Explant Culture Analysis

Sean Nguyen

Overall Objective

Load Libraries

```
library(tidyverse)
library(cowplot)
library(broom)
library(plotly)
```

Import data

Convert data from 'wide' to 'long' format

```
# data
data1 <- data %>%
  gather(Sample,Count,2:37)

# Separate samples by identifiers
data2 <- data1 %>%
  separate(Sample, into=c("Sample_ID","Dilution_factor",
                          "Injection","Tech_rep", sep = "_")) %>%
  select(-`_`)

# Standards
standards1 <- standards %>%
  gather(Sample,Count,2:13)


standards2 <- standards1 %>%
  separate(Sample, into=c("Sample_ID","When","Dilution_factor",
                          "Nano_day","Injection","Tech_Rep", sep = "_")) %>%
  select(-`_`)
```

Factor the data into categorical variables

```
# Refactoring Columns for samples
data2$Sample_ID <- as.factor(data2$Sample_ID)
data2$Dilution_factor <- as.numeric(data2$Dilution_factor)
data2$Injection<- as.factor(data2$Injection)
data2$Tech_rep <- as.numeric(data2$Tech_rep)

data2
```

```
## # A tibble: 36,000 × 6
```

```
##    particle_size Sample_ID Dilution_factor Injection Tech_rep Count
## *          <dbl>    <fctr>           <dbl>    <fctr>    <dbl> <int>
## 1            0.5         1             125         1        0     0
## 2            1.5         1             125         1        0     0
## 3            2.5         1             125         1        0     0
## 4            3.5         1             125         1        0     0
## 5            4.5         1             125         1        0     0
## 6            5.5         1             125         1        0     0
## 7            6.5         1             125         1        0     0
## 8            7.5         1             125         1        0     0
## 9            8.5         1             125         1        0     0
## 10           9.5         1             125         1        0     0
## # ... with 35,990 more rows
```

```
# Refactoring COlumns for key
key$Sample_ID <- as.factor(key$Sample_ID)
key$Animal <- as.factor(key$Animal)
key$Condition <- as.factor(key$Condition)

key
```

```
## # A tibble: 6 × 3
##   Sample_ID Animal Condition
##      <fctr> <fctr>    <fctr>
## 1         1   1373    Normal
## 2         2   1371 lowOxygen
## 3         3   1370    Normal
## 4         4   1370 lowOxygen
## 5         5   1373 lowOxygen
## 6         6   1371    Normal
```

```
# Refactoring columns for standards
standards2$Sample_ID <- as.factor(standards2$Sample_ID)
standards2$When <- as.factor(standards2$When)
standards2$Dilution_factor <- as.numeric(standards2$Dilution_factor)
standards2$Injection <- as.factor(standards2$Injection)
standards2$Nano_day <- as.numeric(standards2$Nano_day)

standards2
```

```
## # A tibble: 12,000 × 8
##    particle_size Sample_ID   When Dilution_factor Nano_day Injection
## *          <dbl>    <fctr> <fctr>           <dbl>    <dbl>    <fctr>
## 1            0.5       std  after             125        1         1
## 2            1.5       std  after             125        1         1
## 3            2.5       std  after             125        1         1
## 4            3.5       std  after             125        1         1
## 5            4.5       std  after             125        1         1
## 6            5.5       std  after             125        1         1
## 7            6.5       std  after             125        1         1
## 8            7.5       std  after             125        1         1
## 9            8.5       std  after             125        1         1
## 10           9.5       std  after             125        1         1
## # ... with 11,990 more rows, and 2 more variables: Tech_Rep <chr>,
## #   Count <int>
```

