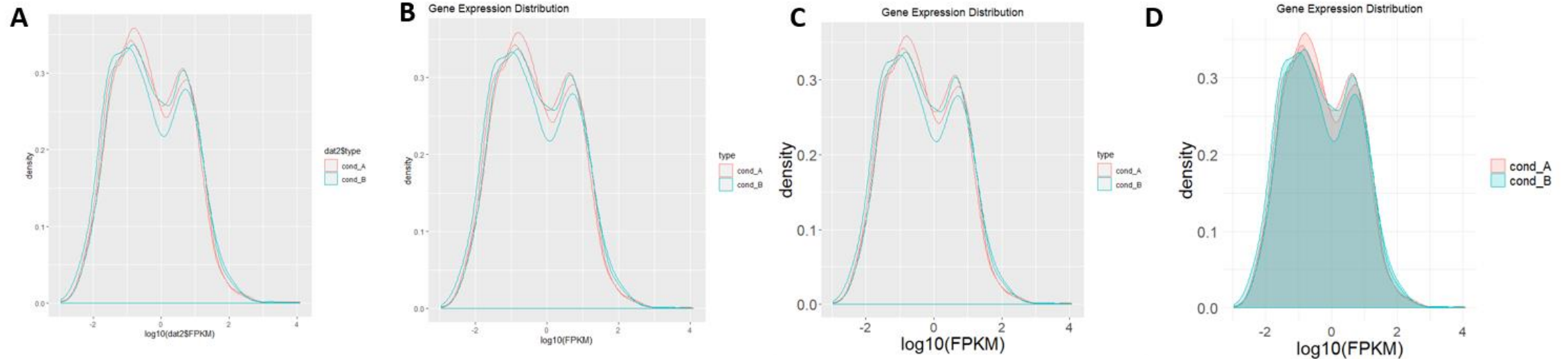
FPKM histogram as an example:

```
> dat = read.delim("yourFPKMtable.tsv", row.names=1)
> dat2plot <- data.frame(geneID = rownames(dat), FPKM = expVals , samples = sampNames, type = conditions)
> ggplot(dat2plot, aes(x = log10(FPKM), element_line = samples, color = type, fill = type)) + geom_density() +
ggtitle("Gene Expression Distribution") + theme(plot.title = element_text(hjust=0.5), axis.text.x =
element_text(size=16), axis.text.y=element_text(size=18), axis.title=element_text(size=22), legend.title =
element_blank(), legend.text = element_text(size = 15)) + theme_minimal()
```



FPKM histogram as an example:

```
> dat = read.delim("yourFPKMtable.tsv", row.names=1)  
> dat2plot <- data.frame(geneID = rownames(dat), FPKM = expVals , samples = sampNames, type = conditions)  
> ggplot(dat2plot, aes(x = log10(FPKM), element_line = samples, color = type, fill = type)) + geom_density(alpha=0.2)  
+ ggtitle("Gene Expression Distribution") + theme(plot.title = element_text(hjust=0.5), axis.text.x =  
element_text(size=16), axis.text.y=element_text(size=18), axis.title=element_text(size=22), legend.title =  
element_blank(), legend.text = element_text(size = 15)) + theme_minimal()
```



Bring your issues on!