The Effect of Soil Moisture Content on the Growth and Hexadecane Remediation Capacity of *Pseudomonas putida*

Y3948024

15-04-2025

1 Experimental overview

This study investigated how moisture level in the growth media affects the proliferation of saprophytic bacterium *Pseudomonas putida* and its ability to degrade hexadecane, a common hydrocarbon pollutant. We aim to advise on efficacy of *P. putida* to clean oil spills in environments with different aridity and humidity, such as the 2024 Libya pipe burst (Rédaction Africanews, 2024) and the recent spills in the Ecuadorian Amazon (Shanna Hanbury, 2025).

Bacterial growth was quantified using colony-forming unit (CFU) counts from spot plates prepared via serial dilution of the soil suspended in $\frac{1}{4}$ strength Ringer solution (Ringer, 1883). 4 replicates were used per spot plate, with 3 plates per sample.

The identity of the bacteria was confirmed by a series of tests, including oxidase testing, light microscopy, culturing on selective media, and PCR followed by gel electrophoresis.

1.1 Description of data

Mean colony counts were normalised to CFU per gram of dry soil. Relevant datasets include:

- General data: `cfu_per_g_d_soil_allgroups.xlsx`
- Group 3 (spot replicate) data: `cfu_per_g_d_soil_group3.xlsx`
- Pre-incubation values: Sheet 4 in the general dataset
- Gel image (PCR): `data_gel_image`

Data files are stored under the directory: `/Y3948024/analysis/data`. This is required for the .rmd to knit successfully.

1.2 Analytical Overview

All data processing and visualisation were conducted in **R version 4.4.1 (2024-06-14) – "Race for Your Life"** via **RStudio**, using **Git** for version control on macOS Sequoia (v15.3.1).

1.2.1 Tests used:

- **Normality**: Shapiro–Wilk test (Shapiro & Wilk, 1965)
- **Variance homogeneity**: Levene's test (Levene, 1960)
- **Non-parametric comparison**: Scheirer–Ray–Hare test (Scheirer et al., 1976)
- **Post-hoc pairwise comparison** Dunn's test (Dunn, 1964)

1.2.2 Packages and Software

All packages were obtained from CRAN (Accessed 14 April 2025). Citations for each package are included inline using standard references.

- **tidyverse** (Wickham et al., 2019)
- **ggplot2** (Wickham, 2016)
- **FSA** (Ogle et al., 2024)
- **readxl** (Wickham and Bryan, 2023)
- **scales** (Wickham and Seidel, 2022)
- **vegan** (Oksanen et al., 2022)
- **rcompanion** (Mangiafico, 2024)
- **ggpubr** (Kassambara, 2023)
- **car** (Fox and Weisberg, 2019)

2 Load required libraries

```
library(tidyverse)      # data manipulation + wrangling (Wickham et al., 2019)
library(ggplot2)        # data visualisation (Wickham, 2016)
library(FSA)            # non-parametric testing (Ogle et al., 2024)
library(readxl)         # read Excel files (Wickham and Bryan, 2023)
library(scales)         # scale customisation for plots (Wickham and Seidel, 2022)
library(vegan)          # multivariate ecology tools (Oksanen et al., 2022)
library(rcompanion)     # statistics for extension work (Mangiafico, 2024)
library(ggpubr)         # ggplot2 enhancements (Kassambara, 2023)
library(car)            # regression diagnostics (Fox and Weisberg, 2019)
```

3 Statistical analysis and visualisation of data from plate replicates

3.1 Read in all Excel sheets and add a new column indicating the group each row came from

```
path <- 'data/cfu_per_g_d_soil_allgroups.xlsx'
data_allgroups <- excel_sheets(path)
data_tidy_list <- lapply(data_allgroups, function(sheet) {
  data <- read_excel(path, sheet = sheet)  # read each sheet
  data$group <- sheet                      # tag with group label
  return(data)
})
```

3.2 Combine all data into a single data frame and coerce variables to factors

```r
data_tidy <- do.call(rbind, data_tidy_list)
data_tidy$plate <- as.factor(data_tidy$plate)
data_tidy$Contamination <- as.factor(data_tidy$Contamination)
data_tidy$group <- as.factor(data_tidy$group)
```

3.3 Summarise data

Calculate means, standard deviation, sample size and standard error

```r
# Exclude pre-incubation data (group 4)
data_summary <- data_tidy %>%
  filter(group != "4") %>%
  group_by(plate, Contamination) %>%
  summarise(mean = mean(cfu_count),
            sd = sd(cfu_count),
            n = n(),
            se = sd / sqrt(n))
```

```r
data_summary
```

```
## # A tibble: 8 x 6
## # Groups:   plate [4]
##   plate Contamination             mean          sd     n          se
##   <fct> <fct>                    <dbl>       <dbl> <int>       <dbl>
## 1 30    With hexadecane     352887896.  357877255.     3  206620530.
## 2 30    Without hexadecane   48226222.   43478923.     3   25102568.
## 3 40    With hexadecane    2645521618. 4348823641.     3 2510794500.
## 4 40    Without hexadecane   86096111.   42394654.     3   24476565.
## 5 50    With hexadecane    1224751382. 1547511269.     3  893456048.
## 6 50    Without hexadecane  338414694.  368077310.     3  212509534.
## 7 60    With hexadecane     505089743.  661967720.     3  382187241.
## 8 60    Without hexadecane  388161944.  495927219.     3  286323713.
```

