Hydrolytic Extracellular Enzyme Activity Calculations

Hava Blair

June 25, 2020

Read in plate templates

You need one completed **template file** for each plate, in CSV format. There is a templtae in the project folder, file name XXXXXX

You also need one completed **metadata file** (data for all plates you are analyzing with this script can be included in the same file). The template is in the project folder, file name XXXXXXXX

Name your template files with the ID of your plate. This will be added as its own column by plater so that you can identify which plate the data is coming from.

Files used with this version of the script should contain all the metadata in the csv file itself, in 4 blocks:

–"template" = standard, blank, and sample IDs. Empty wells (no sample, standard, or check) may be coded as ".", "0" (zero), "NA", or left blank.

–"data" = raw fluorescence data from the plate reader

–"bad_wells" = identifies any wells that have known problems (ex. pipetting errors). May be coded as "bad", "Bad", "x", or "X". Anything with NA (blank) will be kept as "good"

– "conc_uM" = identifies the concentration of the substrate or MUB used in the well. The concentration should be in umol/L (uM). Examples: sub_300 = 300 uM enzyme substrate. mub_1.25 = 1.25 uM MUB standard.

The files you want to process must all be in a single folder, and that folder must not contain any other files R will attempt to read in all files in the designated folder.

Now you are ready to read in your files. If you get an error, check whether you have any extra CSV files in the designated folder.

```
file.names <- dir("./plater-templates/")
file.paths<- paste0("./plater-templates/", file.names)
plates <- read_plates(file.paths)
colnames(plates) <- c("plate", "wells", "id", "fluor", "bad_wells", "conc_uM")
head(plates)
```

```
## # A tibble: 6 x 6
##   plate   wells id            fluor bad_wells conc_uM
##   <chr>   <chr> <chr>         <int> <chr>     <chr>
## 1 mg_2006 A01   BG-sub-blank    501 <NA>      <NA>
## 2 mg_2006 A02   NAG-sub-blank   713 <NA>      <NA>
## 3 mg_2006 A03   BG-assay       1755 <NA>      <NA>
## 4 mg_2006 A04   P-assay       10428 <NA>      <NA>
## 5 mg_2006 A05   NAG-assay       893 <NA>      <NA>
## 6 mg_2006 A06   Cello-assay     540 <NA>      <NA>
```

```r
unique(plates$plate) # To check that all plates read in correctly
```

```
##  [1] "mg_2006" "mg_2008" "mg_2014" "mg_2016" "mg_2022" "mg_2024" "mg_2050"
##  [8] "mg_2051" "mg_2082" "mg_2083" "mg_2090" "mg_2092" "plate_b"
```

```r
#read in plate metadata
plate_metadata <- read.csv("./eea_metadata_mgtest.csv", stringsAsFactors = FALSE)

colnames(plate_metadata) <- c("plate", "ph_buffer", "moist_soil_mass_g",  "vol_buffer_ml", "substrates"

#parse the time columns so we can do math on them
plate_metadata <- plate_metadata %>%
  mutate(time_soil_added = parse_hm(time_soil_added),
         time_naoh_added = parse_hm(time_soil_added),
         time_plate_read = parse_hm(time_plate_read))

# calculate incubation time and convert from seconds to hours
plate_metadata <- plate_metadata %>%
  mutate(inc_time_hr = as.numeric(((time_plate_read - time_soil_added)/60/60)))

plate_metadata
```

```
##      plate ph_buffer moist_soil_mass_g vol_buffer_ml substrates  tin tin_moist
## 1  mg_2006       6.5            0.5190            50    C-B-N-P 2.67     10.00
## 2  mg_2008       6.5            0.4995            50    C-B-N-P 2.68      9.97
## 3  mg_2014       6.5            0.4954            50    C-B-N-P 2.63     12.75
## 4  mg_2016       6.5            0.4977            50    C-B-N-P 2.65     13.20
## 5  mg_2022       6.5            0.5104            50    C-B-N-P 2.62     12.69
## 6  mg_2024       6.5            0.5136            50    C-B-N-P 2.66     12.63
## 7  mg_2050       6.5            0.5287            50    C-B-N-P 2.68     13.00
## 8  mg_2051       6.5            0.4967            50    C-B-N-P 2.67     12.53
## 9  mg_2082       6.5            0.4798            50    C-B-N-P 2.65     12.52
## 10 mg_2083       6.5            0.5242            50    C-B-N-P 2.64     12.63
## 11 mg_2090       6.5            0.4994            50    C-B-N-P 2.68     12.67
## 12 mg_2092       6.5            0.4812            50    C-B-N-P 2.64     12.57
## 13 plate_b       6.5                NA            NA              NA        NA
##    tin_dry time_soil_added time_naoh_added time_plate_read inc_time_hr
## 1     8.77        12:00:00        12:00:00        13:00:00           1
## 2     8.96        13:00:00        13:00:00        14:00:00           1
## 3    11.25        14:00:00        14:00:00        15:00:00           1
## 4    11.76        15:00:00        15:00:00        16:00:00           1
## 5    10.52        16:00:00        16:00:00        17:00:00           1
## 6    10.43        17:00:00        17:00:00        18:00:00           1
## 7    11.79        18:00:00        18:00:00        19:00:00           1
## 8    10.91        19:00:00        19:00:00        20:00:00           1
## 9    11.22        20:00:00        20:00:00        21:00:00           1
## 10   11.32        21:00:00        21:00:00        22:00:00           1
## 11   11.65        22:00:00        22:00:00        23:00:00           1
## 12   11.66        23:00:00        23:00:00        00:00:00         -23
## 13      NA              NA              NA              NA          NA
```

```r
#calculate soil moisture content and dry soil equivalent

plate_metadata <- plate_metadata %>%
  mutate(mc_soil_moist = tin_moist - tin,
         mc_soil_dry = tin_dry - tin,
         soil_water_content = (mc_soil_moist-mc_soil_dry)/mc_soil_dry,
         soil_ov_dry_eq_g = moist_soil_mass_g - (moist_soil_mass_g * soil_water_content))

head(plate_metadata)
```

```
##     plate ph_buffer moist_soil_mass_g vol_buffer_ml substrates  tin tin_moist
## 1 mg_2006       6.5            0.5190            50    C-B-N-P 2.67     10.00
## 2 mg_2008       6.5            0.4995            50    C-B-N-P 2.68      9.97
## 3 mg_2014       6.5            0.4954            50    C-B-N-P 2.63     12.75
## 4 mg_2016       6.5            0.4977            50    C-B-N-P 2.65     13.20
## 5 mg_2022       6.5            0.5104            50    C-B-N-P 2.62     12.69
## 6 mg_2024       6.5            0.5136            50    C-B-N-P 2.66     12.63
##   tin_dry time_soil_added time_naoh_added time_plate_read inc_time_hr
## 1    8.77        12:00:00        12:00:00        13:00:00           1
## 2    8.96        13:00:00        13:00:00        14:00:00           1
## 3   11.25        14:00:00        14:00:00        15:00:00           1
## 4   11.76        15:00:00        15:00:00        16:00:00           1
## 5   10.52        16:00:00        16:00:00        17:00:00           1
## 6   10.43        17:00:00        17:00:00        18:00:00           1
##   mc_soil_moist mc_soil_dry soil_water_content soil_ov_dry_eq_g
## 1          7.33        6.10          0.2016393        0.4143492
## 2          7.29        6.28          0.1608280        0.4191664
## 3         10.12        8.62          0.1740139        0.4091935
## 4         10.55        9.11          0.1580681        0.4190295
## 5         10.07        7.90          0.2746835        0.3702015
## 6          9.97        7.77          0.2831403        0.3681792
```

Remove empty wells and known bad wells

```r
#Remove empty wells
no_missing <- subset(plates, !is.na(id) & id != 0 & id != ".")

# Remove bad wells and keep good ones
no_bad <- subset(no_missing, is.na(bad_wells))

#Create a dataframe with the details about which wells were removed as "bad".  Can write this to CSV no
bad_list <- subset(no_missing, !is.na(bad_wells))

#original PMN script from Miriam has the code below to remove bad wells. Based on the way I have been f
# plates <- subset(plates, Bad_wells != "bad" & Bad_wells != "Bad"
#                  & Bad_wells != "x" & Bad_wells != "X" | is.na(Bad_wells))

wells_removed <- nrow(plates) - nrow(no_bad)

glue("Removed {wells_removed} wells that were missing or bad")
```

```
## Removed 193 wells that were missing or bad
```

```r
# filter out plate b here - it will be used for the emission coefficient calculation

plate_b <- no_bad %>%
  filter(plate == "plate_b")

#clean_data contains all of the plate a data
clean_data <- no_bad %>%
  filter(plate != "plate_b")

clean_nested <- clean_data %>%
  group_by(plate) %>%
  nest()
```

