Statistical Analysis of Plastic Degradation: Effects of Substrate and Concentration Levels

###1. LOAD THE PACKAGES

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(ggplot2)
library(readxl)
```

###2. LOAD & CLEAN THE DATASET

```
#Load the dataset
data <- read_excel("Data for graph MMEAA .xlsx",
    sheet = "Test  run regression plt")
```

```
## New names:
## * `` -> `...6`
## * `` -> `...7`
## * `` -> `...8`
## * `` -> `...10`
```

```r
head(data)
```

```
## # A tibble: 6 x 10
##   `y= uM/min` sub_conc_level substrate enzyme_conc  temp ...6  ...7  ...8
##         <dbl>          <dbl> <chr>           <dbl> <dbl> <lgl> <lgl> <lgl>
## 1        233.            0.2 pNOB           0.0015    60 NA    NA    NA
## 2        227.            0.2 pNOB           0.0015    60 NA    NA    NA
## 3        230.            0.2 pNOB           0.0015    60 NA    NA    NA
## 4        225.            0.1 pNOB           0.0015    60 NA    NA    NA
## 5        199.            0.1 pNOB           0.0015    60 NA    NA    NA
## 6        227.            0.1 pNOB           0.0015    60 NA    NA    NA
## # i 2 more variables: `For Reference Only` <dbl>, ...10 <dbl>
```

```r
#Rename Y, select needed columns, turn into factor
clean_data <- data %>%
  rename(y = `y= uM/min`) %>%
  select(y, sub_conc_level, substrate) %>%
  mutate(sub_conc_level = as.factor(sub_conc_level),
         substrate = as.factor(substrate))

glimpse(clean_data)
```

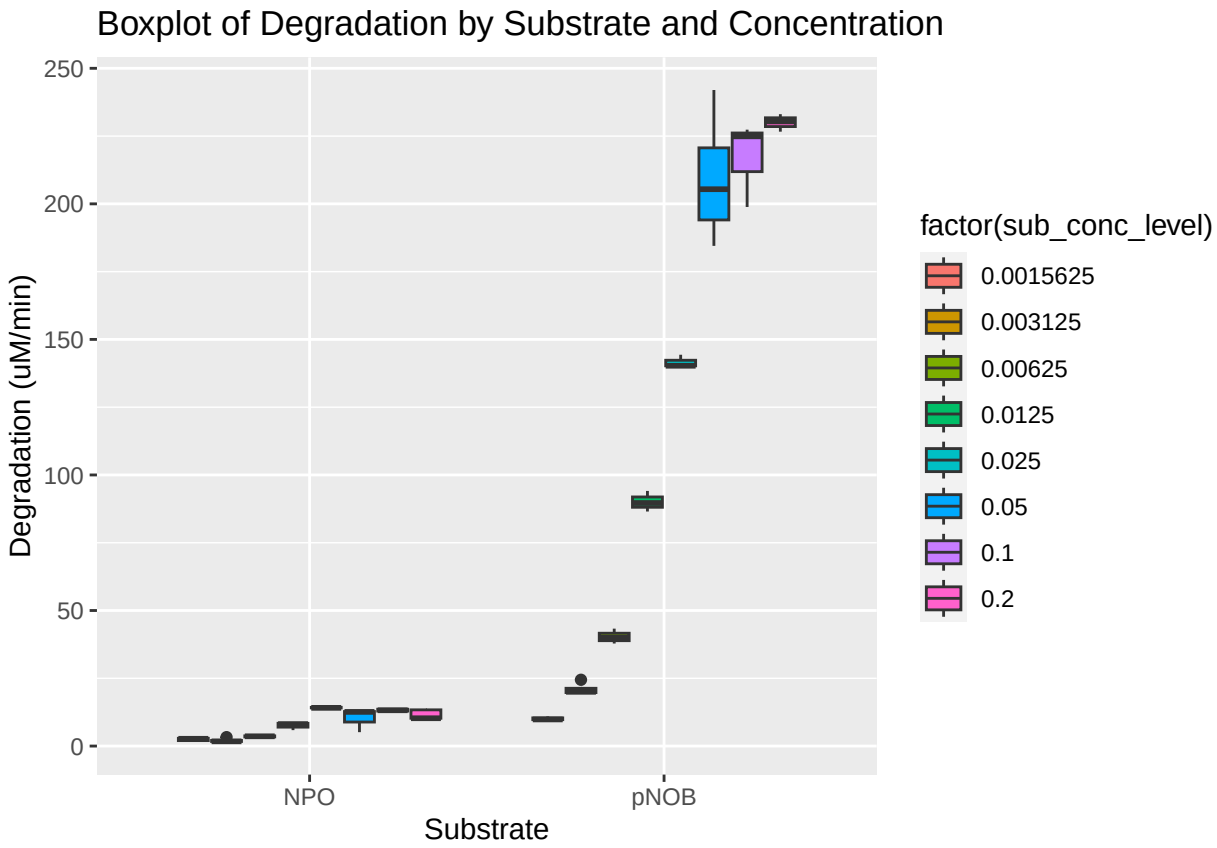
```
## Rows: 57
## Columns: 3
## $ y             <dbl> 233.085700, 226.628600, 230.400000, 224.914300, 198.857~
## $ sub_conc_level <fct> 0.2, 0.2, 0.2, 0.1, 0.1, 0.1, 0.05, 0.05, 0.05, 0.05, 0~
## $ substrate     <fct> pNOB, pNOB, pNOB, pNOB, pNOB, pNOB, pNOB, pNOB, pNOB, p~
```

