C14

Denise Ong

4/2/2020

Load data

```
C14 <- readxl::read_excel("TAN1810 Adriana samples complete _final_AGR_1.0.xlsx", sheet = 2)
C14$EXP <- as.character(C14$EXP)
C14
```

```
## # A tibble: 460 x 50
##    Operator  date julian_day Cycle `CTD#`   STN `Incub Time` EXP   SAMPLE
##    <chr>    <dbl>      <dbl> <dbl> <chr>  <dbl> <chr>        <chr> <chr>
##  1 Adriana  43398        298     1 U9103     15 T24          1     SUR
##  2 Adriana  43398        298     1 U9103     15 T24          1     SUR
##  3 Adriana  43398        298     1 U9103     15 T24          1     SUR
##  4 Adriana  43398        298     1 U9103     15 T24          1     SUR
##  5 Adriana  43398        298     1 U9103     15 T24          1     SUR
##  6 Adriana  43398        298     1 U9103     15 T24          1     SUR
##  7 Adriana  43398        298     1 U9103     15 T24          1     SUR
##  8 Adriana  43398        298     1 U9103     15 T24          1     SUR
##  9 Adriana  43398        298     1 U9103     15 T24          1     SUR
## 10 Adriana  43398        298     1 U9103     15 T24          1     SUR
## # ... with 450 more rows, and 41 more variables: `Biosecurity code` <chr>,
## #   Lat_DD <dbl>, Long_DD <dbl>, `Karl Code` <dbl>, DEPTH <dbl>, `CTD Z
## #   code` <dbl>, `Vial code` <chr>, `Type (D for DNA and S for
## #   scintilation)` <chr>, `BOX NUMBER...18` <dbl>, `Sorting number` <dbl>,
## #   `sorting population` <chr>, `Cells sorted` <dbl>, `Sample name` <chr>,
## #   ...23 <dbl>, ...24 <chr>, ...25 <dbl>, `S#` <dbl>, `Count Time` <dbl>,
## #   CPMA <dbl>, DPM1 <dbl>, SIS <dbl>, tSIE <dbl>, `A:2S%` <dbl>,
## #   MESSAGES <lgl>, `BOX NUMBER...34` <dbl>, Tray <dbl>, Vial <dbl>, DIC <dbl>,
## #   `SA (uCi/mL)` <dbl>, `NPP (mgC d-1; need to double check the units,seems a
## #   bit too large for me)` <dbl>, `SA in DPM (aproximation)` <dbl>,
## #   ...41 <lgl>, ...42 <lgl>, ...43 <lgl>, ...44 <lgl>, ...45 <lgl>,
## #   ...46 <chr>, ...47 <chr>, ...48 <chr>, ...49 <dbl>, ...50 <chr>
```

Load libraries, and narrow down the columns for the dataset.

```
library(dplyr)
library(tidyverse)
library(ggplot2)
# create a new dataset
C14_1 <- select(C14, julian_day, Cycle, STN, EXP, SAMPLE, Lat_DD, Long_DD, DEPTH, 'Vial code', 'sorting
C14_1
```

```
## # A tibble: 460 x 12
##    julian_day Cycle   STN EXP   SAMPLE Lat_DD Long_DD DEPTH `Vial code`
##         <dbl> <dbl> <dbl> <chr> <chr>   <dbl>   <dbl> <dbl> <chr>
##  1        298     1    15 1     SUR     -44.6    175.    12 A
##  2        298     1    15 1     SUR     -44.6    175.    12 A
##  3        298     1    15 1     SUR     -44.6    175.    12 A
##  4        298     1    15 1     SUR     -44.6    175.    12 A
##  5        298     1    15 1     SUR     -44.6    175.    12 A
##  6        298     1    15 1     SUR     -44.6    175.    12 A
##  7        298     1    15 1     SUR     -44.6    175.    12 A
##  8        298     1    15 1     SUR     -44.6    175.    12 B
##  9        298     1    15 1     SUR     -44.6    175.    12 B
## 10        298     1    15 1     SUR     -44.6    175.    12 B
## # ... with 450 more rows, and 3 more variables: `sorting population` <chr>,
## #   `Cells sorted` <dbl>, DPM1 <dbl>
```

To include only essential columns. Can edit code to include other columns that are essential later.

```
# Tidy for essential columns
C14_DPM <- C14_1 %>%
  select ( EXP, SAMPLE, `Vial code`, `sorting population`, `Cells sorted`, DPM1)
C14_DPM
```

```
## # A tibble: 460 x 6
##    EXP   SAMPLE `Vial code` `sorting population` `Cells sorted`  DPM1
##    <chr> <chr>  <chr>       <chr>                         <dbl> <dbl>
##  1 1     SUR    A           Pico                           2000    62
##  2 1     SUR    A           Pico                           4000    93
##  3 1     SUR    A           Pico                          10000   179
##  4 1     SUR    A           Syn                            2000    75
##  5 1     SUR    A           Syn                            4000    48
##  6 1     SUR    A           Syn                           10000   140
##  7 1     SUR    A           Nano                           1000   926
##  8 1     SUR    B           Pico                           2000    61
##  9 1     SUR    B           Pico                           4000    90
## 10 1     SUR    B           Pico                          10000   184
## # ... with 450 more rows
```

First, I focused on experiment 1 only to try 2 different methods I had in mind.

```
#filter for exp 1
C14_E1 <- filter(C14_DPM, EXP==1)
C14_E1
```

```
## # A tibble: 28 x 6
##    EXP   SAMPLE `Vial code` `sorting population` `Cells sorted`  DPM1
##    <chr> <chr>  <chr>       <chr>                         <dbl> <dbl>
##  1 1     SUR    A           Pico                           2000    62
##  2 1     SUR    A           Pico                           4000    93
##  3 1     SUR    A           Pico                          10000   179
##  4 1     SUR    A           Syn                            2000    75
##  5 1     SUR    A           Syn                            4000    48
##  6 1     SUR    A           Syn                           10000   140
```

```
##  7 1     SUR    A           Nano                          1000   926
##  8 1     SUR    B           Pico                          2000    61
##  9 1     SUR    B           Pico                          4000    90
## 10 1     SUR    B           Pico                         10000   184
## # ... with 18 more rows
```

To make calculations easier, I first created a new dataset that would bring vial D to a new column called DPM_dark. I then create another column DPM2 = light - dark. DPM2 would be DPM data that I will use to calculate my PP.

