Correlation

Paul Esker and Felipe Dalla Lana

Background

Correlation analysis is helpful to identify the associations between different variables (measurements). For databases with combinations of qualitative and quantitative data, we use this as a preliminary step to understand the likely relationships, or potential explanatory value of different measurements. We will apply some examples here based on *tidyverse* to estimate the correlation coefficients based on different methods. We will also visualize the associations graphically. Two primary packages we need for this example are *Hmisc* y de *corrplot*. We will also use the package *readr* to read data into R.

```
library(tidyverse)
```

```
## ── Attaching packages ───────────────────────────────────────────────────────────
─────────────────────────────────────────────────────────────────────────────────────
────────────────── tidyverse 1.2.1 ──
```

```
## ✔ ggplot2 3.2.0     ✔ purrr   0.3.2
## ✔ tibble  2.1.3     ✔ dplyr   0.8.2
## ✔ tidyr   0.8.3     ✔ stringr 1.4.0
## ✔ readr   1.3.1     ✔ forcats 0.4.0
```

```
## ── Conflicts ────────────────────────────────────────────────────────────────────────
─────────────────────────────────────────────────────────────────────────────────────
──────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(readr)
```

Database

There are different options for working with data that is in a local folder. For many, the manual options with a data import are easier, but it is also useful to understand how you can directly read data into R. We will use both at time during the workshop, so do not stress too much for now.

```
# Introduce the data to R - in this situation, we apply the function read_csv the most i
mportant item is to know the physical location of the file. In this example, I mainain a
copy in Documents folder on my Mac

correlations <- read_csv("~/Documents/Correlations.csv")
```

```
## Parsed with column specification:
## cols(
##   Treatment = col_double(),
##   Count1 = col_double(),
##   Count2 = col_double(),
##   Yield = col_double(),
##   Protein = col_double(),
##   Oil = col_double()
## )
```

```
correlations
```

Treatment <dbl>	Count1 <dbl>	Count2 <dbl>	Yield <dbl>	Protein <dbl>	Oil <dbl>
1	241.0	241.0	2568.9	35.4	19.6

Treatment <dbl>	Count1 <dbl>	Count2 <dbl>	Yield <dbl>	Protein <dbl>	Oil <dbl>
1	241.0	249.6	2905.3	35.8	19.5
1	241.0	249.6	3186.4	36.2	19.3
2	395.9	482.0	2887.4	36.0	19.0
2	284.0	275.4	3388.9	36.3	19.4
2	292.7	309.9	3481.8	35.5	19.3
3	464.8	473.4	2836.1	35.6	19.5
3	421.8	456.2	3360.6	36.2	19.7
3	370.1	378.7	3569.2	36.4	19.1
4	413.2	447.6	2918.6	33.9	20.0