```r
# Group the data by 'substrate' and calculate the mean of 'y' for each group
means_by_substrate <- clean_data %>%
  group_by(substrate) %>%
  summarize(mean_y = mean(y))
means_by_substrate
```

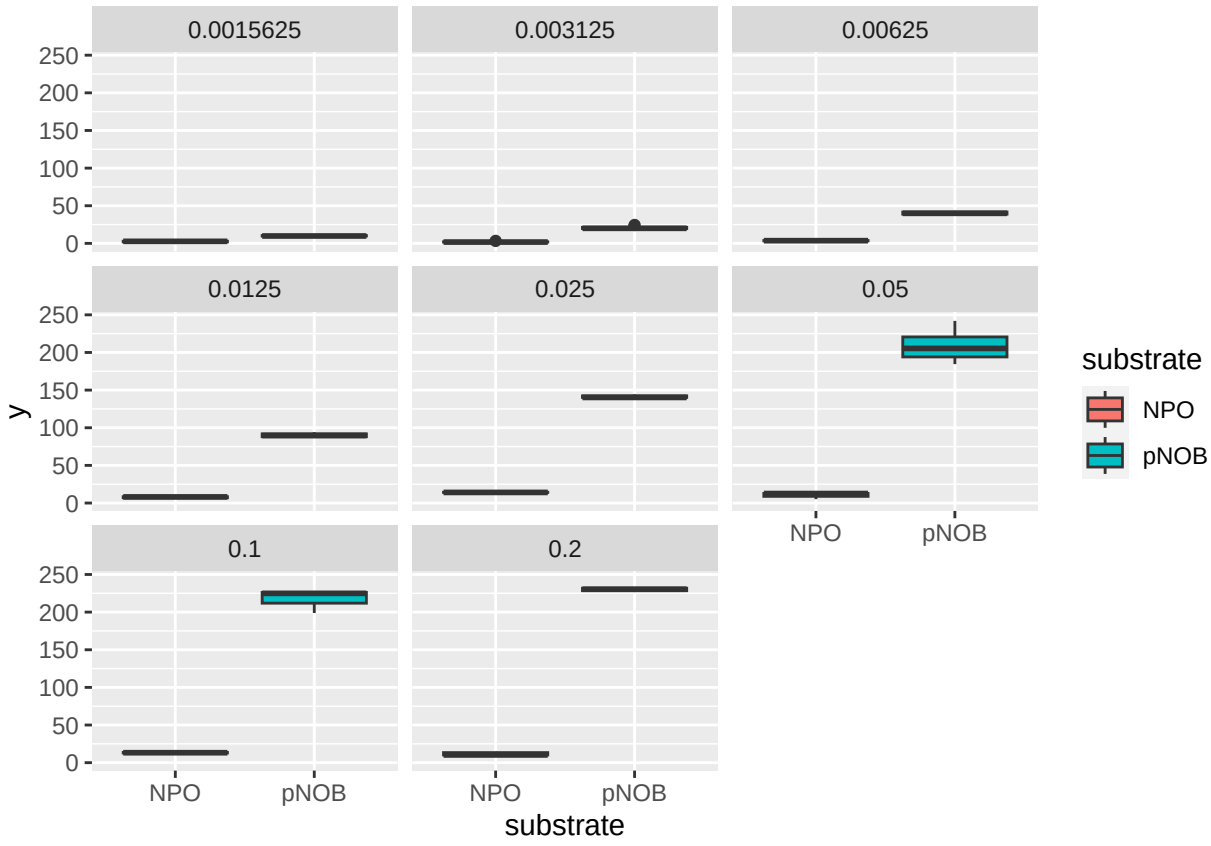
```
## # A tibble: 2 x 2
##   substrate mean_y
##   <fct>      <dbl>
## 1 NPO         7.42
## 2 pNOB      120.
```

###3. EDA (Exploratory Data Analysis)

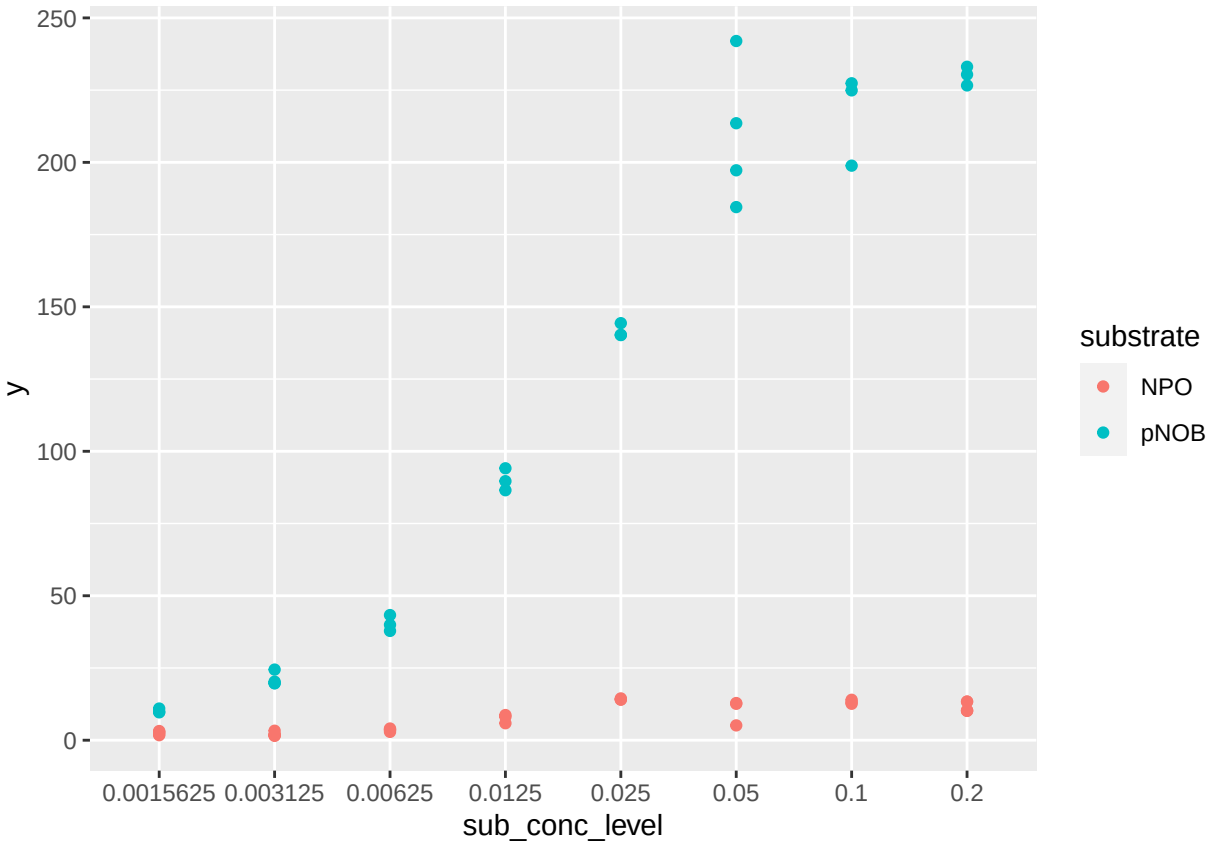
```r
# Boxplot for EDA
ggplot(clean_data, aes(x = substrate, y = y, fill = factor(sub_conc_level))) +
  geom_boxplot() +
  labs(x = "Substrate", y = "Degradation (uM/min)", title = "Boxplot of Degradation by Substrate and Co
```




