Protein assay in RStudio

Anders Jensen

RStudio Tutorial

Protein assays

Data preperation

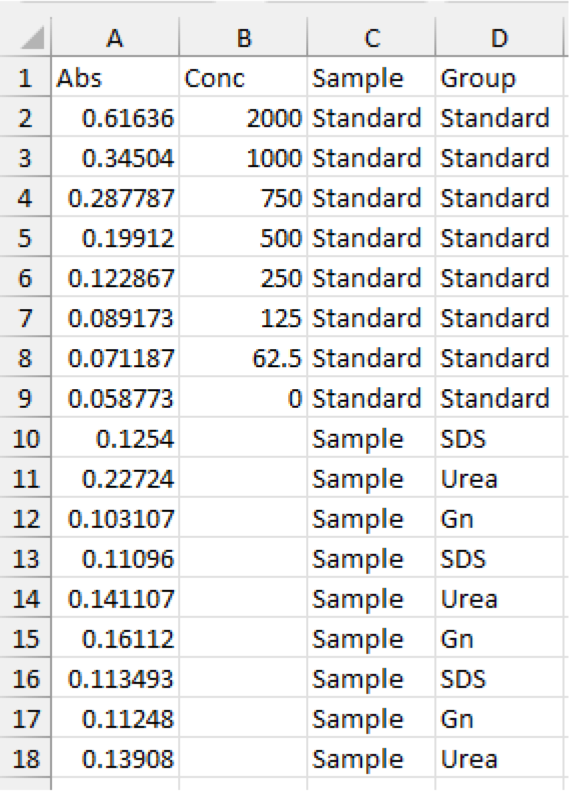
After completing your protein assay and taking measurements be sure to save that data as an excel file.

Then, using excel be sure to work out averages for each sample depending on how you ordered your triplicate samples. You can do this using RStudio, however it is probably easier using excel.

Then and most importantly you need to make a datafram with the following columns;

1) Absorbance - Average absorbance values
2) Concentration - At the stage only needed for standards
3) Sample - If value is a standard or a sample
4) Group - The experimental condition.

The following image shows an example setup:



	A	B	C	D
1	Abs	Conc	Sample	Group
2	0.61636	2000	Standard	Standard
3	0.34504	1000	Standard	Standard
4	0.287787	750	Standard	Standard
5	0.19912	500	Standard	Standard
6	0.122867	250	Standard	Standard
7	0.089173	125	Standard	Standard
8	0.071187	62.5	Standard	Standard
9	0.058773	0	Standard	Standard
10	0.1254		Sample	SDS
11	0.22724		Sample	Urea
12	0.103107		Sample	Gn
13	0.11096		Sample	SDS
14	0.141107		Sample	Urea
15	0.16112		Sample	Gn
16	0.113493		Sample	SDS
17	0.11248		Sample	Gn
18	0.13908		Sample	Urea

Figure 1: Example excel sheet

Loading data into RStudio

In order to load your data into RStudio you need to first set your working directory where your file is saved and then use the **readxl** package as shown below.

```r
# Reading in data

## Load packages
library(ggplot2)
library(readxl)
library(dplyr)

## Setwd
setwd("C:/Users/hlajens2/Desktop/Lab work/Protein assays")

# Read in data
df <- read_xlsx(path = "Tendon extraction buffer test data.xlsx", sheet = 3)
```

You may need to install the package prior to this but you can do that with **install.packages**

After this you need to use your standard to make a linear model for calculating unknown solutions.

```r
# Filter for standards
df_standard <- df %>%
  filter(Sample == "Standard")

# Make model
model <- lm(formula = Abs~Conc, data = df_standard)

# extract coefficients
gradient <- model$coefficients[2]
intercept <- model$coefficients[1]

# plot data
df_standard %>%
  ggplot(aes(x = Conc, y = Abs))+
  geom_point()+
  geom_smooth(method = 'lm')+
  annotate(geom = 'text',
           x = max(df_standard$Conc)/3.4, # adjust if placement is off
           y = max(df_standard$Abs)/1.3, # adjust if placement is off
           label = paste("Absorbance =",
                         round(gradient, digits = 5),
                         "* Concentration +",
                         round(intercept, digits = 4)))+
  annotate(geom = 'text',
           x = max(df_standard$Conc)/3.4, # adjust if placement is off
           y = max(df_standard$Abs)/1.4, # adjust if placement is off
           label = paste("Concentration = Absorbance - ",
                         round(intercept, digits = 4),
                         "/",
                         round(gradient, digits = 5)))

# Save graph (This may change depending on directory)
```

```
ggsave(filename = paste0("name","LOBF", ".png"),
       device = "png",
       dpi = 600,
       path = paste0(name, "/"))
```