1-10 of 54 rows Previous **1** 2 3 4 5 6 Next

Pearson

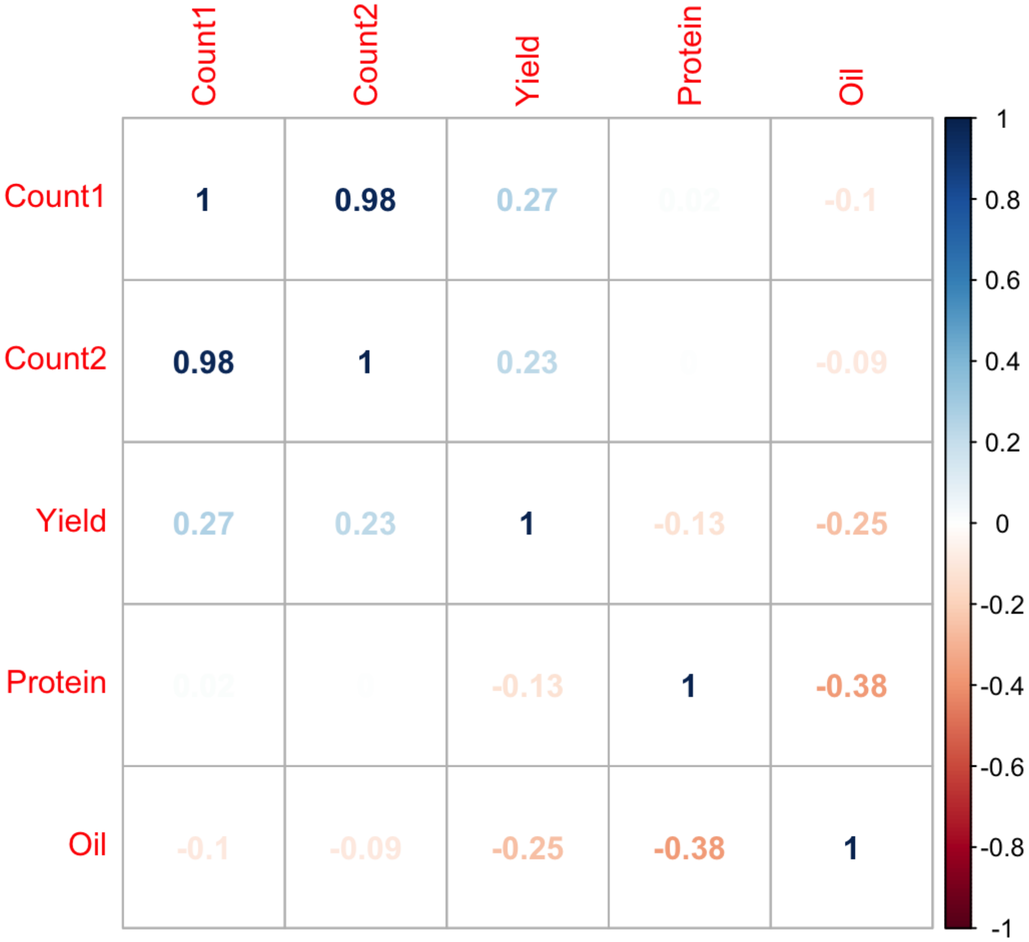
We will begin with the first type of correlation, which is the Pearson correlation. In this situation, we assume that we have quantitative variables. Depending on the database, you may just define the function by calling the name of the database. Nonetheless, we do need to understand our database and “clean” this some, especially to ignore the first column that defines some treatment. We will then use the function *rcorr*. This function allows us to perform two types of analyses: (1) Pearson and (2) Spearman (nonparametric method).

In R, and this is something that will carry throughout different types of models and analyses, there are often different packages and functions that we can use. Each has its advantages and disadvantages, for example, some functions do not provide a test statistic. In other cases the method does not permit the use of some of the graphical methods to visualize the associations.

```
# In this first example, the "select" option is indicating that we will use all columns
 except the first one, which is for treatment

example_cor <- correlations %>%
  select(-Treatment) %>%
  as.matrix() %>%
  rcorr(type = "pearson")

example_cor
```