```
# Create a boxplot for each group
ggplot(clean_data, aes(x = substrate, y = y, fill = substrate)) +
  geom_boxplot() +
  facet_wrap(~sub_conc_level)
```




```
#Dotplot
ggplot(clean_data, aes(x = sub_conc_level, y = y, color = substrate)) +
  geom_point()
```



###4. BALANCED VS UNBALANCED DESIGN

```
#Balanced vs Unbalanced Design?
table(clean_data$sub_conc_level, clean_data$substrate)
```

```
##
##              NPO pNOB
##   0.0015625    5    3
##   0.003125     5    4
##   0.00625      4    3
##   0.0125       3    3
##   0.025        3    3
##   0.05         3    4
##   0.1          3    3
##   0.2          5    3
```

Since we have unbalanced design (i.e. unequal numbers of subjects in each group). There are fundamentally 3 ways to run ANOVA in an unbalanced design (Type I, II, or III). The recommended method is Type III.

###5. TWOWAY ANOVA

```
#Anova
anova_result <- aov(y ~ substrate * sub_conc_level, data = clean_data)
Anova(anova_result, type = "III")
```

```
## Anova Table (Type III tests)
##
```

```
## Response: y
##                            Sum Sq Df  F value Pr(>F)
## (Intercept)                    31  1   0.5111 0.4787
## substrate                     110  1   1.7983 0.1873
## sub_conc_level                671  7   1.5728 0.1709
## substrate:sub_conc_level    97628  7 228.7802 <2e-16 ***
## Residuals                    2499 41
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

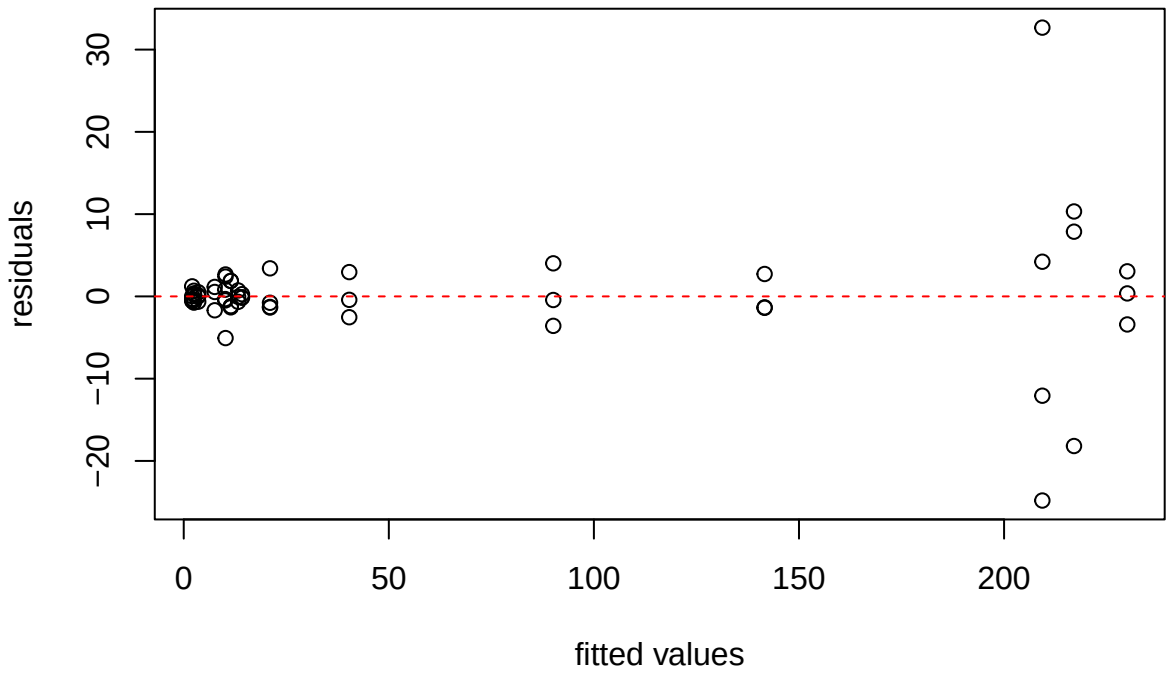
Substrate Type Effect: The main effect of substrate type was found to be statistically significant ($F(1, 53) = 23.93$, $p < 0.001$), indicating that the choice of substrate significantly influenced the rate of plastic degradation. Specifically, it suggests that different substrate types had a notable impact on the 'y' variable.