Problem: this method only works for complete dataset, and I would need to repeat for every experiment and depth. How do I set conditions to make sure that DPM_dark moves to its corresponding population? I would like to have a code that will apply to the entire dataset, without needed to subset the data.

```
# create new column for D, repeating from A to C
# select values for DPM dark
subset1 <- subset(C14_E1, `Vial code` == "D",  select = c("DPM1"))
subset1
```

```
## # A tibble: 7 x 1
##    DPM1
##   <dbl>
## 1    28
## 2    28
## 3    35
## 4    26
## 5    30
## 6    37
## 7   905
```

```
# select for vials A to C
C14_E1 <- filter(C14_E1, `Vial code` != "D")
# create DPM dark column
C14_E1 <- mutate(C14_E1, DPM_dark = rep(subset1$DPM1, 3))
# DPMD2 = DPM light - dark
C14_E1 <- mutate(C14_E1, DPM2 = DPM1 - DPM_dark)
C14_E1
```

```
## # A tibble: 21 x 8
##    EXP   SAMPLE `Vial code` `sorting popula~ `Cells sorted`  DPM1 DPM_dark  DPM2
##    <chr> <chr>  <chr>       <chr>                    <dbl> <dbl>    <dbl> <dbl>
##  1 1     SUR    A           Pico                      2000    62       28    34
##  2 1     SUR    A           Pico                      4000    93       28    65
##  3 1     SUR    A           Pico                     10000   179       35   144
##  4 1     SUR    A           Syn                       2000    75       26    49
##  5 1     SUR    A           Syn                       4000    48       30    18
##  6 1     SUR    A           Syn                      10000   140       37   103
##  7 1     SUR    A           Nano                      1000   926      905    21
##  8 1     SUR    B           Pico                      2000    61       28    33
##  9 1     SUR    B           Pico                      4000    90       28    62
## 10 1     SUR    B           Pico                     10000   184       35   149
```

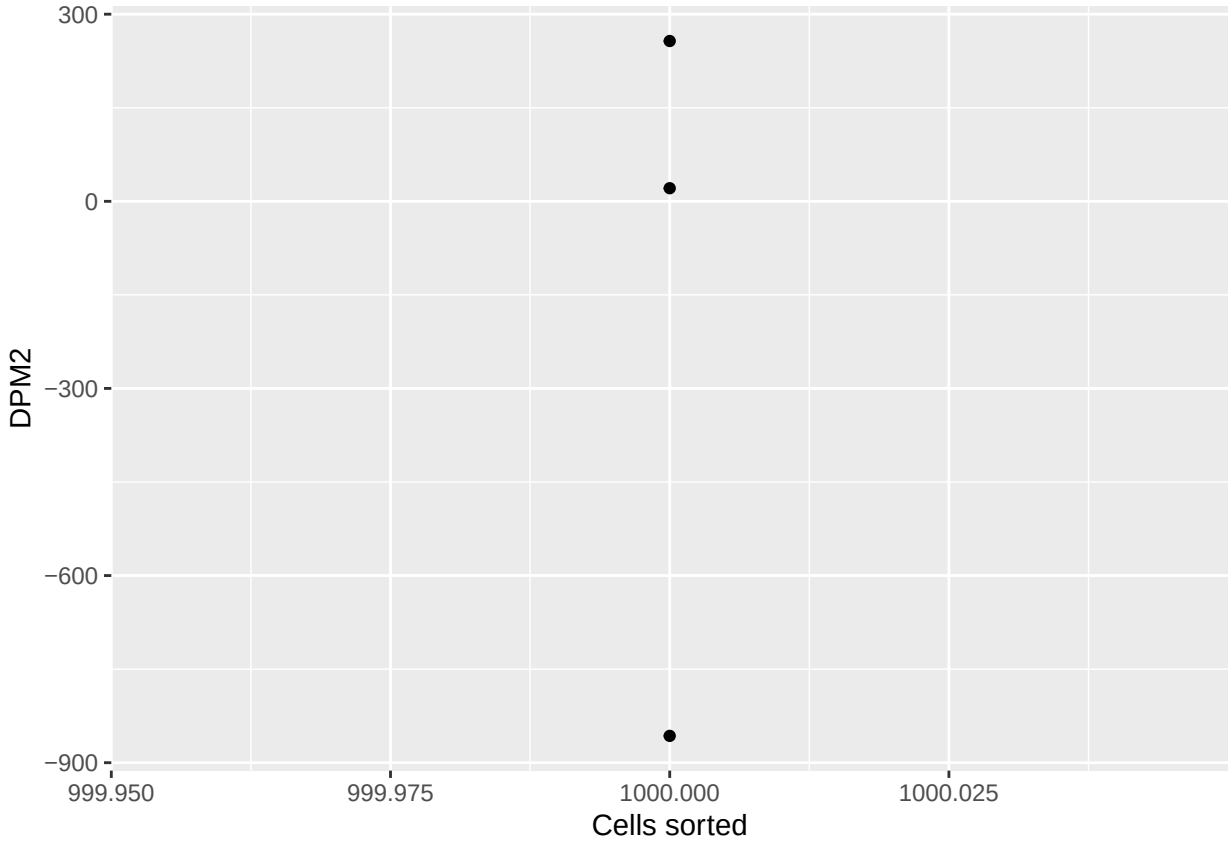
```
## # ... with 11 more rows
```

IGNORE Method 1: Calculate coefficient when vials A to C combined (I have decided not to use this method as it reduces my number of replicates)