B Plate: MUB standard calculations (for emission coefficient calculation)

```r
#exract MUB concentrations from the conc_uM column and drop the empty "bad_wells" column
plate_b <- plate_b %>%
  mutate(conc_uM = as.numeric(str_replace(conc_uM, "mub_", ""))) %>%
  select(-bad_wells)

#calculate the mean fluorescence across technical replicates for each MUB concentration
plate_b_means <- plate_b %>%
  group_by(id) %>%
  filter(str_detect(id, "MUB")) %>%
  summarise(mub_std_fluor = mean(fluor), n = n())

#calculate the plate blank (buffer only)
plate_b_blank <- plate_b %>%
  filter(str_detect(id, "Buf")) %>%
  summarise(fluor_blank = mean(fluor), n = n())

fluor_blank <- plate_b_blank$fluor_blank

#subtract the fluorescence of the plate blank (id = Buf)
plate_b_means <- plate_b_means %>%
  mutate(plate_blank = fluor_blank,
         corr_fluor = mub_std_fluor - plate_blank,
         mub_conc_uM = as.numeric(str_replace(id, "MUB", "")))

plate_b_means
```

```
## # A tibble: 4 x 6
##   id       mub_std_fluor     n plate_blank corr_fluor mub_conc_uM
##   <chr>            <dbl> <int>       <dbl>      <dbl>       <dbl>
## 1 MUB0.16          1486.     4        66.1      1420.       0.16
## 2 MUB0.625         5891.     4        66.1      5825.       0.625
## 3 MUB1.25         11928.     4        66.1     11862.       1.25
## 4 MUB2.5          22628      4        66.1     22562.       2.5
```

A Plates: Calculate plate blank and homogenate blank

```
#function to calculate plate blanks (Buffer only)
get_plate_blank <- function(data){
  buf <- data %>%
    filter(id == "Buf")
    mean(buf$fluor)
}

#apply function to nested df
with_a_blanks <- clean_nested %>%
  mutate(plate_blank = map_dbl(data, get_plate_blank))

#function to calculate the fluor for HOMogenate BLanks (hombl = buffer + homogenate)
get_hombl <- function(data){
  buf_soil <- data %>%
    filter(str_detect(id, "Hombl"))
    mean(buf_soil$fluor)
}

#apply function to nested df
calc_hombl <- with_a_blanks %>%
  mutate(hombl = map_dbl(data, get_hombl))

head(calc_hombl)
```

```
## # A tibble: 6 x 4
## # Groups:   plate [6]
##   plate   data              plate_blank hombl
##   <chr>   <list>                  <dbl> <dbl>
## 1 mg_2006 <tibble [80 x 5]>        61.2  113.
## 2 mg_2008 <tibble [80 x 5]>        60.8  116.
## 3 mg_2014 <tibble [80 x 5]>        68.6  131.
## 4 mg_2016 <tibble [80 x 5]>        59.2  125
## 5 mg_2022 <tibble [80 x 5]>        64.4  121.
## 6 mg_2024 <tibble [80 x 5]>        58.6  128.
```

Plate A: Calculate mean fluor of the MUB standard + soil wells (for quench coeff calculation)

```
#function to average the MUB+soil (quench) standard wells to get mean fluorescence for each MUB concent

quench_std_fun <- function(data, hombl){
  data %>%
    select(wells, id, fluor, conc_uM) %>%
    filter(str_detect(id, "quench")) %>%
    mutate(conc_uM = as.numeric(str_replace(conc_uM, "mub_", ""))) %>%
    group_by(conc_uM) %>%
    summarise(quench_fluor = mean(fluor), n = n(), hombl = hombl, corr_fluor = quench_fluor - hombl)  #
```

```
}

#apply function to nested df
quench_nested <- calc_hombl %>%
  mutate(quench_std_values = map2(data, hombl, quench_std_fun))

head(quench_nested)
```

```
## # A tibble: 6 x 5
## # Groups:   plate [6]
##   plate   data              plate_blank hombl quench_std_values
##   <chr>   <list>                  <dbl> <dbl> <list>
## 1 mg_2006 <tibble [80 x 5]>        61.2  113. <tibble [4 x 5]>
## 2 mg_2008 <tibble [80 x 5]>        60.8  116. <tibble [4 x 5]>
## 3 mg_2014 <tibble [80 x 5]>        68.6  131. <tibble [4 x 5]>
## 4 mg_2016 <tibble [80 x 5]>        59.2  125  <tibble [4 x 5]>
## 5 mg_2022 <tibble [80 x 5]>        64.4  121. <tibble [4 x 5]>
## 6 mg_2024 <tibble [80 x 5]>        58.6  128. <tibble [4 x 5]>
```

Linear model functions

```
#function to run a linear model x = MUB concentration, y = mean fluorescence

lm_mod_ftn <- function(df){
  lm(corr_fluor ~ conc_uM, data = df)
}

# functions to extract linear model details calculated above into a nicer format for putting in our gra

b_fun <- function(mod){
  coefficients(mod)[[1]]
}

slope_fun <- function(mod){
  coefficients(mod)[[2]]
}

r_sq_fun <- function(mod){
  summary(mod)[["r.squared"]]
}

#leaving this max fluorescence function out for now because I'm not sure we need it.
# max_fluor_fun <- function(data){
#   max(data$corr_std_mean)
# }
```

A Plates: Linear model calculations for quench std curves

```
#calculate linear model for quench (A plates)

quench_lm_calcs <- quench_nested %>%
  mutate(quench_lm = map(quench_std_values, lm_mod_ftn))

#Extract linear models details from the homogenate control linear model
quench_lm_details <- quench_lm_calcs %>%
  mutate(intcpt_quench = map_dbl(quench_lm, b_fun), slope_quench = map_dbl(quench_lm, slope_fun), r_squa

#nest the linear model details in a dataframe
nest_quench_stats <- quench_lm_details %>%
  nest(lm_stats_quench = c(intcpt_quench, slope_quench, r_squared_quench))

head(nest_quench_stats)
```

```
## # A tibble: 6 x 7
## # Groups:   plate [6]
##   plate  data       plate_blank hombl quench_std_valu~ quench_lm lm_stats_quench
##   <chr>  <list>           <dbl> <dbl> <list>           <list>    <list>
## 1 mg_20~ <tibble [~        61.2  113. <tibble [4 x 5]> <lm>      <tibble [1 x 3~
## 2 mg_20~ <tibble [~        60.8  116. <tibble [4 x 5]> <lm>      <tibble [1 x 3~
## 3 mg_20~ <tibble [~        68.6  131. <tibble [4 x 5]> <lm>      <tibble [1 x 3~
## 4 mg_20~ <tibble [~        59.2  125  <tibble [4 x 5]> <lm>      <tibble [1 x 3~
## 5 mg_20~ <tibble [~        64.4  121. <tibble [4 x 5]> <lm>      <tibble [1 x 3~
## 6 mg_20~ <tibble [~        58.6  128. <tibble [4 x 5]> <lm>      <tibble [1 x 3~
```

B Plate linear model calculations

```
#fix name of mub concentration column so it works with the lm function
plate_b_rename <- plate_b_means %>%
  rename(conc_uM = mub_conc_uM)

#apply linear model ftn to plate b data
lm_plate_b <- lm_mod_ftn(plate_b_rename)

#save linear model details
intcpt_emis <- b_fun(lm_plate_b)
slope_emis <- slope_fun(lm_plate_b)
rsq_emis <- r_sq_fun(lm_plate_b)

#add linear model details to plate b dataframe
lm_stats_emis <- plate_b_rename %>%
  mutate(intcpt_emis = intcpt_emis, slope_emis = slope_emis, rsq_emis = rsq_emis)

head(lm_stats_emis)
```

```
## # A tibble: 4 x 9
##   id    mub_std_fluor     n plate_blank corr_fluor conc_uM intcpt_emis
```

```
##   <chr>          <dbl> <int>       <dbl>      <dbl>   <dbl>       <dbl>
## 1 MUB0~          1486.     4        66.1      1420.   0.16         186.
## 2 MUB0~          5891.     4        66.1      5825.   0.625        186.
## 3 MUB1~         11928.     4        66.1     11862.   1.25         186.
## 4 MUB2~         22628      4        66.1     22562.   2.5          186.
## # ... with 2 more variables: slope_emis <dbl>, rsq_emis <dbl>
```