3.4 Statistical tests

3.4.1 Fit a linear model on CFU count by moisture level and contamination status

```r
mod <- lm(cfu_count ~ plate + Contamination, data = data_tidy %>% filter(group != "4"))
```

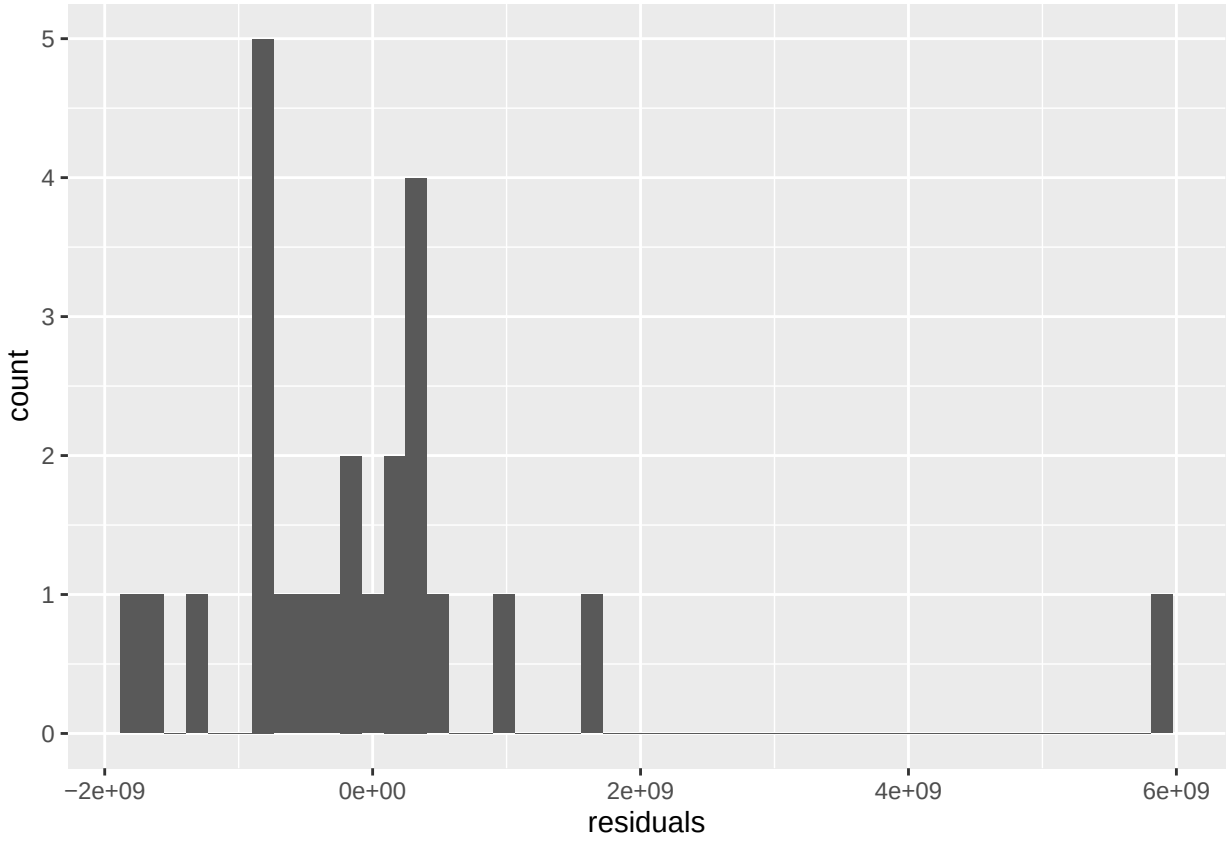
3.4.2 Use the Freedman-Diaconis rule to determine optimal bin width for this histogram

```r
iqr <- IQR(data_tidy$cfu_count)
n <- nrow(data_tidy)
bin_width <- 2 * iqr / (n^(1/3))
```

Bin width = 237686445

3.4.3 Visualise residual distribution to check normality of residuals

```
resid_df <- tibble(residuals = mod$residuals)
ggplot(resid_df, aes(x = residuals)) +
  geom_histogram(binwidth = bin_width)
```



The data appears to have a positive skew, with a long tail on the right. This suggests that the data may not be normally distributed.

3.4.4 Shapiro-Wilk test for normality of residuals

```
shapiro.test(mod$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mod$residuals
## W = 0.73228, p-value = 2.775e-05
```

p = 2.775e-05 < 0.05. Therefore there is evidence to suggest that the residuals are not normally distributed.

3.4.5 Levene's test for homogeneity of variance

```
data_filtered <- data_tidy %>% filter(group != "4")
```

3.4.5.1 Filter out group 4 (pre-incubation) for homogeneity testing

```
levmod <- lm(cfu_count ~ plate + Contamination, data = data_filtered)
leveneTest(residuals(levmod) ~ data_filtered$group)
```