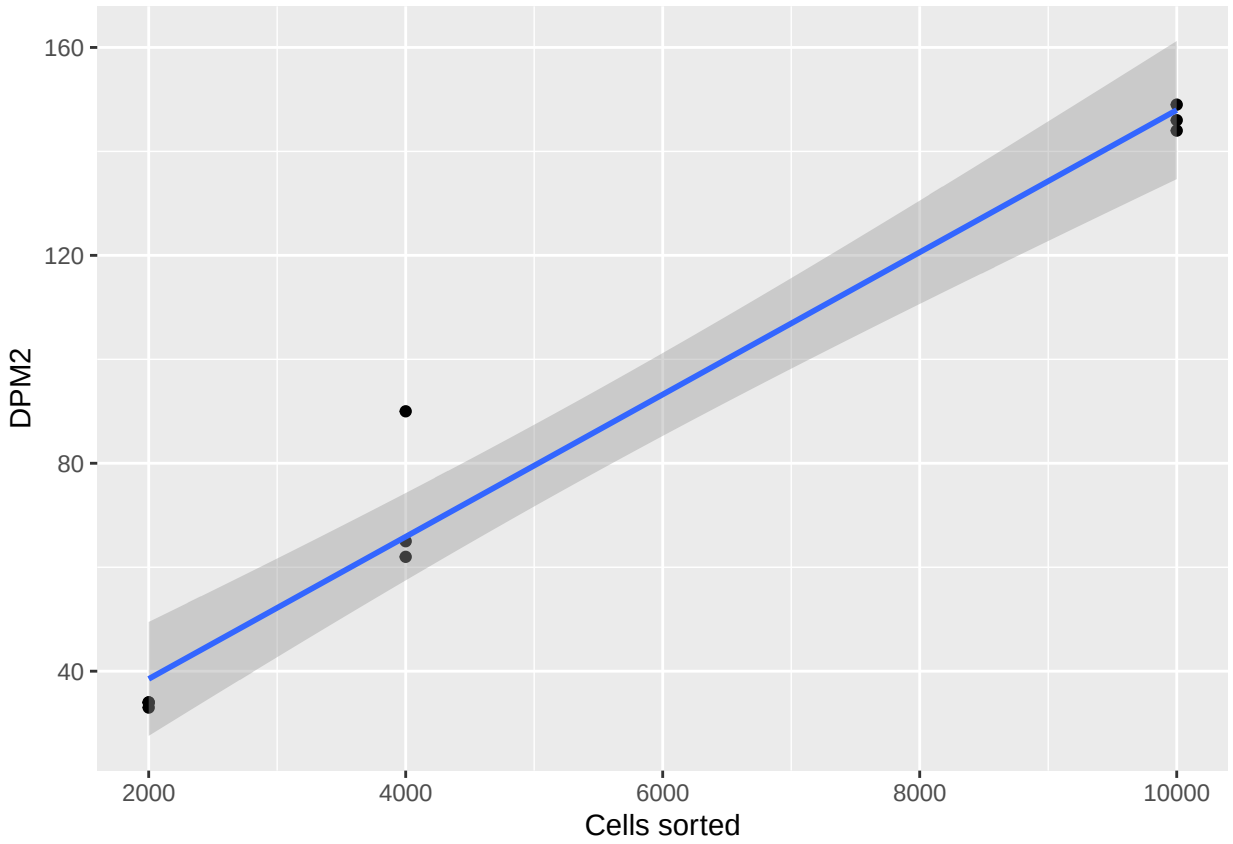
```
E1 <- C14_E1 %>%
  group_by(`sorting population`, `Cells sorted`, `Vial code`) %>%
  summarise(DPM2)
E1
```

```
## # A tibble: 21 x 4
## # Groups:   sorting population, Cells sorted [7]
##    `sorting population` `Cells sorted` `Vial code`  DPM2
##    <chr>                          <dbl> <chr>       <dbl>
##  1 Nano                            1000 A              21
##  2 Nano                            1000 B             257
##  3 Nano                            1000 C            -857
##  4 Pico                            2000 A              34
##  5 Pico                            2000 B              33
##  6 Pico                            2000 C              34
##  7 Pico                            4000 A              65
##  8 Pico                            4000 B              62
##  9 Pico                            4000 C              90
## 10 Pico                           10000 A             144
## # ... with 11 more rows
```

```
E1_nano <- E1 %>%
  ungroup() %>%
  filter(`sorting population` == "Nano")
E1_plot_nano <- ggplot(E1_nano, aes(x = `Cells sorted`, y = DPM2)) + geom_point()
E1_plot_nano
```




```
E1_pico <- E1 %>%
  ungroup() %>%
  filter(`sorting population` == "Pico")
E1_plot_pico <- ggplot(E1_pico, aes(x = `Cells sorted`, y = DPM2)) + geom_point() + geom_smooth(method =
E1_plot_pico
```




```
E1_model_pico <- lm(DPM2 ~ `Cells sorted`, data=E1_pico)
E1_model_pico
```

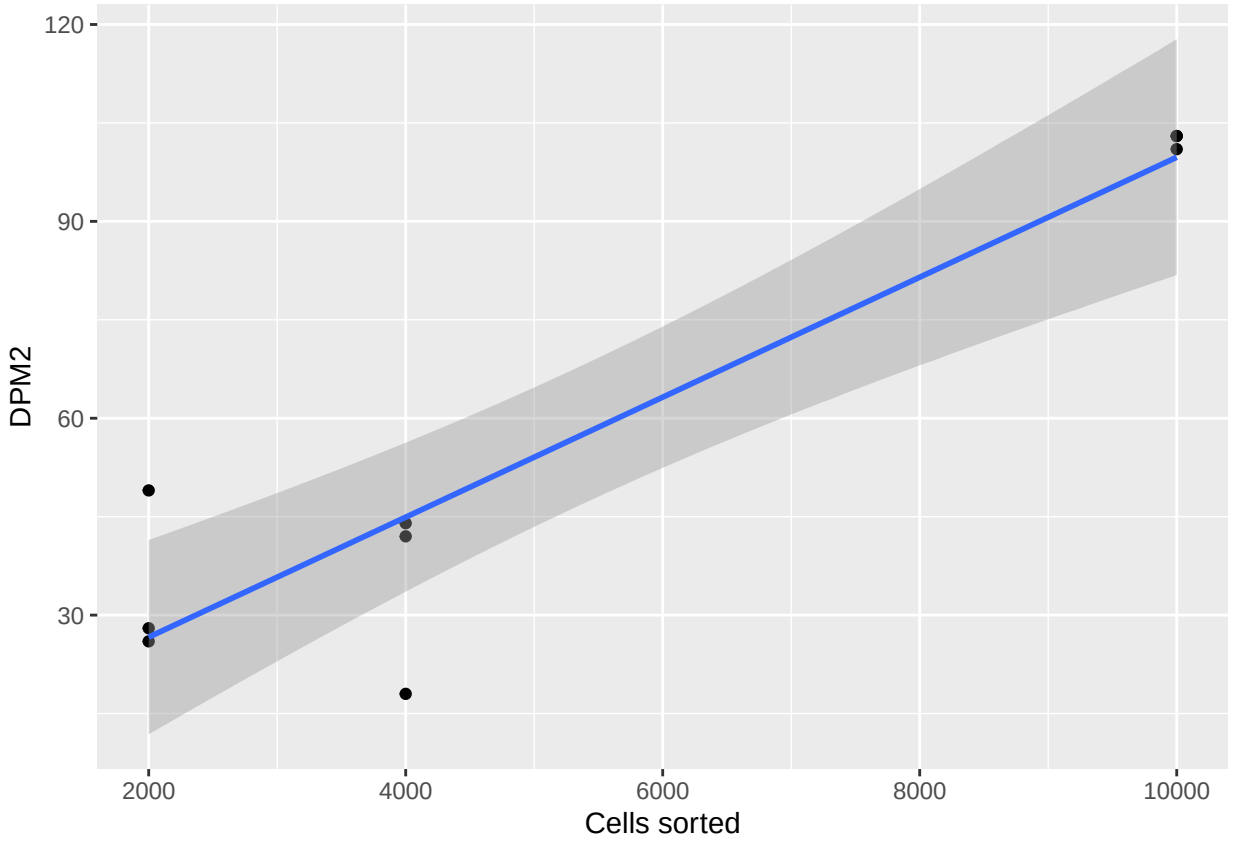
```
##
## Call:
## lm(formula = DPM2 ~ `Cells sorted`, data = E1_pico)
##
## Coefficients:
##    (Intercept)  `Cells sorted`
##       11.15385         0.01368
```

```
summary(E1_model_pico)
```

```
##
## Call:
## lm(formula = DPM2 ~ `Cells sorted`, data = E1_pico)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5128 -4.5128 -3.8718 -0.8718 24.1282
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.115e+01  6.155e+00   1.812    0.113
## `Cells sorted` 1.368e-02  9.732e-04  14.056 2.19e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 9.925 on 7 degrees of freedom
## Multiple R-squared:  0.9658, Adjusted R-squared:  0.9609
## F-statistic: 197.6 on 1 and 7 DF,  p-value: 2.186e-06
```

```
E1_syn <- E1 %>%
  ungroup() %>%
  filter(`sorting population` == "Syn")
E1_plot_syn <- ggplot(E1_syn, aes(x = `Cells sorted`, y = DPM2)) + geom_point() + geom_smooth(method = l
E1_plot_syn
```




```
E1_model_syn <- lm(DPM2 ~ `Cells sorted`, data=E1_syn)
E1_model_syn
```

```
##
## Call:
## lm(formula = DPM2 ~ `Cells sorted`, data = E1_syn)
##
## Coefficients:
##    (Intercept)  `Cells sorted`
##       8.358974        0.009141
```

```
summary(E1_model_syn)
```

```
##
## Call:
## lm(formula = DPM2 ~ `Cells sorted`, data = E1_syn)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -26.9231  -0.9231   1.2308   3.2308  22.3590
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.358974   8.316608   1.005 0.348323
## `Cells sorted` 0.009141   0.001315   6.952 0.000221 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.41 on 7 degrees of freedom
## Multiple R-squared:  0.8735, Adjusted R-squared:  0.8554
## F-statistic: 48.32 on 1 and 7 DF,  p-value: 0.0002209
```

