Lyons Data Import Mini-Project

Oliver d'Pug

Import Temperature and Agitation Data

We use the **readxl** package to import data from the "Data_Complete_Temp_Ag (75%oc)" sheet in "4NP Standard Curve and Data.xlsx". Data are in a similar format to the "Data_Complete_%OC" sheet, so we just rename a few things.

```
  ### Import the entire sheet
  tempag <- read_excel("4NP Standard Curve and Data.xlsx",
                       sheet = "Data_Complete_Temp_Ag (75%oc)",
                       skip = 2)
```

```
## New names:
## * `Mass of dust per trial (g)` -> `Mass of dust per trial (g)...5`
## * `4NP in trial (mg)` -> `4NP in trial (mg)...6`
## * `Mass released (mg)` -> `Mass released (mg)...7`
## * `Mass of dust per trial (g)` -> `Mass of dust per trial (g)...9`
## * `4NP in trial (mg)` -> `4NP in trial (mg)...10`
## * ...
```

```
  names(tempag)
```

```
##  [1] "Days"                          "Mass dosed"
##  [3] "Mass dust dosed(g)"            "Conc on dust (mg/g)"
##  [5] "Mass of dust per trial (g)...5"  "4NP in trial (mg)...6"
##  [7] "Mass released (mg)...7"        "20C calm"
##  [9] "Mass of dust per trial (g)...9"  "4NP in trial (mg)...10"
## [11] "Mass released (mg)...11"       "20 C agit"
## [13] "Mass of dust per trial (g)...13" "4NP in trial (mg)...14"
## [15] "Mass released (mg)...15"       "4 C calm"
## [17] "Mass of dust per trial (g)...17" "4NP in trial (mg)...18"
## [19] "Mass released (mg)...19"       "4 C agit"
```

```
  ### Subset columns 1:4 and 5:8 to get the 20 degree and calm data
  tempag.20calm <- tempag[, c(1:4,5:8)]
  names(tempag.20calm) <- c("Days", "Mass_Dosed", "Dust_Massed_Dosed", "Conc_On_Dust", "Mass_Of_Dust_Pe
  tempag.20calm$Temp <- 20
  tempag.20calm$Agitation <- "Calm"

  ### Subset columns 1:4 and 9:12 to get the 20 degree and agitated data
  tempag.20ag <- tempag[, c(1:4,9:12)]
  names(tempag.20ag) <- c("Days", "Mass_Dosed", "Dust_Massed_Dosed", "Conc_On_Dust", "Mass_Of_Dust_Per_
  tempag.20ag$Temp <- 20
  tempag.20ag$Agitation <- "Agitated"

