ENV 710 Final Project

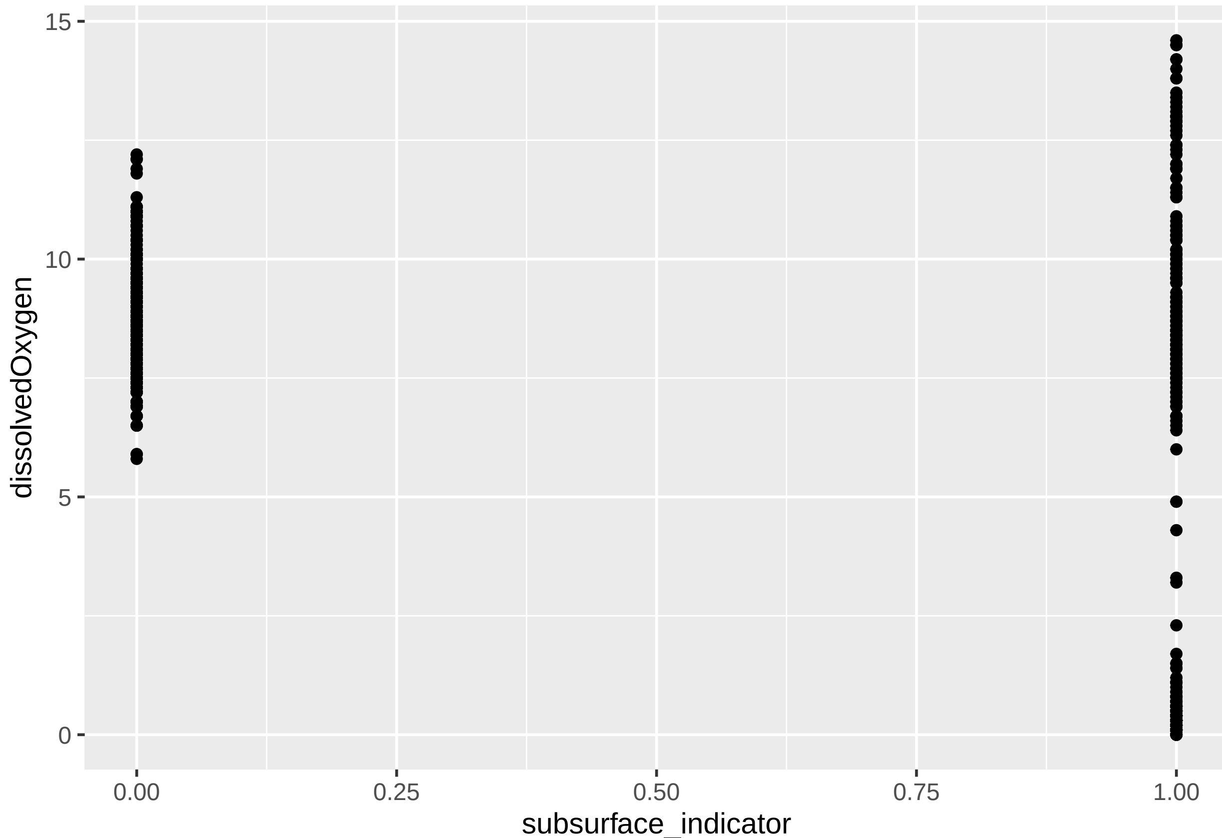
Emma Kaufman, Cara Kuuskvere, Jenn McNeill