Substrate Concentration Level Effect: The main effect of substrate concentration level was not statistically significant ($F(1, 53) = 0.18$, $p = 0.6694$). This suggests that varying the concentration levels within the tested range did not have a significant influence on the rate of plastic degradation.

Interaction Effect: The interaction effect between substrate type and concentration level was highly significant ($F(1, 53) = 45.22$, $p < 0.001$). This implies that the combined influence of both substrate type and concentration level had a substantial impact on 'y' and that the relationship between these factors was not additive.

###6. CHECK ANOVE ASSUMPTIONS: TEST VALIDITY

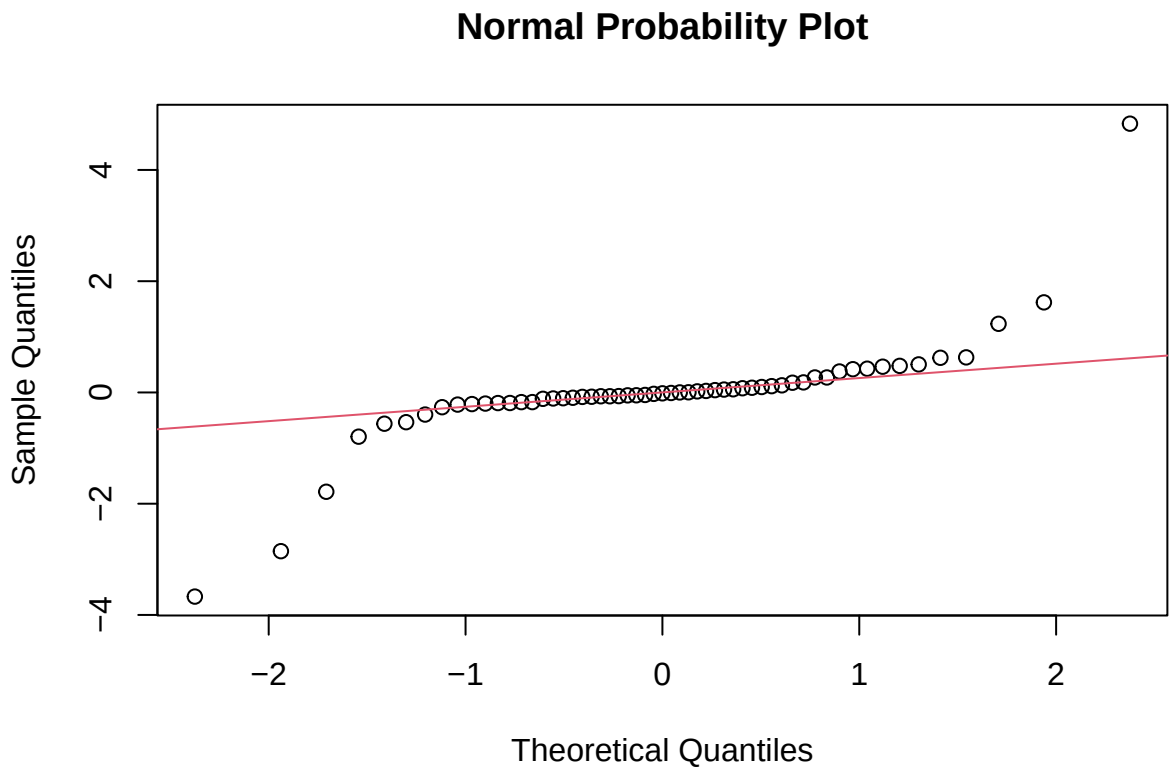
```r
#Residual Plot - Homogeneity of variances
y.res <- residuals(anova_result)
y.fitted <- fitted.values(anova_result)
plot(y.fitted, y.res, xlab = "fitted values", ylab ="residuals")
abline(h = 0, col = "red", lty = 2)
```



```r
#Levene's test - Homogeneity of variances
leveneTest(y ~ substrate * sub_conc_level, data = clean_data)
```