  ### Subset columns 1:4 and 13:16 to get the 4 degree and calm data
  tempag.4calm <- tempag[, c(1:4,13:16)]
```

```
  names(tempag.4calm) <- c("Days", "Mass_Dosed", "Dust_Massed_Dosed", "Conc_On_Dust", "Mass_Of_Dust_Per_
  tempag.4calm$Temp <- 4
  tempag.4calm$Agitation <- "Calm"

  ### Subset columns 1:4 and 17:20 to get the 4 degree and agitated data
  tempag.4ag <- tempag[, c(1:4,17:20)]
  names(tempag.4ag) <- c("Days", "Mass_Dosed", "Dust_Massed_Dosed", "Conc_On_Dust", "Mass_Of_Dust_Per_T
  tempag.4ag$Temp <- 4
  tempag.4ag$Agitation <- "Agitated"


  ### Stack the data to create a single data.frame
  tempag <- rbind(tempag.20calm, tempag.20ag, tempag.4calm, tempag.4ag)

  ### "Study day" is coded as "Days.Trial"
  tempag$Sample <- factor(round((tempag$Days - floor(tempag$Days))*10))
  tempag$Days <- floor(tempag$Days)

  ### For percentage and proportion to numeric
  tempag$Perc_Released <- as.numeric(tempag$Perc_Released)
```

```
## Warning: NAs introduced by coercion
```

```
  tempag$Prop_Released <- tempag$Perc_Released / 100

  tempag <- tempag %>% filter(!is.na(Days))

  ### Remove/delete the temporary individual percentage data.frames
  rm(tempag.20calm, tempag.20ag, tempag.4calm, tempag.4ag)

  ### Write the complete, reformatted data.frame to CSV
  write.csv(tempag, "Four_NP_Perc_Released_Temp_Ag.csv", row.names = FALSE)

  ### Check to make sure everything is balanced and plot it just for fun
  head(tempag)
```

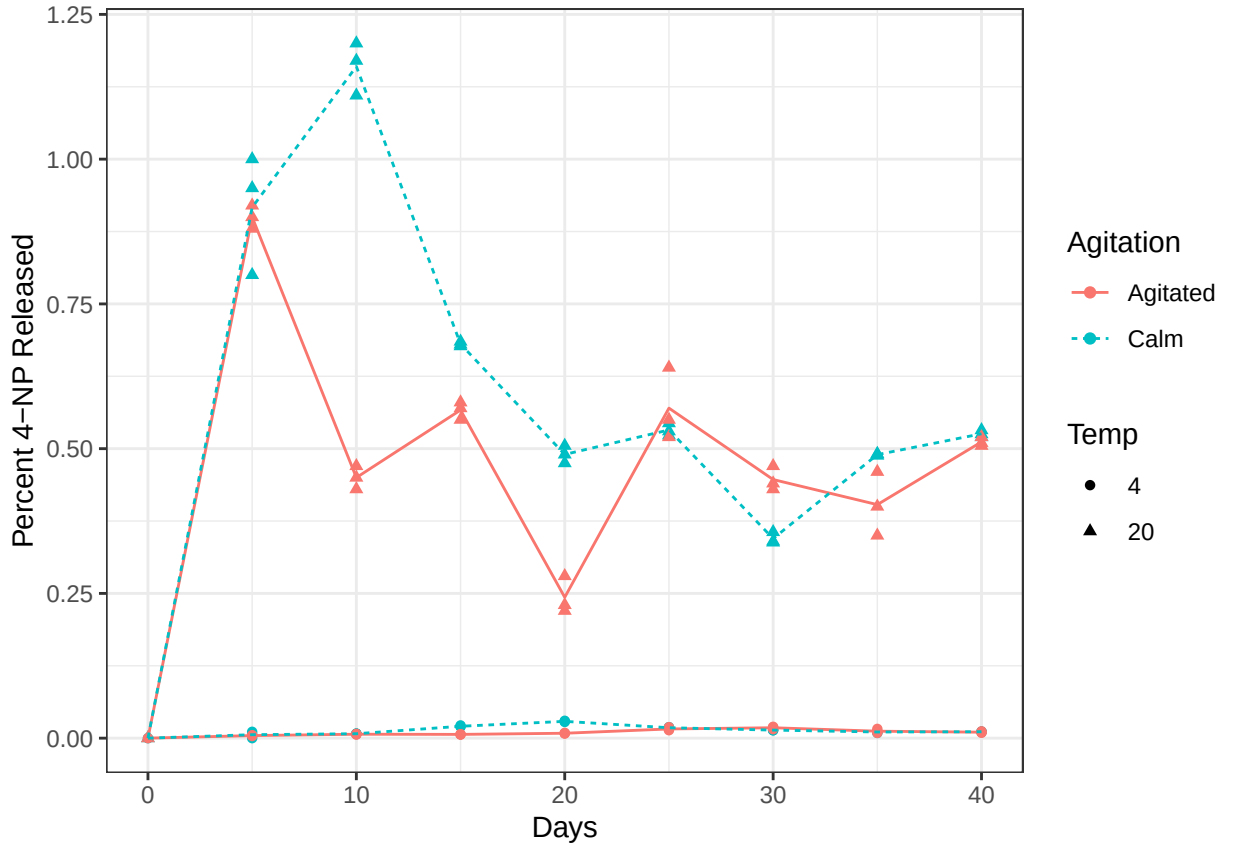
```
## # A tibble: 6 x 12
##    Days Mass_Dosed Dust_Massed_Dos~ Conc_On_Dust Mass_Of_Dust_Pe~
##   <dbl>      <dbl>            <dbl>        <dbl>            <dbl>
## 1     0       NA               NA          NA               NA
## 2     5       18.0              5.5         3.27             1.25
## 3     5       15.6              5.6         2.79             1.22
## 4     5       22.3              7           3.19             1.31
## 5    10       18.0              5.5         3.27             1.2
## 6    10       15.6              5.6         2.79             1.5
## # ... with 7 more variables: FourNP_In_Trial <dbl>, Mass_Released <chr>,
## #   Perc_Released <dbl>, Temp <dbl>, Agitation <chr>, Sample <fct>,
## #   Prop_Released <dbl>
```