Back calculate standards

```
standards2 <- standards2 %>%
  mutate(True_Count=Dilution_factor*Count)

# Set the correct order of 'categorical factors'
standards2$Nano_day <-  factor(standards2$Nano_day, levels=c('1'))
standards2$When <- factor(standards2$When, levels=c('before','after'))

standards2
```

```
## # A tibble: 12,000 × 9
##    particle_size Sample_ID   When Dilution_factor Nano_day Injection
##            <dbl>    <fctr> <fctr>           <dbl>   <fctr>    <fctr>
## 1            0.5       std  after             125        1         1
## 2            1.5       std  after             125        1         1
## 3            2.5       std  after             125        1         1
## 4            3.5       std  after             125        1         1
## 5            4.5       std  after             125        1         1
## 6            5.5       std  after             125        1         1
## 7            6.5       std  after             125        1         1
## 8            7.5       std  after             125        1         1
## 9            8.5       std  after             125        1         1
## 10           9.5       std  after             125        1         1
## # ... with 11,990 more rows, and 3 more variables: Tech_Rep <chr>,
## #   Count <int>, True_Count <dbl>
```

Summarize three technical standard replicates

```
standards3 <- standards2 %>%
  group_by(particle_size,Sample_ID,When,Dilution_factor,Nano_day,Injection) %>%
  summarise( tech_N = length(True_Count),
             tech_mean = mean(True_Count),
             tech_sd = sd(True_Count),
             tech_se = tech_sd/sqrt(tech_N))
standards3
```

```
## Source: local data frame [4,000 x 10]
## Groups: particle_size, Sample_ID, When, Dilution_factor, Nano_day [?]
##
##    particle_size Sample_ID   When Dilution_factor Nano_day Injection
##            <dbl>    <fctr> <fctr>           <dbl>   <fctr>    <fctr>
## 1            0.5       std before             125        1         1
## 2            0.5       std before             125        1         2
## 3            0.5       std  after             125        1         1
## 4            0.5       std  after             125        1         2
## 5            1.5       std before             125        1         1
## 6            1.5       std before             125        1         2
## 7            1.5       std  after             125        1         1
## 8            1.5       std  after             125        1         2
## 9            2.5       std before             125        1         1
## 10           2.5       std before             125        1         2
## # ... with 3,990 more rows, and 4 more variables: tech_N <int>,
```

```
## #   tech_mean <dbl>, tech_sd <dbl>, tech_se <dbl>
```