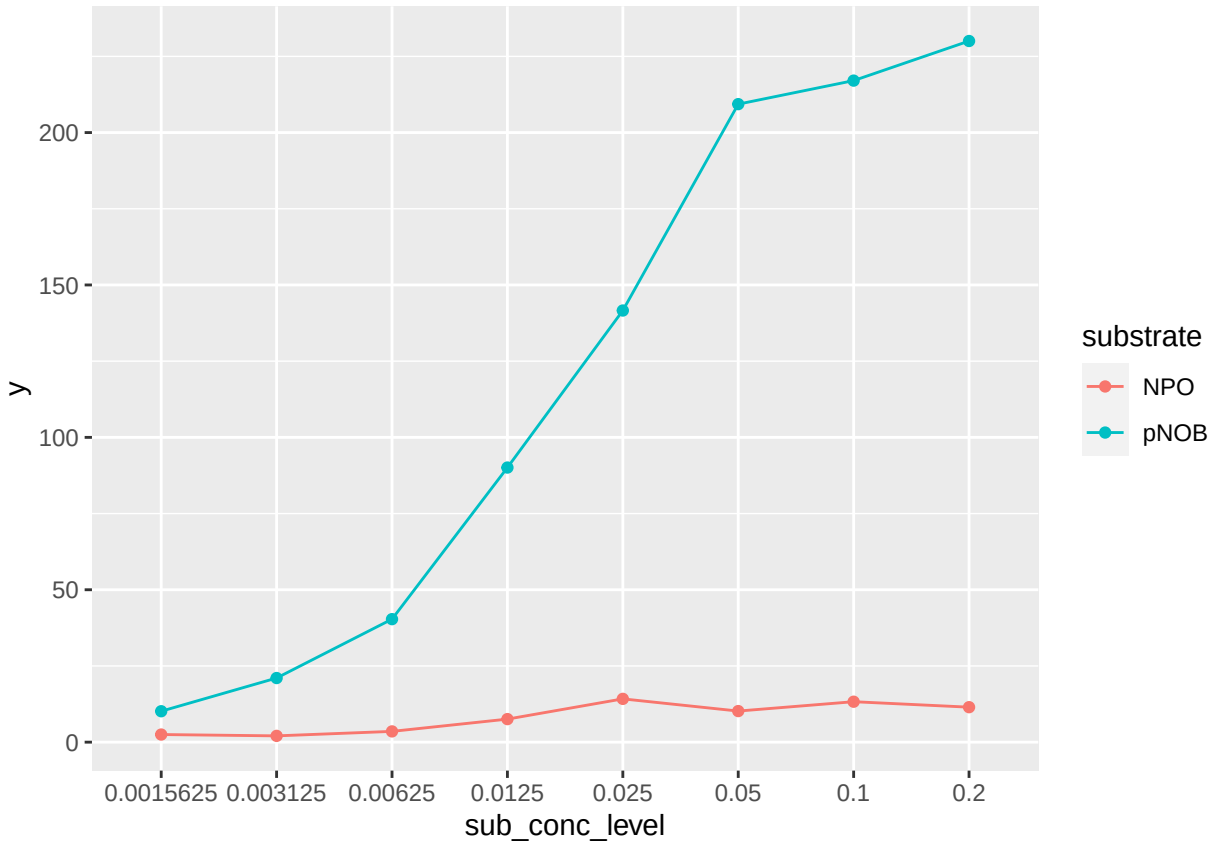
```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value   Pr(>F)
## group 15   3.316 0.001167 **
##       41
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Probability Plot
y.stdres <- rstandard(anova_result)
qqnorm(y.stdres, main = "Normal Probability Plot")
qqline(y.stdres, col = 2)
```




```r
#Interaction Plot
ggplot(clean_data, aes(x = sub_conc_level, y = y, color = substrate, group = substrate)) +
  stat_summary(fun.y = mean, geom = "point") +
  stat_summary(fun.y = mean, geom = "line")
```

```
## Warning: The `fun.y` argument of `stat_summary()` is deprecated as of ggplot2 3.3.0.
## i Please use the `fun` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Residual Plot & Levene's Test: testing homoscedasticity, also known as constant variance.Homoscedasticity means that the variance of the residuals (the differences between observed values and predicted values) is consistent across all levels or groups of the independent variable(s). The spread around 0 is uneven in horizontal sense and p-value for the Levene's Test (0.001167) <0.05, reject the null, there's sufficient evidence to conclude that at least one combination of substrate and sub_conc_level has a different variance compared to the others. -> Homoscedasticity is violated

Probability Plot: This plot can be used to assess whether the standardized residuals follow a normal distribution. If the points closely follow the reference line, it suggests that the residuals are approximately normally distributed. Our case, it is not.-> Residuals ~ N(0, constant var) is violated.

Interaction Plot: we interpret them by looking to see if there is any significant difference in the slopes of the lines, or lines that cross because their slopes have opposite signs. Those would all be indications of an interaction effect. Here it is appear no interaction.

###5. IDENTIFY & DROP OUTLIERS

```
clean_data <- clean_data %>%
  group_by(substrate, sub_conc_level) %>%
  mutate(z_score = scale(y))

