R introduction using tidyverse

Dimitris Papageorgiou

November 4, 2018

References

A lot of this materials was based on material from:

- Hadley Wickham
- Michael Levy

Tidyverse and R

When we do data analysis the usual steps we follow are:
1. Import data
2. Tidy up
3. Transform data (select, filter, transform)
4. Visualize / Analyze
5. Model
6. Export and/or communicate

All the steps above need to be done in a consistent and reproducible way

The very beginning

1. What is R / Rstudio

2. Explanation of the window panes in R studio

3. R code
4. An R package is a collection of functions, data, and documentation that extends the capabilities of base R: **install.packages(tidyverse)**

5. In the begining of every session use the **library("tidyverse")**

What is the tidyverse?

~~Hadleyverse~~ **Hadley Wickam**

The tidyverse is a suite of R tools that follow a tidy philosophy:

Tidy data

Put data in data frames

- Each type of observation gets a data frame
- Each variable gets a column
- Each observation gets a row

Suite of ~20 packages that provide consistent, user-friendly, smart-default tools to do most of what most people do in R.

- Core packages: ggplot2, dplyr, tidyr, readr, purrr, tibble
- Specialized data manipulation: hms, stringr, lubridate, forcats
- Data import: DBI, haven, httr, jsonlite, readxl, rvest, xml2

- Modeling: modelr, broom

Bioconductor

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("limma", version = "3.8")
```

```
## Bioconductor version 3.8 (BiocManager 1.30.4), R 3.5.1 (2018-07-02)
```

```
## Installing package(s) 'limma'
```

```
## package 'limma' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\jimpa\AppData\Local\Temp\RtmpGuEaly\downloaded_packages
```

```
## installation path not writeable, unable to update packages: foreign,
##   lattice, MASS, Matrix, mgcv, survival
```

```
## Update old packages: 'openssl'
```

```
##Load the necessary packages

if (!require("tidyverse",quietly = TRUE))
  install.packages("tidyverse")
```

```
## -- Attaching packages --------------------
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts --- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("tidyverse")

if (!require("readxl",quietly = TRUE))
  install.packages("readxl")
library("readxl") ## Package for importing xls and xlsx files
```

Coding Basics

R operators

Arithmetic

Operator	Description
+	addition
-	subtraction
	multiplication
/	division
^ or **	exponentiation
x %% y	modulus (x mod y) 5%%2 is 1
x %/% y	integer division 5%/%2 is 2

Logical

Operator	Description
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	exactly equal to
!=	not equal to
!x	Not x
x | y	x OR y
x & y	x AND y
isTRUE(x)	test if X is TRUE

```
2^3
```

```
## [1] 8
```

```
cos(45)^2 + sin(45)^2
```

```
## [1] 1
```

```
x <- log2(8) ; y <- "R introduction" ## Assign value to a variable
#(in RStudio use Alt and - to create the assign symbol)

x ; y ## Print the x and y  values
```

```
## [1] 3
```

```
## [1] "R introduction"
```

```
d <- rnorm(10,mean = 0, sd = 1) # Almost everything in R is a function

d <- c(d,5,20)

d <- c(d,"Karim")
```

Data Structures in R

Homogeneous	Heterogenous
vectors	data frames (or tibbles)
matrix	lists
array	

```
v1 <- c(5,10,20) ; v1 # Vector 1
```

```
## [1]  5 10 20
```

```
v2 <- c(30,40,50) ;v2 # Vector 2
```

```
## [1] 30 40 50
```

```
m1 <- matrix(data = c(v1,v2),nrow=3,ncol = 4,byrow = F) # What will happen if I define byrow=True ?
m1
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    5   30    5   30
## [2,]   10   40   10   40
## [3,]   20   50   20   50
```

```
dat1 <- data.frame(v1,v2)

daf <- as_tibble(diamonds)

### Transposable tibble

tribble(
~x, ~y, ~z,
#--/--/----
"a", 2, 3.6,
"b", 1, 8.5
)
```

```
## # A tibble: 2 x 3
##   x         y     z
##   <chr> <dbl> <dbl>
## 1 a         2   3.6
## 2 b         1   8.5
```

```
### You can store everything in a list
List1 <- list(c("a,b,c"),dat1,daf)
```

Subsetting using base R

```
diamonds[1:3,5:7] ## We will focus later on this using dplyr package
```

```
## # A tibble: 3 x 3
##   depth table price
##   <dbl> <dbl> <int>
## 1  61.5    55   326
## 2  59.8    61   326
## 3  56.9    65   327
```

One pipe to rule them all %>% magrittr

Sends the output of the LHS function to the first argument of the RHS function.

```
sum(1:8) %>%
  sqrt()
```

```
## [1] 6
```

```
cos(log10(rnorm(n = 100,mean = 5,10))) # Syntax with base R without using the pipe operator
```

```
## Warning: NaNs produced
```

```
##   [1]        NaN 0.44101588        NaN 0.63957808        NaN 0.47396208
##   [7]        NaN 0.67704301 0.82179974        NaN        NaN 0.25008001
##  [13] 0.34777190 0.29765707 0.22288536        NaN 0.39953524 0.52763333
##  [19] 0.38224111 0.25187311 0.68582723        NaN 0.52462448        NaN
##  [25] 0.47382808 0.65841481 0.73027633 0.23122763 0.33956845 0.83737097
##  [31] 0.85539489 0.74893515 0.38255480 0.27543411 0.59312473        NaN
```

```
##  [37]        NaN 0.73597127 0.44404113        NaN 0.99543820 0.69276302
##  [43] 0.05168359 0.61103843 0.77555792 0.42246706        NaN 0.56436604
##  [49] 0.46307916 0.45474927 0.73608042 0.27320284 0.50302326        NaN
##  [55]        NaN        NaN 0.44365056 0.65372449 0.28523760 0.35521852
##  [61] 0.15615996        NaN        NaN 0.56187373 0.67308822        NaN
##  [67] 0.39573009        NaN 0.48175714        NaN        NaN        NaN
##  [73] 0.25100183        NaN 0.23292237 0.38945641 0.28705558 0.97737011
##  [79]        NaN        NaN        NaN 0.98487056        NaN        NaN
##  [85] 0.47824356 0.87514223 0.46702281 0.98088584 0.48061576 0.55241926
##  [91] 0.50351878        NaN        NaN 0.77293705 0.18486777 0.49155640
##  [97]        NaN 0.18066516        NaN 0.48589341
```

```
rnorm(n = 100,mean = 5,10) %>% log10 %>% cos()
```

```
## Warning in function_list[[i]](value): NaNs produced
```

```
##   [1] 0.5279678 0.2925682       NaN 0.3383412 0.2609484 0.6978835 0.1616503
##   [8] 0.4288363 0.4310854 0.4061190 0.4491406       NaN 0.8039595 0.6748063
##  [15] 0.5688850 0.2343699       NaN       NaN 0.1387920 0.4529559 0.5350968
##  [22]       NaN 0.3278234 0.1270340 0.9106624 0.3104638 0.2017732 0.4045210
##  [29] 0.6296261 0.8098482 0.3856100 0.3816634 0.8271066 0.5401548 0.9904793
##  [36] 0.6041595 0.6201242 0.9990582       NaN       NaN       NaN 0.2409608
##  [43]       NaN 0.2685878 0.4108540 0.7995562       NaN       NaN 0.6016950
##  [50]       NaN 0.3777631       NaN 0.5363669 0.9798754 0.4302576 0.5621174
##  [57] 0.9176298 0.5243271 0.9395010 0.2044851 0.2545878 0.7627489 0.4587975
##  [64] 0.6155051       NaN       NaN 0.5380932       NaN 0.8779793 0.6861561
##  [71] 0.4501363 0.9218198 0.7671553 0.3513882 0.8214803 0.9010489       NaN
##  [78]       NaN       NaN       NaN 0.4320120       NaN 0.2984066 0.7026903
##  [85] 0.1950160       NaN 0.9190025 0.3066046 0.7340950       NaN 0.6619142
##  [92] 0.7813468 0.1064042 0.2754326 0.6973393 0.5346120       NaN       NaN
##  [99]       NaN 0.7070391
```

```
### How is the pipe incorporated for functions with multiple arguments

sum(1:8) %>% sqrt() %>% rnorm(n=20,mean=.,sd=.) ### Just subsitute the dot in the argument
```

```
##  [1]  5.8437360  3.3433304  9.1910635  7.9330384 -2.4567275  5.6071669
##  [7] 12.1134305  7.2999482 11.6792855  1.8051693  5.4514762  9.5034421
## [13] 13.3014595  2.8661394  2.9082737  2.7804742  2.4653299  0.2509689
## [19] 11.6781743  9.2576918
```

Set seed function

Set the seed of R's random number generator, which is useful for creating simulations or random objects that can be reproduced.

```
rnorm(5) ## Gives random numbers everytime it is executed
```

```
## [1] -1.99161845 -1.01268952  0.86194173 -0.18781764  0.02100164
```

```
## Set seed produces the same random numbers all the time ##
set.seed(123)
rnorm (5)
```

```
## [1] -0.56047565 -0.23017749  1.55870831  0.07050839  0.12928774
```

```
## If you run the rnorm only you get the same sequence of random numbers when the seed is set.
```

```
## If you want to reset the seed just

set.seed(Sys.time()) ## everytime it gets a different number
rnorm (5)
```

```
## [1] -1.632393960 -0.003903443 -0.819799488  1.099984916 -0.655763586
```

Importing data into R

We will depend on **readr** and **readxl** instead of the base R functions instead of using the base R code

- ** read_csv() ** reads comma-delimited files
- ** read_csv2() ** reads semicolon-separated files
- ** read_tsv() ** reads tab-delimited files
- ** read_delim() ** read_delim()
- ** read_xls() ** "old excel files" **AVOID IMPORTING EXCEL FILES**

- ** read_xlsx() ** "newer excel files"

Maxquant output files

Irrespectively of the MQ version the output files are all in txt format.

```
## Using base R

#prot <- read.table(choose.files(), header=TRUE, sep="\t") ## Why this is bad ??


### Select the location in your computer of where your file is located

prot <- read.table("C:/Users/jimpa/Documents/R_projects/R_introduction_B230/proteinGroups_Kar_081118.tx


#system.time(prot <- read.table(
#"C:/Users/papageor/OneDrive/R_files/proteinGroups_Kar_081118.txt", header=TRUE, sep="\t"))

#prot <- read_tsv(choose.files(),na = "NaN")

#system.time(prot <- read_tsv(file = "C:/Users/papageor/OneDrive/R_files/proteinGroups_Kar_081118.txt")

### Select the location in your computer of where your file is located

prot <- read_tsv(file = "C:/Users/jimpa/Documents/R_projects/R_introduction_B230/proteinGroups_Kar_0811
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   `Protein IDs` = col_character(),
##   `Majority protein IDs` = col_character(),
##   `Peptide counts (all)` = col_character(),
##   `Peptide counts (razor+unique)` = col_character(),
##   `Peptide counts (unique)` = col_character(),
##   `Protein names` = col_character(),
##   `Gene names` = col_character(),
##   `Fasta headers` = col_character(),
##   `Number of proteins` = col_integer(),
```

```
##   Peptides = col_integer(),
##   `Razor + unique peptides` = col_integer(),
##   `Unique peptides` = col_integer(),
##   `Peptides KRAS_1xIC50_R1` = col_integer(),
##   `Peptides KRAS_1xIC50_R2` = col_integer(),
##   `Peptides KRAS_5xIC50_R1` = col_integer(),
##   `Peptides KRAS_5xIC50_R2` = col_integer(),
##   `Peptides PIK3CA_1xIC50_R1` = col_integer(),
##   `Peptides PIK3CA_1xIC50_R2` = col_integer(),
##   `Peptides PIK3CA_5xIC50_R1` = col_integer(),
##   `Peptides PIK3CA_5xIC50_R2` = col_integer()
##   # ... with 125 more columns
## )
```

```
## See spec(...) for full column specifications.
```

If you open the same file in excel are there any differences ??

Tidy Data

- **The first step is always to figure out what the variables and observations are**
- **Solve two usual problems:**
  - **One variable might be spread across multiple columns**
  - **One observation might be scattered across multiple rows**

Is proteingroups.txt from MQ in a tidy data format ?

Tidy proteingroups.txt

```
colnames(prot) <- str_replace_all(colnames(prot), "\\s",replacement = "_")

## Makes our life for later easier (Replaces space in the column names with _)

colnames(prot)
```

```
##   [1] "Protein_IDs"
##   [2] "Majority_protein_IDs"
##   [3] "Peptide_counts_(all)"
##   [4] "Peptide_counts_(razor+unique)"
##   [5] "Peptide_counts_(unique)"
##   [6] "Protein_names"
##   [7] "Gene_names"
##   [8] "Fasta_headers"
##   [9] "Number_of_proteins"
##  [10] "Peptides"
##  [11] "Razor_+_unique_peptides"
##  [12] "Unique_peptides"
##  [13] "Peptides_KRAS_1xIC50_R1"
##  [14] "Peptides_KRAS_1xIC50_R2"
##  [15] "Peptides_KRAS_5xIC50_R1"
##  [16] "Peptides_KRAS_5xIC50_R2"
##  [17] "Peptides_PIK3CA_1xIC50_R1"
##  [18] "Peptides_PIK3CA_1xIC50_R2"
##  [19] "Peptides_PIK3CA_5xIC50_R1"
```