Summarize standards by injection

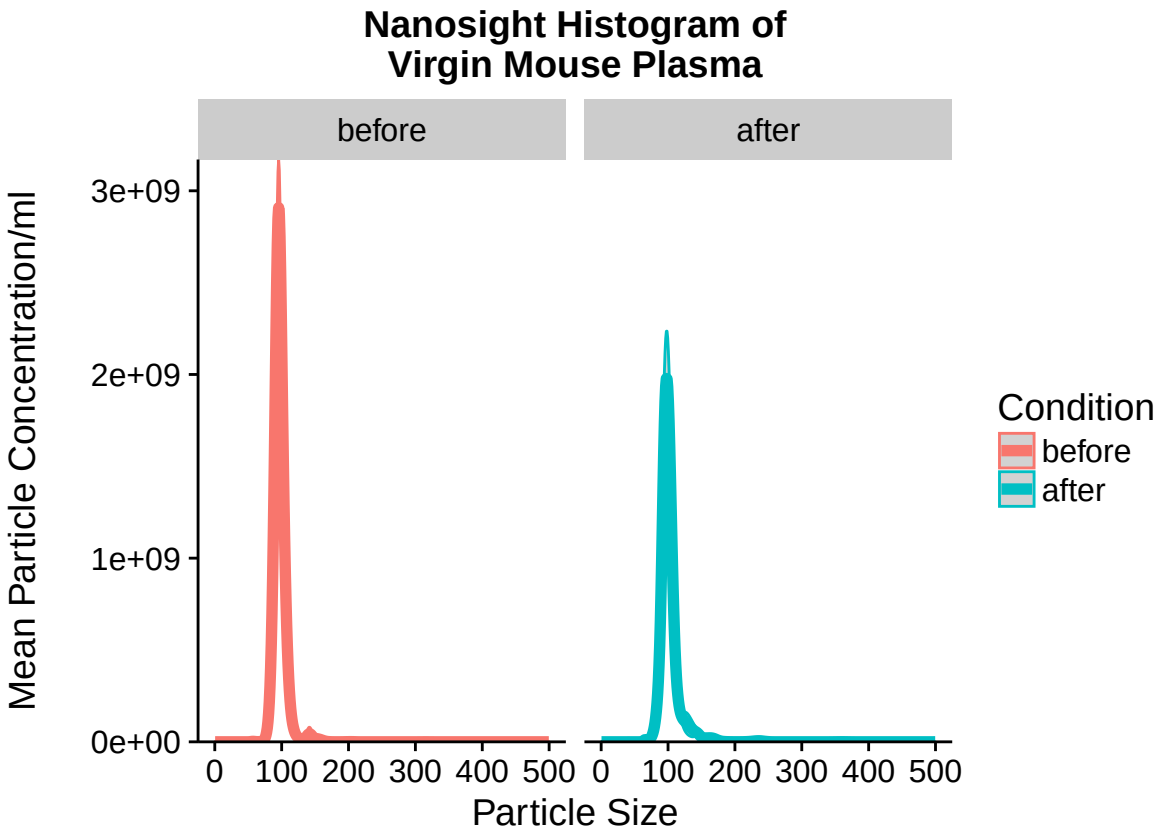
```
standards4 <- standards3 %>%
  group_by(Nano_day,When,particle_size) %>%
  summarise( inj_N = length(tech_mean),
             inj_mean = mean(tech_mean),
             inj_sd = sd(tech_mean),
             inj_se = inj_sd/sqrt(inj_N))
standards4
```

```
## Source: local data frame [2,000 x 7]
## Groups: Nano_day, When [?]
##
##    Nano_day   When particle_size inj_N inj_mean inj_sd inj_se
##      <fctr> <fctr>         <dbl> <int>    <dbl>  <dbl>  <dbl>
## 1         1 before           0.5     2        0      0      0
## 2         1 before           1.5     2        0      0      0
## 3         1 before           2.5     2        0      0      0
## 4         1 before           3.5     2        0      0      0
## 5         1 before           4.5     2        0      0      0
## 6         1 before           5.5     2        0      0      0
## 7         1 before           6.5     2        0      0      0
## 8         1 before           7.5     2        0      0      0
## 9         1 before           8.5     2        0      0      0
## 10        1 before           9.5     2        0      0      0
## # ... with 1,990 more rows
```

Plot before and after plots, facet by experimental day

```
std_plot <- standards4 %>%
  ggplot(aes(x=particle_size,y=inj_mean,color=When))+
  geom_ribbon(aes(ymin=inj_mean-inj_se, ymax=inj_mean+inj_se),
              alpha=0.2,fill = alpha('grey12', 0.2)) + #error bars
  geom_line(size=2) + xlim(0,500)+ #line size, x-axis scale
  scale_y_continuous(expand=c(0,0))+ #set bottom of graph
  xlab("Particle Size") + # X axis label
  ylab("\nMean Particle Concentration/ml\n") + # Y axis label
  ggtitle("Nanosight Histogram of\nVirgin Mouse Plasma")+ #title
  labs(color="Condition")+ #Label table title
  facet_grid(. ~ When)

std_plot
```

```
## Warning: Removed 1000 rows containing missing values (geom_path).
```