# Identify outliers with z-scores
outliers <- clean_data[abs(clean_data$z_score) > 2, ]
outliers
```

```
## # A tibble: 0 x 4
## # Groups:   substrate, sub_conc_level [0]
## # i 4 variables: y <dbl>, sub_conc_level <fct>, substrate <fct>,
## #   z_score <dbl[,1]>
```

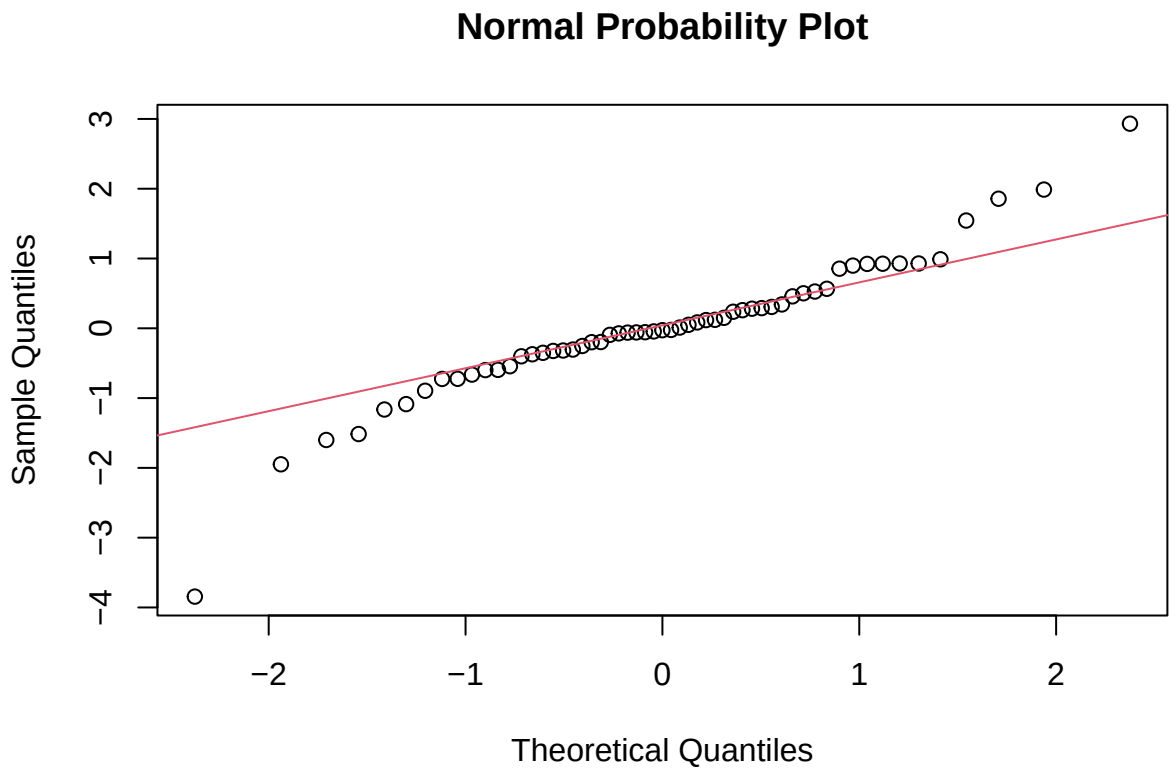
Since the assumptions of Anova are not met and no outliers can be dropped. Three viable options we can go from here: *Transform our variables* Using a non-parametric test like the Kruskal-Wallist test, which does not assume honoscedasticity or normality. This test is suitable for comparing groups when ANOVA assumptions are violated. *Bootstrapping: a small sample size or non-normal residuals, you can use bootstrapping to resample your data and calculate confidence intervals. This approach doesn't rely on parametric assumptions and can provide robust results.

###7. TRANSFORM Y

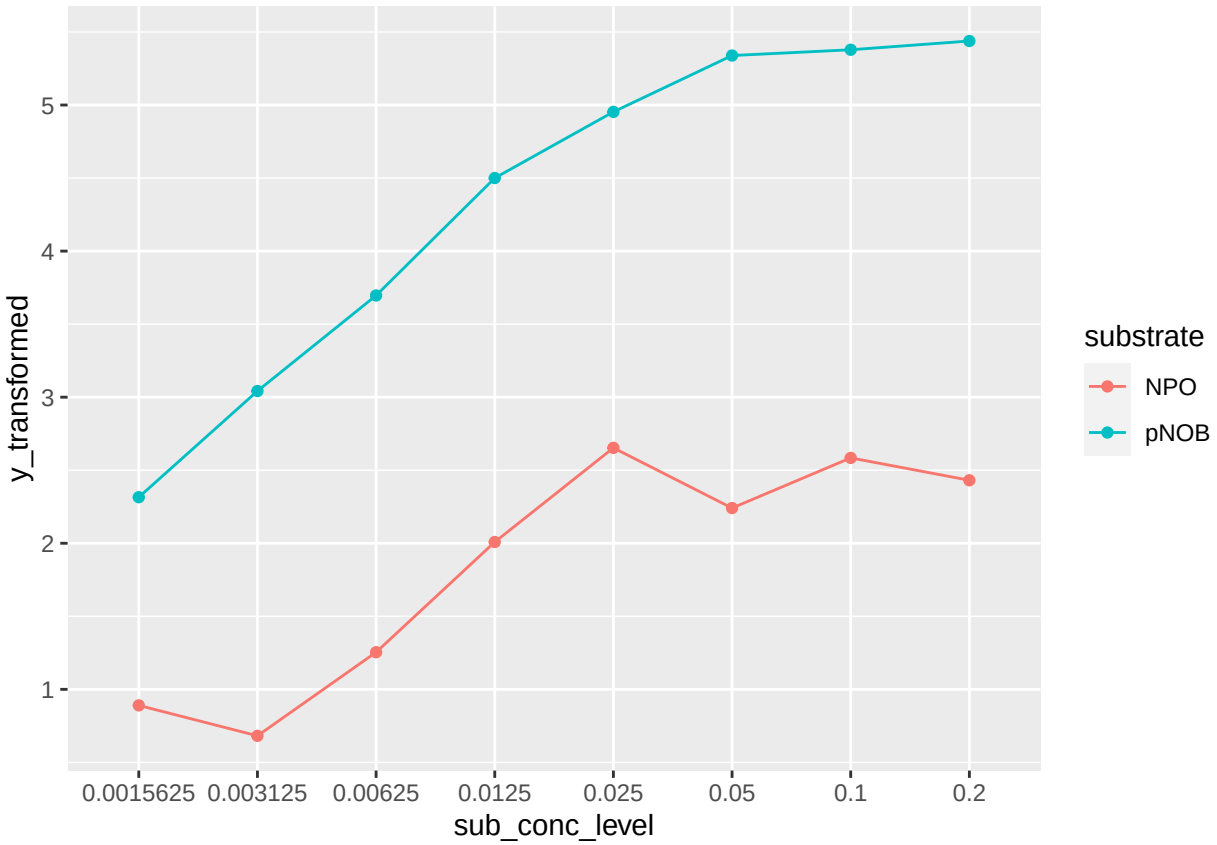
```
# Log-transform the response variable
clean_data$y_transformed <- log(clean_data$y)

# Levene's test for homogeneity of variances
leveneTest(y_transformed ~ substrate * sub_conc_level, data = clean_data)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group 15   0.856  0.614
##       41
```