```
##  [20] "Peptides_PIK3CA_5xIC50_R2"
##  [21] "Razor_+_unique_peptides_KRAS_1xIC50_R1"
##  [22] "Razor_+_unique_peptides_KRAS_1xIC50_R2"
##  [23] "Razor_+_unique_peptides_KRAS_5xIC50_R1"
##  [24] "Razor_+_unique_peptides_KRAS_5xIC50_R2"
##  [25] "Razor_+_unique_peptides_PIK3CA_1xIC50_R1"
##  [26] "Razor_+_unique_peptides_PIK3CA_1xIC50_R2"
##  [27] "Razor_+_unique_peptides_PIK3CA_5xIC50_R1"
##  [28] "Razor_+_unique_peptides_PIK3CA_5xIC50_R2"
##  [29] "Unique_peptides_KRAS_1xIC50_R1"
##  [30] "Unique_peptides_KRAS_1xIC50_R2"
##  [31] "Unique_peptides_KRAS_5xIC50_R1"
##  [32] "Unique_peptides_KRAS_5xIC50_R2"
##  [33] "Unique_peptides_PIK3CA_1xIC50_R1"
##  [34] "Unique_peptides_PIK3CA_1xIC50_R2"
##  [35] "Unique_peptides_PIK3CA_5xIC50_R1"
##  [36] "Unique_peptides_PIK3CA_5xIC50_R2"
##  [37] "Sequence_coverage_[%]"
##  [38] "Unique_+_razor_sequence_coverage_[%]"
##  [39] "Unique_sequence_coverage_[%]"
##  [40] "Mol._weight_[kDa]"
##  [41] "Sequence_length"
##  [42] "Sequence_lengths"
##  [43] "Q-value"
##  [44] "Identification_type_KRAS_1xIC50_R1"
##  [45] "Identification_type_KRAS_1xIC50_R2"
##  [46] "Identification_type_KRAS_5xIC50_R1"
##  [47] "Identification_type_KRAS_5xIC50_R2"
##  [48] "Identification_type_PIK3CA_1xIC50_R1"
##  [49] "Identification_type_PIK3CA_1xIC50_R2"
##  [50] "Identification_type_PIK3CA_5xIC50_R1"
##  [51] "Identification_type_PIK3CA_5xIC50_R2"
##  [52] "Ratio_M/L"
##  [53] "Ratio_M/L_normalized"
##  [54] "Ratio_M/L_variability_[%]"
##  [55] "Ratio_M/L_count"
##  [56] "Ratio_M/L_iso-count"
##  [57] "Ratio_M/L_type"
##  [58] "Ratio_H/L"
##  [59] "Ratio_H/L_normalized"
##  [60] "Ratio_H/L_variability_[%]"
##  [61] "Ratio_H/L_count"
##  [62] "Ratio_H/L_iso-count"
##  [63] "Ratio_H/L_type"
##  [64] "Ratio_H/M"
##  [65] "Ratio_H/M_normalized"
##  [66] "Ratio_H/M_variability_[%]"
##  [67] "Ratio_H/M_count"
##  [68] "Ratio_H/M_iso-count"
##  [69] "Ratio_H/M_type"
##  [70] "Ratio_M/L_KRAS_1xIC50_R1"
##  [71] "Ratio_M/L_normalized_KRAS_1xIC50_R1"
##  [72] "Ratio_M/L_variability_[%]_KRAS_1xIC50_R1"
##  [73] "Ratio_M/L_count_KRAS_1xIC50_R1"
```

```
##  [74] "Ratio_M/L_iso-count_KRAS_1xIC50_R1"
##  [75] "Ratio_M/L_type_KRAS_1xIC50_R1"
##  [76] "Ratio_H/L_KRAS_1xIC50_R1"
##  [77] "Ratio_H/L_normalized_KRAS_1xIC50_R1"
##  [78] "Ratio_H/L_variability_[%]_KRAS_1xIC50_R1"
##  [79] "Ratio_H/L_count_KRAS_1xIC50_R1"
##  [80] "Ratio_H/L_iso-count_KRAS_1xIC50_R1"
##  [81] "Ratio_H/L_type_KRAS_1xIC50_R1"
##  [82] "Ratio_H/M_KRAS_1xIC50_R1"
##  [83] "Ratio_H/M_normalized_KRAS_1xIC50_R1"
##  [84] "Ratio_H/M_variability_[%]_KRAS_1xIC50_R1"
##  [85] "Ratio_H/M_count_KRAS_1xIC50_R1"
##  [86] "Ratio_H/M_iso-count_KRAS_1xIC50_R1"
##  [87] "Ratio_H/M_type_KRAS_1xIC50_R1"
##  [88] "Ratio_M/L_KRAS_1xIC50_R2"
##  [89] "Ratio_M/L_normalized_KRAS_1xIC50_R2"
##  [90] "Ratio_M/L_variability_[%]_KRAS_1xIC50_R2"
##  [91] "Ratio_M/L_count_KRAS_1xIC50_R2"
##  [92] "Ratio_M/L_iso-count_KRAS_1xIC50_R2"
##  [93] "Ratio_M/L_type_KRAS_1xIC50_R2"
##  [94] "Ratio_H/L_KRAS_1xIC50_R2"
##  [95] "Ratio_H/L_normalized_KRAS_1xIC50_R2"
##  [96] "Ratio_H/L_variability_[%]_KRAS_1xIC50_R2"
##  [97] "Ratio_H/L_count_KRAS_1xIC50_R2"
##  [98] "Ratio_H/L_iso-count_KRAS_1xIC50_R2"
##  [99] "Ratio_H/L_type_KRAS_1xIC50_R2"
## [100] "Ratio_H/M_KRAS_1xIC50_R2"
## [101] "Ratio_H/M_normalized_KRAS_1xIC50_R2"
## [102] "Ratio_H/M_variability_[%]_KRAS_1xIC50_R2"
## [103] "Ratio_H/M_count_KRAS_1xIC50_R2"
## [104] "Ratio_H/M_iso-count_KRAS_1xIC50_R2"
## [105] "Ratio_H/M_type_KRAS_1xIC50_R2"
## [106] "Ratio_M/L_KRAS_5xIC50_R1"
## [107] "Ratio_M/L_normalized_KRAS_5xIC50_R1"
## [108] "Ratio_M/L_variability_[%]_KRAS_5xIC50_R1"
## [109] "Ratio_M/L_count_KRAS_5xIC50_R1"
## [110] "Ratio_M/L_iso-count_KRAS_5xIC50_R1"
## [111] "Ratio_M/L_type_KRAS_5xIC50_R1"
## [112] "Ratio_H/L_KRAS_5xIC50_R1"
## [113] "Ratio_H/L_normalized_KRAS_5xIC50_R1"
## [114] "Ratio_H/L_variability_[%]_KRAS_5xIC50_R1"
## [115] "Ratio_H/L_count_KRAS_5xIC50_R1"
## [116] "Ratio_H/L_iso-count_KRAS_5xIC50_R1"
## [117] "Ratio_H/L_type_KRAS_5xIC50_R1"
## [118] "Ratio_H/M_KRAS_5xIC50_R1"
## [119] "Ratio_H/M_normalized_KRAS_5xIC50_R1"
## [120] "Ratio_H/M_variability_[%]_KRAS_5xIC50_R1"
## [121] "Ratio_H/M_count_KRAS_5xIC50_R1"
## [122] "Ratio_H/M_iso-count_KRAS_5xIC50_R1"
## [123] "Ratio_H/M_type_KRAS_5xIC50_R1"
## [124] "Ratio_M/L_KRAS_5xIC50_R2"
## [125] "Ratio_M/L_normalized_KRAS_5xIC50_R2"
## [126] "Ratio_M/L_variability_[%]_KRAS_5xIC50_R2"
## [127] "Ratio_M/L_count_KRAS_5xIC50_R2"
```

```
## [128] "Ratio_M/L_iso-count_KRAS_5xIC50_R2"
## [129] "Ratio_M/L_type_KRAS_5xIC50_R2"
## [130] "Ratio_H/L_KRAS_5xIC50_R2"
## [131] "Ratio_H/L_normalized_KRAS_5xIC50_R2"
## [132] "Ratio_H/L_variability_[%]_KRAS_5xIC50_R2"
## [133] "Ratio_H/L_count_KRAS_5xIC50_R2"
## [134] "Ratio_H/L_iso-count_KRAS_5xIC50_R2"
## [135] "Ratio_H/L_type_KRAS_5xIC50_R2"
## [136] "Ratio_H/M_KRAS_5xIC50_R2"
## [137] "Ratio_H/M_normalized_KRAS_5xIC50_R2"
## [138] "Ratio_H/M_variability_[%]_KRAS_5xIC50_R2"
## [139] "Ratio_H/M_count_KRAS_5xIC50_R2"
## [140] "Ratio_H/M_iso-count_KRAS_5xIC50_R2"
## [141] "Ratio_H/M_type_KRAS_5xIC50_R2"
## [142] "Ratio_M/L_PIK3CA_1xIC50_R1"
## [143] "Ratio_M/L_normalized_PIK3CA_1xIC50_R1"
## [144] "Ratio_M/L_variability_[%]_PIK3CA_1xIC50_R1"
## [145] "Ratio_M/L_count_PIK3CA_1xIC50_R1"
## [146] "Ratio_M/L_iso-count_PIK3CA_1xIC50_R1"
## [147] "Ratio_M/L_type_PIK3CA_1xIC50_R1"
## [148] "Ratio_H/L_PIK3CA_1xIC50_R1"
## [149] "Ratio_H/L_normalized_PIK3CA_1xIC50_R1"
## [150] "Ratio_H/L_variability_[%]_PIK3CA_1xIC50_R1"
## [151] "Ratio_H/L_count_PIK3CA_1xIC50_R1"
## [152] "Ratio_H/L_iso-count_PIK3CA_1xIC50_R1"
## [153] "Ratio_H/L_type_PIK3CA_1xIC50_R1"
## [154] "Ratio_H/M_PIK3CA_1xIC50_R1"
## [155] "Ratio_H/M_normalized_PIK3CA_1xIC50_R1"
## [156] "Ratio_H/M_variability_[%]_PIK3CA_1xIC50_R1"
## [157] "Ratio_H/M_count_PIK3CA_1xIC50_R1"
## [158] "Ratio_H/M_iso-count_PIK3CA_1xIC50_R1"
## [159] "Ratio_H/M_type_PIK3CA_1xIC50_R1"
## [160] "Ratio_M/L_PIK3CA_1xIC50_R2"
## [161] "Ratio_M/L_normalized_PIK3CA_1xIC50_R2"
## [162] "Ratio_M/L_variability_[%]_PIK3CA_1xIC50_R2"
## [163] "Ratio_M/L_count_PIK3CA_1xIC50_R2"
## [164] "Ratio_M/L_iso-count_PIK3CA_1xIC50_R2"
## [165] "Ratio_M/L_type_PIK3CA_1xIC50_R2"
## [166] "Ratio_H/L_PIK3CA_1xIC50_R2"
## [167] "Ratio_H/L_normalized_PIK3CA_1xIC50_R2"
## [168] "Ratio_H/L_variability_[%]_PIK3CA_1xIC50_R2"
## [169] "Ratio_H/L_count_PIK3CA_1xIC50_R2"
## [170] "Ratio_H/L_iso-count_PIK3CA_1xIC50_R2"
## [171] "Ratio_H/L_type_PIK3CA_1xIC50_R2"
## [172] "Ratio_H/M_PIK3CA_1xIC50_R2"
## [173] "Ratio_H/M_normalized_PIK3CA_1xIC50_R2"
## [174] "Ratio_H/M_variability_[%]_PIK3CA_1xIC50_R2"
## [175] "Ratio_H/M_count_PIK3CA_1xIC50_R2"
## [176] "Ratio_H/M_iso-count_PIK3CA_1xIC50_R2"
## [177] "Ratio_H/M_type_PIK3CA_1xIC50_R2"
## [178] "Ratio_M/L_PIK3CA_5xIC50_R1"
## [179] "Ratio_M/L_normalized_PIK3CA_5xIC50_R1"
## [180] "Ratio_M/L_variability_[%]_PIK3CA_5xIC50_R1"
## [181] "Ratio_M/L_count_PIK3CA_5xIC50_R1"
```

```
## [182] "Ratio_M/L_iso-count_PIK3CA_5xIC50_R1"
## [183] "Ratio_M/L_type_PIK3CA_5xIC50_R1"
## [184] "Ratio_H/L_PIK3CA_5xIC50_R1"
## [185] "Ratio_H/L_normalized_PIK3CA_5xIC50_R1"
## [186] "Ratio_H/L_variability_[%]_PIK3CA_5xIC50_R1"
## [187] "Ratio_H/L_count_PIK3CA_5xIC50_R1"
## [188] "Ratio_H/L_iso-count_PIK3CA_5xIC50_R1"
## [189] "Ratio_H/L_type_PIK3CA_5xIC50_R1"
## [190] "Ratio_H/M_PIK3CA_5xIC50_R1"
## [191] "Ratio_H/M_normalized_PIK3CA_5xIC50_R1"
## [192] "Ratio_H/M_variability_[%]_PIK3CA_5xIC50_R1"
## [193] "Ratio_H/M_count_PIK3CA_5xIC50_R1"
## [194] "Ratio_H/M_iso-count_PIK3CA_5xIC50_R1"
## [195] "Ratio_H/M_type_PIK3CA_5xIC50_R1"
## [196] "Ratio_M/L_PIK3CA_5xIC50_R2"
## [197] "Ratio_M/L_normalized_PIK3CA_5xIC50_R2"
## [198] "Ratio_M/L_variability_[%]_PIK3CA_5xIC50_R2"
## [199] "Ratio_M/L_count_PIK3CA_5xIC50_R2"
## [200] "Ratio_M/L_iso-count_PIK3CA_5xIC50_R2"
## [201] "Ratio_M/L_type_PIK3CA_5xIC50_R2"
## [202] "Ratio_H/L_PIK3CA_5xIC50_R2"
## [203] "Ratio_H/L_normalized_PIK3CA_5xIC50_R2"
## [204] "Ratio_H/L_variability_[%]_PIK3CA_5xIC50_R2"
## [205] "Ratio_H/L_count_PIK3CA_5xIC50_R2"
## [206] "Ratio_H/L_iso-count_PIK3CA_5xIC50_R2"
## [207] "Ratio_H/L_type_PIK3CA_5xIC50_R2"
## [208] "Ratio_H/M_PIK3CA_5xIC50_R2"
## [209] "Ratio_H/M_normalized_PIK3CA_5xIC50_R2"
## [210] "Ratio_H/M_variability_[%]_PIK3CA_5xIC50_R2"
## [211] "Ratio_H/M_count_PIK3CA_5xIC50_R2"
## [212] "Ratio_H/M_iso-count_PIK3CA_5xIC50_R2"
## [213] "Ratio_H/M_type_PIK3CA_5xIC50_R2"
## [214] "Sequence_coverage_KRAS_1xIC50_R1_[%]"
## [215] "Sequence_coverage_KRAS_1xIC50_R2_[%]"
## [216] "Sequence_coverage_KRAS_5xIC50_R1_[%]"
## [217] "Sequence_coverage_KRAS_5xIC50_R2_[%]"
## [218] "Sequence_coverage_PIK3CA_1xIC50_R1_[%]"
## [219] "Sequence_coverage_PIK3CA_1xIC50_R2_[%]"
## [220] "Sequence_coverage_PIK3CA_5xIC50_R1_[%]"
## [221] "Sequence_coverage_PIK3CA_5xIC50_R2_[%]"
## [222] "Intensity"
## [223] "Intensity_L"
## [224] "Intensity_M"
## [225] "Intensity_H"
## [226] "Intensity_KRAS_1xIC50_R1"
## [227] "Intensity_L_KRAS_1xIC50_R1"
## [228] "Intensity_M_KRAS_1xIC50_R1"
## [229] "Intensity_H_KRAS_1xIC50_R1"
## [230] "Intensity_KRAS_1xIC50_R2"
## [231] "Intensity_L_KRAS_1xIC50_R2"
## [232] "Intensity_M_KRAS_1xIC50_R2"
## [233] "Intensity_H_KRAS_1xIC50_R2"
## [234] "Intensity_KRAS_5xIC50_R1"
## [235] "Intensity_L_KRAS_5xIC50_R1"
```