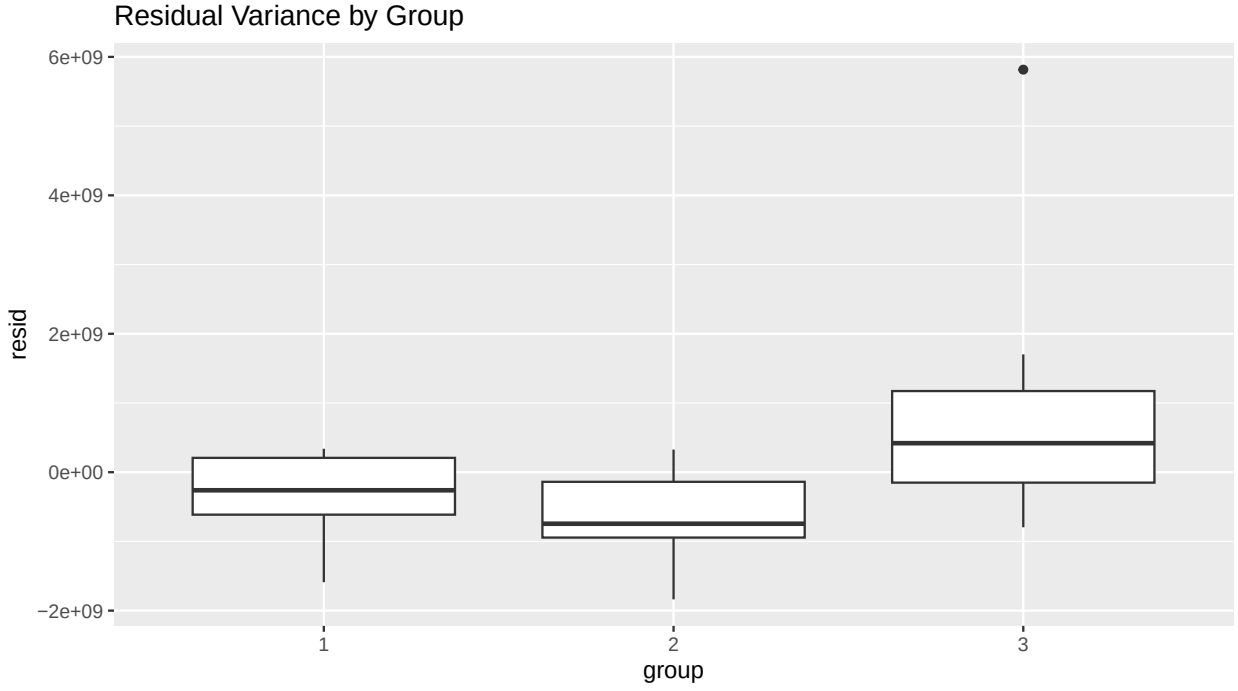
3.4.5.2 Fit linear model and conduct Levene's test

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  2  1.3126 0.2903
##       21
```

$p = 0.2903 > 0.05$. Therefore, there is not enough evidence to suggest that the variances are not homogenous across groups.

```
data_filtered$resid <- residuals(levmod)
ggplot(data_filtered, aes(x = group, y = resid)) +
  geom_boxplot() +
  labs(title = "Residual Variance by Group")
```

3.4.5.3 Plot boxplot of residuals grouped by replicate group to visualise the spread of variance



3.5 Testing for differences in mean CFU count between soil moisture levels and contamination status

3.5.1 We will perform a non-parametric test, eg. the Scheirer-Ray-Hare test, to observe overall significance

```
scheirerRayHare(cfu_count ~ plate * Contamination, data = filter(data_tidy, group != 4))
```

```
##
## DV:  cfu_count
## Observations:  24
## D:  1
## MS total:  50

##                      Df Sum Sq      H p.value
## plate                 3  98.33 1.9667 0.57935
## Contamination         1  80.67 1.6133 0.20402
## plate:Contamination   3  52.33 1.0467 0.78996
## Residuals            16 918.67
```

p(Plate:Contamination) = 0.78996 > 0.05, therefore there is not enough evidence to suggest that soil moisture content affects the growth of P. putida, using data from all groups.

3.5.2 Post-hoc pairwise testing

3.5.2.1 Post-hoc for Contamination

```
dunnTest(cfu_count ~ Contamination, data = filter(data_tidy, group != 4), method =
 ↪  "bonferroni")
```

3.5.2.2

```
## Dunn (1964) Kruskal-Wallis multiple comparison

##   p-values adjusted with the Bonferroni method.

##                              Comparison        Z   P.unadj     P.adj
## 1 With hexadecane - Without hexadecane 1.270171 0.2040239 0.2040239
```

p.adj = 0.2040239 > 0.05, therefore there is not enough evidence to suggest that hexadecane presence does affect growth of P. putida.

3.5.2.3 Post-hoc for Plate

3.5.2.3.1 Contamination = With hexadecane

```
dunnTest(cfu_count ~ plate, data = filter(data_tidy, Contamination == "With hexadecane",
 ↪  group != 4), method = "bh")
```

3.5.2.3.2

```
## Dunn (1964) Kruskal-Wallis multiple comparison

##   p-values adjusted with the Benjamini-Hochberg method.

##   Comparison          Z P.unadj P.adj
## 1    30 - 40 -0.1132277 0.90985     1
## 2    30 - 50 -0.1132277 0.90985     1
## 3    40 - 50  0.0000000 1.00000     1
## 4    30 - 60  0.0000000 1.00000     1
## 5    40 - 60  0.1132277 0.90985     1
## 6    50 - 60  0.1132277 0.90985     1
```

3.5.2.3.3 Contamination = Without hexadecane

```
dunnTest(cfu_count ~ plate, data = filter(data_tidy, Contamination == "Without
 ↪  hexadecane", group != 4), method = "bh")
```

3.5.2.3.4