```
# Probability plot for normality
y_stdres <- rstandard(aov(y_transformed ~ substrate * sub_conc_level, data = clean_data))
qqnorm(y_stdres, main = "Normal Probability Plot")
qqline(y_stdres, col = 2)
```




```
#Interaction Plot
ggplot(clean_data, aes(x = sub_conc_level, y = y_transformed, color = substrate, group = substrate)) +
  stat_summary(fun.y = mean, geom = "point") +
  stat_summary(fun.y = mean, geom = "line")
```



After transforming y to log(y), the p-value of the Levene's Test (0.614) $>$ 0.05, we fail reject the null hypothesis (the variances are equal across groups). This suggests that there are no significant variance differences among groups, meeting the assumption of homoscedasticity.

Probability Plot: the points mostly stay around the line, except for some extreme points at the two tails, suggesting that while the residuals are approximately normally distributed overall. This indicates that the assumption of normality of residuals is reasonably met.

###8. TWOWAY ANOVA USING LOG(Y)

```
# Run ANOVA with transformed data
anova_result_transformed <- aov(y_transformed ~ substrate * sub_conc_level, data = clean_data)

# Summary of ANOVA results
summary(anova_result_transformed)
```

```
##                          Df Sum Sq Mean Sq F value  Pr(>F)
## substrate                 1  95.31   95.31 2556.21 < 2e-16 ***
## sub_conc_level            7  46.97    6.71  179.97 < 2e-16 ***
## substrate:sub_conc_level  7   3.49    0.50   13.38 7.7e-09 ***
## Residuals                41   1.53    0.04
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In the transformed dataset, the two independent variables, 'substrate' and 'sub_conc_level,' along with their interaction, have a significant effect on the dependent variable 'y_transformed' as evidenced by the extremely low p-values (all $<$ 2e-16). This indicates that the type of substrate, concentration levels, and their interaction all play a significant role in explaining the variability in the transformed 'y' values. Additionally, the residuals

show low variability within groups, supporting the assumptions of homoscedasticity and normality, further validating the results.

Interesting observation: While the interaction plot shows that the lines for 'pNOB' and 'NPO' never cross, indicating an absence of a qualitative interaction (where the effect of one variable depends on the level of another), ANOVA's significance of the interaction term suggests that there is still a quantitative interaction present.

In other words, while the two substrates may not have fundamentally different effects on the outcome, the strength or magnitude of their effects may vary depending on the concentration levels. This implies that the interaction between 'substrate' and 'sub_conc_level' may not change the nature of the relationship between these variables, but it does affect the extent or degree to which they influence the outcome 'y_transformed.' This is an important finding and highlights the importance of considering both qualitative and quantitative aspects of interactions in your analysis.

###9. POST-HOC TEST:

```
# Load the necessary library for post-hoc tests
library(TukeyC)

# Perform Tukey's HSD post-hoc test on Substrate
posthoc <- TukeyC::TukeyC(anova_result_transformed)
summary(posthoc)
```

```
## Goups of means at sig.level = 0.05
##      Means G1 G2
## pNOB  4.33  a
## NPO   1.84     b
##
## Matrix of the difference of means above diagonal and
## respective p-values of the Tukey test below diagonal values
##      pNOB  NPO
## pNOB    0 2.49
## NPO     0 0.00
```

```
# Perform Tukey's HSD post-hoc test on sub_conc_level
posthoc_sub_conc <- TukeyHSD(anova_result_transformed, which = "sub_conc_level")
print(posthoc_sub_conc)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = y_transformed ~ substrate * sub_conc_level, data = clean_data)
##
## $sub_conc_level
##                            diff        lwr       upr     p adj
## 0.003125-0.0015625  0.12601589 -0.1735148 0.4255466 0.8766696
## 0.00625-0.0015625   0.73701694  0.4179846 1.0560493 0.0000001
## 0.0125-0.0015625    1.50529436  1.1723846 1.8382041 0.0000000
## 0.025-0.0015625     2.05395499  1.7210453 2.3868647 0.0000000
## 0.05-0.0015625      2.07660262  1.7575703 2.3956349 0.0000000
## 0.1-0.0015625       2.23224236  1.8993326 2.5651521 0.0000000
## 0.2-0.0015625       2.13438250  1.8261680 2.4425970 0.0000000
## 0.00625-0.003125    0.61100105  0.3003500 0.9216521 0.0000046
```

```
## 0.0125-0.003125     1.37927846  1.0543918 1.7041651 0.0000000
## 0.025-0.003125      1.92793909  1.6030525 2.2528257 0.0000000
## 0.05-0.003125       1.95058673  1.6399357 2.2612378 0.0000000
## 0.1-0.003125        2.10622646  1.7813398 2.4311131 0.0000000
## 0.2-0.003125        2.00836661  1.7088359 2.3078973 0.0000000
## 0.0125-0.00625      0.76827742  0.4253280 1.1112269 0.0000003
## 0.025-0.00625       1.31693804  0.9739886 1.6598875 0.0000000
## 0.05-0.00625        1.33958568  1.0100905 1.6690809 0.0000000
## 0.1-0.00625         1.49522541  1.1522760 1.8381749 0.0000000
## 0.2-0.00625         1.39736556  1.0783332 1.7163979 0.0000000
## 0.025-0.0125        0.54866063  0.1927652 0.9045561 0.0003577
## 0.05-0.0125         0.57130827  0.2283588 0.9142577 0.0001020
## 0.1-0.0125          0.72694800  0.3710525 1.0828435 0.0000021
## 0.2-0.0125          0.62908815  0.2961784 0.9619979 0.0000102
## 0.05-0.025          0.02264764 -0.3203018 0.3655971 0.9999989
## 0.1-0.025           0.17828737 -0.1776081 0.5341828 0.7479780
## 0.2-0.025           0.08042752 -0.2524822 0.4133372 0.9937321
## 0.1-0.05            0.15563973 -0.1873097 0.4985892 0.8289694
## 0.2-0.05            0.05777988 -0.2612524 0.3768122 0.9989701
## 0.2-0.1            -0.09785985 -0.4307696 0.2350499 0.9802279
```