Method 2: calculate regression coefficient for each vial

This method is the one that I would like to use. The idea that I have is to group the data by sorting population (nano, pico, syn) and then by vial code. I would like to use a map function to create a new column that creates an LM function for each sorting population and vial, such that lm(DPM2~cells sorted), and then corresponding columns for the slope, p-value or each LM.
Problem: I took some examples online to try and fit to my data but the code doesnt work.

```
library(dplyr)
library(broom)
library(tsibble)
```

```
# This is the code example that I used. The link is https://community.rstudio.com/t/how-to-calculate-sl
df %>%
    group_nest(Source.Name, visbility, soundvolume) %>%
    mutate(model = map(data, ~lm(m ~ stim_ending_t, data = .x))) %>%
    mutate(slope = map_dbl(model, ~tidy(.x)$estimate[2]))
```

```
C14_E1_ts <- as.tibble(C14_E1)
C14_E1_ts
E1_trial <- C14_E1_ts %>%
  ungroup() %>%
  group_by(`sorting population`, `Vial code`)  %>%
  mutate(model = map(data, ~lm(DPM2~`Cells sorted`, data = .))) %>%
  mutate(slope = map_dbl(model, ~tidy(.)$estimate[2]))
E1_trial
```

Daniel solution

A few things:

- Please read in detail R for data science chapter on grouping data: https://r4ds.had.co.nz/transform.html
- Please read in detail R for data science chpater on joining data: https://r4ds.had.co.nz/relational-data.html
- You can also go back to my class here: https://vaulot.github.io/course-ntu-data-science-2020/R-session-04-data_wrangling.html
- When you group data you must explicty tell which operation you want to perform (e.g. n() for counting, sum(), mean ()). If not no need to group data
- Make your code general from the start so that to extend to all cases, no need to rewrite it.

- It is much better to rename immediately your varibales in a clean way

Read the data

```
C14_DPM <- readxl::read_excel("TAN1810 Adriana samples complete _final_AGR_1.0.xlsx", sheet = "Sheet1")
     select ( EXP, SAMPLE, `Vial code`, `sorting population`, `Cells sorted`, DPM1) %>%   # Can add mor
     rename(exp = EXP,
            sample = SAMPLE,
            vial = `Vial code`,
            population = `sorting population`,
            cells_sorted = `Cells sorted`,
            dpm = DPM1) %>%
     filter(exp == 1)  # You will just need to remove this line to make the code valid for all experime
C14_DPM
```

```
## # A tibble: 28 x 6
##      exp sample vial  population cells_sorted   dpm
##    <dbl> <chr>  <chr> <chr>             <dbl> <dbl>
##  1     1 SUR    A     Pico               2000    62
##  2     1 SUR    A     Pico               4000    93
##  3     1 SUR    A     Pico              10000   179
##  4     1 SUR    A     Syn                2000    75
##  5     1 SUR    A     Syn                4000    48
##  6     1 SUR    A     Syn               10000   140
##  7     1 SUR    A     Nano               1000   926
##  8     1 SUR    B     Pico               2000    61
##  9     1 SUR    B     Pico               4000    90
## 10     1 SUR    B     Pico              10000   184
## # ... with 18 more rows
```

Substract the dark DPM

You need to make join based on the common variables

```
 C14_DPM_Dark <- C14_DPM %>%
  filter(vial == "D") %>%
  rename(dpm_dark = dpm) %>%
  select(-vial) # Remove the vial column
C14_DPM_Dark
```

```
## # A tibble: 7 x 5
##     exp sample population cells_sorted dpm_dark
```

```
##   <dbl> <chr>  <chr>             <dbl>    <dbl>
## 1     1 SUR    Pico               2000       28
## 2     1 SUR    Pico               4000       28
## 3     1 SUR    Pico              10000       35
## 4     1 SUR    Syn                2000       26
## 5     1 SUR    Syn                4000       30
## 6     1 SUR    Syn               10000       37
## 7     1 SUR    Nano               1000      905
```

```
C14_DPM_corrected <- left_join(C14_DPM,C14_DPM_Dark ) %>%
  filter (vial != "D") %>%
  mutate(dpm_corrected = dpm - dpm_dark) %>%
  filter(dpm_corrected >= 0) # Remove negative values

C14_DPM_corrected
```