```
  table(tempag$Days, tempag$Temp, tempag$Agitation)
```

```
## , ,  = Agitated
##
##
##      4 20
```

```
##   0  1  1
##   5  3  3
##   10 3  3
##   15 3  3
##   20 3  3
##   25 3  3
##   30 3  3
##   35 3  3
##   40 3  3
##
## , ,  = Calm
##
##
##      4 20
##   0  1  1
##   5  3  3
##   10 3  3
##   15 3  3
##   20 3  3
##   25 3  3
##   30 3  3
##   35 3  3
##   40 3  3
```

```
ggplot(tempag, aes(x=Days, y=Perc_Released,
                   colour=factor(Agitation), shape=factor(Temp),
                   linetype=factor(Agitation))) +
       geom_point() +
       stat_summary(fun="mean", geom="line") +
       #stat_summary(fun="mean", geom="point", shape = "-", size=10) +
       theme_bw() +
       labs(shape = "Temp", linetype = "Agitation", colour = "Agitation") +
       ylab("Percent 4-NP Released")
```




```
  head(tempag)
```

```
## # A tibble: 6 x 12
##    Days Mass_Dosed Dust_Massed_Dos~ Conc_On_Dust Mass_Of_Dust_Pe~
##   <dbl>      <dbl>            <dbl>        <dbl>            <dbl>
## 1     0       NA               NA          NA               NA
## 2     5       18.0              5.5         3.27             1.25
## 3     5       15.6              5.6         2.79             1.22
## 4     5       22.3              7           3.19             1.31
## 5    10       18.0              5.5         3.27             1.2
## 6    10       15.6              5.6         2.79             1.5
## # ... with 7 more variables: FourNP_In_Trial <dbl>, Mass_Released <chr>,
## #   Perc_Released <dbl>, Temp <dbl>, Agitation <chr>, Sample <fct>,
## #   Prop_Released <dbl>
```

```
  dim(tempag)
```

```
## [1] 100  12
```

Import Size Data Using Ranges

We use the **readxl** package to import data from selected ranges of the "Data_Complete_Size" sheet in "4NP Standard Curve and Data.xlsx".