```
## [236] "Intensity_M_KRAS_5xIC50_R1"
## [237] "Intensity_H_KRAS_5xIC50_R1"
## [238] "Intensity_KRAS_5xIC50_R2"
## [239] "Intensity_L_KRAS_5xIC50_R2"
## [240] "Intensity_M_KRAS_5xIC50_R2"
## [241] "Intensity_H_KRAS_5xIC50_R2"
## [242] "Intensity_PIK3CA_1xIC50_R1"
## [243] "Intensity_L_PIK3CA_1xIC50_R1"
## [244] "Intensity_M_PIK3CA_1xIC50_R1"
## [245] "Intensity_H_PIK3CA_1xIC50_R1"
## [246] "Intensity_PIK3CA_1xIC50_R2"
## [247] "Intensity_L_PIK3CA_1xIC50_R2"
## [248] "Intensity_M_PIK3CA_1xIC50_R2"
## [249] "Intensity_H_PIK3CA_1xIC50_R2"
## [250] "Intensity_PIK3CA_5xIC50_R1"
## [251] "Intensity_L_PIK3CA_5xIC50_R1"
## [252] "Intensity_M_PIK3CA_5xIC50_R1"
## [253] "Intensity_H_PIK3CA_5xIC50_R1"
## [254] "Intensity_PIK3CA_5xIC50_R2"
## [255] "Intensity_L_PIK3CA_5xIC50_R2"
## [256] "Intensity_M_PIK3CA_5xIC50_R2"
## [257] "Intensity_H_PIK3CA_5xIC50_R2"
## [258] "Only_identified_by_site"
## [259] "Reverse"
## [260] "Potential_contaminant"
## [261] "id"
## [262] "Peptide_IDs"
## [263] "Peptide_is_razor"
## [264] "Mod._peptide_IDs"
## [265] "Evidence_IDs"
## [266] "MS/MS_IDs"
## [267] "Best_MS/MS"
## [268] "AHA->DAB_site_IDs"
## [269] "AHA->HS_site_IDs"
## [270] "Met->AHA_site_IDs"
## [271] "Oxidation_(M)_site_IDs"
## [272] "AHA->DAB_site_positions"
## [273] "AHA->HS_site_positions"
## [274] "Met->AHA_site_positions"
## [275] "Oxidation_(M)_site_positions"
```

dplyr

Common data(frame) manipulation tasks.

Four core "verbs": filter, select, arrange, group_by + summarize, plus many more convenience functions.

Filter

```
# Remove contaminants, reverse hits and only identified by site

prot_f <- prot %>%

  filter(Only_identified_by_site != "+",Reverse != "+",Potential_contaminant != "+")
```

```
prot_f
```

```
## # A tibble: 3,804 x 275
##    Protein_IDs Majority_protei~ `Peptide_counts~ `Peptide_counts~
##    <chr>       <chr>            <chr>            <chr>
##  1 A0A096LP01  A0A096LP01       2                2
##  2 A0FGR8      A0FGR8           28               28
##  3 A1L0T0      A1L0T0           13               13
##  4 A2A288      A2A288           2                2
##  5 A2A3N6      A2A3N6           7                2
##  6 A2RRP1      A2RRP1           5                5
##  7 A3KMH1      A3KMH1           27               27
##  8 A4D1E9      A4D1E9           7                7
##  9 A5PLL7      A5PLL7           3                3
## 10 A5YKK6      A5YKK6           9                9
## # ... with 3,794 more rows, and 271 more variables:
## #   `Peptide_counts_(unique)` <chr>, Protein_names <chr>,
## #   Gene_names <chr>, Fasta_headers <chr>, Number_of_proteins <int>,
## #   Peptides <int>, `Razor_+_unique_peptides` <int>,
## #   Unique_peptides <int>, Peptides_KRAS_1xIC50_R1 <int>,
## #   Peptides_KRAS_1xIC50_R2 <int>, Peptides_KRAS_5xIC50_R1 <int>,
## #   Peptides_KRAS_5xIC50_R2 <int>, Peptides_PIK3CA_1xIC50_R1 <int>,
## #   Peptides_PIK3CA_1xIC50_R2 <int>, Peptides_PIK3CA_5xIC50_R1 <int>,
## #   Peptides_PIK3CA_5xIC50_R2 <int>,
## #   `Razor_+_unique_peptides_KRAS_1xIC50_R1` <int>,
## #   `Razor_+_unique_peptides_KRAS_1xIC50_R2` <int>,
## #   `Razor_+_unique_peptides_KRAS_5xIC50_R1` <int>,
## #   `Razor_+_unique_peptides_KRAS_5xIC50_R2` <int>,
## #   `Razor_+_unique_peptides_PIK3CA_1xIC50_R1` <int>,
## #   `Razor_+_unique_peptides_PIK3CA_1xIC50_R2` <int>,
## #   `Razor_+_unique_peptides_PIK3CA_5xIC50_R1` <int>,
## #   `Razor_+_unique_peptides_PIK3CA_5xIC50_R2` <int>,
## #   Unique_peptides_KRAS_1xIC50_R1 <int>,
## #   Unique_peptides_KRAS_1xIC50_R2 <int>,
## #   Unique_peptides_KRAS_5xIC50_R1 <int>,
## #   Unique_peptides_KRAS_5xIC50_R2 <int>,
## #   Unique_peptides_PIK3CA_1xIC50_R1 <int>,
## #   Unique_peptides_PIK3CA_1xIC50_R2 <int>,
## #   Unique_peptides_PIK3CA_5xIC50_R1 <int>,
## #   Unique_peptides_PIK3CA_5xIC50_R2 <int>, `Sequence_coverage_[%]` <dbl>,
## #   `Unique_+_razor_sequence_coverage_[%]` <dbl>,
## #   `Unique_sequence_coverage_[%]` <dbl>, `Mol._weight_[kDa]` <dbl>,
## #   Sequence_length <int>, Sequence_lengths <chr>, `Q-value` <dbl>,
## #   Identification_type_KRAS_1xIC50_R1 <chr>,
## #   Identification_type_KRAS_1xIC50_R2 <chr>,
## #   Identification_type_KRAS_5xIC50_R1 <chr>,
## #   Identification_type_KRAS_5xIC50_R2 <chr>,
## #   Identification_type_PIK3CA_1xIC50_R1 <chr>,
## #   Identification_type_PIK3CA_1xIC50_R2 <chr>,
## #   Identification_type_PIK3CA_5xIC50_R1 <chr>,
## #   Identification_type_PIK3CA_5xIC50_R2 <chr>, `Ratio_M/L` <dbl>,
## #   `Ratio_M/L_normalized` <dbl>, `Ratio_M/L_variability_[%]` <dbl>,
## #   `Ratio_M/L_count` <int>, `Ratio_M/L_iso-count` <int>,
```

```
## #   `Ratio_M/L_type` <chr>, `Ratio_H/L` <dbl>,
## #   `Ratio_H/L_normalized` <dbl>, `Ratio_H/L_variability_[%]` <dbl>,
## #   `Ratio_H/L_count` <int>, `Ratio_H/L_iso-count` <int>,
## #   `Ratio_H/L_type` <chr>, `Ratio_H/M` <dbl>,
## #   `Ratio_H/M_normalized` <dbl>, `Ratio_H/M_variability_[%]` <dbl>,
## #   `Ratio_H/M_count` <int>, `Ratio_H/M_iso-count` <int>,
## #   `Ratio_H/M_type` <chr>, `Ratio_M/L_KRAS_1xIC50_R1` <dbl>,
## #   `Ratio_M/L_normalized_KRAS_1xIC50_R1` <dbl>,
## #   `Ratio_M/L_variability_[%]_KRAS_1xIC50_R1` <dbl>,
## #   `Ratio_M/L_count_KRAS_1xIC50_R1` <int>,
## #   `Ratio_M/L_iso-count_KRAS_1xIC50_R1` <int>,
## #   `Ratio_M/L_type_KRAS_1xIC50_R1` <chr>,
## #   `Ratio_H/L_KRAS_1xIC50_R1` <dbl>,
## #   `Ratio_H/L_normalized_KRAS_1xIC50_R1` <dbl>,
## #   `Ratio_H/L_variability_[%]_KRAS_1xIC50_R1` <dbl>,
## #   `Ratio_H/L_count_KRAS_1xIC50_R1` <int>,
## #   `Ratio_H/L_iso-count_KRAS_1xIC50_R1` <int>,
## #   `Ratio_H/L_type_KRAS_1xIC50_R1` <chr>,
## #   `Ratio_H/M_KRAS_1xIC50_R1` <dbl>,
## #   `Ratio_H/M_normalized_KRAS_1xIC50_R1` <dbl>,
## #   `Ratio_H/M_variability_[%]_KRAS_1xIC50_R1` <dbl>,
## #   `Ratio_H/M_count_KRAS_1xIC50_R1` <int>,
## #   `Ratio_H/M_iso-count_KRAS_1xIC50_R1` <int>,
## #   `Ratio_H/M_type_KRAS_1xIC50_R1` <chr>,
## #   `Ratio_M/L_KRAS_1xIC50_R2` <dbl>,
## #   `Ratio_M/L_normalized_KRAS_1xIC50_R2` <dbl>,
## #   `Ratio_M/L_variability_[%]_KRAS_1xIC50_R2` <dbl>,
## #   `Ratio_M/L_count_KRAS_1xIC50_R2` <int>,
## #   `Ratio_M/L_iso-count_KRAS_1xIC50_R2` <int>,
## #   `Ratio_M/L_type_KRAS_1xIC50_R2` <chr>,
## #   `Ratio_H/L_KRAS_1xIC50_R2` <dbl>,
## #   `Ratio_H/L_normalized_KRAS_1xIC50_R2` <dbl>,
## #   `Ratio_H/L_variability_[%]_KRAS_1xIC50_R2` <dbl>,
## #   `Ratio_H/L_count_KRAS_1xIC50_R2` <int>,
## #   `Ratio_H/L_iso-count_KRAS_1xIC50_R2` <int>,
## #   `Ratio_H/L_type_KRAS_1xIC50_R2` <chr>,
## #   `Ratio_H/M_KRAS_1xIC50_R2` <dbl>,
## #   `Ratio_H/M_normalized_KRAS_1xIC50_R2` <dbl>,
## #   `Ratio_H/M_variability_[%]_KRAS_1xIC50_R2` <dbl>,
## #   `Ratio_H/M_count_KRAS_1xIC50_R2` <int>,
## #   `Ratio_H/M_iso-count_KRAS_1xIC50_R2` <int>, ...
```

Select

- starts_with("abc") matches names that begin with "abc"
- ends_with("xyz") matches names that end with "xyz"
- contains("ijk") matches names that contain "ijk"
- matches("(.)\1") selects variables that match a regular expression.
- num_range("x", 1:3) matches x1 , x2 , and x3

```
# Select columns that we will need for further processing

prot_f1 <- prot_f %>% select(Protein_IDs,Majority_protein_IDs,

                             Protein_names,Gene_names,
```

```
                              Fasta_headers,Number_of_proteins)

# Isn't there a faster way ?

prot_f1 <- prot_f %>%

  select(contains("Protein"),Gene_names:Number_of_proteins,

                              starts_with("Peptides_"),

                              matches("^Sequence_coverage_[^[]"), `Mol._weight_[kDa]`,

                              starts_with("Identification"),

         matches("Ratio_./._[^vit]"),

                              matches("^Intensity_._."))
```

Split Protein IDs and Gene names

```
prot_f1 <- prot_f1 %>%

  mutate(Protein_IDs = str_split(Protein_IDs,";",simplify = T)[,1],

                               Gene_names = str_split(Gene_names,";",simplify = T)[,1])



##prot_f1$Protein_IDs <- str_split(string = prot_f1$Protein_IDs, pattern = ";",simplify = T)[,1]
#prot_f1$Gene_names <- str_split(string = prot_f1$Gene_names, pattern = ";",simplify = T)[,1]
```

Tidying up the variables

We observe that variables (both categorical and numerical are spread across the table)

```
Peptides_tb <- prot_f1 %>% select(Protein_IDs:Peptides_PIK3CA_5xIC50_R2) %>%

  gather(Peptides_KRAS_1xIC50_R1:Peptides_PIK3CA_5xIC50_R2 ,key = "Experiment",
         value = "Peptide_Number") %>%

  mutate(Experiment = str_remove_all(Experiment,pattern = "Peptides_"))


Seq_cov_tb <- prot_f1 %>% select(Protein_IDs:Fasta_headers,starts_with("Seq")) %>%

  gather(starts_with("Seq") ,key = "Experiment", value = "Seq_cov_[%]") %>%

  mutate(Experiment = str_remove_all(Experiment,pattern = "Sequence_coverage_")) %>%

  mutate(Experiment = str_remove_all(Experiment,pattern ="_\\[%]"))

Id_type_tb <- prot_f1 %>% select(Protein_IDs:Fasta_headers,starts_with("Ident")) %>%
```

```
  gather(starts_with("Ident") ,key = "Experiment", value = "Ident_type") %>%

  mutate(Experiment = str_remove_all(Experiment,pattern = "Identification_type_"))

## Gather the intensities

Intensity_tb_L <- prot_f1 %>% select(Protein_IDs:Fasta_headers,starts_with("Intensity")) %>%

  gather(starts_with("Intensity_L") ,key = "Experiment", value = "Intensity_L") %>%

  mutate(Experiment = str_remove_all(Experiment,pattern = "Intensity_L_")) %>%

  select(Protein_IDs:Fasta_headers,Experiment:Intensity_L)


Intensity_tb_M <-  prot_f1 %>% select(Protein_IDs:Fasta_headers,starts_with("Intensity")) %>%

  gather(starts_with("Intensity_M") ,key = "Experiment", value = "Intensity_M") %>%

  mutate(Experiment = str_remove_all(Experiment,pattern = "Intensity_M_")) %>%

    select(Protein_IDs:Fasta_headers,Experiment:Intensity_M)


 Intensity_tb_H <- prot_f1 %>% select(Protein_IDs:Fasta_headers,starts_with("Intensity")) %>%

   gather(starts_with("Intensity_H") ,key = "Experiment", value = "Intensity_H") %>%

  mutate(Experiment = str_remove_all(Experiment,pattern = "Intensity_H_")) %>%

  select(Protein_IDs:Fasta_headers,Experiment:Intensity_H)


 Intensity_tb <- left_join(Intensity_tb_L,Intensity_tb_M) %>% left_join(Intensity_tb_H)
```

```
## Joining, by = c("Protein_IDs", "Majority_protein_IDs", "Protein_names", "Number_of_proteins", "Gene_
## Joining, by = c("Protein_IDs", "Majority_protein_IDs", "Protein_names", "Number_of_proteins", "Gene_
```

```
#%>%
  #mutate(Channel = if_else(str_detect(Experiment,"\\_L_")==TRUE,"Light",
   #                      if_else(str_detect(Experiment,"\\_M_")==TRUE,"Medium","Heavy")),
    #  Experiment = str_remove_all(Experiment,pattern = "Intensity_._"))