2024-04-10

```
lakes_raw <- read.csv("./Data/Raw/NTL-LTER_Lake_Nutrients_Raw.csv",stringsAsFactors = TRUE)
do_raw <- read.csv("Data/Raw/NTL-LTER_Lake_ChemistryPhysics_Raw.csv",stringsAsFactors = TRUE)

lakes_raw$sampledate <- mdy(lakes_raw$sampledate)
do_raw$sampledate <- mdy(do_raw$sampledate)

lakes_processed <- lakes_raw %>%
  select(lakename, sampledate, depth, tn_ug, tp_ug, nh34, no23, po4) %>%
  na.omit() %>%
  filter(lakename %in% c("Paul Lake", "West Long Lake", "Peter Lake", "East Long Lake")) %>%
  filter(year(sampledate) >= 1994 & year(sampledate) <= 1999) %>%
  mutate(year = factor(year(sampledate)),
         month = factor(month(sampledate)),
         subsurface = factor(ifelse(depth > 1, "subsurface", "surface")),
         season = factor(ifelse(month == 5, "spring", ifelse(month == 8, "summer", NA)))) %>%
  filter(month %in% c(5,6,7,8)) %>%
  mutate(depth = round(depth * 2) / 2)

lakes_processed$subsurface_indicator <- ifelse(lakes_processed$subsurface == "subsurface", 1, 0)
lakes_processed$summer_indicator <- ifelse(lakes_processed$season == "summer", 1, 0)

lakes_processed <- left_join(lakes_processed, select(do_raw, lakename, sampledate, depth, dissolvedOxyge

lake_1994 <- lakes_processed %>%
  filter(year %in% c(1994)) %>%
  filter(po4<20) %>%
  filter(tp_ug<100)
```