Standards particle concentrations from each experimental day

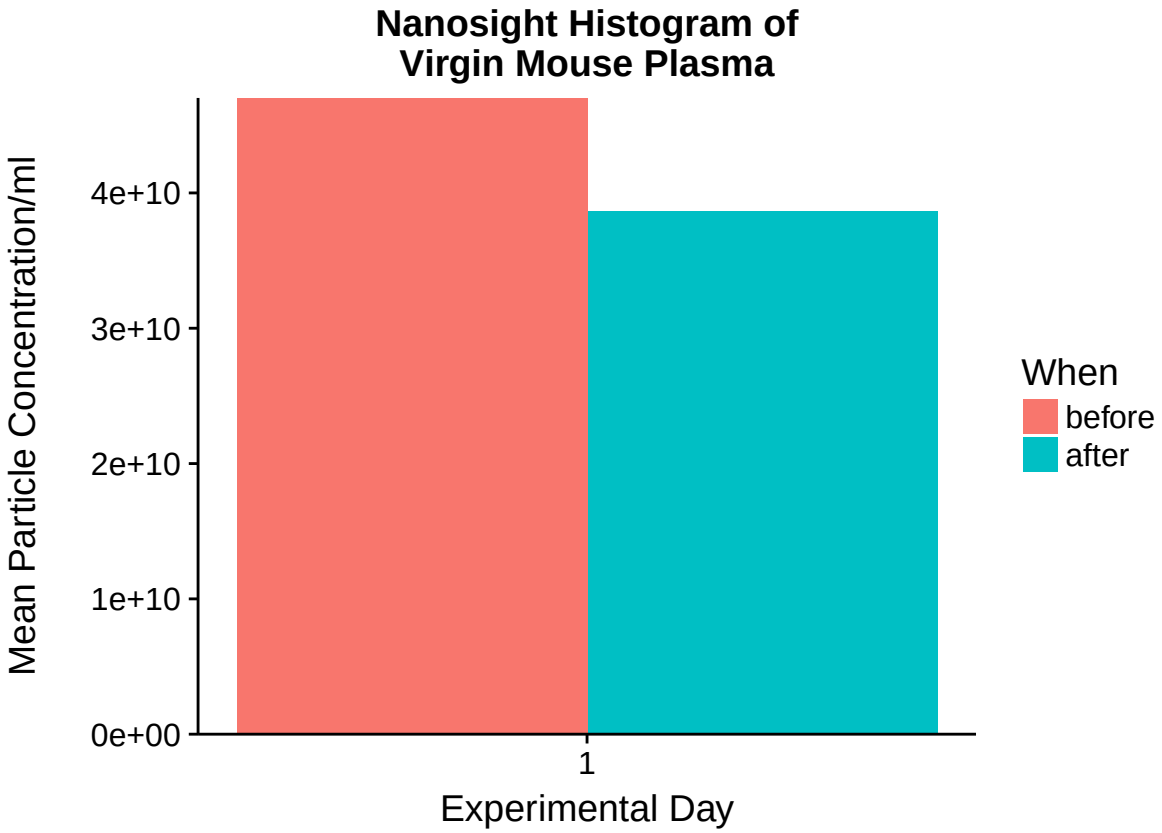
```
standards_df <- standards4 %>%
  group_by(Nano_day,When) %>%
  summarise(total=sum(inj_mean))

standards_df
```

```
## Source: local data frame [2 x 3]
## Groups: Nano_day [?]
##
##   Nano_day   When       total
##      <fctr> <fctr>       <dbl>
## 1        1 before 47020454229
## 2        1  after 38623334583
```

Bar graph of standards particle concentrations

```
standards_df %>%
  ggplot(aes(x=Nano_day,y=total,fill=When))+
  geom_col(position="dodge")+
  scale_y_continuous(expand=c(0,0))+ #set bottom of graph
  xlab("Experimental Day") + # X axis label
```

```
  ylab("\nMean Particle Concentration/ml\n") + # Y axis label
  ggtitle("Nanosight Histogram of\nVirgin Mouse Plasma")+ #title
  labs(color="When") #Label table title
```