```
## Dunn (1964) Kruskal-Wallis multiple comparison

##   p-values adjusted with the Benjamini-Hochberg method.

##   Comparison          Z    P.unadj     P.adj
## 1    30 - 40 -0.5661385 0.57129962 0.6855595
## 2    30 - 50 -1.6984156 0.08942936 0.5365762
## 3    40 - 50 -1.1322770 0.25751798 0.5150360
## 4    30 - 60 -1.5851878 0.11292366 0.3387710
## 5    40 - 60 -1.0190493 0.30817955 0.4622693
## 6    50 - 60  0.1132277 0.90985003 0.9098500
```

With hexadecane: No significant differences. Without hexadecane: No significant differences.

Therefore there is not enough evidence to suggest that soil moisture content affects the growth of P. putida, using meaned data from all 3 groups.

3.6 Visualise group-level CFU count with mean, SE, and LOESS lines

3.6.1 Design a custom theme for the plot

```
theme_custom <- theme(
  panel.spacing.x = unit(1, "lines"),
  legend.position = 'right',
  panel.grid.major.y = element_line(colour = "#e3e1e1", linetype = 1),
  panel.grid.major.x = element_line(colour = "#e3e1e1", linetype = 1),
  panel.grid.minor.y = element_line(colour = "#e3e1e1", linetype = 1),
  axis.text.x = element_text(hjust = 0.5),
  plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
  plot.margin = margin(10, 10, 10, 10),
  plot.subtitle = element_text(vjust = -250, hjust = 1)
)
```

3.6.2 Create plot

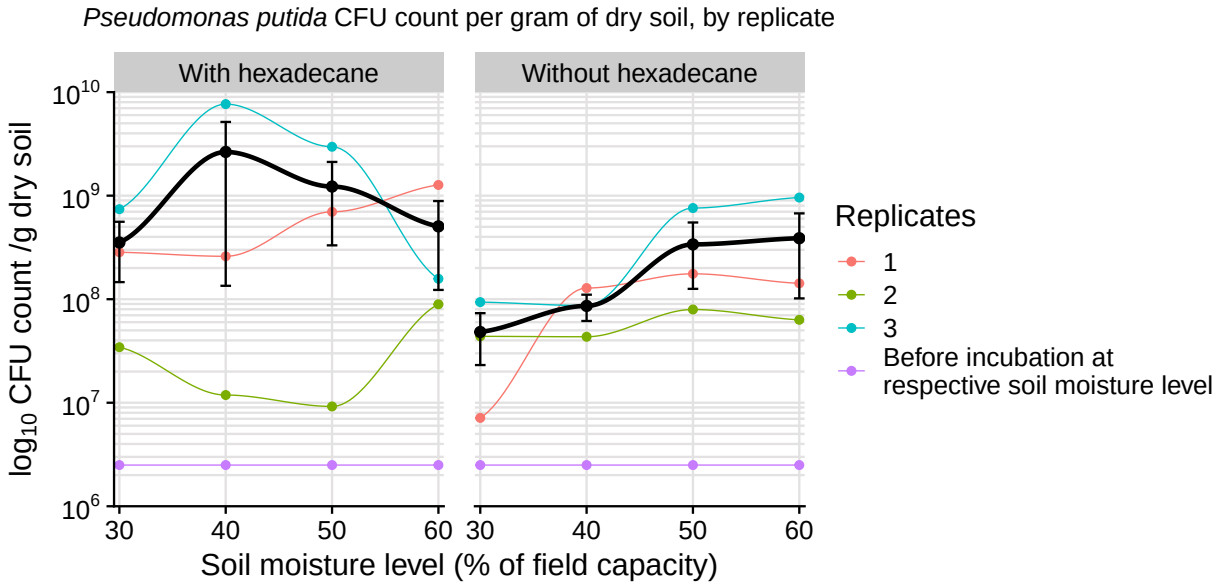
```
mean_CFU_count <- ggplot(data = data_tidy) +

  #### Plotting replicate data points
  geom_point(aes(x = factor(plate),
                 y = cfu_count,
                 colour = group)) +

  #### Plotting LOESS regression lines for replicates from data_tidy
  geom_smooth(aes(x = factor(plate),
                  y = cfu_count,
                  colour = group,
                  group = interaction(group, Contamination)),
              method = "loess",
              se = FALSE,
              linewidth = 0.3) +

  #### Plotting means from data_summary
  geom_point(data = data_summary,
             aes(x = factor(plate),
                 y = mean),
             colour = "black",
             size = 2) +

  #### Plotting error bars for means from data_summary
  geom_errorbar(data = data_summary,
                aes(x = factor(plate),
                    ymin = mean - se,
                    ymax = mean + se),
                width = 0.1,
                colour = "black") +

  #### Plotting LOESS regression lines for the means
  geom_smooth(data = data_summary,
              aes(x = factor(plate),
                  y = mean,
                  group = Contamination),
              method = "loess",
              colour = "black",
```

```
              se = FALSE,
              linewidth = 1) +

  #### Adding labels and changing axis
  labs(title = expression(italic("Pseudomonas putida")~"\nCFU count per gram of dry soil,
   ↪  by replicate"),
       x = expression("Soil moisture level (% of field capacity)"),
       y = expression("log"[10]~"CFU count /g dry soil"),
       colour = "Replicates") +