Findings: Here are the pairs of concentration levels that are NOT significantly different from each other (adj p-value >0.05): (0.05 & 0.025) (0.1 & 0.025) (0.2 & 0.025)
(0.1 & 0.05)
(0.2 & 0.05)
(0.2 & 0.1) (0.003125 & 0.0015625) This means that when it comes to how these concentrations affect the outcome we're studying, they are quite similar and don't stand out as significantly different from each other.

The concentration level with the highest mean (if significantly different) can be considered the "best" in terms of achieving the highest y value, indicating better degradation. And that is "0.2", this suggests that a concentration of 0.2 is likely the most effective in achieving a higher log(y) value, essentially higher y value, indicating better degradation.

###10. MULTIPLE REGRESSION

```r
# Create dummy variables for categorical variables
dummy_data <- model.matrix(~ substrate + sub_conc_level, data = clean_data)

# Create the multiple regression model
lm_model <- lm(y ~ dummy_data, data = clean_data)

# Summarize the regression results
summary(lm_model)
```

```
##
## Call:
## lm(formula = y ~ dummy_data, data = clean_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -62.769 -42.945   8.732  37.556  72.593
##
## Coefficients: (1 not defined because of singularities)
##                                  Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                         -34.870     16.788  -2.077 0.043173 *
## dummy_data(Intercept)                    NA         NA      NA       NA
## dummy_datasubstratepNOB              107.291     12.250   8.758 1.61e-11 ***
## dummy_datasub_conc_level0.003125    -2.320     22.209  -0.104 0.917226
## dummy_datasub_conc_level0.00625      8.201     23.647   0.347 0.730242
## dummy_datasub_conc_level0.0125      30.052     24.714   1.216 0.229933
## dummy_datasub_conc_level0.025       59.133     24.714   2.393 0.020684 *
## dummy_datasub_conc_level0.05        97.548     23.760   4.106 0.000156 ***
## dummy_datasub_conc_level0.1         96.382     24.714   3.900 0.000299 ***
## dummy_datasub_conc_level0.2         88.072     22.836   3.857 0.000342 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 45.67 on 48 degrees of freedom
## Multiple R-squared:  0.7338, Adjusted R-squared:  0.6894
## F-statistic: 16.54 on 8 and 48 DF,  p-value: 1.942e-11
```

```
# Regression with interaction term
lm_model_interaction <- lm(y ~ substrate * sub_conc_level, data = clean_data)
summary(lm_model_interaction)
```

```
##
## Call:
## lm(formula = y ~ substrate * sub_conc_level, data = clean_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.814  -1.211  -0.101   1.150  32.671
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            2.4963     3.4918   0.715  0.47871
## substratepNOB                          7.6465     5.7020   1.341  0.18730
## sub_conc_level0.003125                -0.4366     4.9381  -0.088  0.92998
## sub_conc_level0.00625                  1.0338     5.2376   0.197  0.84451
## sub_conc_level0.0125                   5.0618     5.7020   0.888  0.37987
## sub_conc_level0.025                   11.7018     5.7020   2.052  0.04657 *
## sub_conc_level0.05                     7.7008     5.7020   1.351  0.18425
## sub_conc_level0.1                     10.7703     5.7020   1.889  0.06600 .
## sub_conc_level0.2                      8.9779     4.9381   1.818  0.07636 .
## substratepNOB:sub_conc_level0.003125  11.3309     7.7425   1.463  0.15096
## substratepNOB:sub_conc_level0.00625   29.1795     8.2507   3.537  0.00102 **
## substratepNOB:sub_conc_level0.0125    74.8906     8.5530   8.756 6.23e-11 ***
## substratepNOB:sub_conc_level0.025    119.7745     8.5530  14.004  < 2e-16 ***
## substratepNOB:sub_conc_level0.05     191.4849     8.2507  23.208  < 2e-16 ***
## substratepNOB:sub_conc_level0.1      196.1344     8.5530  22.932  < 2e-16 ***
## substratepNOB:sub_conc_level0.2      210.9173     8.0639  26.156  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.808 on 41 degrees of freedom
## Multiple R-squared:  0.9934, Adjusted R-squared:  0.9909
## F-statistic: 408.5 on 15 and 41 DF,  p-value: < 2.2e-16
```

Regression without Interaction Term (It assesses the main effects of each variable on the plastic degradation rate): The overall model fits the data reasonably well, with an adjusted R-squared of 0.6894. This value indicates that about 68.94% of the variation in plastic degradation can be explained by the combination of substrate and concentration level.

Main Effects:

dummy_datasubstratepNOB is statistically significant: the coefficient 107.291 uM/min means using pNOB results in an increase of 107.291 uM/min in the degradation rate (y) compared to NPO.

sub_conc_level: * 0.003125, 0.00625, 0.0125: are not statistically significant, thus no effect on the degradation rate (y) * 0.025, 0.05. 0.1, 0.2: are statiscically significant, 0.05 is the most effective concentration level because it results in the highest increase in y among the tested levels (97.548 uM/min)

Regression with Interaction Term (It captures the joint effects of both variables on the plastic degradation rate): The model's overall fit is excellent, with an Adjusted R-squared value of 0.9909. This indicates that the model explains approximately 99.09% of the variance in the plastic degradation rate.