Intraassay variability

```
Intra.assay_cv <- standards_df %>%
  group_by(Nano_day) %>%
  summarise(Day_N = length(total),
            Day_mean = mean(total),
            Day_sd = sd(total),
            Day_se = Day_sd/sqrt(Day_N),
           Day_cv = Day_sd/Day_mean )
Intra.assay_cv
```

```
## # A tibble: 1 × 6
##   Nano_day Day_N    Day_mean     Day_sd     Day_se    Day_cv
##     <fctr> <int>       <dbl>      <dbl>      <dbl>     <dbl>
## 1        1     2 42821894406 5937660244 4198559823 0.1386594
```

Sample analysis

Back calculate the original concentration of the sample

```
data2 <- data2 %>%
  mutate(True_Count = Dilution_factor*Count)
data2
```

```
## # A tibble: 36,000 × 7
##    particle_size Sample_ID Dilution_factor Injection Tech_rep Count
##            <dbl>    <fctr>           <dbl>    <fctr>    <dbl> <int>
## 1            0.5         1             125         1        0     0
## 2            1.5         1             125         1        0     0
## 3            2.5         1             125         1        0     0
## 4            3.5         1             125         1        0     0
## 5            4.5         1             125         1        0     0
## 6            5.5         1             125         1        0     0
## 7            6.5         1             125         1        0     0
## 8            7.5         1             125         1        0     0
## 9            8.5         1             125         1        0     0
## 10           9.5         1             125         1        0     0
## # ... with 35,990 more rows, and 1 more variables: True_Count <dbl>
```

Average three technical readings

```
data3 <- data2 %>%
  group_by(particle_size,Sample_ID,Dilution_factor,Injection) %>%
  summarise( tech_N = length(True_Count),
             tech_mean = mean(True_Count),
             tech_sd = sd(True_Count),
             tech_se = tech_sd/sqrt(tech_N))
data3
```

```
## Source: local data frame [12,000 x 8]
## Groups: particle_size, Sample_ID, Dilution_factor [?]
##
##    particle_size Sample_ID Dilution_factor Injection tech_N tech_mean
##            <dbl>    <fctr>           <dbl>    <fctr>  <int>     <dbl>
## 1            0.5         1             125         1      3         0
## 2            0.5         1             125         2      3         0
## 3            0.5         2             125         1      3         0
## 4            0.5         2             125         2      3         0
## 5            0.5         3             125         1      3         0
## 6            0.5         3             125         2      3         0
## 7            0.5         4             125         1      3         0
## 8            0.5         4             125         2      3         0
## 9            0.5         5             125         1      3         0
## 10           0.5         5             125         2      3         0
## # ... with 11,990 more rows, and 2 more variables: tech_sd <dbl>,
## #   tech_se <dbl>
```

Summarize samples by injection (average both injections)

```
data4 <- data3 %>%
  group_by(particle_size,Sample_ID,Dilution_factor) %>%
  summarise( inj_N = length(tech_mean),
             inj_mean = mean(tech_mean),
             inj_sd = sd(tech_mean),
             inj_se = inj_sd/sqrt(inj_N))
data4
```

```
## Source: local data frame [6,000 x 7]
## Groups: particle_size, Sample_ID [?]
##
##    particle_size Sample_ID Dilution_factor inj_N inj_mean inj_sd inj_se
##            <dbl>    <fctr>           <dbl> <int>    <dbl>  <dbl>  <dbl>
## 1            0.5         1             125     2        0      0      0
## 2            0.5         2             125     2        0      0      0
## 3            0.5         3             125     2        0      0      0
## 4            0.5         4             125     2        0      0      0
## 5            0.5         5             125     2        0      0      0
## 6            0.5         6             125     2        0      0      0
## 7            1.5         1             125     2        0      0      0
## 8            1.5         2             125     2        0      0      0
## 9            1.5         3             125     2        0      0      0
## 10           1.5         4             125     2        0      0      0
## # ... with 5,990 more rows
```

```
# Average technical replicates and merge with key
merge <- left_join(key,data3, by= "Sample_ID")

merge
```

```
## # A tibble: 12,000 × 10
##    Sample_ID Animal Condition particle_size Dilution_factor Injection
##       <fctr> <fctr>    <fctr>         <dbl>           <dbl>    <fctr>
## 1          1   1373    Normal           0.5             125         1
## 2          1   1373    Normal           0.5             125         2
## 3          1   1373    Normal           1.5             125         1
## 4          1   1373    Normal           1.5             125         2
## 5          1   1373    Normal           2.5             125         1
## 6          1   1373    Normal           2.5             125         2
## 7          1   1373    Normal           3.5             125         1
## 8          1   1373    Normal           3.5             125         2
## 9          1   1373    Normal           4.5             125         1
## 10         1   1373    Normal           4.5             125         2
## # ... with 11,990 more rows, and 4 more variables: tech_N <int>,
## #   tech_mean <dbl>, tech_sd <dbl>, tech_se <dbl>
```

```
# Average injection replicates and merge with key
merge1 <- left_join(key,data4, by= "Sample_ID")

merge1
```