Plate B: Plot standard curve (emission)

```
## consider writing this into a function so that multiple B plates (from multiple runs) could be proces

  b_plot <- ggplot(lm_stats_emis) +
    geom_point(aes(x = conc_uM, y = corr_fluor)) +
    geom_abline(aes(slope = slope_emis, intercept = intcpt_emis)) +
    geom_text(data = lm_stats_emis, aes(x = 1.0, y = 20000, label = paste("R^2=", round(rsq_emis, digit
    geom_text(aes(x = 1.0, y = 17000, label = paste(" y= ", round(slope_emis, digits = 2), "x", " + ",
    labs(title = glue("Plate B (emission) standard curve: MUB conc vs. fluorescence"))

b_plot
```

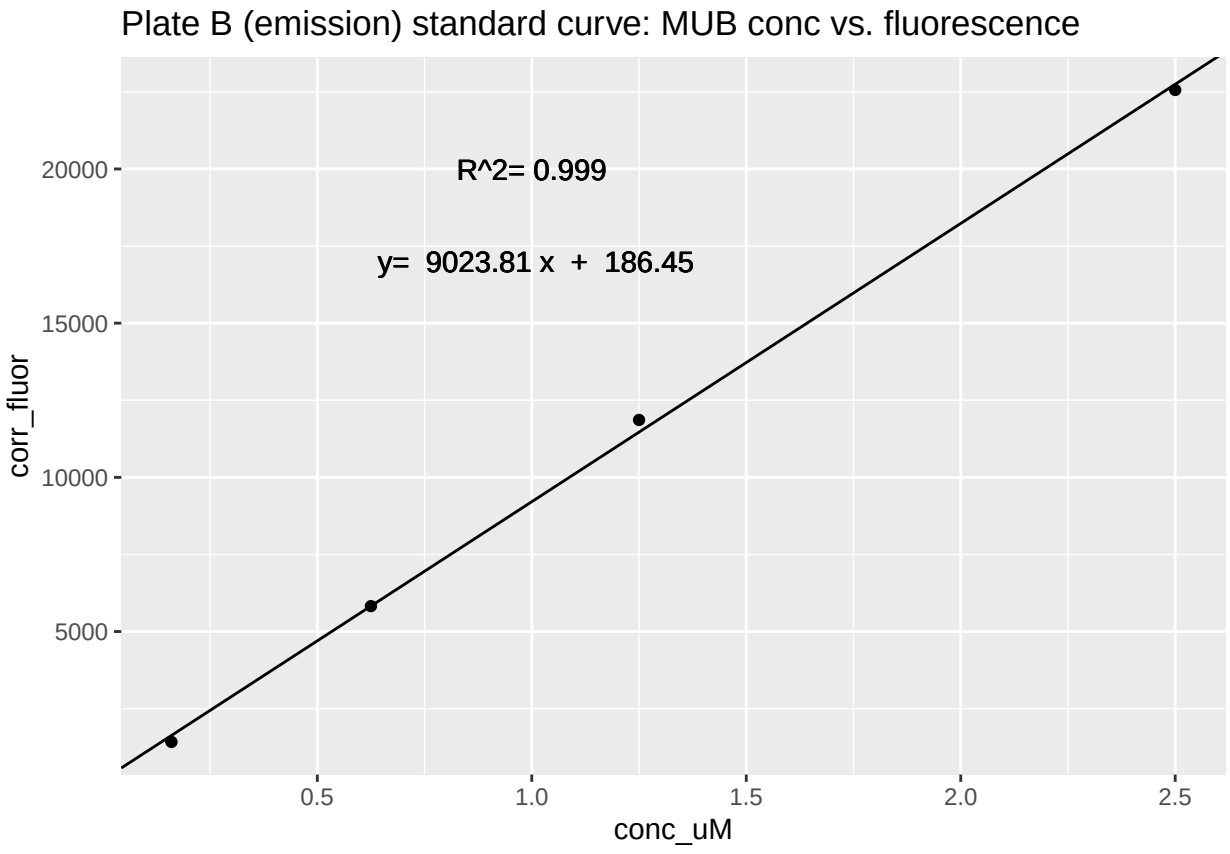
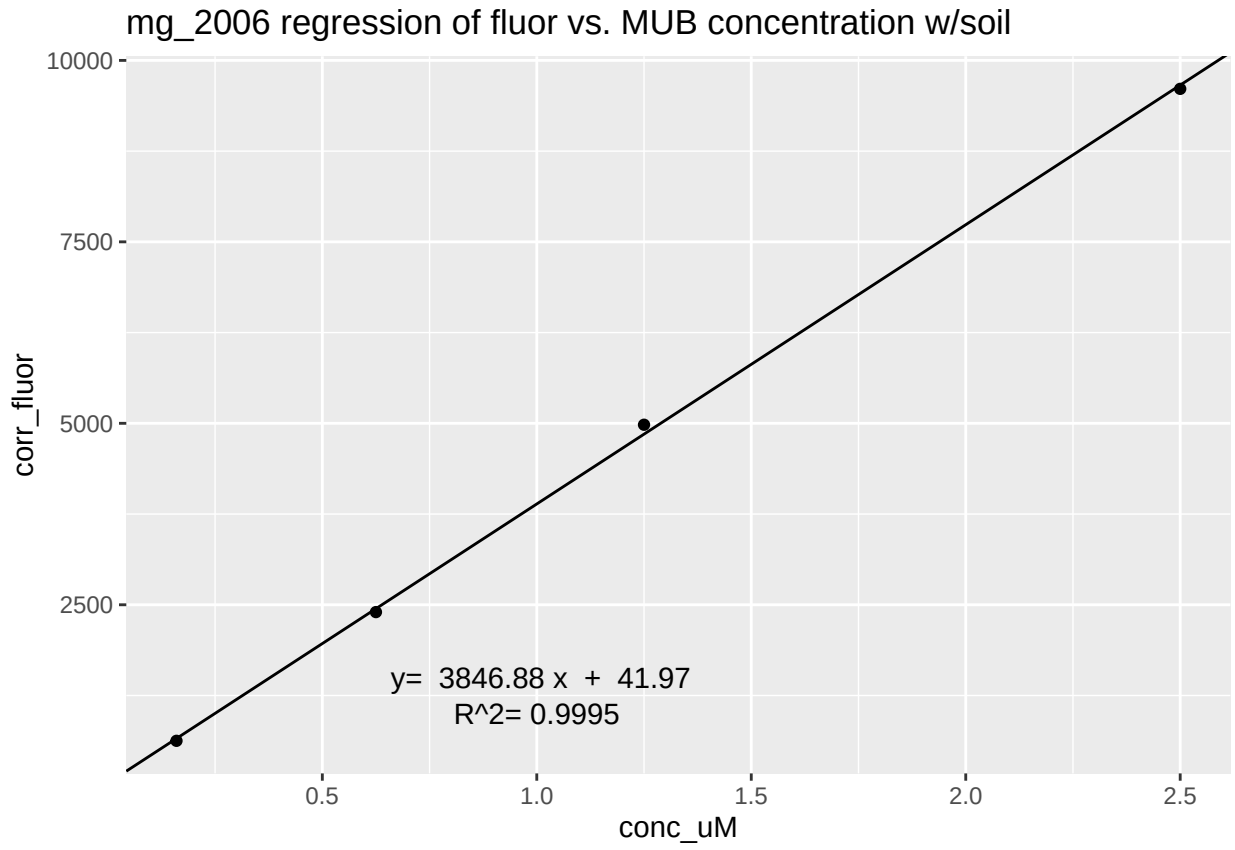