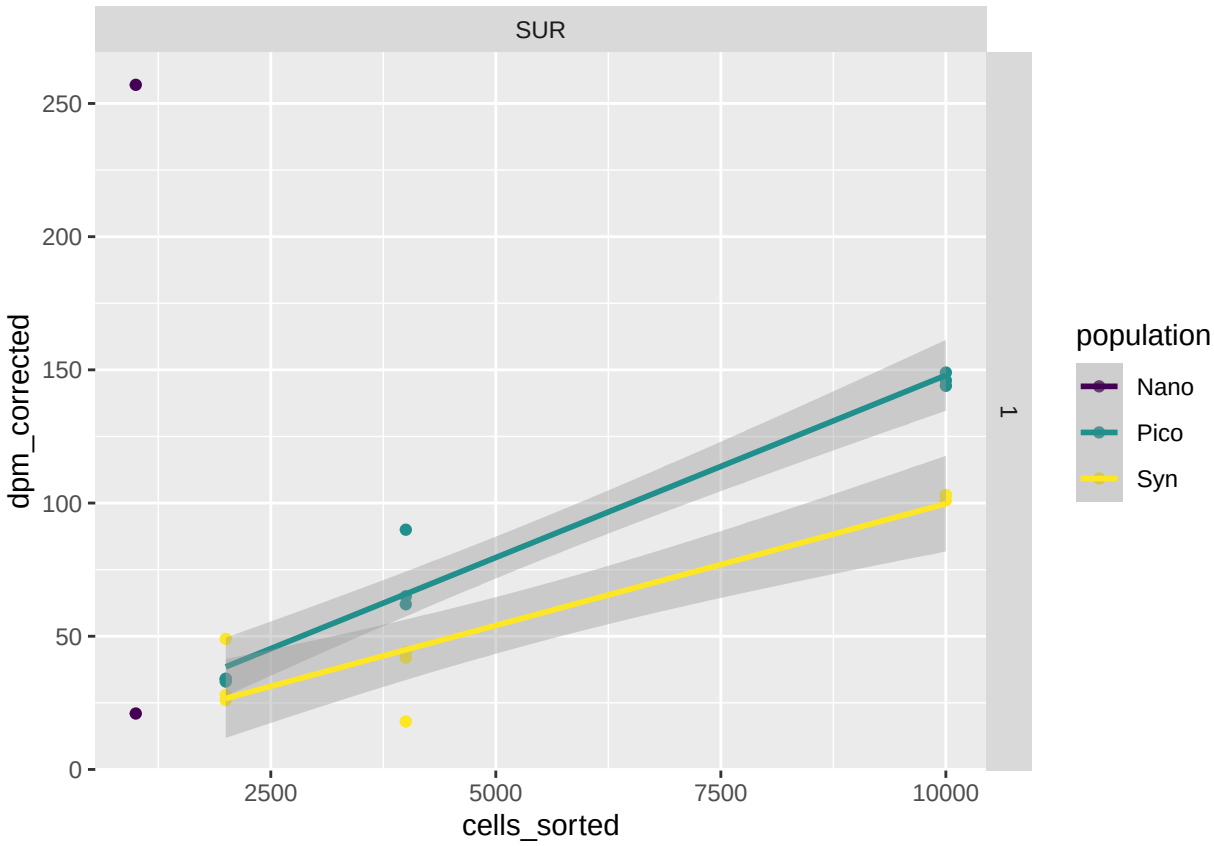
```
## # A tibble: 20 x 8
##      exp sample vial  population cells_sorted   dpm dpm_dark dpm_corrected
##    <dbl> <chr>  <chr> <chr>             <dbl> <dbl>    <dbl>          <dbl>
##  1     1 SUR    A     Pico               2000    62       28             34
##  2     1 SUR    A     Pico               4000    93       28             65
##  3     1 SUR    A     Pico              10000   179       35            144
##  4     1 SUR    A     Syn                2000    75       26             49
##  5     1 SUR    A     Syn                4000    48       30             18
##  6     1 SUR    A     Syn               10000   140       37            103
##  7     1 SUR    A     Nano               1000   926      905             21
##  8     1 SUR    B     Pico               2000    61       28             33
##  9     1 SUR    B     Pico               4000    90       28             62
## 10     1 SUR    B     Pico              10000   184       35            149
## 11     1 SUR    B     Syn                2000    54       26             28
## 12     1 SUR    B     Syn                4000    72       30             42
## 13     1 SUR    B     Syn               10000   138       37            101
## 14     1 SUR    B     Nano               1000  1162      905            257
## 15     1 SUR    C     Pico               2000    62       28             34
## 16     1 SUR    C     Pico               4000   118       28             90
## 17     1 SUR    C     Pico              10000   181       35            146
## 18     1 SUR    C     Syn                2000    52       26             26
## 19     1 SUR    C     Syn                4000    74       30             44
## 20     1 SUR    C     Syn               10000   140       37            103
```

Method 1 - Compute lm by grouping ABC together

Plots

Do plots for each group. One regression line based on the EXP, Sample and Populations

```
ggplot(data = C14_DPM_corrected, aes(x=cells_sorted, y=dpm_corrected, color=population)) +
  geom_point() + stat_smooth(method="lm") +
  facet_grid(rows=vars(exp), cols=vars(sample)) +
  scale_color_viridis_d()
```



Do linear model

$y = ax + b$

See: https://cran.r-project.org/web/packages/broom/vignettes/broom_and_dplyr.html

```
C14_DPM_model_1 <- C14_DPM_corrected %>%
  group_by(exp, sample, population) %>%
  tidyr::nest() %>%
  mutate(
    fit = purrr::map(data, ~ lm(dpm_corrected ~ cells_sorted, data = .x)),
    tidied = purrr::map(fit, tidy)
  ) %>%
  unnest(tidied)

C14_DPM_model_output_1 <- C14_DPM_model_1 %>%
  select(exp:population, term, estimate) %>%
  pivot_wider(names_from="term", values_from ="estimate" ) %>%
  rename (a = cells_sorted, b = `(Intercept)`)

C14_DPM_model_output_1
```

```
## # A tibble: 3 x 5
## # Groups:   exp, sample, population [3]
##     exp sample population      b        a
##   <dbl> <chr>  <chr>       <dbl>    <dbl>
## 1     1 SUR    Pico         11.2   0.0137
## 2     1 SUR    Syn           8.36  0.00914
```