##Intensity_tb <- prot_f1 %>% select(Protein_IDs:Fasta_headers,starts_with("Intensity")) %>%
  ##gather(starts_with("Intensity") ,key = "Experiment", value = "Intensity") %>%
  ##mutate(Channel = if_else(str_detect(Experiment,"\\_L_")==TRUE,"Light",
    ##                     if_else(str_detect(Experiment,"\\_M_")==TRUE,"Medium","Heavy")),
      ## Experiment = str_remove_all(Experiment,pattern = "Intensity_._"))


######## Normalized Silac Ratios ##################
#### H/L
```

```
Silac_tb_norm_HL <- prot_f1 %>% select(Protein_IDs:Fasta_headers,starts_with("Ratio"),-("Ratio_M/L_norma

  gather(contains("H/L_normalized"),key = "Experiment",value = "Ratio_norm_H/L") %>%


  mutate(Experiment = str_remove_all(Experiment,pattern = "Ratio_H/L_normalized_")) %>%

  select(Protein_IDs:Fasta_headers,Experiment:"Ratio_norm_H/L")

## M/L

Silac_tb_norm_ML <- prot_f1 %>%

  select(Protein_IDs:Fasta_headers,starts_with("Ratio"),-("Ratio_M/L_normalized":"Ratio_H/M_count")) %>%

  gather(contains("M/L_normalized"),key = "Experiment",value = "Ratio_norm_M/L") %>%


  mutate(Experiment = str_remove_all(Experiment,pattern = "Ratio_M/L_normalized_")) %>%

  select(Protein_IDs:Fasta_headers,Experiment:"Ratio_norm_M/L")

## H/M

Silac_tb_norm_HM <- prot_f1 %>% select(Protein_IDs:Fasta_headers,starts_with("Ratio"),-("Ratio_M/L_norma

  gather(contains("H/M_normalized"),key = "Experiment",value = "Ratio_norm_H/M") %>%


  mutate(Experiment = str_remove_all(Experiment,pattern = "Ratio_H/M_normalized_")) %>%

  select(Protein_IDs:Fasta_headers,Experiment:"Ratio_norm_H/M")

### Gather the Silac_ratios

Silac_tb_norm <- left_join(Silac_tb_norm_HL,Silac_tb_norm_ML) %>%

  left_join(Silac_tb_norm_HM)
```

```
## Joining, by = c("Protein_IDs", "Majority_protein_IDs", "Protein_names", "Number_of_proteins", "Gene_n
## Joining, by = c("Protein_IDs", "Majority_protein_IDs", "Protein_names", "Number_of_proteins", "Gene_n
```

Speed up the tidying up of the variables with gather and spread

```
### Fix the counts

Silac_tb_count <- prot_f1 %>% select(Protein_IDs:Fasta_headers,starts_with("Ratio"),-("Ratio_M/L_normali

  gather(contains("count"),key = "Experiment",value = "Ratio_count") %>%

 ### mutate(Ratio_type = if_else(str_detect(Experiment,"\\_H/L_")==TRUE,"H/L",
                             ### if_else(str_detect(Experiment,"\\_M/L_")==TRUE,"M/L","H/M"))) %>%
```

```
  select(Protein_IDs:Fasta_headers,Experiment:Ratio_count) %>%

  extract(Experiment, c("Ratio_type","Experiment"),"([^.].{15})(.*)") %>%

  spread(Ratio_type,value = Ratio_count)


### Tidy the not normalized value

Silac_tb_unnorm <- prot_f1 %>% select(Protein_IDs:Fasta_headers,starts_with("Ratio"),-("Ratio_M/L_normal
  gather(seq(from = 7,to = 76, by = 3),key = "Experiment",value = "Ratio_unnorm") %>%

  select(Protein_IDs:Fasta_headers,Experiment:Ratio_unnorm) %>%

   extract(Experiment, c("Ratio_type","Experiment"),"([^.].{9})(.*)") %>%

  spread(Ratio_type,value = Ratio_unnorm)

### Create our final table

table_merg <- left_join(Silac_tb_norm,Silac_tb_count) %>%
  left_join(Silac_tb_unnorm) %>% left_join(Intensity_tb) %>% left_join(Peptides_tb) %>%
  left_join(Id_type_tb) %>% left_join(Seq_cov_tb)
```

```
## Joining, by = c("Protein_IDs", "Majority_protein_IDs", "Protein_names", "Number_of_proteins", "Gene_
## Joining, by = c("Protein_IDs", "Majority_protein_IDs", "Protein_names", "Number_of_proteins", "Gene_
## Joining, by = c("Protein_IDs", "Majority_protein_IDs", "Protein_names", "Number_of_proteins", "Gene_
## Joining, by = c("Protein_IDs", "Majority_protein_IDs", "Protein_names", "Number_of_proteins", "Gene_
## Joining, by = c("Protein_IDs", "Majority_protein_IDs", "Protein_names", "Number_of_proteins", "Gene_
## Joining, by = c("Protein_IDs", "Majority_protein_IDs", "Protein_names", "Number_of_proteins", "Gene_
```

Data exploration

Excel Comparison

```
table_merg %>% group_by(Experiment) %>%  summarize(Peptides = sum(Peptide_Number))
```

```
## # A tibble: 8 x 2
##   Experiment       Peptides
##   <chr>               <int>
## 1 KRAS_1xIC50_R1      18746
## 2 KRAS_1xIC50_R2      18414
## 3 KRAS_5xIC50_R1      18384
## 4 KRAS_5xIC50_R2      18018
## 5 PIK3CA_1xIC50_R1    17945
## 6 PIK3CA_1xIC50_R2    14852
## 7 PIK3CA_5xIC50_R1    18224
## 8 PIK3CA_5xIC50_R2    17719
```

```
### Why the petide numbers are completely off ??

table_merg %>% filter(Gene_names == "CTCF") %>%

  group_by(Experiment,Gene_names,Ident_type)  %>%
```

```
  summarize(Peptides = sum(Peptide_Number)) %>%

  arrange(desc(Peptides))
```

```
## # A tibble: 8 x 4
## # Groups:   Experiment, Gene_names [8]
##   Experiment       Gene_names Ident_type  Peptides
##   <chr>            <chr>      <chr>          <int>
## 1 PIK3CA_1xIC50_R1 CTCF       By MS/MS           4
## 2 PIK3CA_1xIC50_R2 CTCF       By MS/MS           2
## 3 PIK3CA_5xIC50_R1 CTCF       By MS/MS           1
## 4 PIK3CA_5xIC50_R2 CTCF       By MS/MS           1
## 5 KRAS_1xIC50_R1   CTCF       By matching        0
## 6 KRAS_1xIC50_R2   CTCF       By matching        0
## 7 KRAS_5xIC50_R1   CTCF       By matching        0
## 8 KRAS_5xIC50_R2   CTCF       By matching        0
```

```
 table_merg %>%
   group_by(Gene_names,Experiment,Ident_type,Intensity_L,Intensity_M,Intensity_H) %>%

   filter(Ident_type == "By matching",Gene_names == "DDX39A") %>%

   summarize(Peptides = sum(Peptide_Number)) %>%

   arrange(desc(Peptides))
```

```
## # A tibble: 4 x 7
## # Groups:   Gene_names, Experiment, Ident_type, Intensity_L, Intensity_M
## #   [4]
##   Gene_names Experiment Ident_type Intensity_L Intensity_M Intensity_H
##   <chr>      <chr>      <chr>            <dbl>       <dbl>       <dbl>
## 1 DDX39A     KRAS_5xIC~ By matchi~     6897000    11349000     8359400
## 2 DDX39A     PIK3CA_1x~ By matchi~           0           0           0
## 3 DDX39A     PIK3CA_5x~ By matchi~           0           0           0
## 4 DDX39A     PIK3CA_5x~ By matchi~    15783000    30887000    27892000
## # ... with 1 more variable: Peptides <int>
```

```
table_merg %>% group_by(Gene_names,Experiment) %>% filter (Gene_names == "SUZ12") %>% select (`Seq_cov_
```

```
## Adding missing grouping variables: `Gene_names`, `Experiment`
```

```
## # A tibble: 8 x 3
## # Groups:   Gene_names, Experiment [8]
##   Gene_names Experiment       `Seq_cov_[%]`
##   <chr>      <chr>                    <dbl>
## 1 SUZ12      KRAS_5xIC50_R2             3
## 2 SUZ12      KRAS_1xIC50_R1             2.7
## 3 SUZ12      KRAS_5xIC50_R1             2.7
## 4 SUZ12      PIK3CA_1xIC50_R2           2.3
## 5 SUZ12      KRAS_1xIC50_R2             1.4
## 6 SUZ12      PIK3CA_1xIC50_R1           1.4
## 7 SUZ12      PIK3CA_5xIC50_R1           1.4
## 8 SUZ12      PIK3CA_5xIC50_R2           1.4
```

```
write_tsv(table_merg,"table_merg.txt",na = "NA")
```

You can also save to clipboard

Instead of specifying a path just add "clipboard"

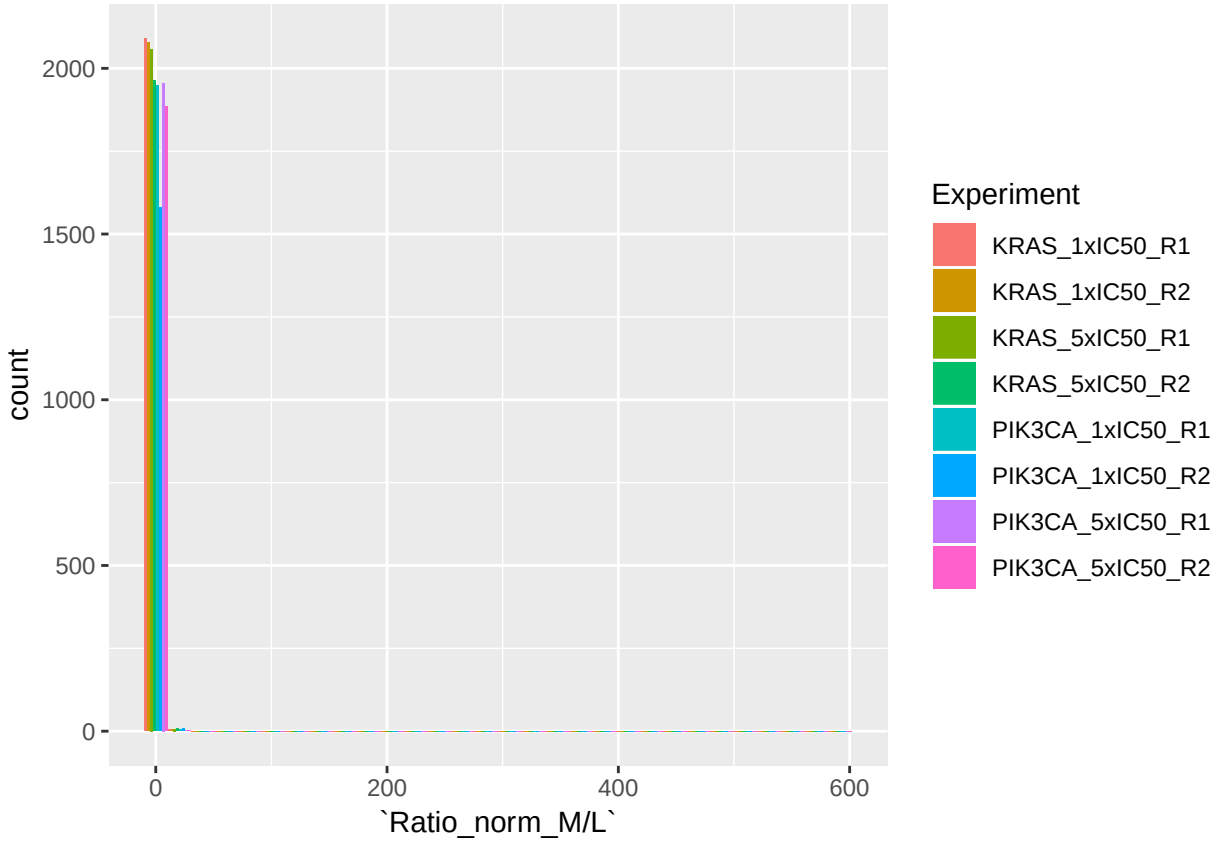
Introduction to ggplot2

Basic concepts of ggplot:

```
ggplot(data = table_merg) +
  geom_histogram(mapping = aes(x = `Ratio_norm_M/L`,fill = Experiment),position = "dodge")
```

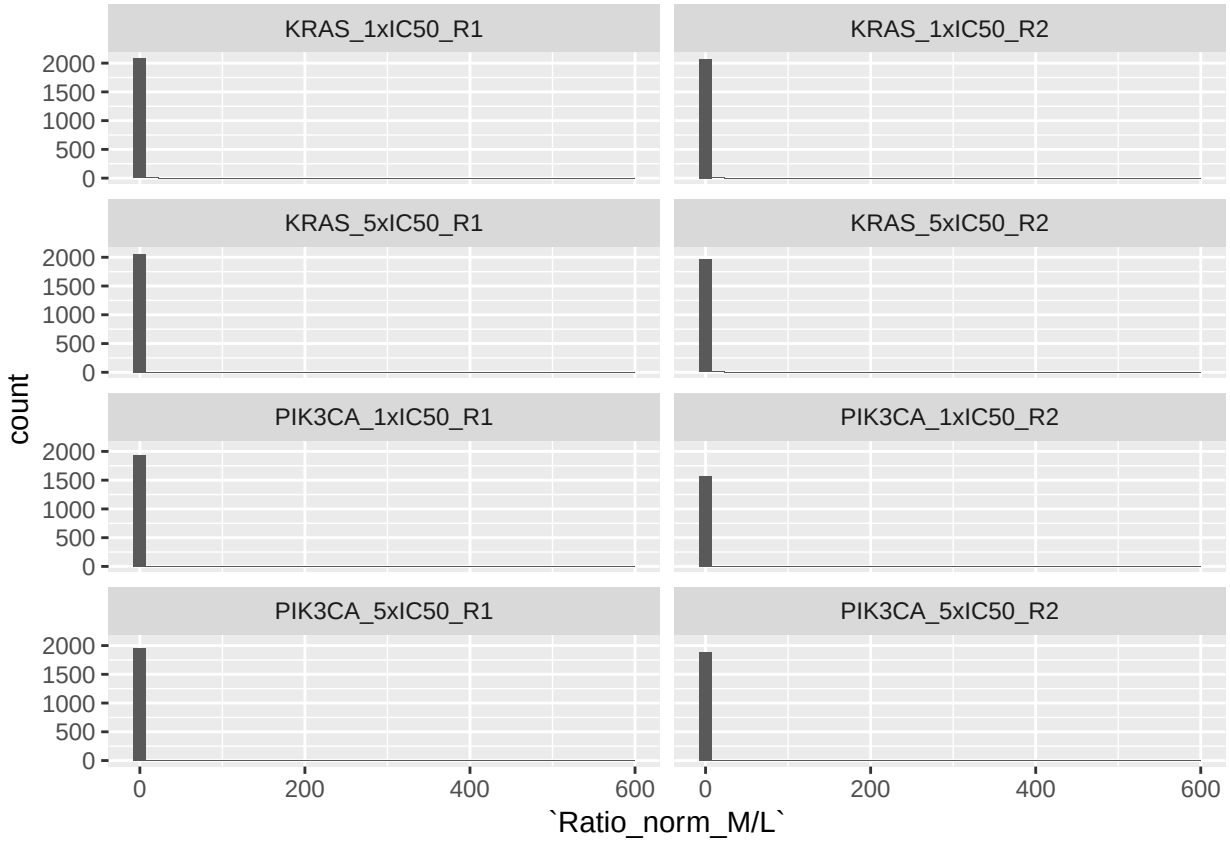
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 14818 rows containing non-finite values (stat_bin).
```




```
ggplot(data = table_merg) +
  geom_histogram(mapping = aes(x = `Ratio_norm_M/L`),bins = 40)+
  facet_wrap(~Experiment,ncol = 2)
```

```
## Warning: Removed 14818 rows containing non-finite values (stat_bin).
```