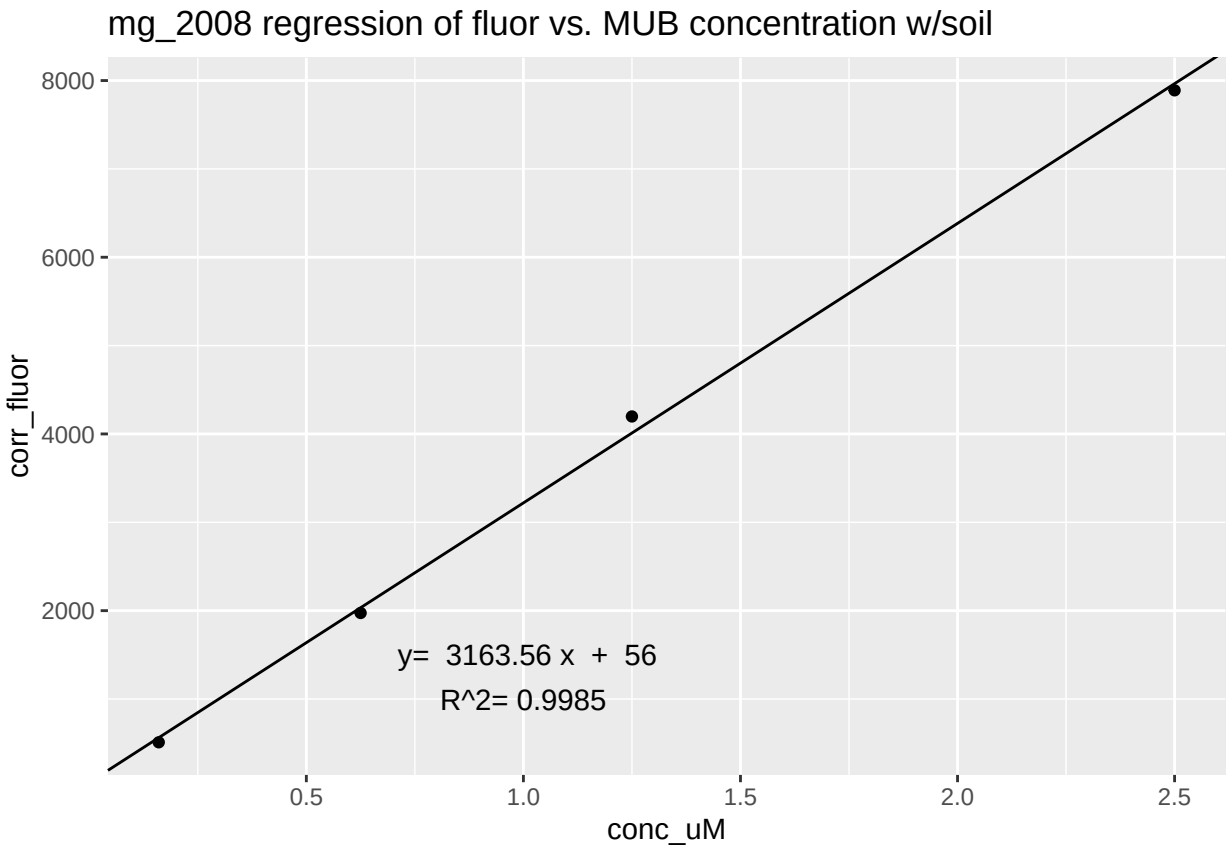
Plate A: plot quench standard curves

```
plot_homog_fun <- function(quench_std_values, lm_stats_quench, plate){
  g <- ggplot() +
  geom_point(data = quench_std_values, aes(x = conc_uM, y = corr_fluor)) +
    geom_abline(data = lm_stats_quench, aes(slope = slope_quench, intercept = intcpt_quench)) +
    geom_text(data = lm_stats_quench, aes(x = 1.0, y = 1000, label = paste("R^2=", round(r_squared_quen
    geom_text(data = lm_stats_quench, aes(x = 1.0, y = 1500, label = paste(" y= ", round(slope_quench, 
    labs(title = glue("{plate} regression of fluor vs. MUB concentration w/soil"))
  return(g)
}

homog_plots <- nest_quench_stats %>%
  mutate(plot_homog_control = pmap(list(quench_std_values, lm_stats_quench, plate), plot_homog_fun))

homog_plots$plot_homog_control
```