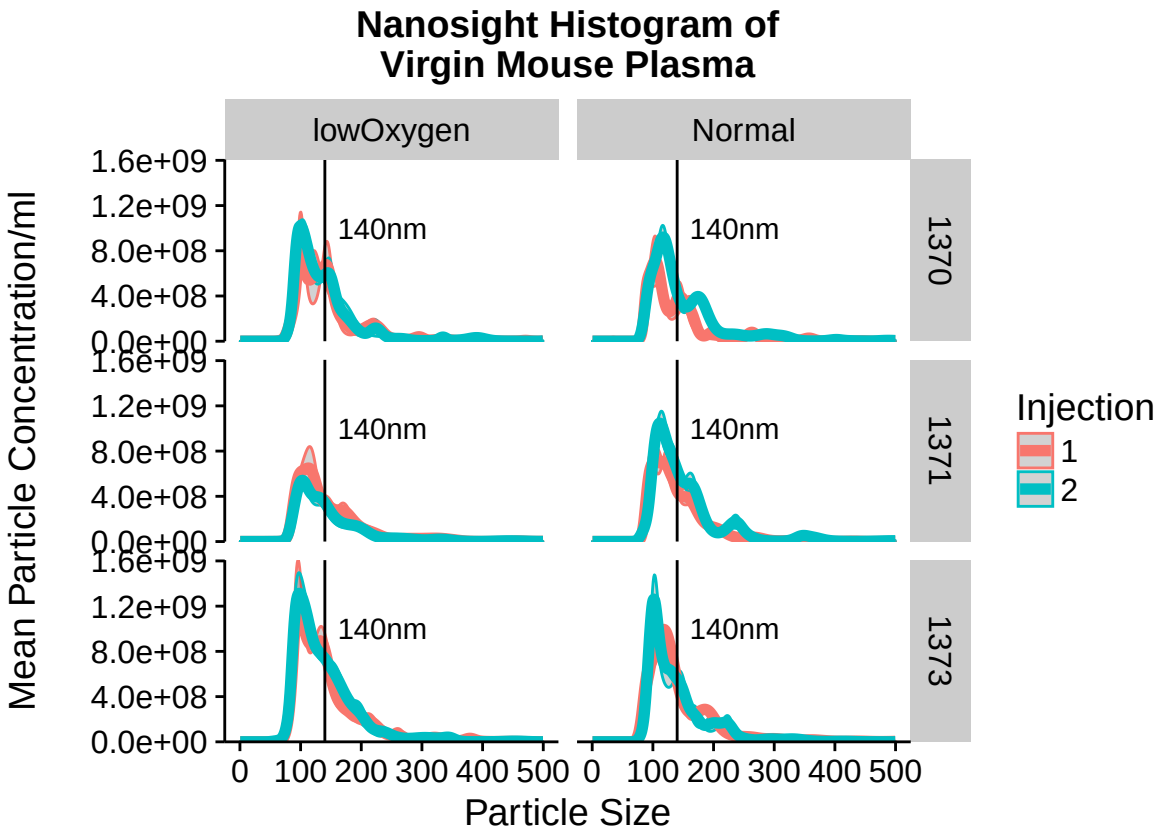
```
## # A tibble: 6,000 × 9
##    Sample_ID Animal Condition particle_size Dilution_factor inj_N inj_mean
```

```
##       <fctr> <fctr>    <fctr>         <dbl>           <dbl> <int>    <dbl>
## 1          1   1373    Normal           0.5             125     2        0
## 2          1   1373    Normal           1.5             125     2        0
## 3          1   1373    Normal           2.5             125     2        0
## 4          1   1373    Normal           3.5             125     2        0
## 5          1   1373    Normal           4.5             125     2        0
## 6          1   1373    Normal           5.5             125     2        0
## 7          1   1373    Normal           6.5             125     2        0
## 8          1   1373    Normal           7.5             125     2        0
## 9          1   1373    Normal           8.5             125     2        0
## 10         1   1373    Normal           9.5             125     2        0
## # ... with 5,990 more rows, and 2 more variables: inj_sd <dbl>,
## #   inj_se <dbl>
```