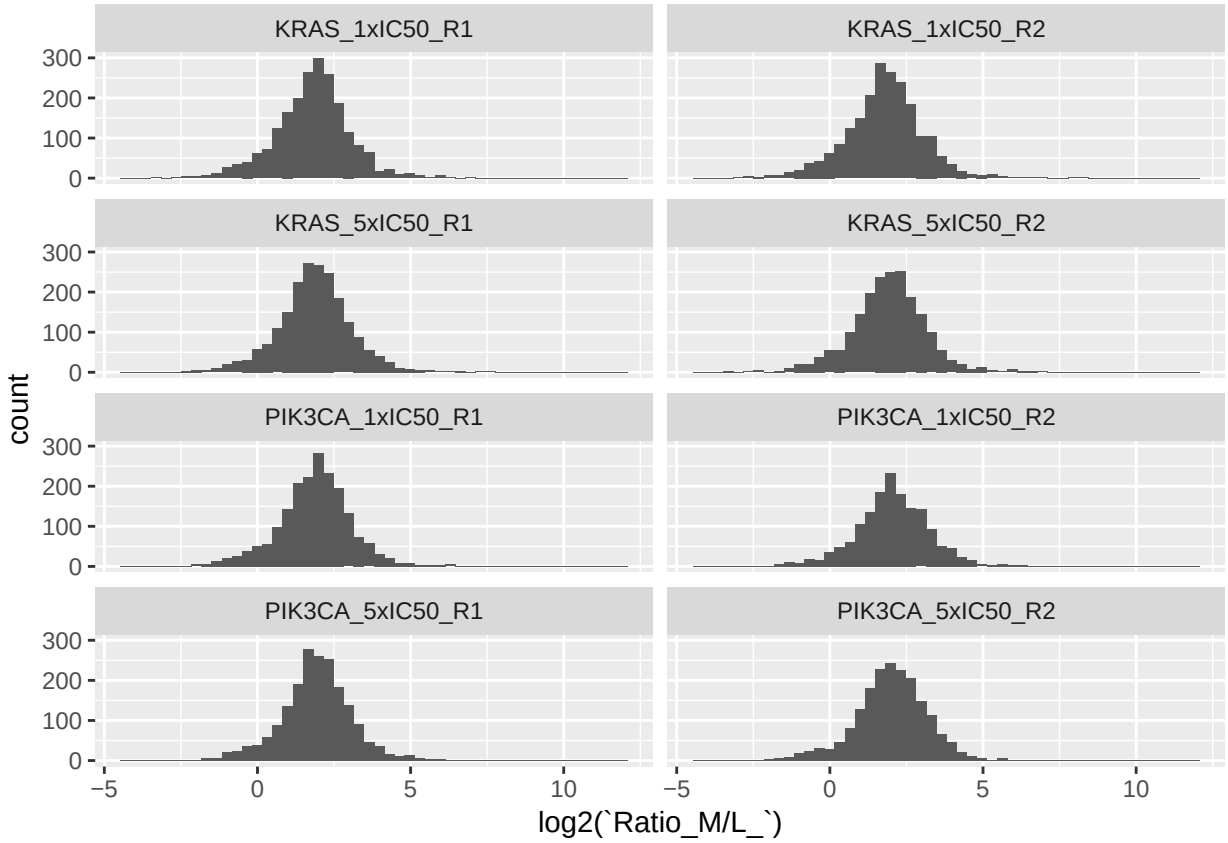



```
## ggplot can handle missing values
```

Appears that our data are not following a normal distribution. Log transform

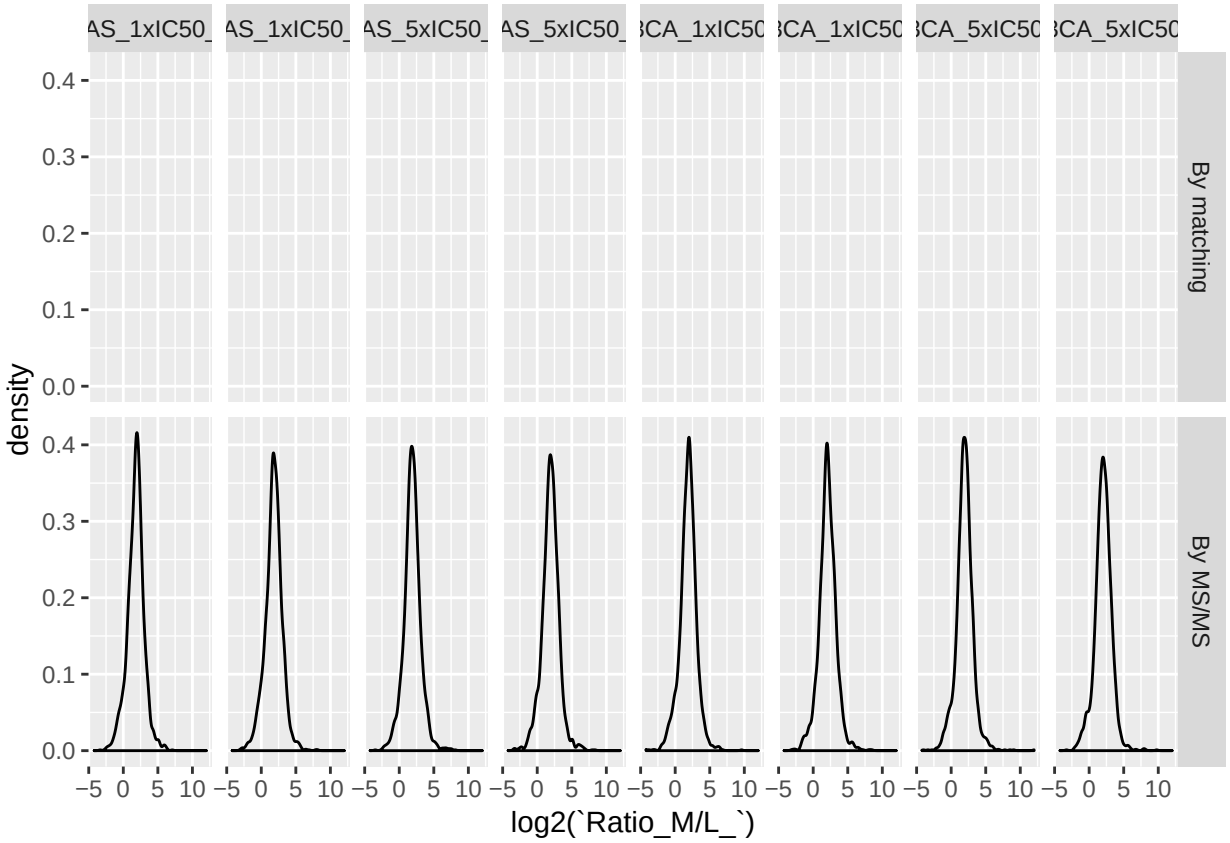
```
ggplot(data = table_merg) +
  geom_histogram(mapping = aes(x = log2(`Ratio_M/L_`)),bins = 50) +
  facet_wrap(~Experiment,ncol = 2)
```

```
## Warning: Removed 14818 rows containing non-finite values (stat_bin).
```




```
ggplot(data = table_merg) +
  geom_density(mapping = aes(x = log2(`Ratio_M/L_`))) +
  facet_grid(Ident_type ~ Experiment)
```

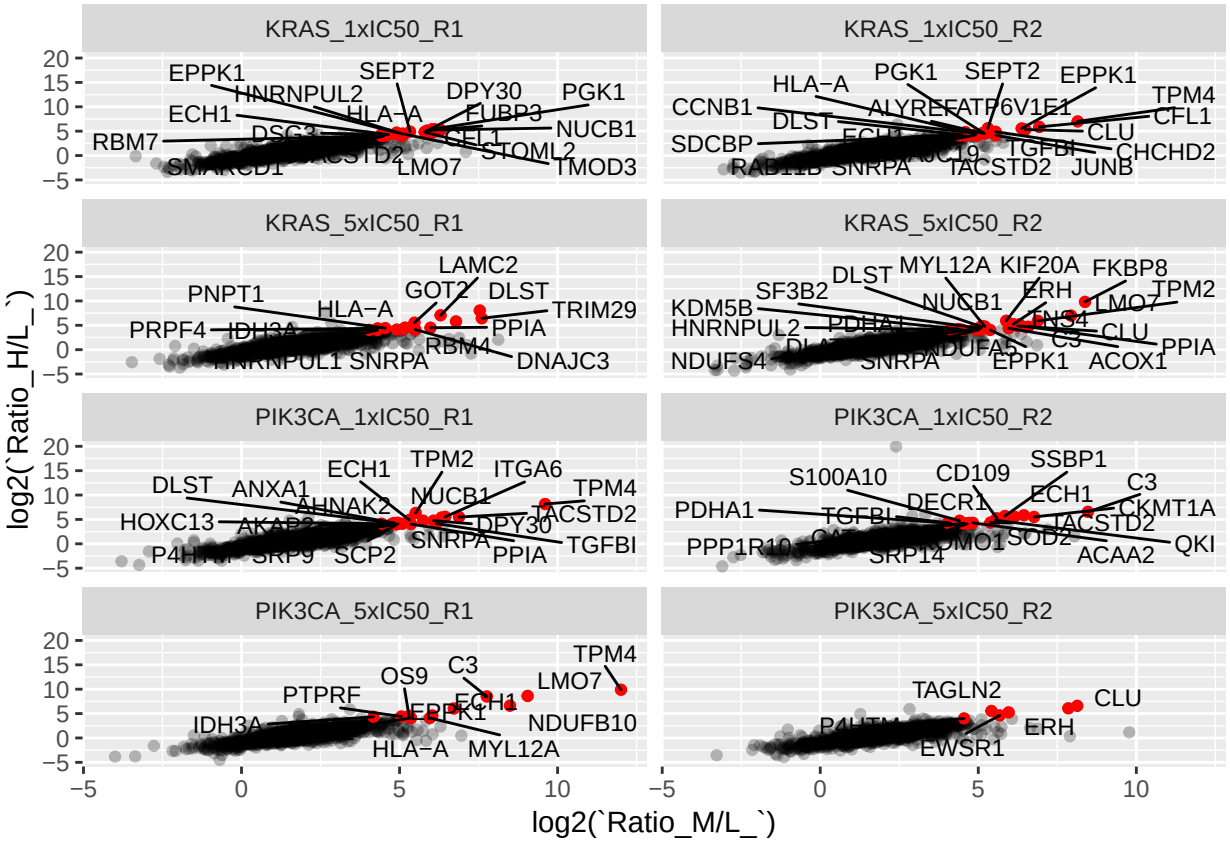
```
## Warning: Removed 14818 rows containing non-finite values (stat_density).
```




```
library("ggrepel") ### Quick way of adding labels to plots

ggplot(data = table_merg,aes(x = log2(`Ratio_M/L_`) , y = log2(`Ratio_H/L_`))) +

  geom_point(alpha = 0.25) +

    geom_point(data = table_merg %>% filter(log2(`Ratio_M/L_`)>4 & log2(`Ratio_H/L_`) >4), color = "red

  geom_text_repel(data = table_merg %>% filter(log2(`Ratio_M/L_`)>4 & log2(`Ratio_H/L_`) >4), mapping =
facet_wrap(~Experiment,ncol = 2.5)
```

```
## Warning: Coercing `ncol` to be an integer.
```

```
## Warning: Removed 14940 rows containing missing values (geom_point).
```



Loops

```
means <- vector ("double",ncol(table_merg %>% group_by(Experiment)))
for (i in seq_along(table_merg)) {
  means[[i]] <- mean(table_merg[[i]], na.rm = T)
}
```

```
## Warning in mean.default(table_merg[[i]], na.rm = T): argument is not
## numeric or logical: returning NA

## Warning in mean.default(table_merg[[i]], na.rm = T): argument is not
## numeric or logical: returning NA

## Warning in mean.default(table_merg[[i]], na.rm = T): argument is not
## numeric or logical: returning NA

## Warning in mean.default(table_merg[[i]], na.rm = T): argument is not
## numeric or logical: returning NA

## Warning in mean.default(table_merg[[i]], na.rm = T): argument is not
## numeric or logical: returning NA

## Warning in mean.default(table_merg[[i]], na.rm = T): argument is not
## numeric or logical: returning NA

## Warning in mean.default(table_merg[[i]], na.rm = T): argument is not
```

```
## numeric or logical: returning NA
```

```
print(means)
```

```
##  [1]           NA           NA           NA 1.128549e+00           NA
##  [6]           NA           NA 1.806581e+02 1.167639e+00 1.150466e+00
## [11] 3.377760e+00 3.378187e+00 3.409273e+00 6.854368e+01 5.651591e-01
## [16] 6.035793e+00 1.531941e+08 6.834835e+08 3.438022e+08 4.676065e+00
## [21]           NA 1.322201e+01
```

```
medians <- vector ("double",ncol(table_merg))
for (i in seq_along(table_merg)) {
  medians[[i]] <- median(table_merg[[i]], na.rm = T)
}
```

```
## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA
```

```
## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA
```

```
print(medians)
```

```
##  [1]           NA           NA           NA 1.00000e+00           NA
##  [6]           NA           NA 8.95470e-01 8.76840e-01 1.03600e+00
## [11] 2.00000e+00 2.00000e+00 2.00000e+00 1.94980e+00 5.09410e-01
## [16] 3.79555e+00 1.34770e+07 4.53315e+07 2.17575e+07 2.00000e+00
## [21]           NA 7.50000e+00
```