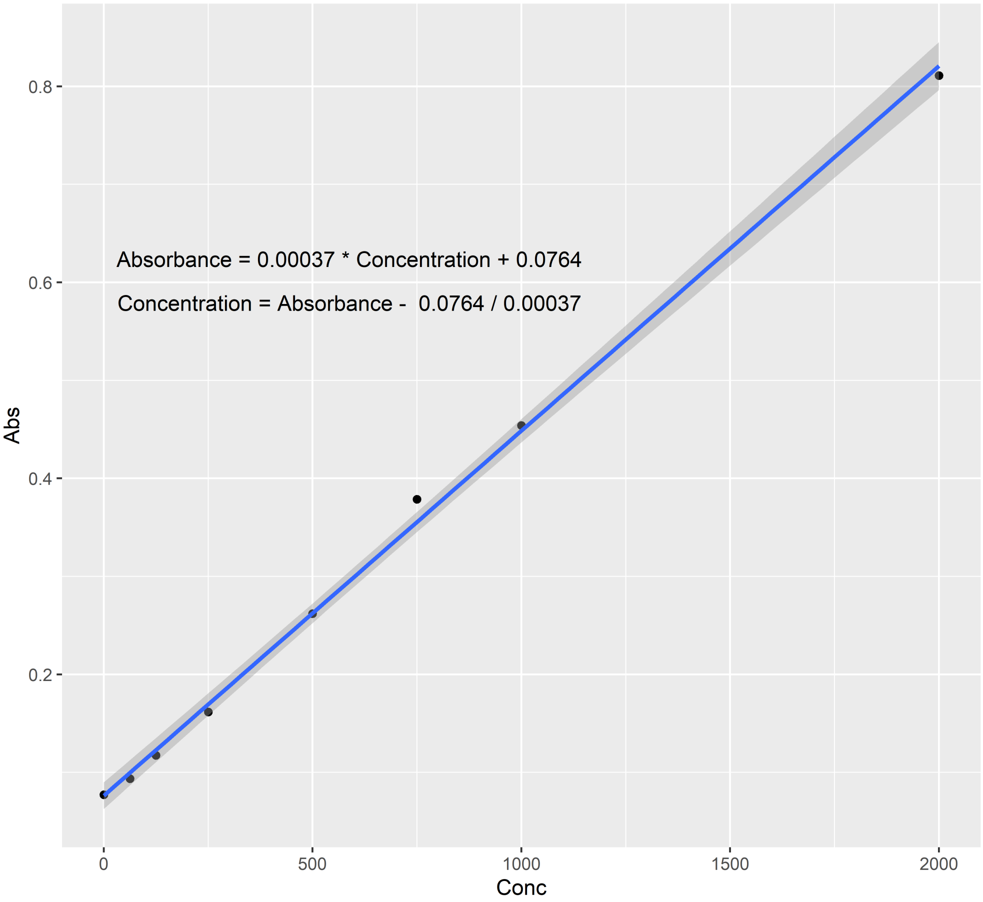


Figure 2: Linear model

Now that you have this information you can begin to work out the unknown values.

```r
# Use the model to calculate unknown concentrations
df_samples <- df %>%
  filter(Sample == "Sample") %>%
  mutate(Conc = (Abs - intercept)/gradient)

# Make box plot of data
df_samples %>%
  ggplot(aes(x = Group, y = Conc))+
  geom_boxplot(aes(fill = Group))+
  geom_jitter()+
  theme_classic()+
  labs(x = "Extraction buffer", y = "Concentration (ug/mL)")


# Save graph (This may change depending on directory)
ggsave(filename = paste0(name,"RES", ".png"),
       device = "png",
       dpi = 600,
       path = paste0(name, "/"))
```