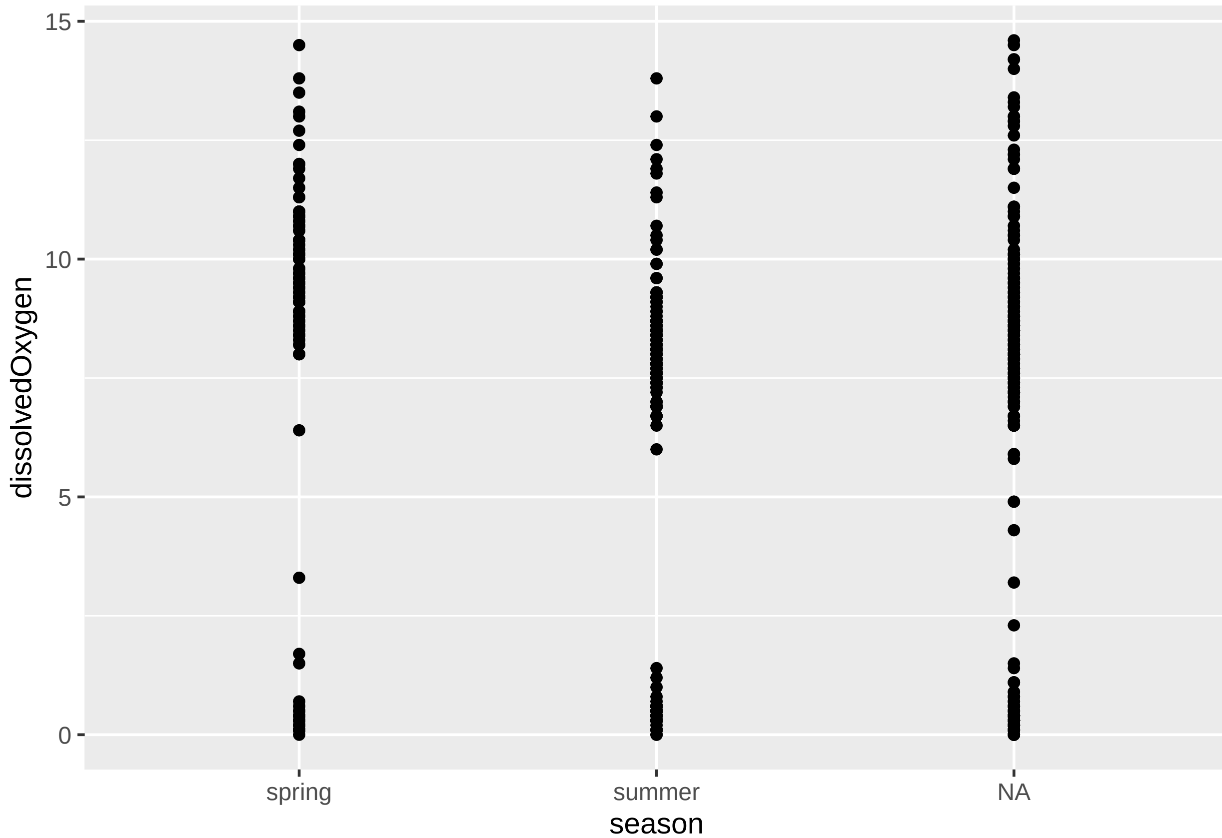
```
ggplot(lakes_processed)+
  geom_point(aes(y=dissolvedOxygen,x=subsurface_indicator))
```

```
## Warning: Removed 14 rows containing missing values (`geom_point()`).
```




```r
ggplot(lakes_processed)+
  geom_point(aes(y=dissolvedOxygen,x=season))
```

```
## Warning: Removed 14 rows containing missing values (‘geom_point()‘).
```




```r
# Holdout 20% of data for testing, 80% for training
lakes_processed$holdout <- rbinom(n=nrow(lakes_processed),size=1,prob=0.2)
lakes_training <- lakes_processed %>% filter(holdout==0)
lakes_testing <- lakes_processed %>% filter(holdout==1)
```

```r
# lakes_raw <- read.csv("./Data/Raw/NTL-LTER_Lake_Nutrients_Raw.csv",stringsAsFactors = TRUE)
# lakes_new <- read_csv("Data/Raw/NTL-LTER_Lake_ChemistryPhysics_Raw.csv")
#
# lakes_raw$sampledate <- mdy(lakes_raw$sampledate)
#
# lakes_processed <- lakes_raw %>%
#   select(lakename, sampledate, depth, tn_ug, tp_ug, nh34, no23, po4) %>%
#   na.omit() %>%
#   filter(lakename %in% c("Paul Lake", "West Long Lake", "Peter Lake", "East Long Lake")) %>%
#   filter(year(sampledate) >= 1994 & year(sampledate) <= 1999) %>%
#   mutate(year = factor(year(sampledate)),
#          month = factor(month(sampledate)),
#          deep = factor(ifelse(depth > 6, 1, 0))) %>%
#   filter(month %in% c(5,6,7,8))
#
# lake_1994 <- lakes_processed %>%
#   filter(year %in% c(1994)) %>%
#   filter(po4<20) %>%
#   filter(tp_ug<100)
#
```

```r
# lakes_new$sampledate <- mdy(lakes_new$sampledate)
#
# lakes_chem_processed <- lakes_new %>%
#   select(lakename, sampledate, depth, temperature_C, dissolvedOxygen) %>%
#   na.omit() %>%
#   filter(lakename %in% c("Paul Lake", "West Long Lake", "Peter Lake", "East Long Lake")) %>%
#   filter(year(sampledate) == 1994) %>%
#   mutate(year = factor(year(sampledate)),
#          month = factor(month(sampledate)),
#          deep = factor(ifelse(depth > 6, 1, 0))) %>%
#   filter(month %in% c(5,6,7,8))
```

```r
two_season_clean<- lakes_processed %>%
  na.omit()

two_season_fit <- lm(dissolvedOxygen ~ summer_indicator + tp_ug + summer_indicator*tp_ug,
                     data= two_season_clean)

summary(two_season_fit)
```

```
##
## Call:
## lm(formula = dissolvedOxygen ~ summer_indicator + tp_ug + summer_indicator *
##     tp_ug, data = two_season_clean)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.445 -0.762  0.283  1.440  6.777
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            10.216614   0.331299  30.838  < 2e-16 ***
## summer_indicator       -1.522394   0.447201  -3.404 0.000764 ***
## tp_ug                  -0.062867   0.005435 -11.567  < 2e-16 ***
## summer_indicator:tp_ug  0.018353   0.006472   2.836 0.004918 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.796 on 270 degrees of freedom
## Multiple R-squared:  0.5379, Adjusted R-squared:  0.5328
## F-statistic: 104.8 on 3 and 270 DF,  p-value: < 2.2e-16
```

a. What is the estimate of β_0 and what does it represent in terms of DO (dissolved oxygen)?