** Avoid using for loops **

Use the Purrr package instead

purrr is kind of like dplyr for lists. It helps you repeatedly apply functions.

```
library("purrr")
```

map is a slightly improved version of lapply and it is quite powerfull and only returns a list

```
map(1:4, log)
```

```
## [[1]]
## [1] 0
##
```

```
## [[2]]
## [1] 0.6931472
##
## [[3]]
## [1] 1.098612
##
## [[4]]
## [1] 1.386294
```

```
map(1:4, log, base = 2) # Argument
```

```
## [[1]]
## [1] 0
##
## [[2]]
## [1] 1
##
## [[3]]
## [1] 1.584963
##
## [[4]]
## [1] 2
```

```
map(1:4, ~ log(4, base = .x)) # formula, map(1:4, function(x) log(4, base = x))
```

```
## [[1]]
## [1] Inf
##
## [[2]]
## [1] 2
##
## [[3]]
## [1] 1.26186
##
## [[4]]
## [1] 1
```

```
map_dbl(c(1:4,0), log, base = 2)
```

```
## [1] 0.000000 1.000000 1.584963 2.000000     -Inf
```

```
means <- map_dbl(mtcars,mean)
medians <- map_dbl(mtcars,median)
```

Transform the data frame filter missing values

```
table_merg_f <- table_merg %>% mutate_at(vars(`Ratio_norm_H/L`:`Ratio_norm_H/M`,`Ratio_H/L_`:`Ratio_M/L_
  mutate_at(vars(Intensity_L:Intensity_H),log10)

## Change NaN and -Inf to NAs

#install.packages("naniar")
library("naniar")

table_merg_f <- table_merg_f %>% replace_with_na_all(condition = ~.x == -Inf)
### Easy to do but quite slow
```

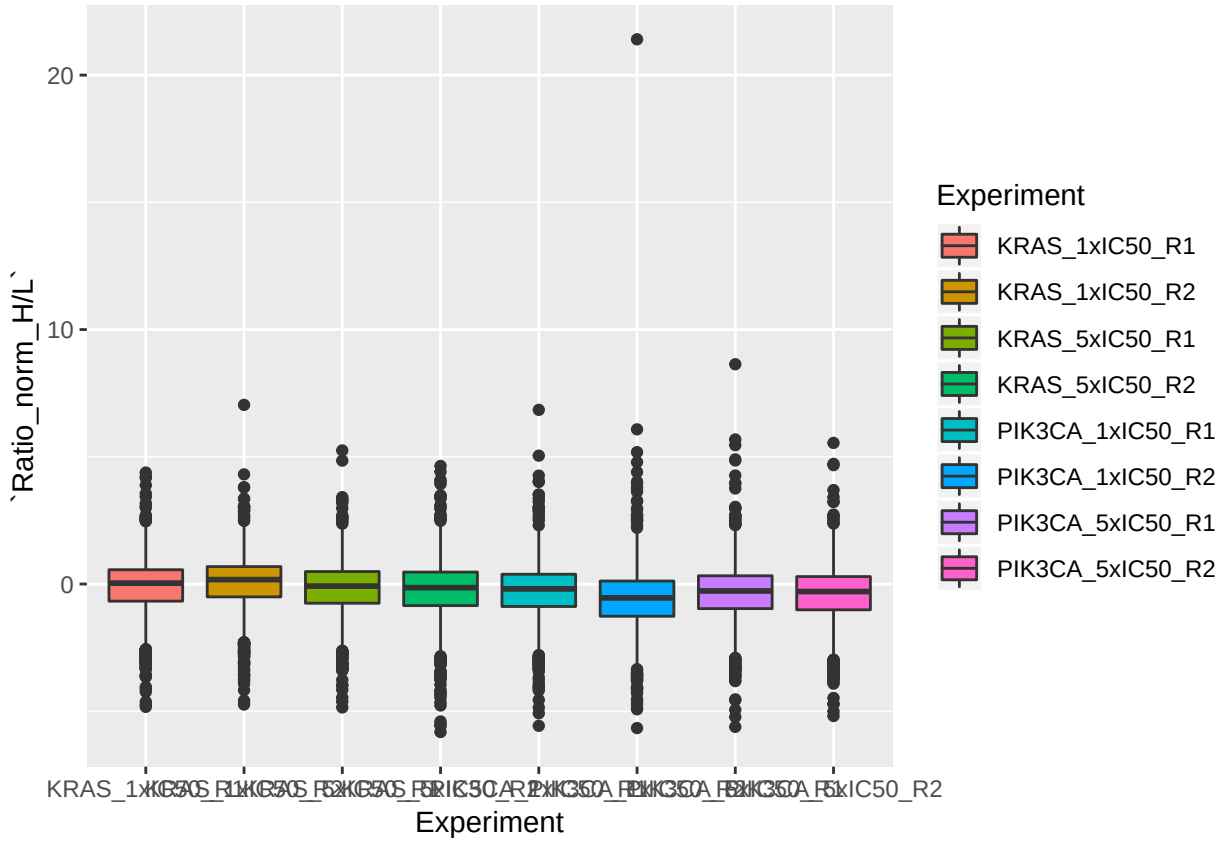
```
is.na(table_merg_f) <- sapply(table_merg_f,is.infinite)
```

More Visualization

Boxplots

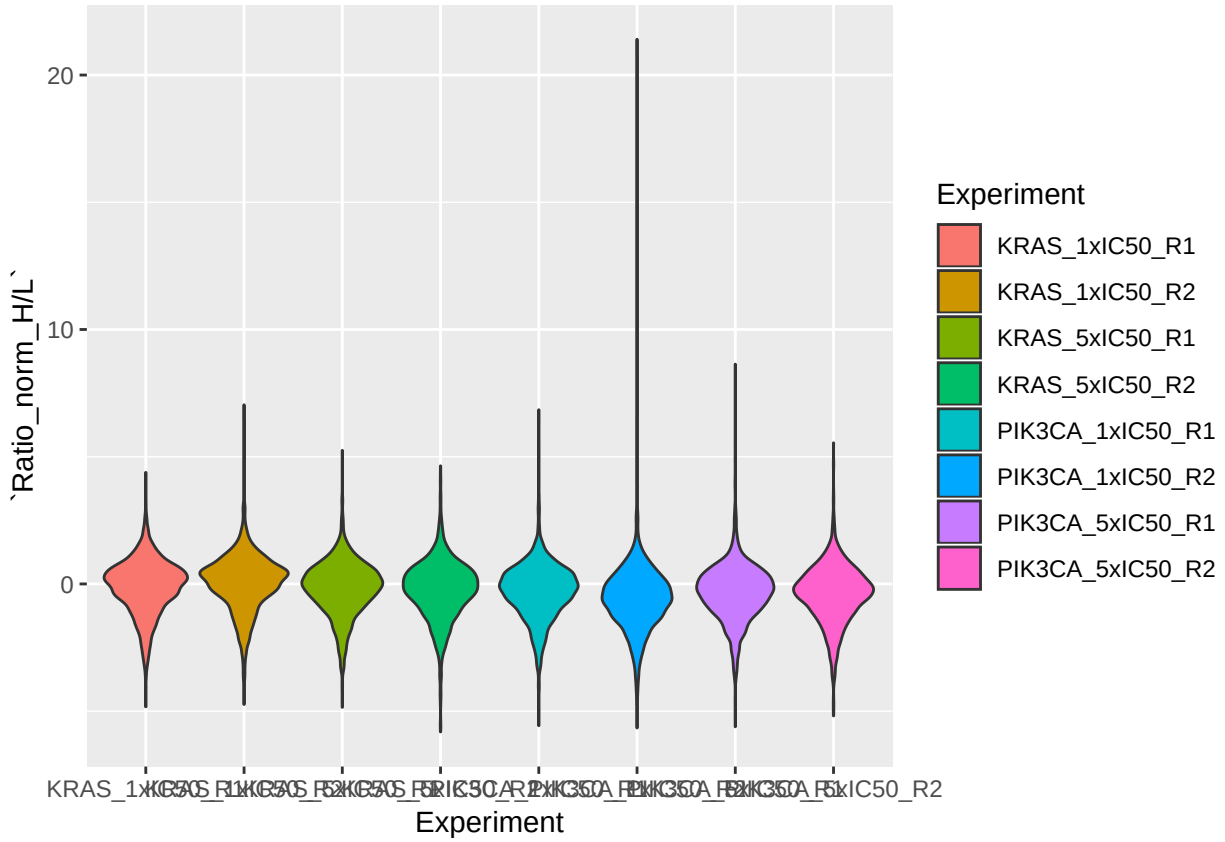
```
ggplot(data = table_merg_f, aes(x =  Experiment ,y = `Ratio_norm_H/L`,fill = Experiment))+
  geom_boxplot()
```

```
## Warning: Removed 14938 rows containing non-finite values (stat_boxplot).
```

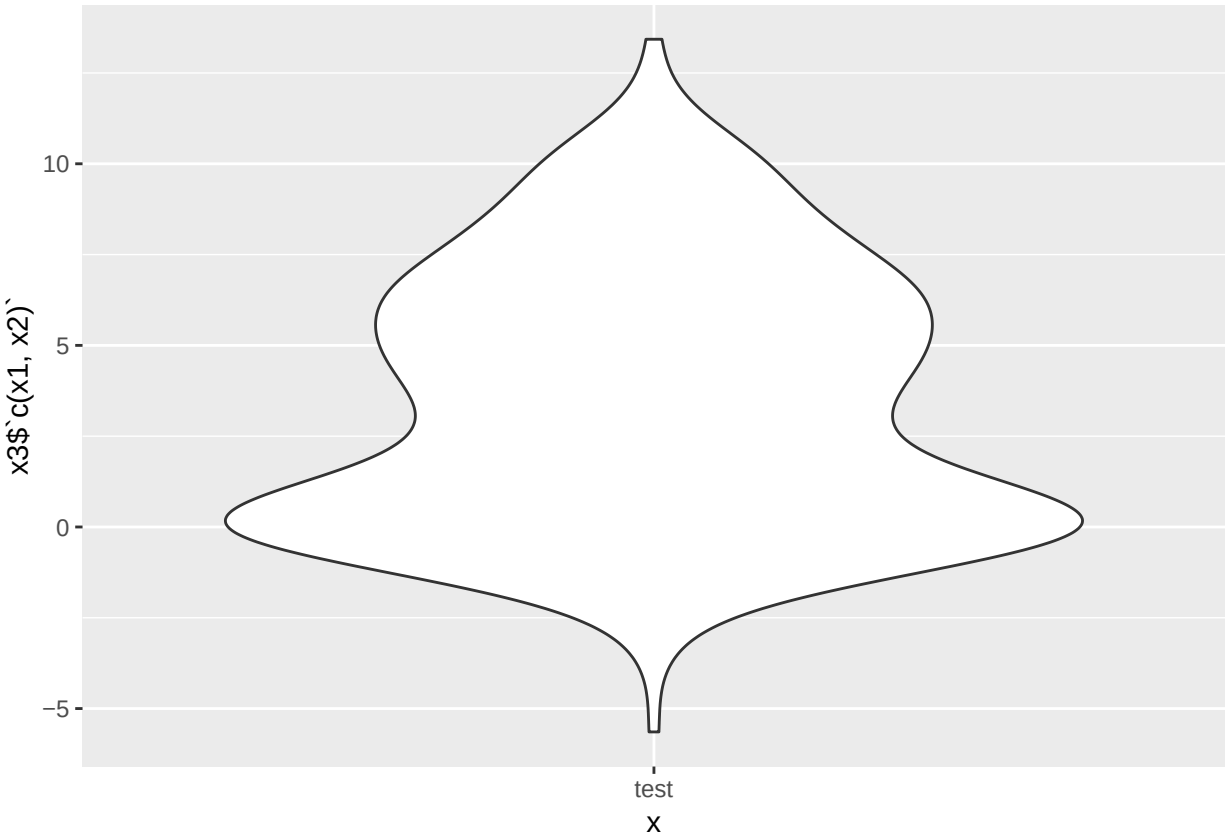



```
ggplot(data = table_merg_f, aes(x =  Experiment ,y = `Ratio_norm_H/L`,fill = Experiment))+
  geom_violin()
```

```
## Warning: Removed 14938 rows containing non-finite values (stat_ydensity).
```

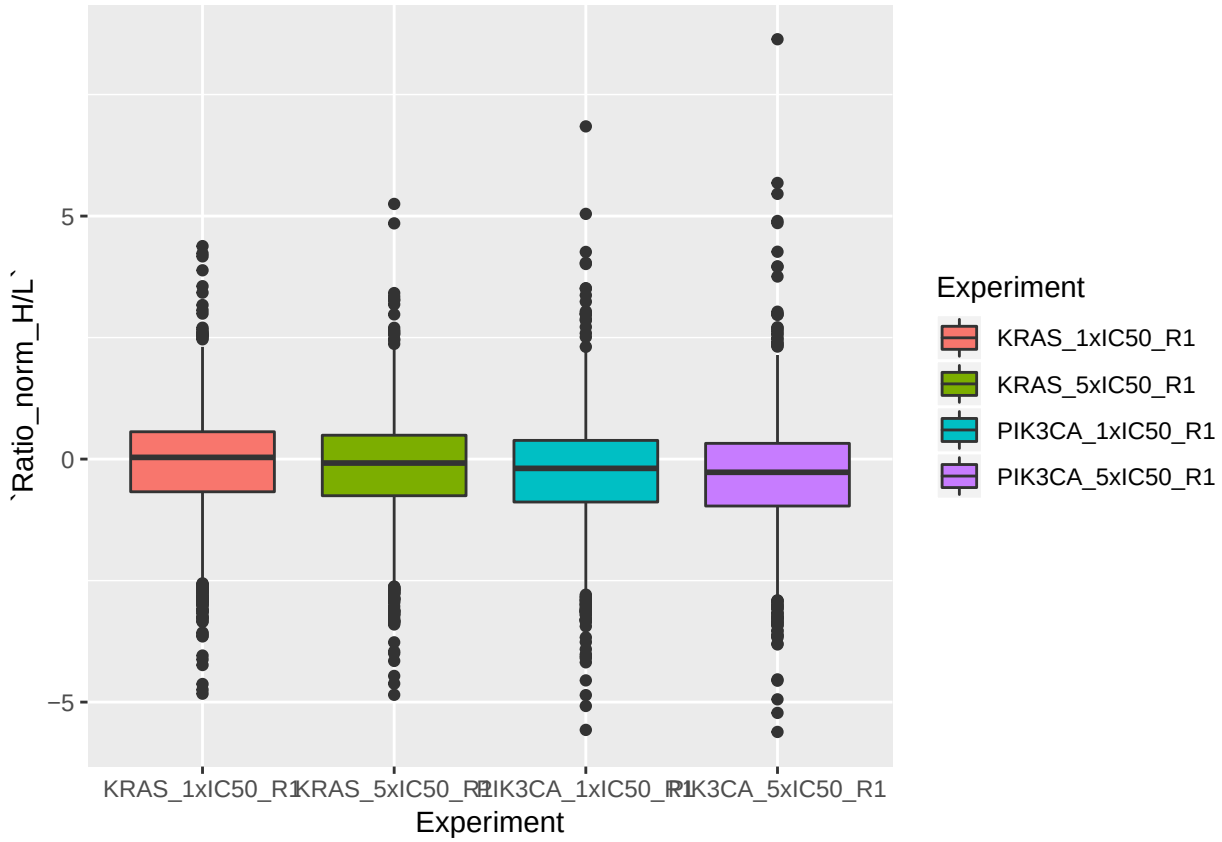



```
###Bimodal distribution

x1 <-  rnorm(100,mean=0 ,sd=1)
x2 <-  rnorm (200, mean=5, sd = 3)
x3 <- data_frame(c(x1,x2))

ggplot(data = x3, aes(x= "test",y = x3$`c(x1, x2)`))+
  geom_violin()
```




```
ggplot(data = table_merg_f %>% filter(str_detect(Experiment,"R1")),aes(x =  Experiment ,y = `Ratio_norm
 geom_boxplot()
```

```
## Warning: Removed 7198 rows containing non-finite values (stat_boxplot).
```