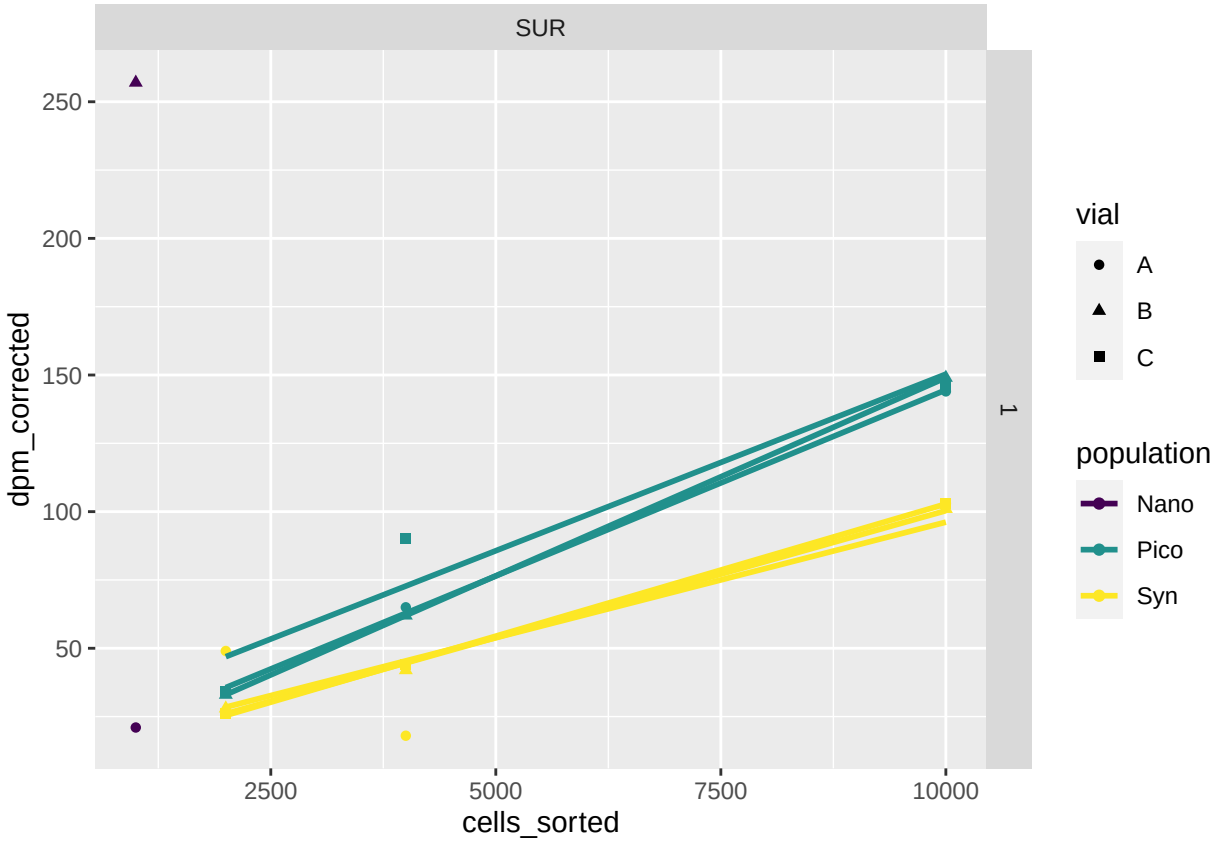
```
## 3     1 SUR    Nano       139    NA
```

Method 2 - Compute lm for ABC separately

Plots

Do plots for each group. One regression line based on the EXP, Sample and Populations

```
ggplot(data = C14_DPM_corrected, aes(x=cells_sorted, y=dpm_corrected, color=population, shape=vial)) +
 geom_point() + stat_smooth(method="lm", se=FALSE) +
 facet_grid(rows=vars(exp), cols=vars(sample)) +
 scale_color_viridis_d()
```



Do linear model

$y = ax + b$

See: https://cran.r-project.org/web/packages/broom/vignettes/broom_and_dplyr.html

```
C14_DPM_model_2 <- C14_DPM_corrected %>%
 group_by(exp, sample, population, vial) %>%
 tidyr::nest() %>%
 mutate(
   fit = purrr::map(data, ~ lm(dpm_corrected ~ cells_sorted, data = .x)),
   tidied = purrr::map(fit, tidy)
 ) %>%
 unnest(tidied)

C14_DPM_model_output_2 <- C14_DPM_model_2 %>%
```

```
  select(exp:population, term, estimate) %>%
  pivot_wider(names_from="term", values_from ="estimate" ) %>%
  rename (a = cells_sorted, b = `(Intercept)`)

C14_DPM_model_output_2
```

```
## # A tibble: 8 x 6
## # Groups:   exp, sample, vial, population [8]
##     exp sample vial  population      b        a
##   <dbl> <chr>  <chr> <chr>       <dbl>    <dbl>
## 1     1 SUR    A     Pico         8.38  0.0136
## 2     1 SUR    A     Syn         11.5   0.00846
## 3     1 SUR    A     Nano        21    NA
## 4     1 SUR    B     Pico         4.    0.0145
## 5     1 SUR    B     Syn          7.46  0.00929
## 6     1 SUR    B     Nano       257    NA
## 7     1 SUR    C     Pico        21.1   0.0129
## 8     1 SUR    C     Syn          6.08  0.00967
```

Denise edits

I made some additional changes. I filtered out Nano population, so the data will only have Pico and Syn. I also removed the filter for experiment 1, so the code is the entire dataset. I only used method 2 (compute lm for ABC separately). This set of edits will continue until the computation of PP as I have added the DIC and SA values from Andres, joining tables based on station no and depth (m).

Final problems:

- From experiment 6: SUR: vial C: pico: 2000 cells sorted -> corrected DPM value = 1485, much higher than DPM of 4000 and 10000 cells counted.
- What should I do with negative slope values that result in negative pp values?

Read the data

```
library(dplyr)
library(tidyverse)
library(ggplot2)
library(broom)
library(tsibble)
```

```
C14_DPM <- readxl::read_excel("TAN1810 Adriana samples complete _final_AGR_1.0.xlsx", sheet = "Sheet1")
     select ( Cycle, EXP, STN, DEPTH, SAMPLE, `Vial code`, `sorting population`, `Cells sorted`, DPM1) %
  # Can add more variables latter
  # station and depth for adding updated DIC values
     rename(cycle = Cycle,
            exp = EXP,
            station = STN,
            depth = DEPTH,
            sample = SAMPLE,
            vial = `Vial code`,
            population = `sorting population`,
            cells_sorted = `Cells sorted`,
            dpm = DPM1) %>%
    filter(population != "Nano")

C14_DPM
```

```
## # A tibble: 385 x 9
##    cycle   exp station depth sample vial  population cells_sorted   dpm
##    <dbl> <dbl>   <dbl> <dbl> <chr>  <chr> <chr>             <dbl> <dbl>
##  1     1     1      15    12 SUR    A     Pico               2000    62
##  2     1     1      15    12 SUR    A     Pico               4000    93
##  3     1     1      15    12 SUR    A     Pico              10000   179
##  4     1     1      15    12 SUR    A     Syn                2000    75
##  5     1     1      15    12 SUR    A     Syn                4000    48
##  6     1     1      15    12 SUR    A     Syn               10000   140
##  7     1     1      15    12 SUR    B     Pico               2000    61
##  8     1     1      15    12 SUR    B     Pico               4000    90
##  9     1     1      15    12 SUR    B     Pico              10000   184
## 10     1     1      15    12 SUR    B     Syn                2000    54
## # ... with 375 more rows
```