β_0 for this fit is 10.22 mg/L, which represents the expected value of DO in spring with zero total phosphorus.

b. What is the estimate of β_1 and what does it represent in terms of DO?

β_1 for this fit is -1.52, which represents the difference in DO between spring and summer with zero total phosphorus. In summer the expected dissolved oxygen is 8.7 mg/L.

c. What is the estimate of β_2 and what does it represent in terms of DO?

β_2 for this fit is -0.063 (mg/L)/ug, which represents the expected change in DO for each unit increase in total phosphorus during spring.

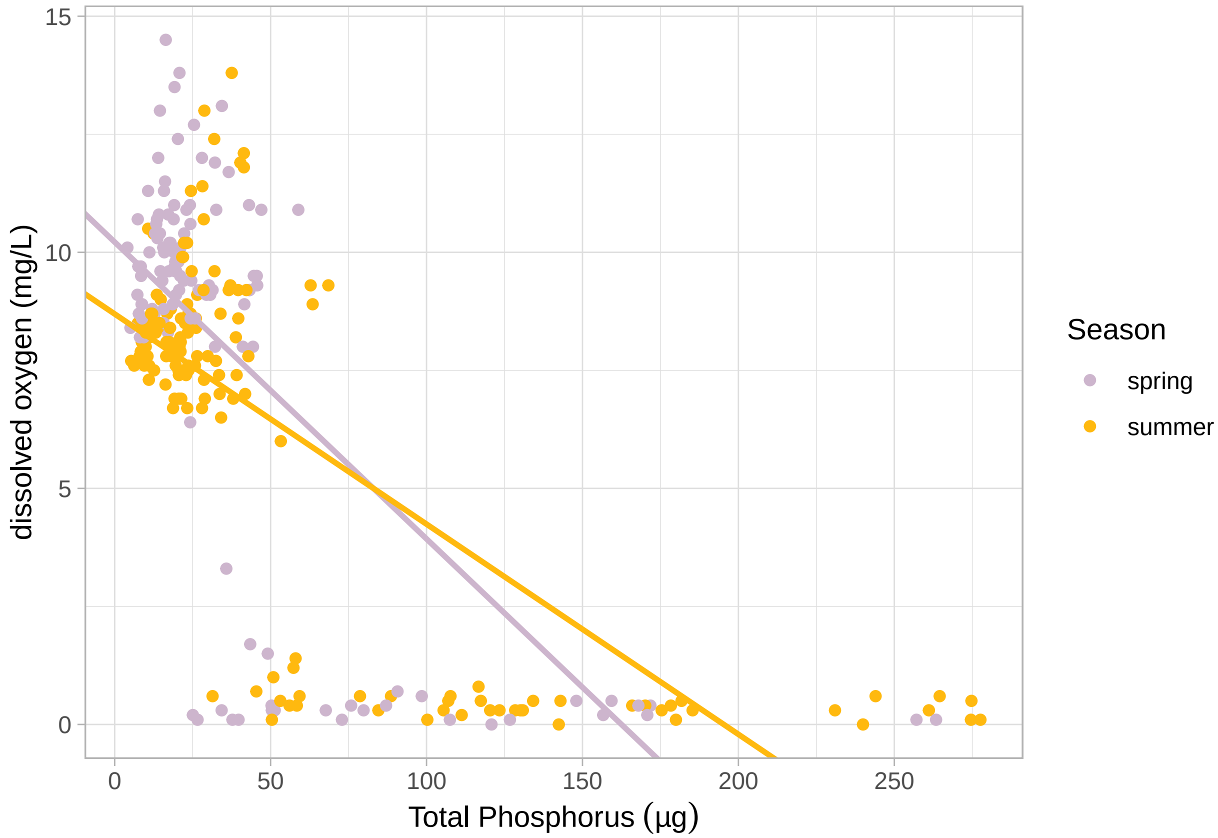
d. What is the estimate of β_3 and what does it represent in terms of DO?