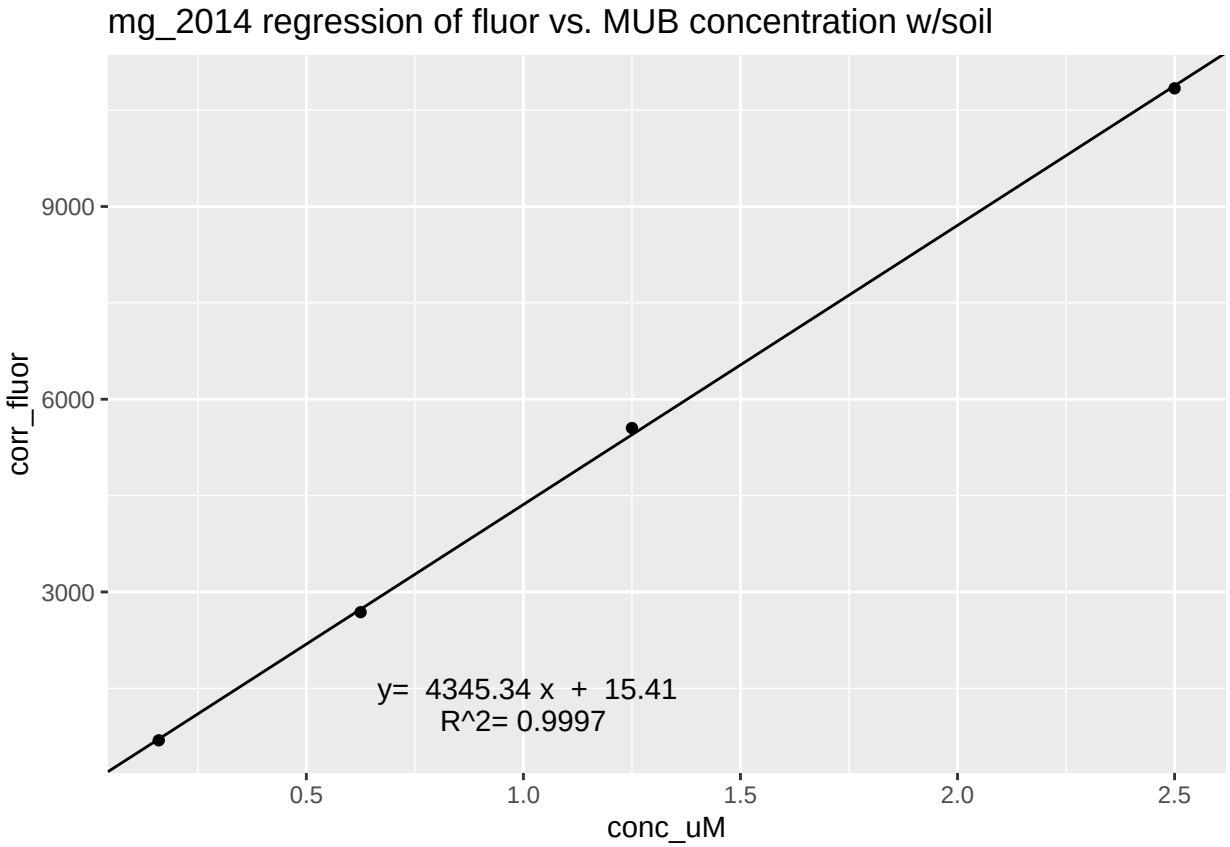
```
## [[1]]
```



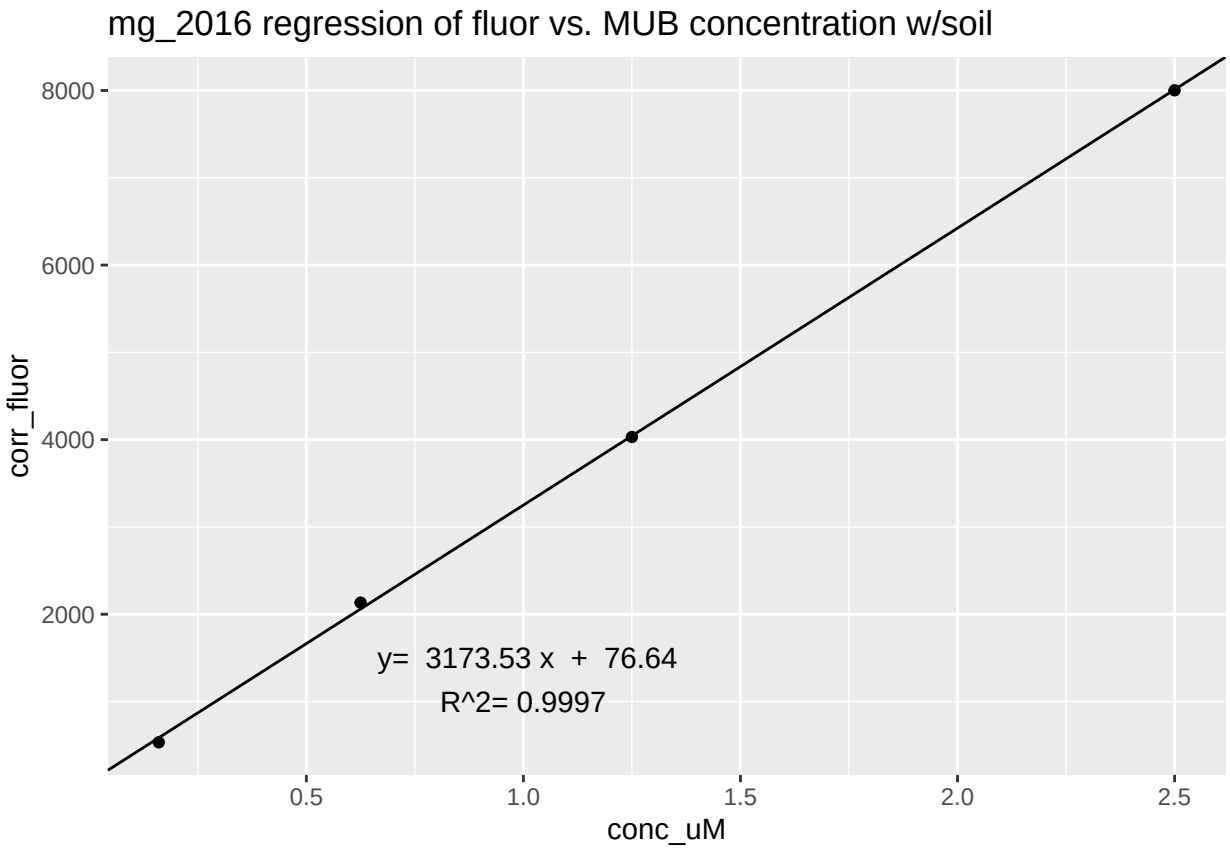

```
##
## [[2]]
```


mg_2008 regression of fluor vs. MUB concentration w/soil
corr_fluor
8000
6000
4000
2000
0.5
1.0
1.5
2.0
2.5
conc_uM
y= 3163.56 x + 56
R^2= 0.9985

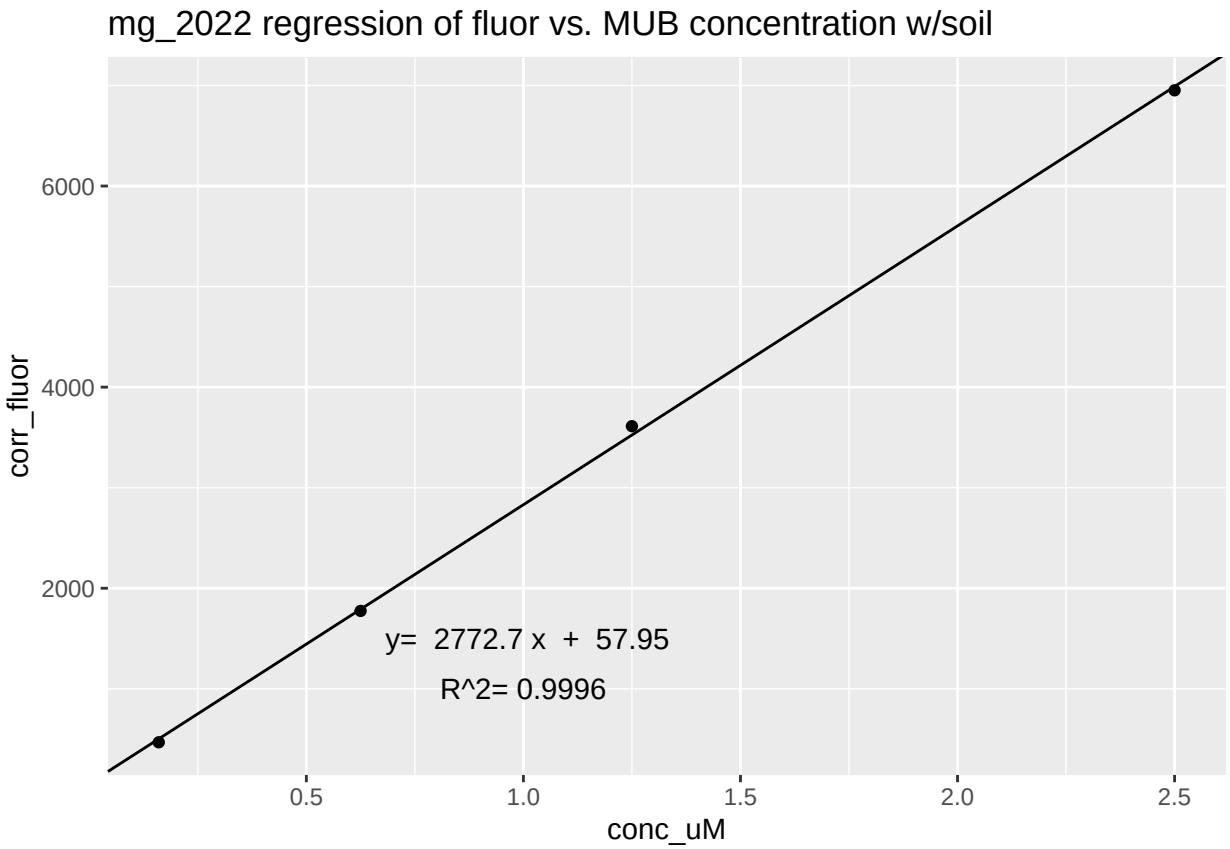
```
##
## [[3]]
```


mg_2014 regression of fluor vs. MUB concentration w/soil
corr_fluor
9000
6000
3000
0.5
1.0
1.5
2.0
2.5
conc_uM
y= 4345.34 x + 15.41
R^2= 0.9997