  #### Faceting by Contamination
  facet_grid(~ Contamination) +

  #### Adjusting the axes and legend
  scale_x_discrete(labels = c("30", "40", "50", "60"),
                   expand = c(0, 0)
  ) +
  scale_y_log10(
    labels = trans_format("log10", math_format(10^.x)),
    expand = c(0, 0),
    limits = c(10^6, 10^10),
    breaks = c(10^6, 10^7, 10^8, 10^9, 10^10),
    minor_breaks = rep(1:9, each = 1) * 10 ^ rep(6:9, times = 9)) +
  scale_colour_manual(values = c('1' = '#F8766D', '2' = '#7CAE00', '3' = '#00BFC4', '4' =
   ↪  '#C77CFF'),
                      labels = c('1', '2', '3',
                                 'Before incubation at \nrespective soil moisture
                                  ↪  level')) +

  #### Theming
  cowplot::theme_cowplot() +
  theme_custom
theme(axis.text.x = element_text(angle = 0, hjust = 1))
```

```
## List of 1
##  $ axis.text.x:List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : num 0
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

```
mean_CFU_count
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



4 Statistical analysis and visualisation of data from spot replicates

4.1 Read in group 3 data from Excel file

```
group3 <- read_excel('data/cfu_per_g_d_soil_group3.xlsx')
```

4.2 Convert relevant columns to factors for analysis

```
# Convert relevant columns to factors for analysis
group3$Plate <- as.factor(group3$Plate)
group3$Contamination <- as.factor(group3$Contamination)
group3$spot <- as.factor(group3$spot)
```

4.3 Summarise data

Calculate means, standard deviation, sample size and standard error

```
data_summary2 <- group3 %>%
  group_by(Plate, Contamination) %>%
  summarise(mean2 = mean(cfu_count),
```

```
            sd2 = sd(cfu_count),
            n2 = n(),
            se2 = sd2 / sqrt(n2))
```

```
## `summarise()` has grouped output by 'Plate'. You can override using the
## `.groups` argument.
```

4.4 Statistical tests

4.4.1 Fit a linear model on CFU count by moisture level and contamination status

```
mod2 <- lm(cfu_count ~ Plate + Contamination, data = group3)
```

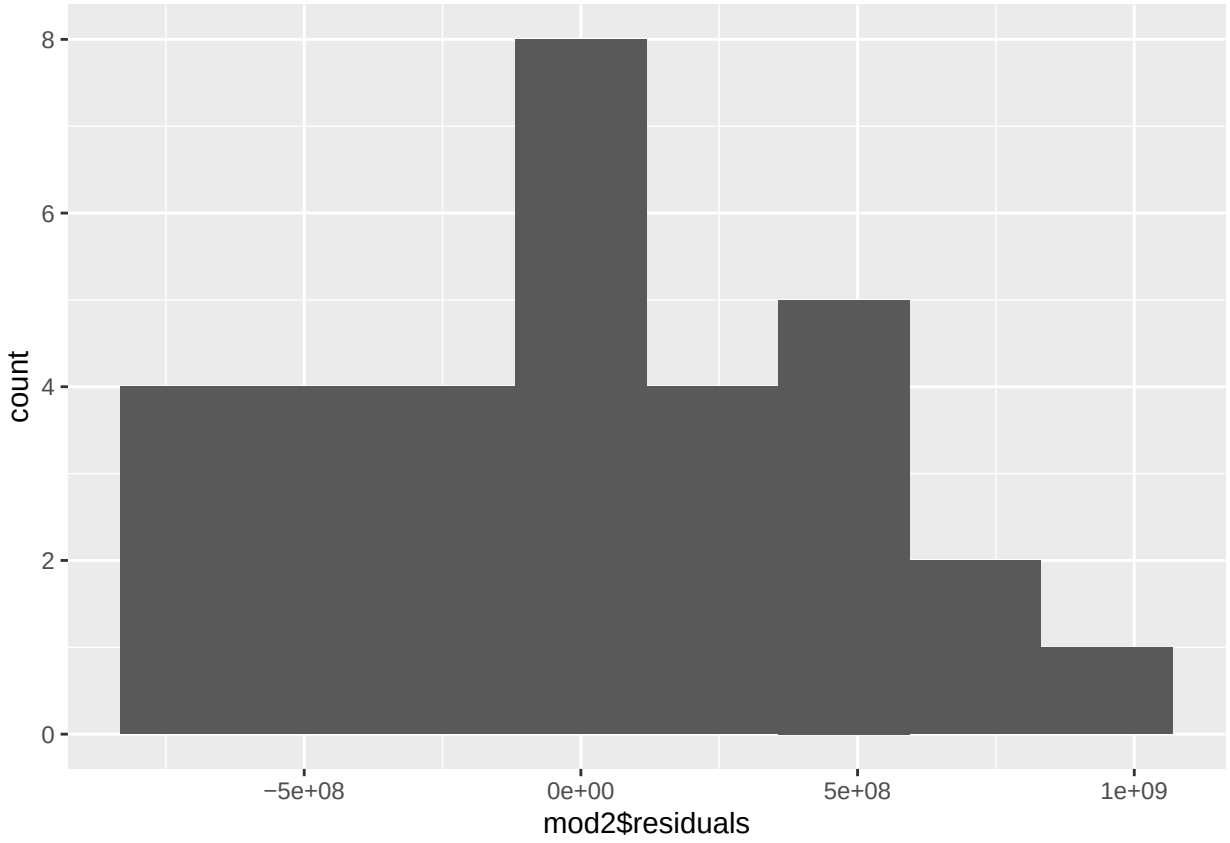
4.4.2 Use the Freedman-Diaconis rule to determine optimal bin width for histogram

```
iqr2 <- IQR(group3$cfu_count)
n2 <- nrow(group3)
bin_width2 <- 2 * iqr2 / (n2^(1/3))
```

Bin width = 237686445

4.4.3 Visualise residual distribution to check normality of residuals