Substract the dark DPM

You need to make join based on the common variables

```
 C14_DPM_Dark <- C14_DPM %>%
  filter(vial == "D") %>%
  rename(dpm_dark = dpm) %>%
  select(-vial) # Remove the vial column
C14_DPM_Dark
```

```
## # A tibble: 103 x 8
##    cycle   exp station depth sample population cells_sorted dpm_dark
##    <dbl> <dbl>   <dbl> <dbl> <chr>  <chr>             <dbl>    <dbl>
##  1     1     1      15    12 SUR    Pico               2000       28
##  2     1     1      15    12 SUR    Pico               4000       28
##  3     1     1      15    12 SUR    Pico              10000       35
##  4     1     1      15    12 SUR    Syn                2000       26
##  5     1     1      15    12 SUR    Syn                4000       30
##  6     1     1      15    12 SUR    Syn               10000       37
##  7     1     2      24    12 SUR    Pico               2000       25
##  8     1     2      24    12 SUR    Pico               4000       28
##  9     1     2      24    12 SUR    Pico              10000       29
## 10     1     2      24    12 SUR    Syn                2000       29
## # ... with 93 more rows
```

```
C14_DPM_corrected <- left_join(C14_DPM,C14_DPM_Dark ) %>%
  filter (vial != "D") %>%
  mutate(dpm_corrected = dpm - dpm_dark) %>%
  filter(dpm_corrected >= 0) %>% # Remove negative values
  filter(!(exp == 2 & sample == "DCM" & vial == "A")) # I removed this row as there was only one value

C14_DPM_corrected
```

```
## # A tibble: 246 x 11
##    cycle   exp station depth sample vial  population cells_sorted   dpm dpm_dark
##    <dbl> <dbl>   <dbl> <dbl> <chr>  <chr> <chr>             <dbl> <dbl>    <dbl>
##  1     1     1      15    12 SUR    A     Pico               2000    62       28
##  2     1     1      15    12 SUR    A     Pico               4000    93       28
##  3     1     1      15    12 SUR    A     Pico              10000   179       35
##  4     1     1      15    12 SUR    A     Syn                2000    75       26
##  5     1     1      15    12 SUR    A     Syn                4000    48       30
##  6     1     1      15    12 SUR    A     Syn               10000   140       37
##  7     1     1      15    12 SUR    B     Pico               2000    61       28
##  8     1     1      15    12 SUR    B     Pico               4000    90       28
##  9     1     1      15    12 SUR    B     Pico              10000   184       35
## 10     1     1      15    12 SUR    B     Syn                2000    54       26
## # ... with 236 more rows, and 1 more variable: dpm_corrected <dbl>
```

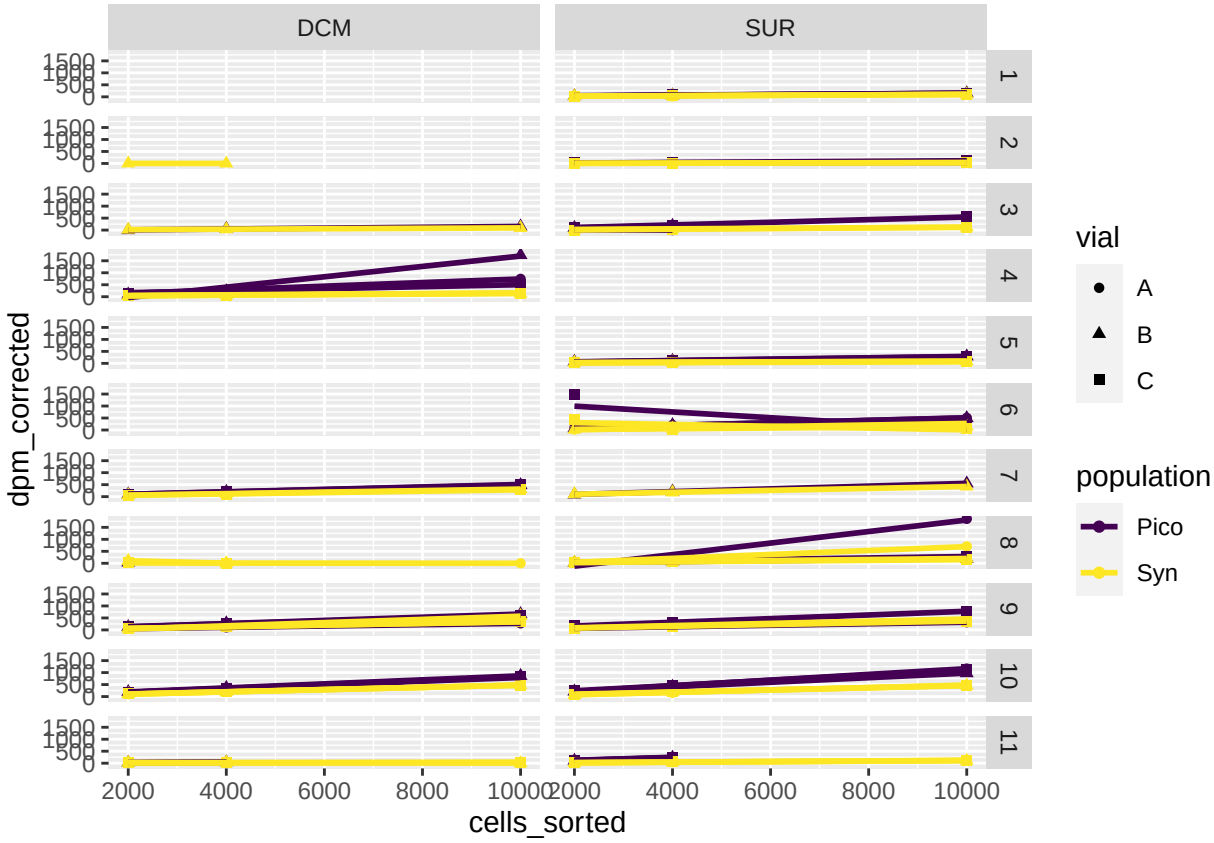
Method 2 - Compute lm for ABC separately

Plots

Do plots for each group. One regression line based on the EXP, Sample and Populations

```
 ggplot(data = C14_DPM_corrected, aes(x=cells_sorted, y=dpm_corrected, color=population, shape=vial)) +
  geom_point() + stat_smooth(method="lm", se=FALSE) +
  facet_grid(rows=vars(exp), cols=vars(sample)) +
```

```
  scale_color_viridis_d()
```



need to check exp 6: SUR: Pico: C: 2000 cells sorted. Value is too high. Slope value is negative.