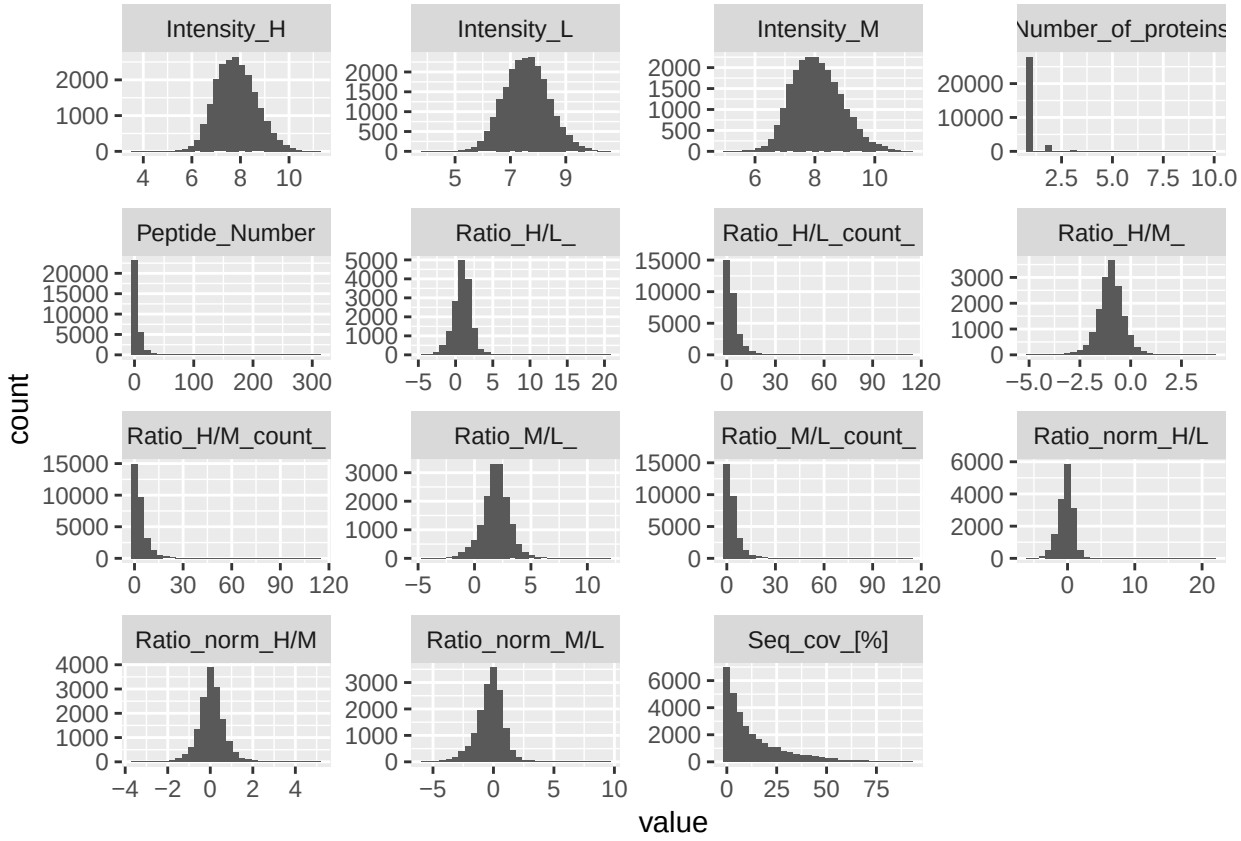


Quick visualization

```
table_merg_f %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

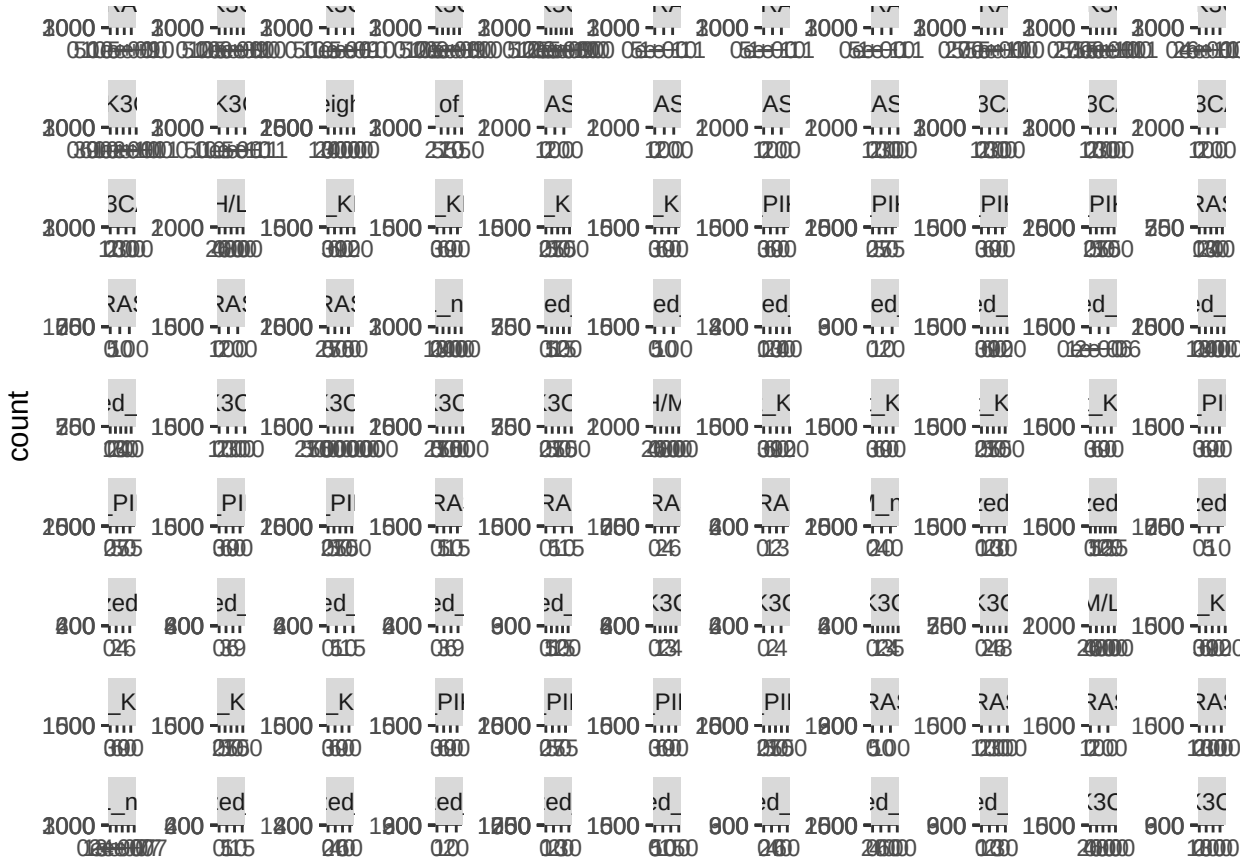
```
## Warning: Removed 114645 rows containing non-finite values (stat_bin).
```




```
prot_f1 %>% keep(is.numeric) %>%
  gather() %>% ggplot(aes(value))+
  facet_wrap(~ key, scales = "free") +
    geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 90735 rows containing non-finite values (stat_bin).
```




```
### R scatter plots

table_merg_f <-  table_merg_f %>% mutate (Replicate = ifelse(str_detect(Experiment,"R1"),paste("R1"),"R

library("GGally")
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:dplyr':
##
##     nasa
```

```
table_merg_f %>% select(Protein_IDs, Experiment,`Ratio_norm_H/L`) %>% spread(key = "Experiment",value =
```

```
## Warning: Removed 1715 rows containing non-finite values (stat_density).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 1969 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2005 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2077 rows containing missing values
```

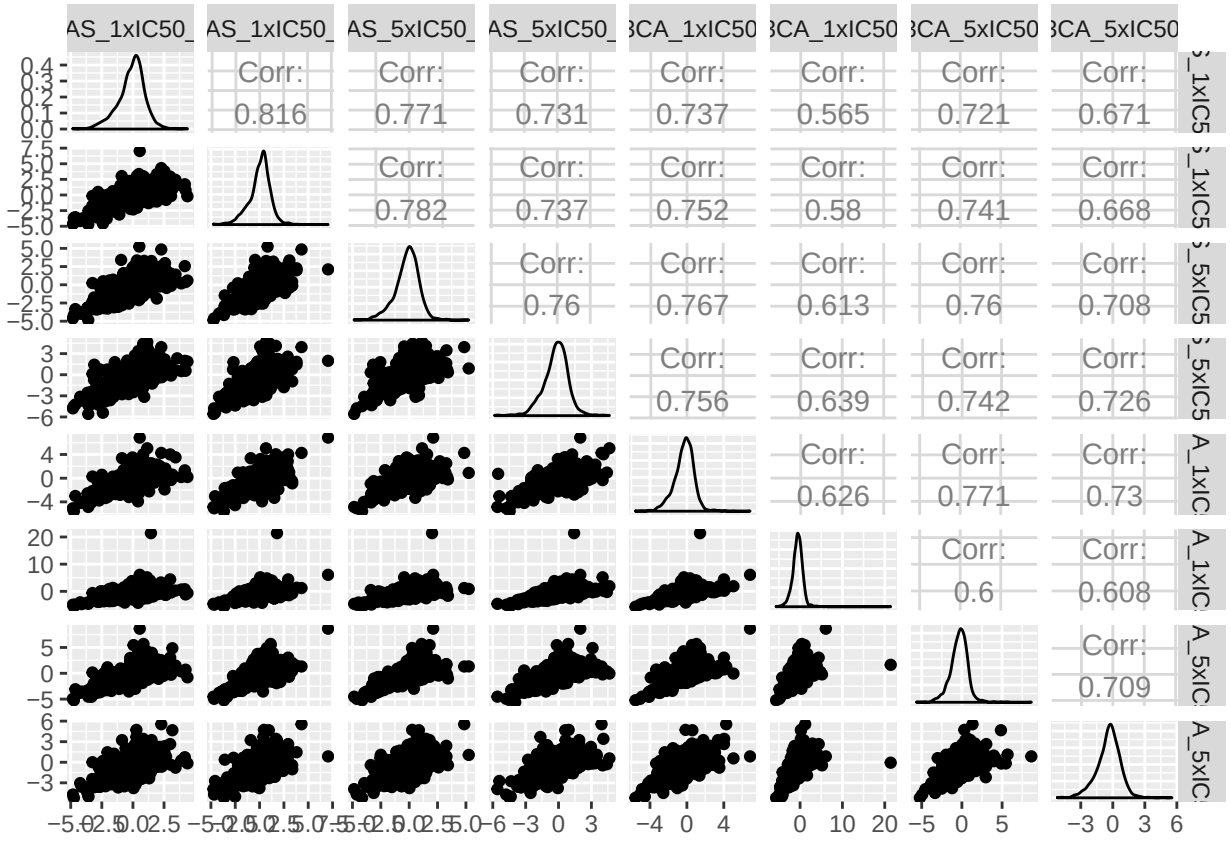
```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2116 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
```

```
## "pearson", : Removed 2405 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2120 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2207 rows containing missing values

## Warning: Removed 1969 rows containing missing values (geom_point).

## Warning: Removed 1727 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2024 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2085 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2124 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2405 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2118 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2218 rows containing missing values

## Warning: Removed 2005 rows containing missing values (geom_point).

## Warning: Removed 2024 rows containing missing values (geom_point).

## Warning: Removed 1755 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2079 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2133 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2400 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2112 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2207 rows containing missing values

## Warning: Removed 2077 rows containing missing values (geom_point).

## Warning: Removed 2085 rows containing missing values (geom_point).

## Warning: Removed 2079 rows containing missing values (geom_point).

## Warning: Removed 1841 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2128 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2393 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2119 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2204 rows containing missing values

## Warning: Removed 2116 rows containing missing values (geom_point).

## Warning: Removed 2124 rows containing missing values (geom_point).

## Warning: Removed 2133 rows containing missing values (geom_point).

## Warning: Removed 2128 rows containing missing values (geom_point).

## Warning: Removed 1867 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2392 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2100 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2190 rows containing missing values

## Warning: Removed 2405 rows containing missing values (geom_point).

## Warning: Removed 2405 rows containing missing values (geom_point).

## Warning: Removed 2400 rows containing missing values (geom_point).

## Warning: Removed 2393 rows containing missing values (geom_point).

## Warning: Removed 2392 rows containing missing values (geom_point).

## Warning: Removed 2228 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2393 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2391 rows containing missing values