```
ggplot(group3, aes(x = mod2$residuals)) +
  geom_histogram(binwidth = bin_width2)
```



4.4.4 Shapiro-Wilk test for normality of residuals

```
shapiro.test(mod2$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mod2$residuals
## W = 0.96822, p-value = 0.4519
```

$p = 0.4519 > 0.05$, there is not evidence to suggest that the data is not normally distributed.

4.4.5 Levene's test for homogeneity of variance

```
levmod2 <- lm(cfu_count ~ Plate + Contamination, data = group3)
leveneTest(residuals(levmod2) ~ group3$spot)
```

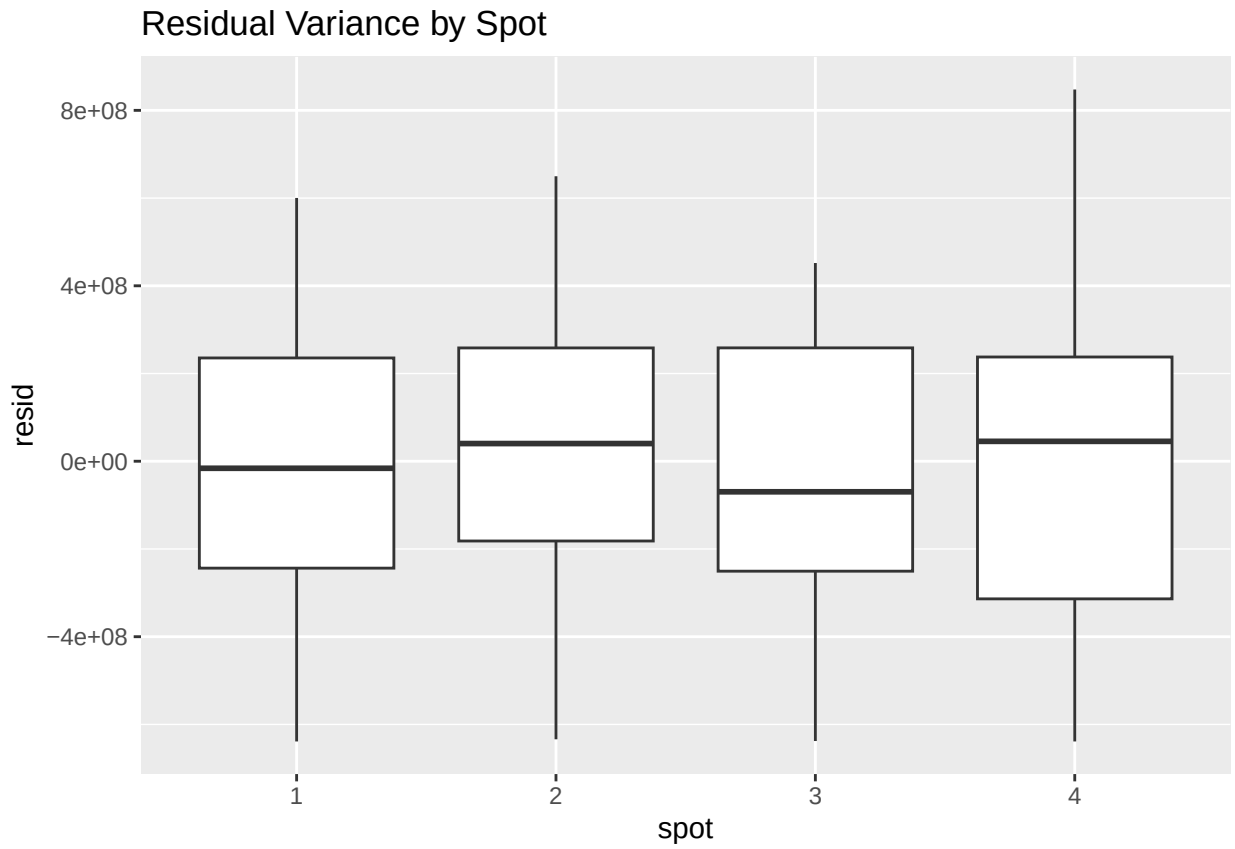
4.4.5.1 Fit linear model and conduct Levene's test

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  3  0.0223 0.9954
##       28
```

$p = 0.9954 > 0.05$, therefore there is not enough evidence to suggest that the data is not homogeneously distributed.

4.4.6 Plot boxplot of residuals grouped by spot to visualise the spread of variance

```
group3$resid <- residuals(levmod2)
ggplot(group3, aes(x = spot, y = resid)) +
  geom_boxplot() +
  labs(title = "Residual Variance by Spot")
```



4.5 Testing for differences in mean CFU count between soil moisture levels and contamination status

4.5.1 We will perform a non-parametric test, eg. the Scheirer-Ray-Hare test, to observe overall significance

```r
scheirerRayHare(cfu_count ~ Plate * Contamination, data = group3)
```

```
##
## DV:  cfu_count
## Observations:  32
## D:  0.9998167
## MS total:  88

##                      Df  Sum Sq       H  p.value
## Plate                 3  336.25  3.8217 0.281369
## Contamination         1  903.12 10.2647 0.001356
## Plate:Contamination   3 1328.13 15.0951 0.001737
## Residuals            24  160.00
```

p(Plate:Contamination) = 0.001737 < 0.05, therefore there is evidence to suggest that soil moisture content does affect growth of P. putida.