Do linear model

$y = ax + b$

See: https://cran.r-project.org/web/packages/broom/vignettes/broom_and_dplyr.html

```
C14_DPM_model_3 <- C14_DPM_corrected %>%
  group_by(cycle, exp, station, depth, sample, population, vial) %>%
  tidyr::nest() %>%
  mutate(
    fit = purrr::map(data, ~ lm(dpm_corrected ~ cells_sorted, data = .x)),
    tidied = purrr::map(fit, tidy)
  ) %>%
  unnest(tidied)

C14_DPM_model_output_3 <- C14_DPM_model_3 %>%
  select(exp:population, term, estimate) %>%
  pivot_wider(names_from="term", values_from ="estimate" ) %>%
  rename (slope = cells_sorted, intercept = `(Intercept)`)

C14_DPM_model_output_3
```

```
## # A tibble: 88 x 9
## # Groups:   cycle, exp, station, depth, sample, vial, population [88]
##    cycle   exp station depth sample vial  population intercept   slope
##    <dbl> <dbl>   <dbl> <dbl> <chr>  <chr> <chr>          <dbl>   <dbl>
##  1     1     1      15    12 SUR    A     Pico           8.38  0.0136
##  2     1     1      15    12 SUR    A     Syn           11.5   0.00846
##  3     1     1      15    12 SUR    B     Pico           4.    0.0145
##  4     1     1      15    12 SUR    B     Syn            7.46  0.00929
##  5     1     1      15    12 SUR    C     Pico          21.1   0.0129
##  6     1     1      15    12 SUR    C     Syn            6.08  0.00967
##  7     1     2      24    12 SUR    A     Pico          15.9   0.00258
##  8     1     2      24    12 SUR    A     Syn           -3.38  0.00438
##  9     1     2      24    12 SUR    C     Pico          -6.77  0.0120
## 10     1     2      24    12 SUR    C     Syn            0.769 0.00298
## # ... with 78 more rows
```

Include DIC and SA values

Import DIC data from Andres. I will use this data to join with the DPM model output. The corrected data frame will have the DIC and SA values. The data will join based on the common columns: station and depth.

Create DIC table

```
DIC_data <- readxl::read_excel("Chla NPP Raw TAN1810 Sept.xlsx", sheet = "Compiled TAN1810 NPP data")

# excel sheet not optimal for R, have to delete columns and make row 5 headers.
names(DIC_data) <- DIC_data[5,] # make row 5 header
DIC_data <- DIC_data [-c(1, 2, 3, 4, 5), c(1:34)]
DIC_data <- DIC_data [, -c(1, 27)] # remove repeated columns
colnames(DIC_data) # check
```

```
##  [1] "Mean contol activity DPM"           "Date"
##  [3] "Time Start"                         "Time End"
##  [5] "Time  days"                         "Station"
##  [7] "Site"                               "Latitude"
##  [9] "Longitude"                          "Depth m"
## [11] "DATE"                               "TIME"
## [13] "P#"                                 "PID"
## [15] "S#"                                 "Count Time"
## [17] "A:2S%"                              "CPMA"
## [19] "SIS"                                "tSIE"
## [21] "DPM1"                               "DPM to full volume"
## [23] "Sample ID"                          "Comments"
## [25] "Volume filtered of bottles 320mls"  "DIC"
## [27] "Additive Chla Total for comparison" "Chla>0.2\r\nmg/m3"
## [29] "Chla>2\r\nmg/m3"                    "Chla>20\r\n mg/m3"
## [31] "Unique code or sample identifier"   "SA"
```

```
# select for station, depth, DIC, SA. Change all columns to integer to join with DPM output.
DIC_data_corrected <- select(DIC_data, Station, `Depth m`, DIC, SA) %>%
  unique() %>% # delete repeated rows
  rename(station = Station,
         depth = `Depth m`) %>%
  filter(depth != "35/40?") %>% # filter out character to change to integer
  mutate_if(is.character, as.double) # change to double (same format as DPM dataset)
```

```
DIC_data_corrected
```

```
## # A tibble: 101 x 4
##    station depth   DIC     SA
##      <dbl> <dbl> <dbl>  <dbl>
##  1      15     5  25.9 21500.
##  2      15    12  25.9 21087.
##  3      24     5  25.9 21089.
##  4      24    12  25.7 20958.
##  5      24    20  25.9 20453.
##  6      24    30  25.9 20761.
##  7      24    40  26.2 21436.
##  8      24    50  26.1 21132.
##  9      39     5  25.7 22025.
## 10      39    12  25.8 18035.
## # ... with 91 more rows
```

Merge DIC/SA table and DPM output

```
# merge tables
final_cal <- left_join(C14_DPM_model_output_3, DIC_data_corrected)
final_cal
```

```
## # A tibble: 88 x 11
## # Groups:   cycle, exp, station, depth, sample, vial, population [88]
##    cycle   exp station depth sample vial  population intercept   slope   DIC
##    <dbl> <dbl>   <dbl> <dbl> <chr>  <chr> <chr>          <dbl>   <dbl> <dbl>
##  1     1     1      15    12 SUR    A     Pico           8.38  0.0136   25.9
##  2     1     1      15    12 SUR    A     Syn           11.5   0.00846  25.9
##  3     1     1      15    12 SUR    B     Pico           4.    0.0145   25.9
##  4     1     1      15    12 SUR    B     Syn            7.46  0.00929  25.9
##  5     1     1      15    12 SUR    C     Pico          21.1   0.0129   25.9
##  6     1     1      15    12 SUR    C     Syn            6.08  0.00967  25.9
##  7     1     2      24    12 SUR    A     Pico          15.9   0.00258  25.7
##  8     1     2      24    12 SUR    A     Syn           -3.38  0.00438  25.7
##  9     1     2      24    12 SUR    C     Pico          -6.77  0.0120   25.7
## 10     1     2      24    12 SUR    C     Syn            0.769 0.00298  25.7
## # ... with 78 more rows, and 1 more variable: SA <dbl>
```

Calculate PP value, based on Daniel's formula found here:

https://vaulot.netlify.com/2018/05/20/compute-primary-production-based-on-single-cell-c14-uptake/

```
final_cal <- mutate(final_cal, pp = DIC*slope*(1/(SA*24))*10^9*1.05) %>%
  ungroup() %>%
  select( -c(station, depth))
#add missing rows for data analysis later
final_cal <- complete(final_cal, nesting(cycle, exp), sample, vial, population)

final_cal
```

```
## # A tibble: 132 x 10
##    cycle   exp sample vial  population intercept    slope   DIC     SA    pp
##    <dbl> <dbl> <chr>  <chr> <chr>          <dbl>    <dbl> <dbl>  <dbl> <dbl>
```

```
##  1     1     1 DCM    A     Pico           NA    NA        NA      NA    NA
##  2     1     1 DCM    A     Syn            NA    NA        NA      NA    NA
##  3     1     1 DCM    B     Pico           NA    NA        NA      NA    NA
##  4     1     1 DCM    B     Syn            NA    NA        NA      NA    NA
##  5     1     1 DCM    C     Pico           NA    NA        NA      NA    NA
##  6     1     1 DCM    C     Syn            NA    NA        NA      NA    NA
##  7     1     1 SUR    A     Pico            8.38  0.0136   25.9 21087.  732.
##  8     1     1 SUR    A     Syn            11.5   0.00846  25.9 21087.  455.
##  9     1     1 SUR    B     Pico            4.    0.0145   25.9 21087.  779.
## 10     1     1 SUR    B     Syn             7.46  0.00929  25.9 21087.  499.
## # ... with 122 more rows
```

Some notes from 14 April:

- For the purposes of Eleanor's class, I will use method 2 (compute lm values for ABC separately) for more replicates, but method 1 will be used for actual calculations later.
- No need to check p value of the slope.
- Will not set intercept to 0.