Quick visualizations

Graphing all samples

```
sample_plot <- merge %>%
  ggplot(aes(x=particle_size, y=tech_mean,color=Injection ))+ #plot
  geom_ribbon(aes(ymin=tech_mean-tech_se,
                  ymax=tech_mean+tech_se),
                  alpha=0.2,fill = alpha('grey12', 0.2)) + #error bars
  geom_line(size=2.0) + xlim(0,500)+ #line size, x-axis scale
  scale_y_continuous(expand=c(0,0))+ #set bottom of graph
  xlab("Particle Size") + # X axis label
  ylab("\nMean Particle Concentration/ml\n") + # Y axis label
  ggtitle("Nanosight Histogram of\nVirgin Mouse Plasma")+ #title
  labs(color="Injection")+ #Label table title
  facet_grid(Animal ~ Condition)+
  geom_vline(xintercept = 140)+
  annotate("text", x= 235, y = 1E9, label= "140nm")

sample_plot
```



Particle concentration values for each of the samples

```
merge2 <- merge1 %>%
  group_by(Animal,Condition) %>%
  summarise(particle_conc=sum(inj_mean))
merge2
```

```
## Source: local data frame [6 x 3]
## Groups: Animal [?]
##
##   Animal Condition particle_conc
##   <fctr>    <fctr>         <dbl>
## 1   1370 lowOxygen   62711510438
## 2   1370    Normal   54048424354
## 3   1371 lowOxygen   46143447083
## 4   1371    Normal   65833232625
## 5   1373 lowOxygen   93505022625
## 6   1373    Normal   70609587271
```

Summary statistics of particle concentration (averaging n=6 for each time point)