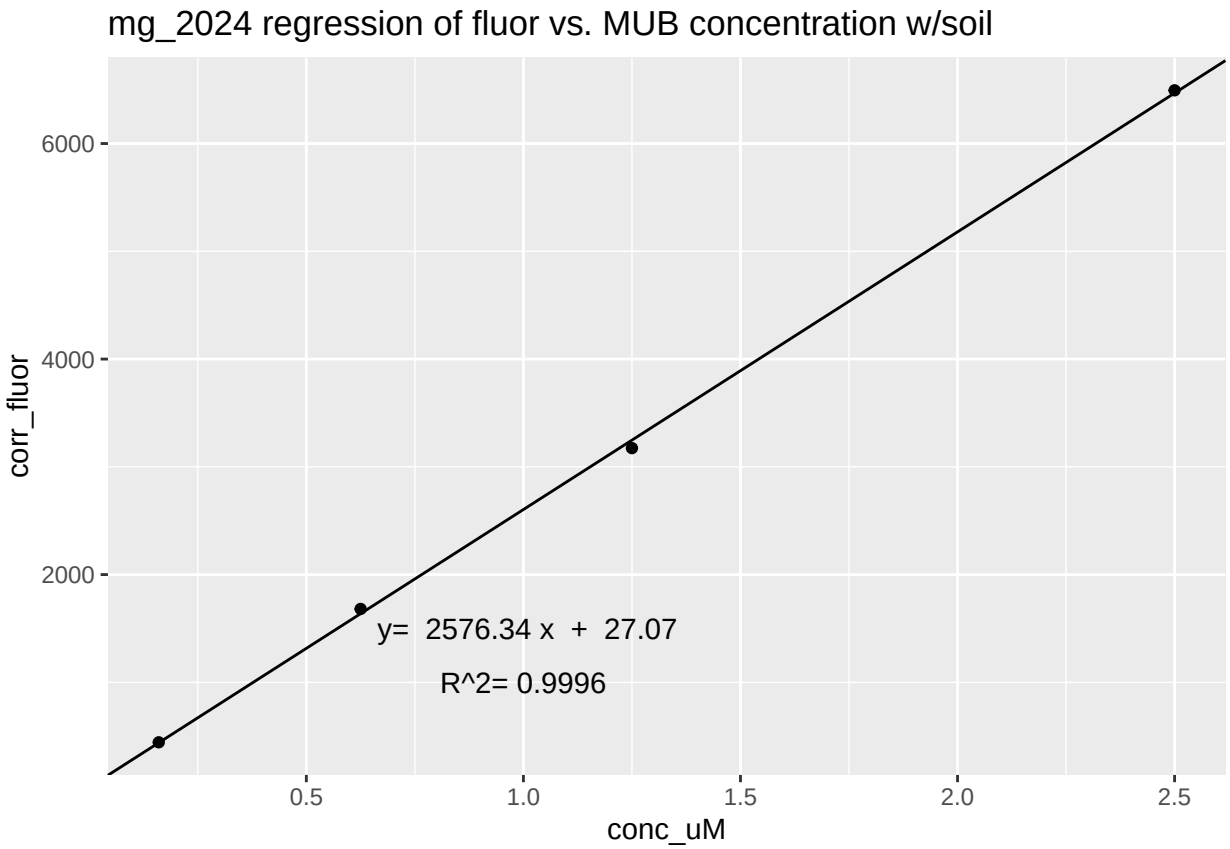
```
##
## [[4]]
```


mg_2016 regression of fluor vs. MUB concentration w/soil
corr_fluor
8000
6000
4000
2000
0.5
1.0
1.5
2.0
2.5
conc_uM
y= 3173.53 x + 76.64
R^2= 0.9997

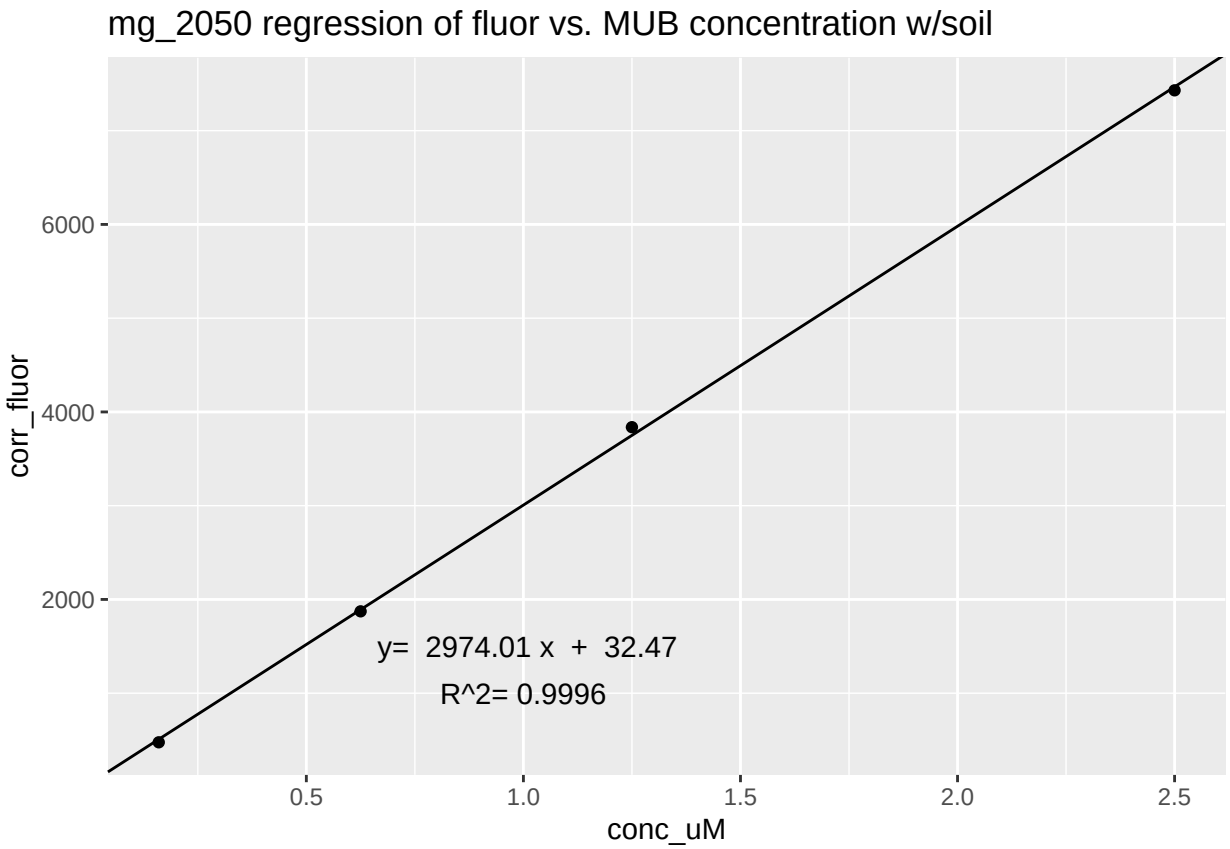
```
##
## [[5]]
```


mg_2022 regression of fluor vs. MUB concentration w/soil
corr_fluor
6000
4000
2000
0.5
1.0
1.5
2.0
2.5
conc_uM
y= 2772.7 x + 57.95
R^2= 0.9996

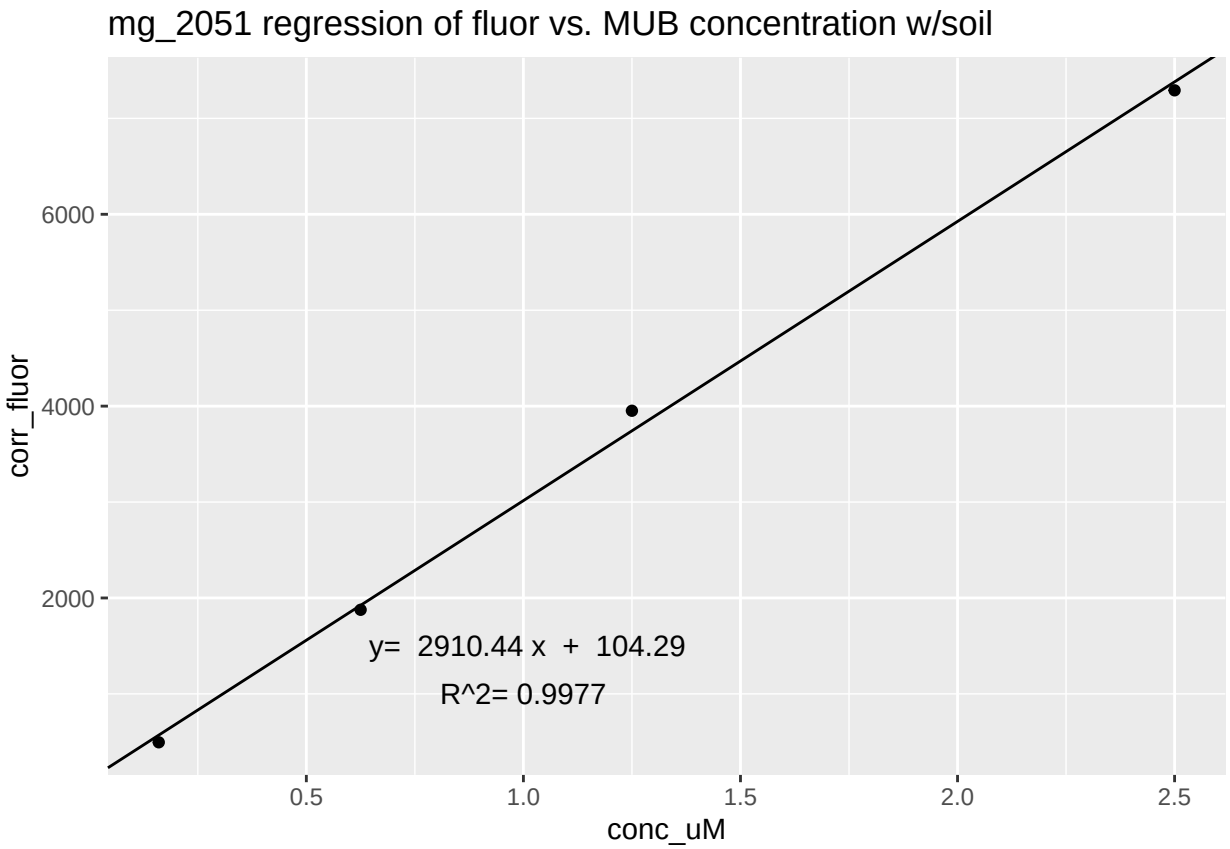
```
##
## [[6]]
```


mg_2024 regression of fluor vs. MUB concentration w/soil
corr_fluor
6000
4000
2000
0.5
1.0
1.5
2.0
2.5
conc_uM
y= 2576.34 x + 27.07
R^2= 0.9996