```
  ### Import the 63 um data
  size.63 <- read_excel("4NP Standard Curve and Data.xlsx",
                        sheet = "Data_Complete_Size",
                        range = "A4:H28",
```

```
                      col_names =c("Days", "Mass_Dosed", "Dust_Massed_Dosed", "Conc_On_Dust", "Mass_Of_D
  names(size.63)
```

```
## [1] "Days"                   "Mass_Dosed"             "Dust_Massed_Dosed"
## [4] "Conc_On_Dust"           "Mass_Of_Dust_Per_Trial" "FourNP_In_Trial"
## [7] "Mass_Released"          "Perc_Released"
```

```
  size.63$Size <- 63

  ### Import the 125 um data
  size.125 <- read_excel("4NP Standard Curve and Data.xlsx",
                      sheet = "Data_Complete_Size",
                      range = "A33:H57",
                      col_names =c("Days", "Mass_Dosed", "Dust_Massed_Dosed", "Conc_On_Dust", "Mass_Of_D
  size.125$Size <- 125

  ### Import the 250 um data
  size.250 <- read_excel("4NP Standard Curve and Data.xlsx",
                      sheet = "Data_Complete_Size",
                      range = "A61:H85",
                      col_names =c("Days", "Mass_Dosed", "Dust_Massed_Dosed", "Conc_On_Dust", "Mass_Of_D
  size.250$Size <- 250

  ### Import the 500 um data
  size.500 <- read_excel("4NP Standard Curve and Data.xlsx",
                      sheet = "Data_Complete_Size",
                      range = "A89:H113",
                      col_names =c("Days", "Mass_Dosed", "Dust_Massed_Dosed", "Conc_On_Dust", "Mass_Of_D
  size.500$Size <- 500


  ### Stack the 63, 125, 250, and 500 um data to create a single data.frame
  dustsize <- rbind(size.63, size.125, size.250, size.500)
  dustsize$Sample <- factor(round((dustsize$Days - floor(dustsize$Days))*10))
  dustsize$Days <- floor(dustsize$Days)
  dustsize$Perc_Released <- as.numeric(dustsize$Perc_Released)
  dustsize$Prop_Released <- dustsize$Perc_Released / 100

  ### Remove/delete the temporary individual percentage data.frames
  rm(size.63, size.125, size.250, size.500)

  ### Write the complete, reformatted data.frame to CSV
  write.csv(dustsize, "Four_NP_Size.csv", row.names = FALSE)

  ### Check to make sure everything is balanced and plot it just for fun
  head(dustsize)
```

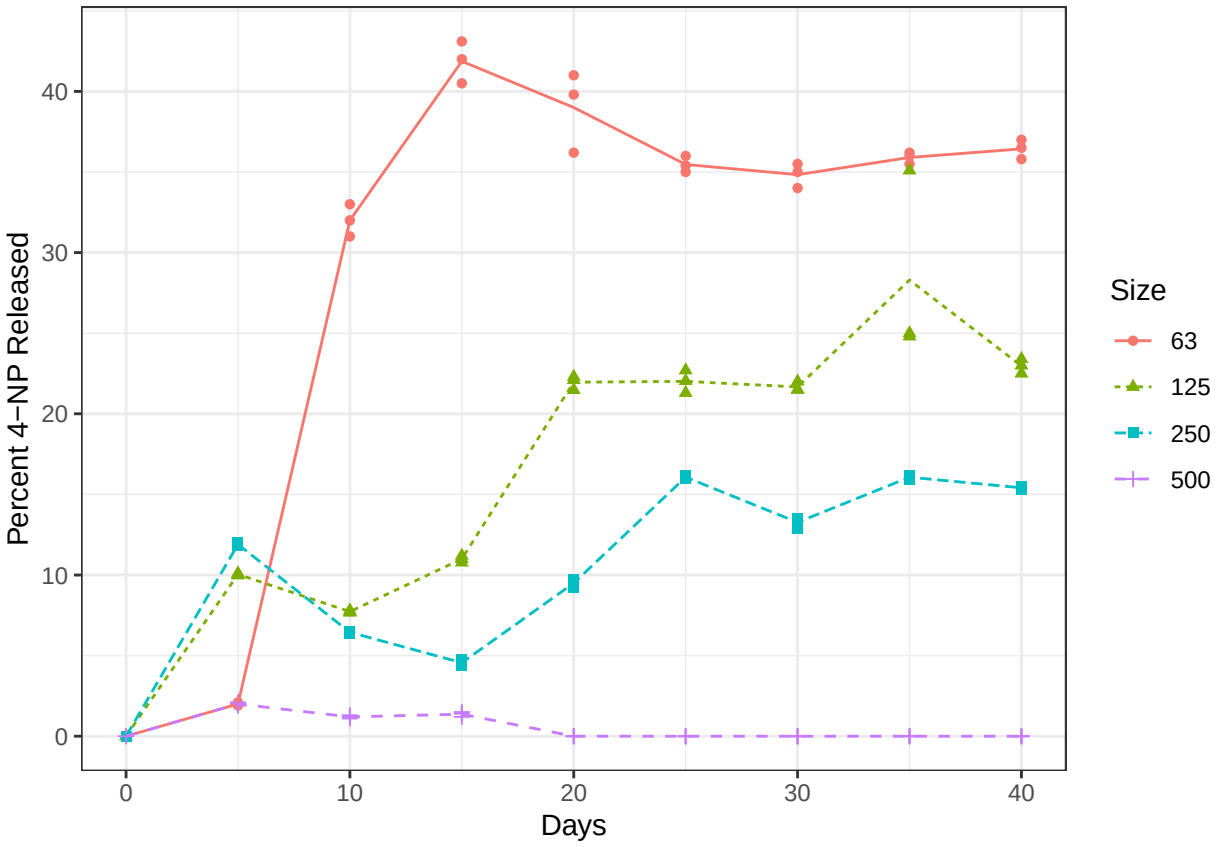
```