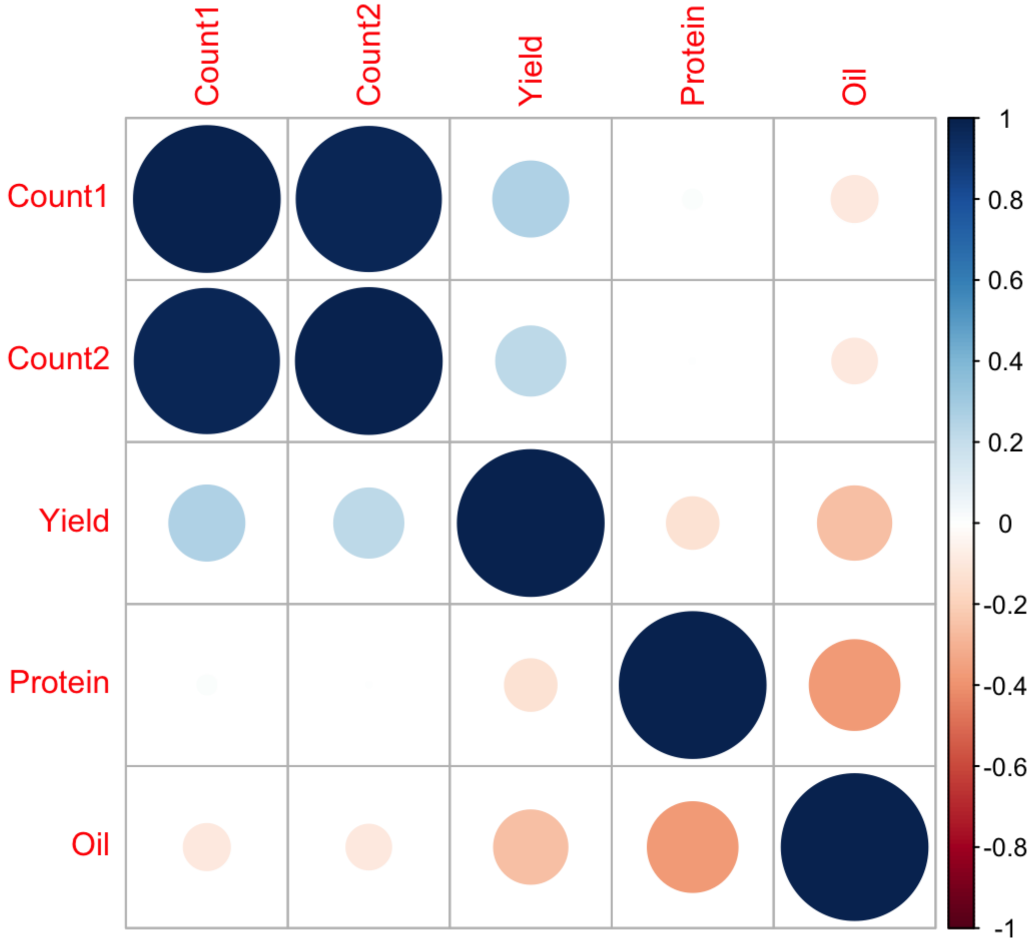
```
##         Count1 Count2 Yield Protein   Oil
## Count1    1.00   0.98  0.27    0.02 -0.10
## Count2    0.98   1.00  0.23    0.00 -0.09
## Yield     0.27   0.23  1.00   -0.13 -0.25
## Protein   0.02   0.00 -0.13    1.00 -0.38
## Oil      -0.10  -0.09 -0.25   -0.38  1.00
## 
## n= 54 
## 
## 
## P
##         Count1 Count2 Yield  Protein Oil   
## Count1         0.0000 0.0527 0.9077  0.4726
## Count2  0.0000        0.1017 0.9952  0.4999
## Yield   0.0527 0.1017        0.3677  0.0646
## Protein 0.9077 0.9952 0.3677         0.0047
## Oil     0.4726 0.4999 0.0646 0.0047
```

```
# We will now apply the function corrplot, which is in the package "corrplot" to look at the associations

example_cor2 <- correlations %>%
  select(-Treatment) %>%
  as.matrix() %>%
  cor(method = "pearson")

corrplot(example_cor2, method="number")
```



	Count1	Count2	Yield	Protein	Oil
Count1	1	0.98	0.27	0.02	-0.1
Count2	0.98	1	0.23	0	-0.09
Yield	0.27	0.23	1	-0.13	-0.25
Protein	0.02	0	-0.13	1	-0.38
Oil	-0.1	-0.09	-0.25	-0.38	1

```
corrplot(example_cor2, method="circle")
```