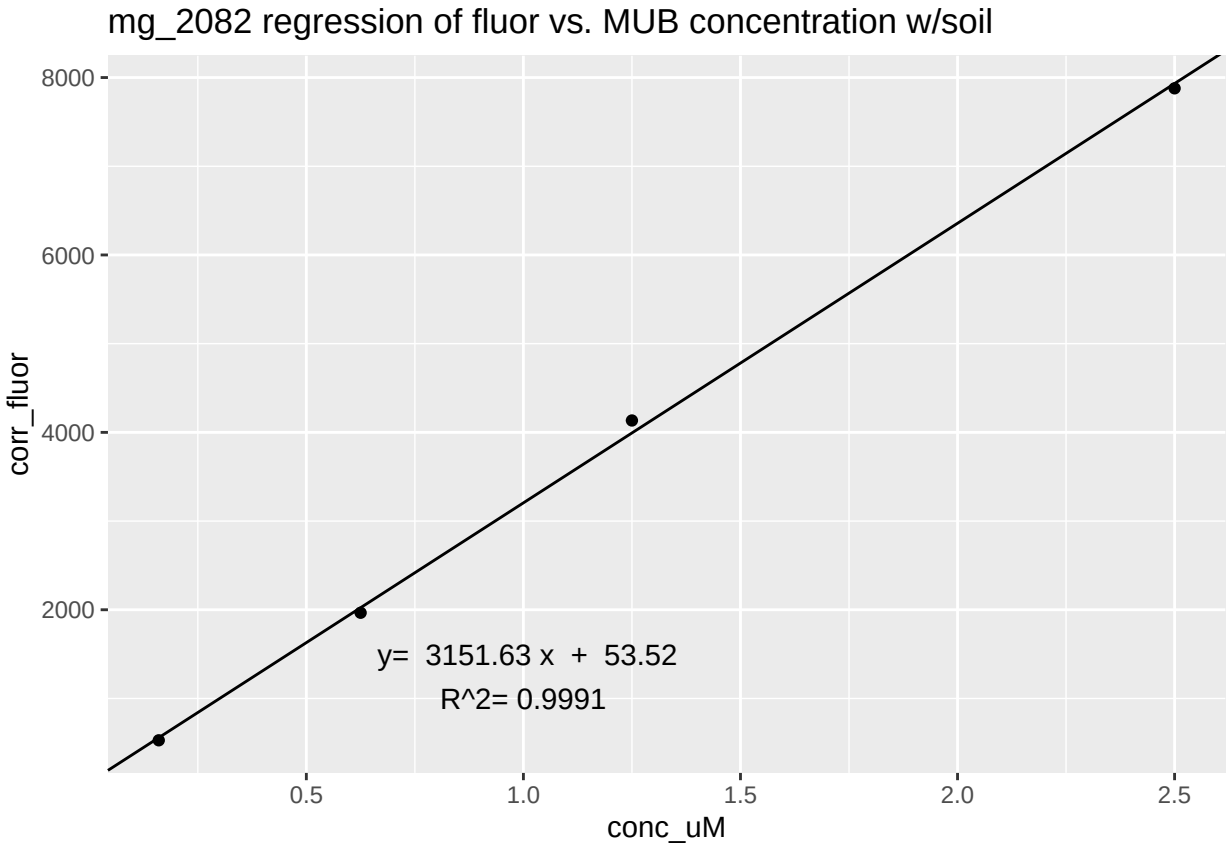
```
##
## [[7]]
```


mg_2050 regression of fluor vs. MUB concentration w/soil
corr_fluor
6000
4000
2000
y= 2974.01 x + 32.47
R^2= 0.9996
0.5
1.0
1.5
2.0
2.5
conc_uM

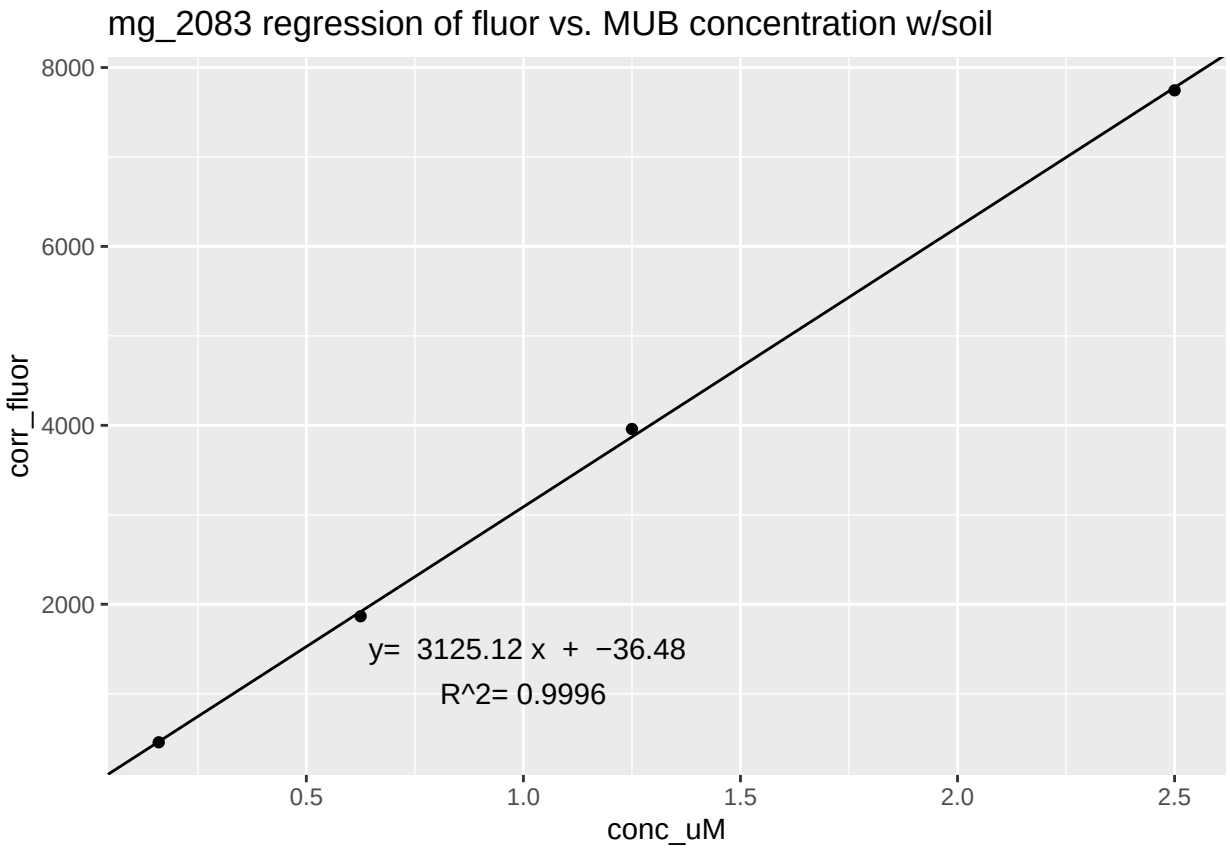
```
##
## [[8]]
```


mg_2051 regression of fluor vs. MUB concentration w/soil
corr_fluor
6000
4000
2000
y= 2910.44 x + 104.29
R^2= 0.9977
0.5
1.0
1.5
2.0
2.5
conc_uM