## Warning: Removed 2120 rows containing missing values (geom_point).

## Warning: Removed 2118 rows containing missing values (geom_point).

## Warning: Removed 2112 rows containing missing values (geom_point).

## Warning: Removed 2119 rows containing missing values (geom_point).

## Warning: Removed 2100 rows containing missing values (geom_point).

## Warning: Removed 2393 rows containing missing values (geom_point).

## Warning: Removed 1861 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2190 rows containing missing values

## Warning: Removed 2207 rows containing missing values (geom_point).

## Warning: Removed 2218 rows containing missing values (geom_point).

## Warning: Removed 2207 rows containing missing values (geom_point).

## Warning: Removed 2204 rows containing missing values (geom_point).
```

```
## Warning: Removed 2190 rows containing missing values (geom_point).
## Warning: Removed 2391 rows containing missing values (geom_point).
## Warning: Removed 2190 rows containing missing values (geom_point).
## Warning: Removed 1944 rows containing non-finite values (stat_density).
```

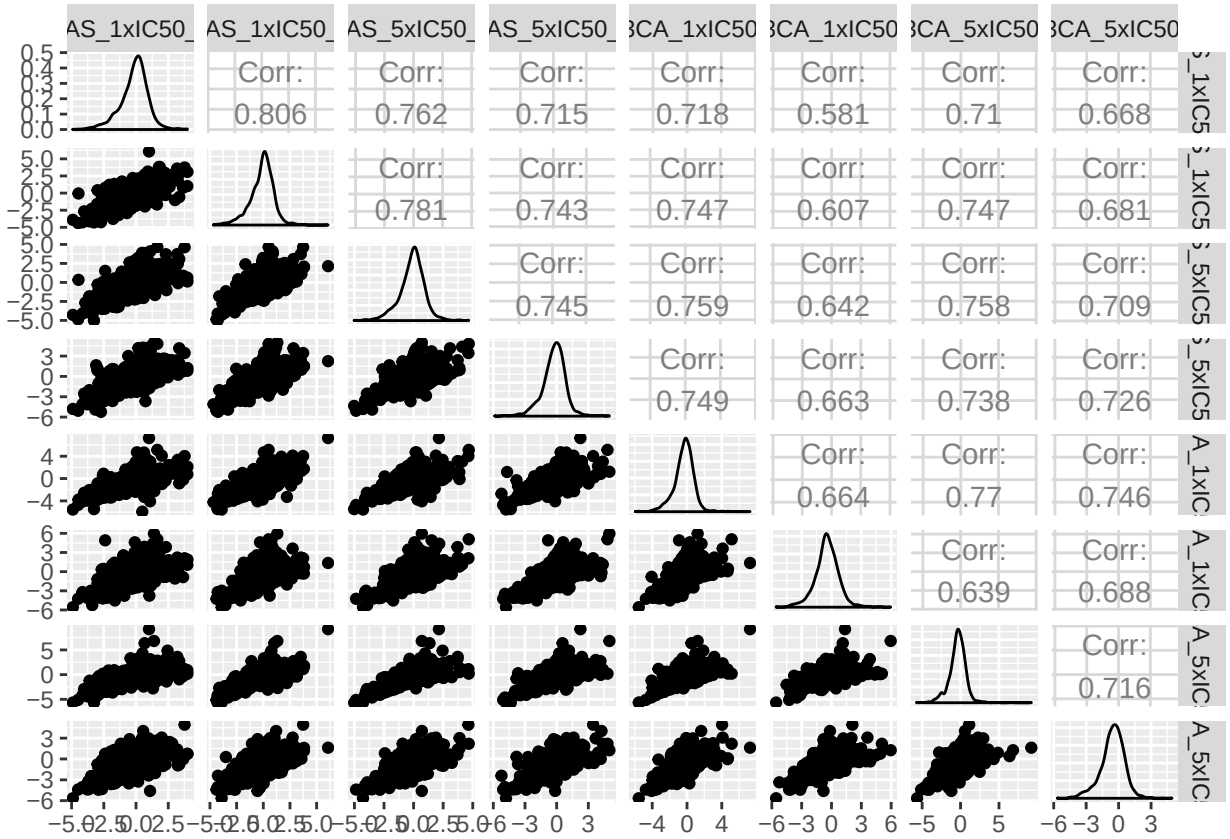



```
table_merg_f %>% select(Protein_IDs, Experiment,`Ratio_norm_M/L`) %>% spread(key = "Experiment",value =
```

```
## Warning: Removed 1708 rows containing non-finite values (stat_density).
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 1963 rows containing missing values
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 1996 rows containing missing values
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2072 rows containing missing values
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2103 rows containing missing values
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2395 rows containing missing values
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2108 rows containing missing values
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2186 rows containing missing values
```

```
## Warning: Removed 1963 rows containing missing values (geom_point).

## Warning: Removed 1720 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2013 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2078 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2109 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2392 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2104 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2200 rows containing missing values

## Warning: Removed 1996 rows containing missing values (geom_point).

## Warning: Removed 2013 rows containing missing values (geom_point).

## Warning: Removed 1739 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2069 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2118 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2388 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2099 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2186 rows containing missing values

## Warning: Removed 2072 rows containing missing values (geom_point).

## Warning: Removed 2078 rows containing missing values (geom_point).

## Warning: Removed 2069 rows containing missing values (geom_point).

## Warning: Removed 1831 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2113 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2381 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2106 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2184 rows containing missing values

## Warning: Removed 2103 rows containing missing values (geom_point).

## Warning: Removed 2109 rows containing missing values (geom_point).
```

```
## Warning: Removed 2118 rows containing missing values (geom_point).

## Warning: Removed 2113 rows containing missing values (geom_point).

## Warning: Removed 1848 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2376 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2082 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2168 rows containing missing values

## Warning: Removed 2395 rows containing missing values (geom_point).

## Warning: Removed 2392 rows containing missing values (geom_point).

## Warning: Removed 2388 rows containing missing values (geom_point).

## Warning: Removed 2381 rows containing missing values (geom_point).

## Warning: Removed 2376 rows containing missing values (geom_point).

## Warning: Removed 2213 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2376 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2378 rows containing missing values

## Warning: Removed 2108 rows containing missing values (geom_point).

## Warning: Removed 2104 rows containing missing values (geom_point).

## Warning: Removed 2099 rows containing missing values (geom_point).

## Warning: Removed 2106 rows containing missing values (geom_point).

## Warning: Removed 2082 rows containing missing values (geom_point).

## Warning: Removed 2376 rows containing missing values (geom_point).

## Warning: Removed 1842 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2168 rows containing missing values

## Warning: Removed 2186 rows containing missing values (geom_point).

## Warning: Removed 2200 rows containing missing values (geom_point).

## Warning: Removed 2186 rows containing missing values (geom_point).

## Warning: Removed 2184 rows containing missing values (geom_point).

## Warning: Removed 2168 rows containing missing values (geom_point).

## Warning: Removed 2378 rows containing missing values (geom_point).

## Warning: Removed 2168 rows containing missing values (geom_point).

## Warning: Removed 1917 rows containing non-finite values (stat_density).
```




```
table_merg_f %>% select(Protein_IDs, Experiment,`Ratio_norm_H/M`) %>% spread(key = "Experiment",value =
```

```
## Warning: Removed 1715 rows containing non-finite values (stat_density).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 1969 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2005 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2077 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2116 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2405 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2120 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2208 rows containing missing values
```

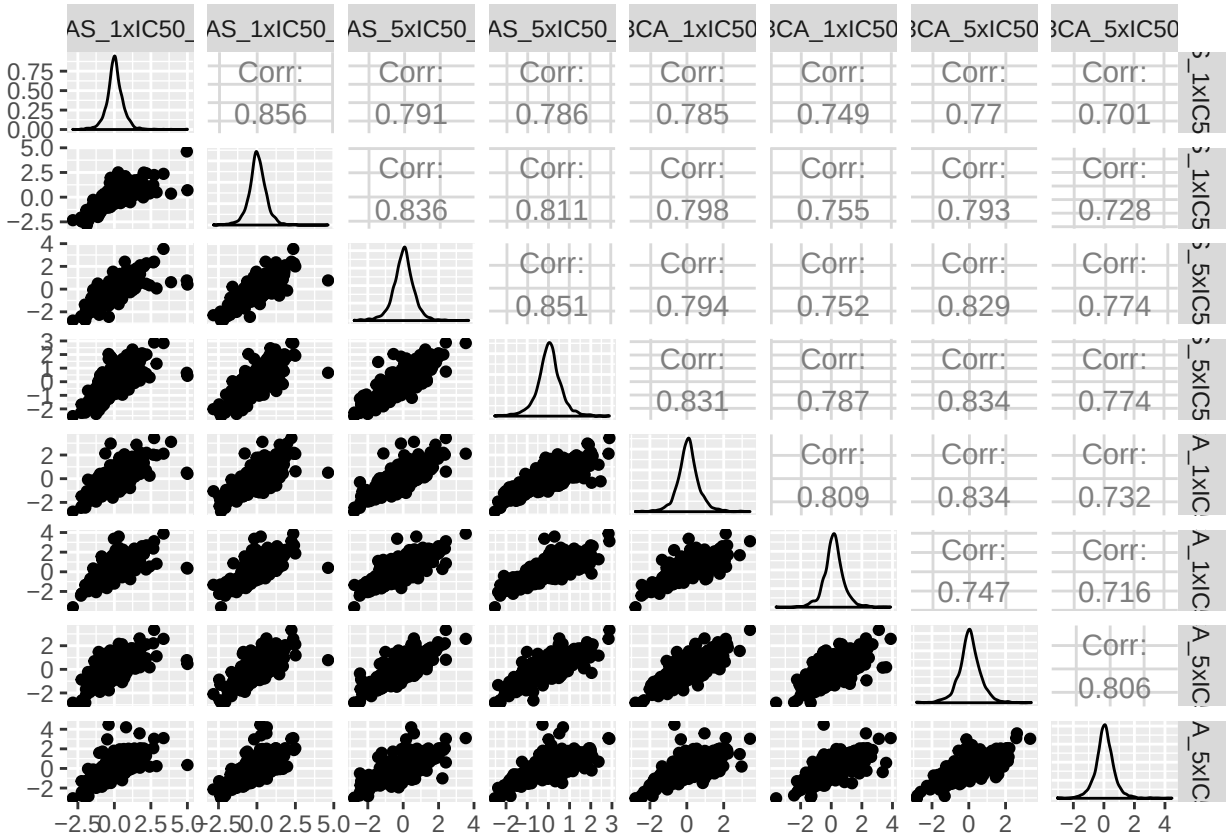
```
## Warning: Removed 1969 rows containing missing values (geom_point).
```

```
## Warning: Removed 1727 rows containing non-finite values (stat_density).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2024 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2085 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2124 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2405 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2118 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2219 rows containing missing values

## Warning: Removed 2005 rows containing missing values (geom_point).

## Warning: Removed 2024 rows containing missing values (geom_point).

## Warning: Removed 1755 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2079 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2133 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2400 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2112 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2208 rows containing missing values

## Warning: Removed 2077 rows containing missing values (geom_point).

## Warning: Removed 2085 rows containing missing values (geom_point).

## Warning: Removed 2079 rows containing missing values (geom_point).

## Warning: Removed 1841 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2128 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2393 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2119 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2205 rows containing missing values

## Warning: Removed 2116 rows containing missing values (geom_point).

## Warning: Removed 2124 rows containing missing values (geom_point).

## Warning: Removed 2133 rows containing missing values (geom_point).

## Warning: Removed 2128 rows containing missing values (geom_point).

## Warning: Removed 1867 rows containing non-finite values (stat_density).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2392 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2100 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2191 rows containing missing values

## Warning: Removed 2405 rows containing missing values (geom_point).

## Warning: Removed 2405 rows containing missing values (geom_point).

## Warning: Removed 2400 rows containing missing values (geom_point).

## Warning: Removed 2393 rows containing missing values (geom_point).

## Warning: Removed 2392 rows containing missing values (geom_point).

## Warning: Removed 2228 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2393 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2392 rows containing missing values

## Warning: Removed 2120 rows containing missing values (geom_point).

## Warning: Removed 2118 rows containing missing values (geom_point).

## Warning: Removed 2112 rows containing missing values (geom_point).

## Warning: Removed 2119 rows containing missing values (geom_point).

## Warning: Removed 2100 rows containing missing values (geom_point).

## Warning: Removed 2393 rows containing missing values (geom_point).

## Warning: Removed 1861 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 2190 rows containing missing values

## Warning: Removed 2208 rows containing missing values (geom_point).

## Warning: Removed 2219 rows containing missing values (geom_point).

## Warning: Removed 2208 rows containing missing values (geom_point).

## Warning: Removed 2205 rows containing missing values (geom_point).

## Warning: Removed 2191 rows containing missing values (geom_point).

## Warning: Removed 2392 rows containing missing values (geom_point).

## Warning: Removed 2190 rows containing missing values (geom_point).

## Warning: Removed 1945 rows containing non-finite values (stat_density).
```




```
funs <- list(mean, median ,sd) # In R you can store everything in a list

  funs %>% map(~table_merg_f%>% map_dbl(.x ),.funs)
```

```
## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(.x[[i]], ...): argument is not numeric or logical:
```

```
## returning NA
## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]):
## argument is not numeric or logical: returning NA
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm
## = na.rm): NAs introduced by coercion

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm
## = na.rm): NAs introduced by coercion

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm
## = na.rm): NAs introduced by coercion

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm
## = na.rm): NAs introduced by coercion

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm
## = na.rm): NAs introduced by coercion

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm
## = na.rm): NAs introduced by coercion

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm
## = na.rm): NAs introduced by coercion

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm
## = na.rm): NAs introduced by coercion
## [[1]]
##          Protein_IDs Majority_protein_IDs        Protein_names
##                   NA                   NA                   NA
##   Number_of_proteins           Gene_names        Fasta_headers
##             1.128549                   NA                   NA
```

```
##           Experiment       Ratio_norm_H/L       Ratio_norm_M/L
##                   NA                   NA                   NA
##       Ratio_norm_H/M     Ratio_H/L_count_     Ratio_H/M_count_
##                   NA             3.377760             3.378187
##     Ratio_M/L_count_           Ratio_H/L_           Ratio_H/M_
##             3.409273                   NA                   NA
##           Ratio_M/L_          Intensity_L          Intensity_M
##                   NA                   NA                   NA
##          Intensity_H       Peptide_Number           Ident_type
##                   NA             4.676065                   NA
##          Seq_cov_[%]            Replicate
##            13.222010                   NA
##
## [[2]]
##          Protein_IDs Majority_protein_IDs        Protein_names
##                   NA                   NA                   NA
##   Number_of_proteins           Gene_names        Fasta_headers
##                  1.0                   NA                   NA
##           Experiment       Ratio_norm_H/L       Ratio_norm_M/L
##                   NA                   NA                   NA
##       Ratio_norm_H/M     Ratio_H/L_count_     Ratio_H/M_count_
##                   NA                  2.0                  2.0
##     Ratio_M/L_count_           Ratio_H/L_           Ratio_H/M_
##                  2.0                   NA                   NA
##           Ratio_M/L_          Intensity_L          Intensity_M
##                   NA                   NA                   NA
##          Intensity_H       Peptide_Number           Ident_type
##                   NA                  2.0                   NA
##          Seq_cov_[%]            Replicate
##                  7.5                   NA
##
## [[3]]
##          Protein_IDs Majority_protein_IDs        Protein_names
##                   NA                   NA                   NA
##   Number_of_proteins           Gene_names        Fasta_headers
##            0.5469966                   NA                   NA
##           Experiment       Ratio_norm_H/L       Ratio_norm_M/L
##                   NA                   NA                   NA
##       Ratio_norm_H/M     Ratio_H/L_count_     Ratio_H/M_count_
##                   NA             5.5348762            5.5350163
##     Ratio_M/L_count_           Ratio_H/L_           Ratio_H/M_
##             5.5553991                   NA                   NA
##           Ratio_M/L_          Intensity_L          Intensity_M
##                   NA                   NA                   NA
##          Intensity_H       Peptide_Number           Ident_type
##                   NA            9.5402629                   NA
##          Seq_cov_[%]            Replicate
##           15.4108825                   NA
```

If we end up having more time we will continue with more exploratory data analysis and ploting stuff