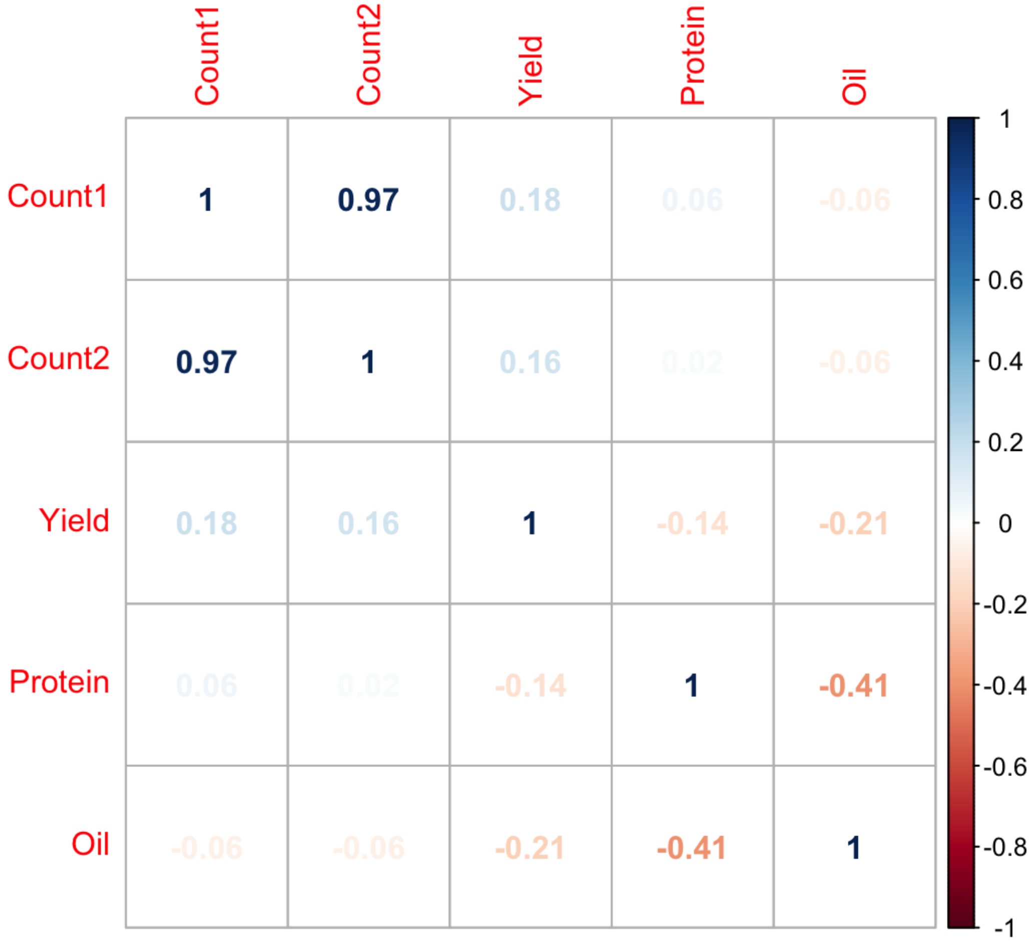
Spearman

This is a non-parametric rank-order correlation analysis.