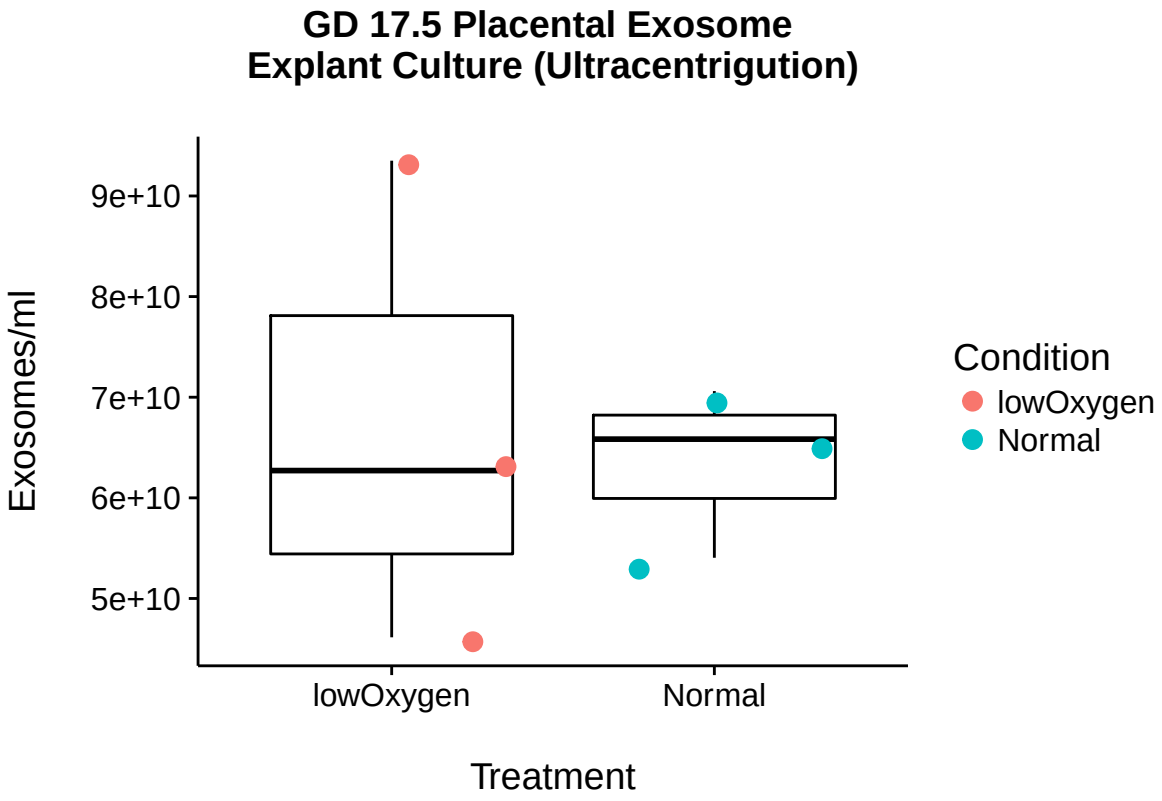
```
merge3 <- merge2 %>%
  group_by(Condition) %>%
```

```
  summarise(Condition_N=length(particle_conc),
            Condition_mean = mean(particle_conc),
            Condition_sd = sd(particle_conc),
            Condition_se = Condition_sd/sqrt(Condition_N))
merge3
```

```
## # A tibble: 2 × 5
##   Condition Condition_N Condition_mean Condition_sd Condition_se
##       <fctr>       <int>          <dbl>        <dbl>        <dbl>
## 1 lowOxygen           3    67453326715  24034211567  13876158518
## 2    Normal           3    63497081417   8524155743   4921423613
```

Boxplot

```
plot1 <- merge2 %>%
  #filter(!Animal== '1371'|!Condition == 'lowOxygen') %>%
  group_by(Condition) %>%
  ggplot(aes(x= Condition, y = particle_conc, color=Condition)) +
  geom_boxplot(colour="black",fill=NA) +
  geom_point(aes(text = paste("Animal:", Animal)),
             position='jitter',size=3)+
  xlab("\nTreatment\n") + # X axis label
  ylab("\nExosomes/ml\n") + # Y axis label
  ggtitle("GD 17.5 Placental Exosome \nExplant Culture (Ultracentrigution)\n")+ #title
  labs(color="Condition") # Label table title


plot1
```



##Interactive Plot

```
# ggplotly(plot1)
```

Statistics

```
fit <- t.test(particle_conc ~ Condition,data=merge2)

tidy(fit)
```

```
##     estimate   estimate1   estimate2 statistic   p.value parameter
## 1 3956245299 67453326715 63497081417  0.268711 0.8087912  2.495319
##       conf.low   conf.high                 method alternative
## 1 -48744278892 56656769489 Welch Two Sample t-test   two.sided
```