β_3 for this fit is 0.02 (mg/L)/ug, which represents the adjustment to the slope for each unit increase in total phosphorus during the summer. In the summer the expected change in DO for each unit increase in total phosphorus is -0.045 (mg/L)/ug.

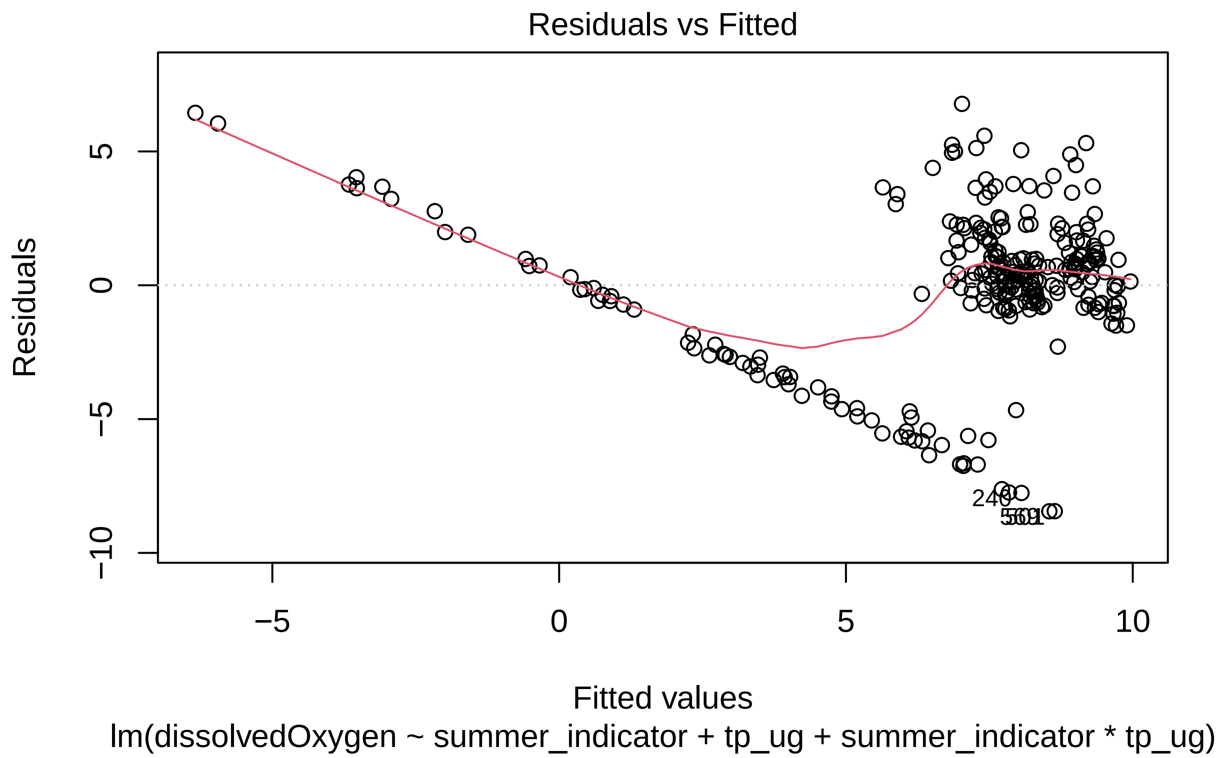
```
Two_Season_P <- ggplot(two_season_clean, aes(x=tp_ug, y=dissolvedOxygen, color=season))+
  geom_point()+
  geom_abline(intercept = coef(two_season_fit)[1],
              slope = coef(two_season_fit)[3],
              color = "thistle3", size = 1)+
  geom_abline(intercept = coef(two_season_fit)[1] + coef(two_season_fit)[2],