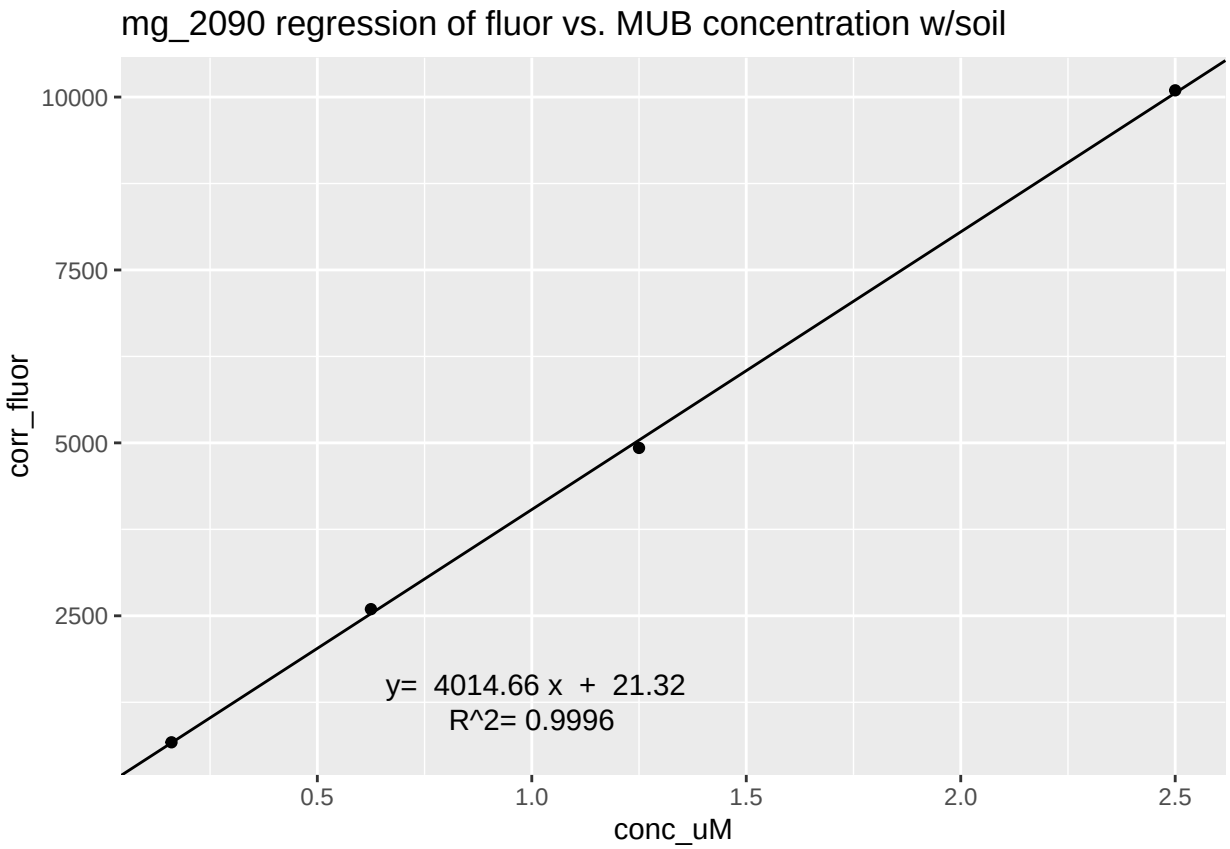
```
##
## [[9]]
```


mg_2082 regression of fluor vs. MUB concentration w/soil
corr_fluor
8000
6000
4000
2000
0.5
1.0
1.5
2.0
2.5
conc_uM
y= 3151.63 x + 53.52
R^2= 0.9991

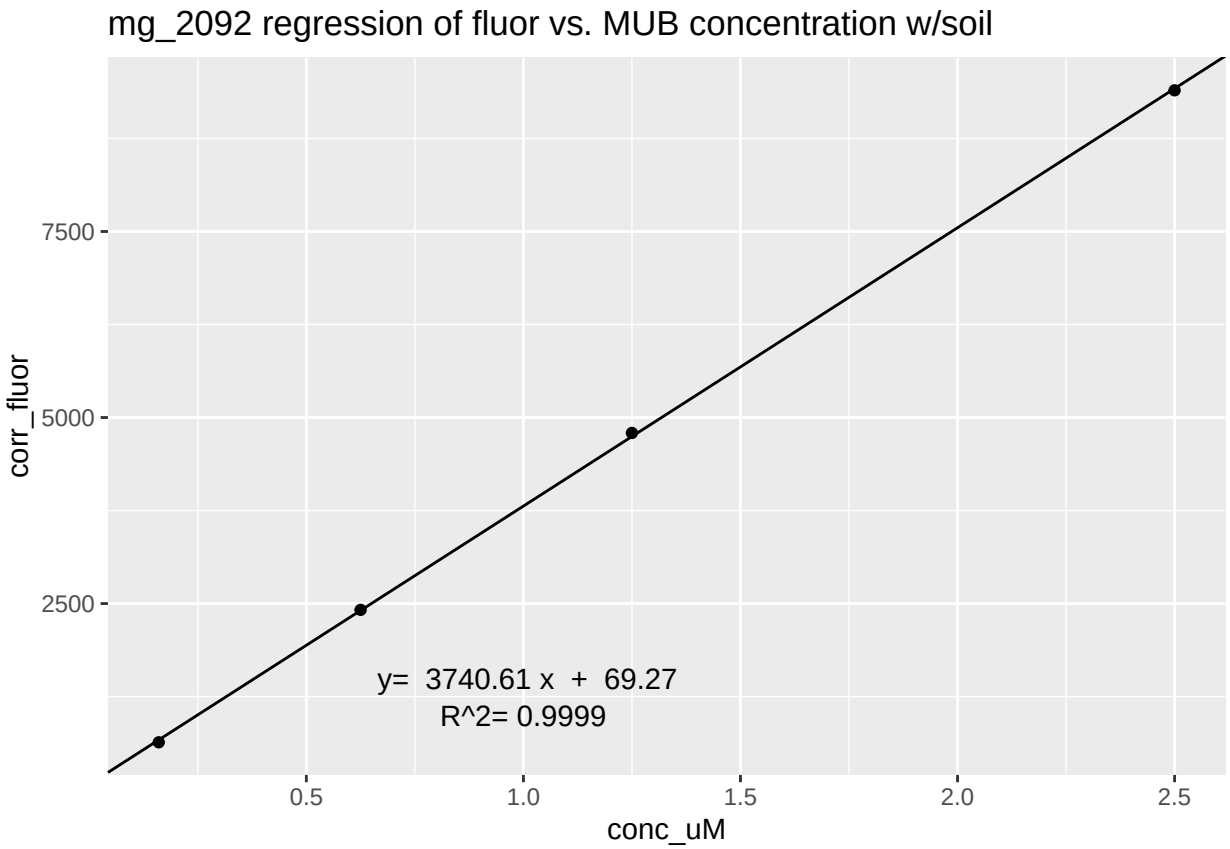
```
##
## [[10]]
```


mg_2083 regression of fluor vs. MUB concentration w/soil
8000
6000
4000
2000
corr_fluor
y= 3125.12 x + −36.48
R^2= 0.9996
0.5
1.0
1.5
2.0
2.5
conc_uM

```
##
## [[11]]
```


mg_2090 regression of fluor vs. MUB concentration w/soil
corr_fluor
10000
7500
5000
2500
y= 4014.66 x + 21.32
R^2= 0.9996
0.5
1.0
1.5
2.0
2.5
conc_uM

```
##
## [[12]]
```



Calculate emission coefficient

The **emission coefficient** is ((DEFINITION)) Formula for emission coefficient is: (nice formatted formula)

The emission coefficient is the slope (m) from plate b (fluorescence vs MUB conc NO SOIL) divided by the assay volume

Pay attention to units! **slope (m) units** from the standard curves = fluor/uM = fluor/ (umol/L) = fluor L / umol = fluor mL / nmol

assay volume is 250uL = 0.00025 L = 0.250 mL