4.5.2 Post-hoc pairwise testing

4.5.2.1 Post-hoc for Contamination

```r
dunnTest(cfu_count ~ Contamination, data = group3, method = "bonferroni")
```

4.5.2.2

```
## Dunn (1964) Kruskal-Wallis multiple comparison

##   p-values adjusted with the Bonferroni method.

##                              Comparison        Z     P.unadj       P.adj
## 1 With hexadecane - Without hexadecane 3.203852 0.001356023 0.001356023
```

p.adj = 0.001356023 < 0.05, therefore there is evidence to suggest that hexadecane presence does affect growth of P. putida.

4.5.2.3 Post-hoc for Plate

4.5.2.3.1 Contamination = With hexadecane

```
dunnTest(cfu_count ~ Plate, data = filter(group3, Contamination == "With hexadecane"),
 ↪  method = "bh")
```

4.5.2.3.2

```
## Dunn (1964) Kruskal-Wallis multiple comparison

##   p-values adjusted with the Benjamini-Hochberg method.

##   Comparison         Z      P.unadj       P.adj
## 1    30 - 40 -2.376354 0.0174846744 0.034969349
## 2    30 - 50 -1.188177 0.2347636628 0.234763663
## 3    40 - 50  1.188177 0.2347636628 0.281716395
## 4    30 - 60  1.188177 0.2347636628 0.352145494
## 5    40 - 60  3.564531 0.0003645072 0.002187043
## 6    50 - 60  2.376354 0.0174846744 0.052454023
```

4.5.2.3.3 Contamination = Without hexadecane

```
dunnTest(cfu_count ~ Plate, data = filter(group3, Contamination == "Without hexadecane"),
 ↪  method = "bh")
```

4.5.2.3.4

```
## Dunn (1964) Kruskal-Wallis multiple comparison

##   p-values adjusted with the Benjamini-Hochberg method.

##   Comparison          Z     P.unadj      P.adj
## 1    30 - 40  0.2972629 0.766265789 0.76626579
## 2    30 - 50 -1.8578932 0.063184176 0.09477626
## 3    40 - 50 -2.1551562 0.031149616 0.06229923
## 4    30 - 60 -2.6010505 0.009293876 0.02788163
## 5    40 - 60 -2.8983135 0.003751754 0.02251052
## 6    50 - 60 -0.7431573 0.457386453 0.54886374
```

With hexadecane: Only significant difference between 30-40% and 40-60%, p.adj = 0.034969349 and 0.002187043, respectively. Without hexadecane: Only significant difference between 30-60% and 40-60%, p.adj = 0.02788163 and 0.02251052, respectively. Therefore there is evidence to suggest that soil moisture content does affect the growth of P. putida, using data from group 3. However, as p.adj (With hexadecane, 30-40%) $>$ 0.025 and the difference is positive, we cannot conclude that the difference between 30% and 40% is significant with a one-tailed test. The same is true for p.adj (Without hexadecane, 30-60%).

4.6 Visualise group 3 CFU count with group means, SE, smoothed lines, and statistical annotations

4.6.1 Preparing statistical annotations.

```
pairwise_p <- data.frame(
  group1 = c("30", "40", "30", "40"),
  group2 = c("40", "60", "60", "60"),
  y.position = c(2.05e9, 2.10e9, 300000000, 500000000),
  p.adj = c(0.034969349, 0.002187043, 0.02788163, 0.02251052),
  Contamination = c("With hexadecane", "With hexadecane",
                    "Without hexadecane", "Without hexadecane")
)

pairwise_p$label <- paste0("p.adj = ", signif(pairwise_p$p.adj, 3))
```