## # A tibble: 6 x 11
##    Days Mass_Dosed Dust_Massed_Dos~ Conc_On_Dust Mass_Of_Dust_Pe~
##   <dbl>      <dbl>            <dbl>        <dbl>            <dbl>
## 1     0       NA               NA          NA                NA
## 2     5       18.0              5.5         3.27              1.1
## 3     5       15.6              5.6         2.79              0.8
## 4     5       22.3              7           3.19              1.2
## 5    10       18.0              5.5         3.27              1.2
```

```
## 6    10       15.6              5.6         2.79              0.9
## # ... with 6 more variables: FourNP_In_Trial <dbl>, Mass_Released <dbl>,
## #   Perc_Released <dbl>, Size <dbl>, Sample <fct>, Prop_Released <dbl>
```

```
table(dustsize$Days, dustsize$Size)
```

```
##
##      63 125 250 500
##   0   1   1   1   1
##   5   3   3   3   3
##   10  3   3   3   3
##   15  3   3   3   3
##   20  3   3   3   3
##   25  3   3   3   3
##   30  3   3   3   3
##   35  3   3   3   3
##   40  3   3   3   3
```

```
ggplot(dustsize, aes(x=Days, y=Perc_Released, group=factor(Size),
                     colour=factor(Size), shape=factor(Size),
                     linetype=factor(Size))) +
       geom_point() +
       stat_summary(fun="mean", geom="line") +
       #stat_summary(fun="mean", geom="point", shape = "-", size=10) +
       theme_bw() +
       labs(shape = "Size", linetype = "Size", colour = "Size") +
       ylab("Percent 4-NP Released")
```



```
  head(dustsize)
```

```
## # A tibble: 6 x 11
##    Days Mass_Dosed Dust_Massed_Dos~ Conc_On_Dust Mass_Of_Dust_Pe~
##   <dbl>      <dbl>            <dbl>        <dbl>            <dbl>
## 1     0       NA               NA          NA                NA
## 2     5       18.0              5.5         3.27              1.1
## 3     5       15.6              5.6         2.79              0.8
## 4     5       22.3              7           3.19              1.2
## 5    10       18.0              5.5         3.27              1.2
## 6    10       15.6              5.6         2.79              0.9
## # ... with 6 more variables: FourNP_In_Trial <dbl>, Mass_Released <dbl>,
## #   Perc_Released <dbl>, Size <dbl>, Sample <fct>, Prop_Released <dbl>
```

```
  dim(dustsize)
```

```
## [1] 100  11
```