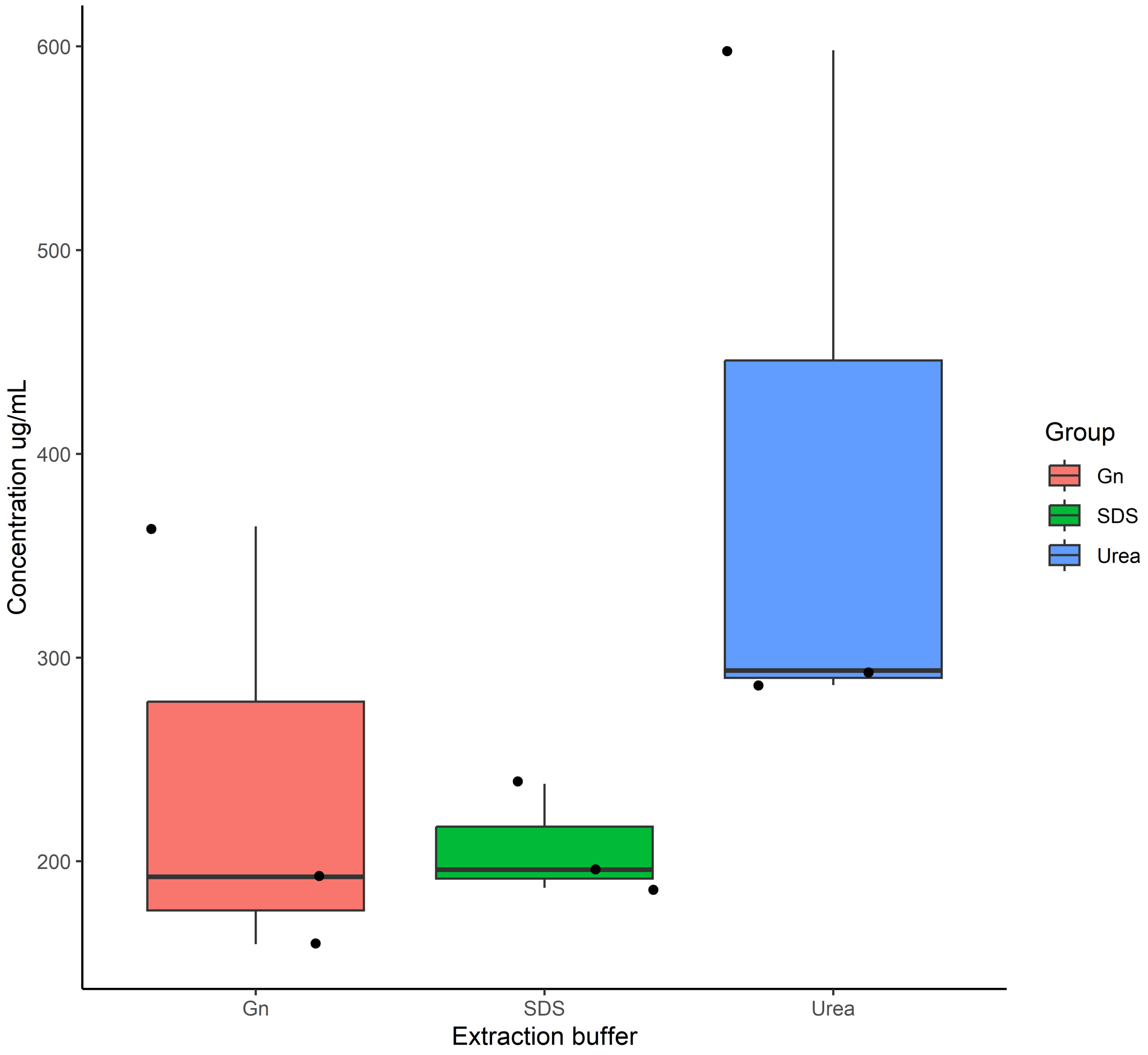


Figure 3: Box plot

The example used in this handbook is a lab experiment testing the effect of three different extraction buffers on protein extraction. Before attempting this you need to make sure your data is formatted to suit your work. If you are having any problems with this or the code is not working then visit the KnowHow page for more help.

I hope this was useful, there are lots more R tutorials on my website feel free to email me to request more RStudio tutorials.