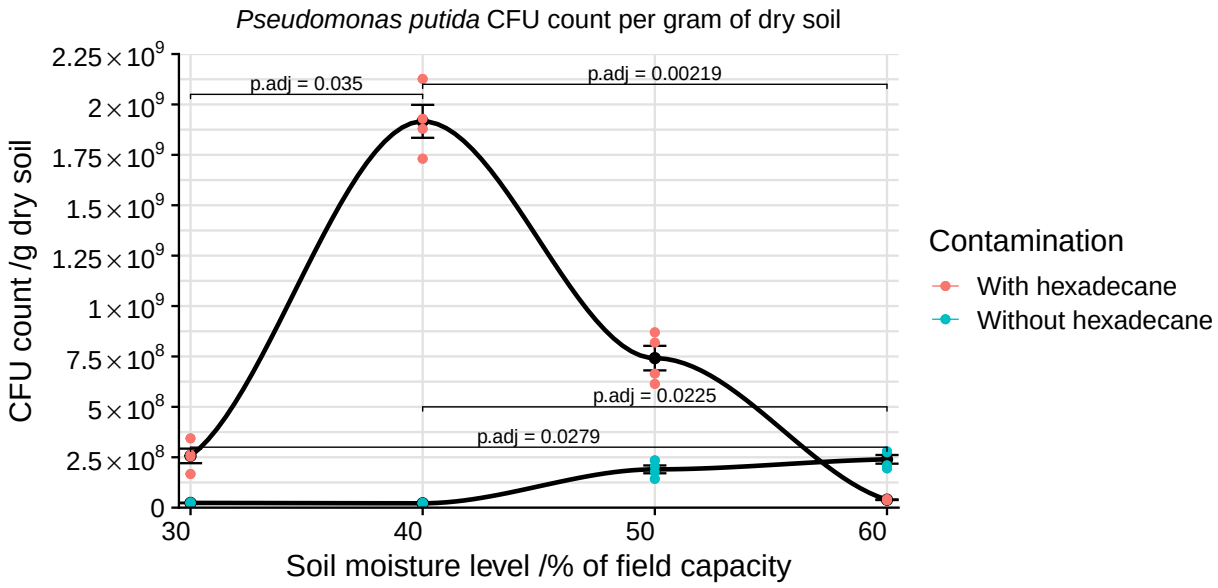
4.6.2 Create plot

```
mean_CFU_count2 <- ggplot() +

  #### Plotting data points.
  geom_point(data = group3,
             aes(x = factor(Plate),
                 y = cfu_count,
                 colour = Contamination)) +

  #### Plotting LOESS regression lines from data_tidy.
  geom_smooth(data = group3,
              aes(x = factor(Plate),
                  y = cfu_count,
                  colour = Contamination),
              method = "loess",
              se = FALSE,
              linewidth = 0.3) +

  #### Plotting means from data_summary.
  geom_point(data = data_summary2,
             aes(x = factor(Plate),
                 y = mean2),
             colour = 'black',
             size = 2) +

  #### Plotting error bars for means from data_summary.
  geom_errorbar(data = data_summary2,
                aes(x = factor(Plate),
                    ymin = mean2 - se2,
                    ymax = mean2 + se2),
                width = 0.1,
                colour = "black") +

  #### Plotting LOESS regression lines for the means.
```

```r
geom_smooth(data = data_summary2,
            aes(x = factor(Plate),
                y = mean2,
                group = Contamination),
            method = "loess",
            colour = "black",
            se = FALSE,
            linewidth = 1) +

#### Adding labels and changing axis.
labs(title = expression(italic("Pseudomonas putida")~"\nCFU count per gram of dry
 ↪  soil"),
     x = expression("Soil moisture level /% of field capacity"),
     y = expression("CFU count /g dry soil"),
     colour = "Contamination") +

#### Adjusting the axes and legend.
scale_x_discrete(labels = c("30", "40", "50", "60"),
                 expand = c(0, 0)
) +
scale_y_continuous(
  labels = function(x) {
    sapply(x, function(val) {
      if (val == 0) {
        "0"
      } else {
        formatted <- formatC(val / 10^floor(log10(val)), digits = 2, format = "f")
        exponent <- floor(log10(val))
        parse(text = paste0(formatted, " %*% 10^", exponent))
      }
    })
  },
  breaks = seq(0, 2.25e9, by = 2.5e8),
  limits = c(0, 2.25e9),
  expand = c(0, 0)
) +
scale_colour_manual(values = c('With hexadecane' = '#F8766D', 'Without hexadecane' =
 ↪  '#00BFC4', '4' = '#C77CFF'),
                    labels = c('With hexadecane', 'Without hexadecane',
                               'Before incubation at \nrespective soil moisture
                                ↪  level')) +
geom_point(data = group3,
           aes(x = factor(Plate),
               y = cfu_count,
               colour = Contamination)) +

#### Adding statistical annotations.
stat_pvalue_manual(pairwise_p,
                   label = "label",
                   xmin = "group1", xmax = "group2",
                   y.position = "y.position",
                   tip.length = 0.01,
                   size = 3,
                   bracket.size = 0.3) +
```

```
  #### Theming.
  cowplot::theme_cowplot() +
  theme_custom +
theme(axis.text.x = element_text(angle = 0, hjust = 1))

mean_CFU_count2
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



4.7 References


```
Dunn, O.J. (1964) 'Multiple comparisons using rank sums', *Technometrics*, 6(3), pp.
 ↪  241-252.
Fox, J. and Weisberg, S. (2019) *An R Companion to Applied Regression*. 3rd edn. Thousand
 ↪  Oaks, CA: Sage.
Kassambara, A. (2023) *ggpubr: 'ggplot2'-Based Publication Ready Plots*. R package
 ↪  version 0.6.0.
Levene, H., 1960. **Robust tests for equality of variances*. In: I. Olkin, ed. 1960.
 ↪  Contributions to probability and statistics: essays in honor of Harold Hotelling.
 ↪  Stanford: Stanford University Press, pp.278-292.
Mangiafico, S. (2024) *rcompanion: Functions to Support Extension Education Program
 ↪  Evaluation*. R package version 2.4.30.
Ogle, D.H. et al. (2024) *FSA: Fisheries Stock Analysis*. R package version 0.9.5.
Oksanen, J. et al. (2022) *vegan: Community Ecology Package*. R package version 2.6-4.
R Core Team (2024) *R: A Language and Environment for Statistical Computing*. Vienna: R
 ↪  Foundation for Statistical Computing.
Scheirer, C.J., Ray, W.S. and Hare, N. (1976) 'The analysis of ranked data derived from
 ↪  completely randomized factorial designs', *Biometrics*, 32(2), pp. 429-434.
```

Shapiro, S.S. and Wilk, M.B. (1965) 'An analysis of variance test for normality (complete samples)', *Biometrika*, 52(3/4), pp. 591-611.
Wickham, H. (2016) *ggplot2: Elegant Graphics for Data Analysis*. New York: Springer.
Wickham, H. et al. (2019) *Welcome to the tidyverse*. R package version 2.0.0.
Wickham, H. and Bryan, J. (2023) *readxl: Read Excel Files*. R package version 1.4.3.
Wickham, H. and Seidel, D. (2022) *scales: Scale Functions for Visualization*. R package version 1.3.0.