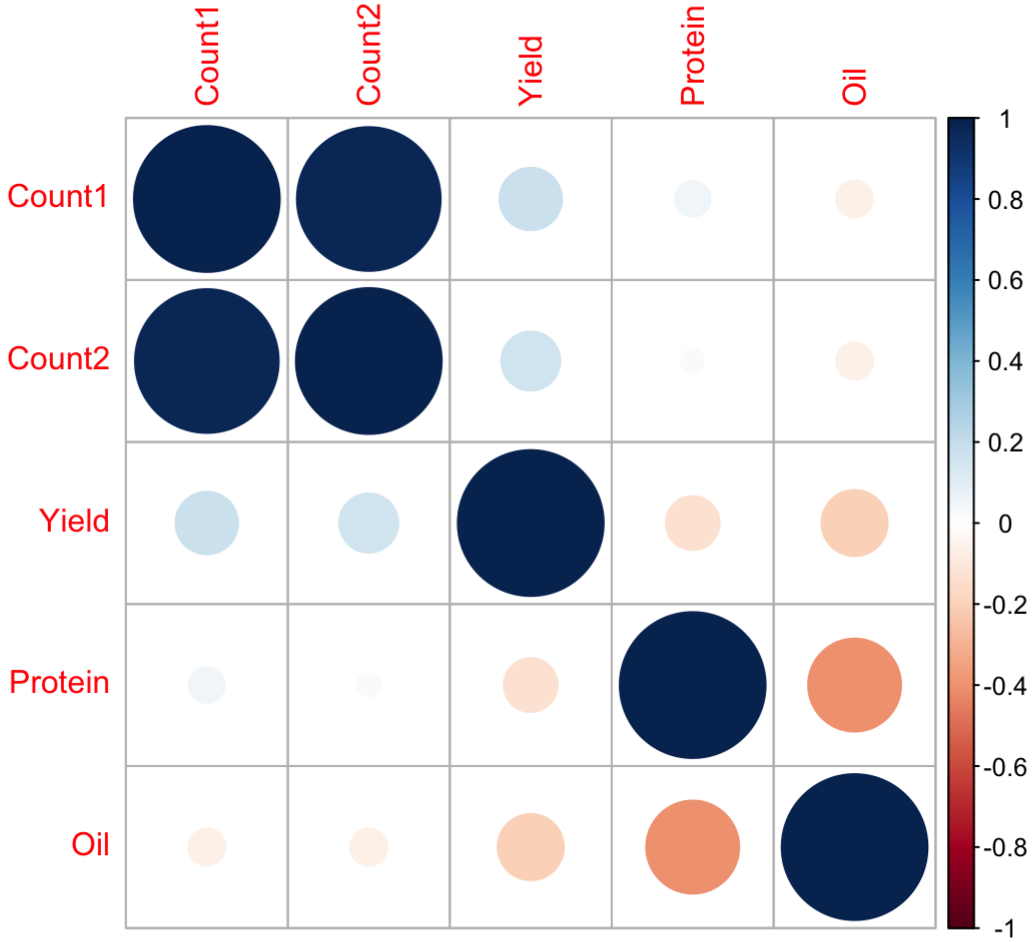
```
# Following again from our example.

example_corB <- correlations %>%
  select(-Treatment) %>%
  as.matrix() %>%
  rcorr(type = "spearman")

example_corB
```

```
##         Count1 Count2 Yield Protein   Oil
## Count1    1.00   0.97  0.18    0.06 -0.06
## Count2    0.97   1.00  0.16    0.02 -0.06
## Yield     0.18   0.16  1.00   -0.14 -0.21
## Protein   0.06   0.02 -0.14    1.00 -0.41
## Oil      -0.06  -0.06 -0.21   -0.41  1.00
##
## n= 54
##
##
## P
##         Count1 Count2 Yield  Protein Oil
## Count1         0.0000 0.1843 0.6741  0.6511
## Count2  0.0000        0.2367 0.8616  0.6477
## Yield   0.1843 0.2367        0.3269  0.1366
## Protein 0.6741 0.8616 0.3269         0.0023
## Oil     0.6511 0.6477 0.1366 0.0023
```

```
# Graphically, following from our initial example.

example_corB2 <- correlations %>%
  select(-Treatment) %>%
  as.matrix() %>%
  cor(method = "spearman")

corrplot(example_corB2, method="number")
```



```
corrplot(example_corB2, method="circle")
```



Summary

The goal of this introductory example was to provide some of the tools we can apply to calculate different correlation coefficients and graph the results. Remember that with these examples we assume a linear correlation so the intepretation of the results need to consider the biological associations as well (think about this for a correlation coefficient of 0 that has a curvilinear relationship).