              slope = coef(two_season_fit)[3] + coef(two_season_fit)[4],
              color = "darkgoldenrod1", size = 1)+
  scale_color_manual(values = c("spring" = "thistle3", "summer" = "darkgoldenrod1"))+
  scale_x_continuous(breaks = seq(0, 300, by = 50))+
  scale_y_continuous(breaks = seq(0, 15, by = 5))+
  xlab(expression("Total Phosphorus" ~ (µg)))+
  ylab("dissolved oxygen (mg/L)")+
  theme_light()+
  theme(legend.position="right")+
  labs(color = "Season")
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

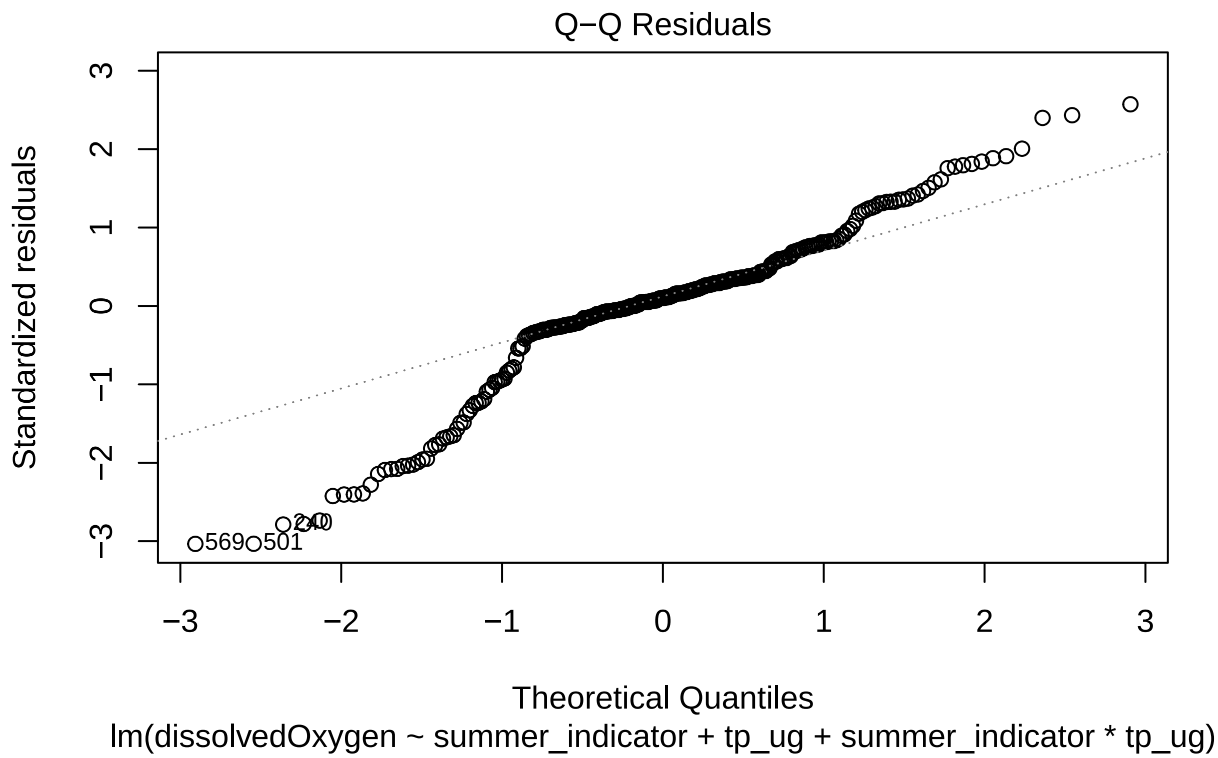
```
Two_Season_P
```

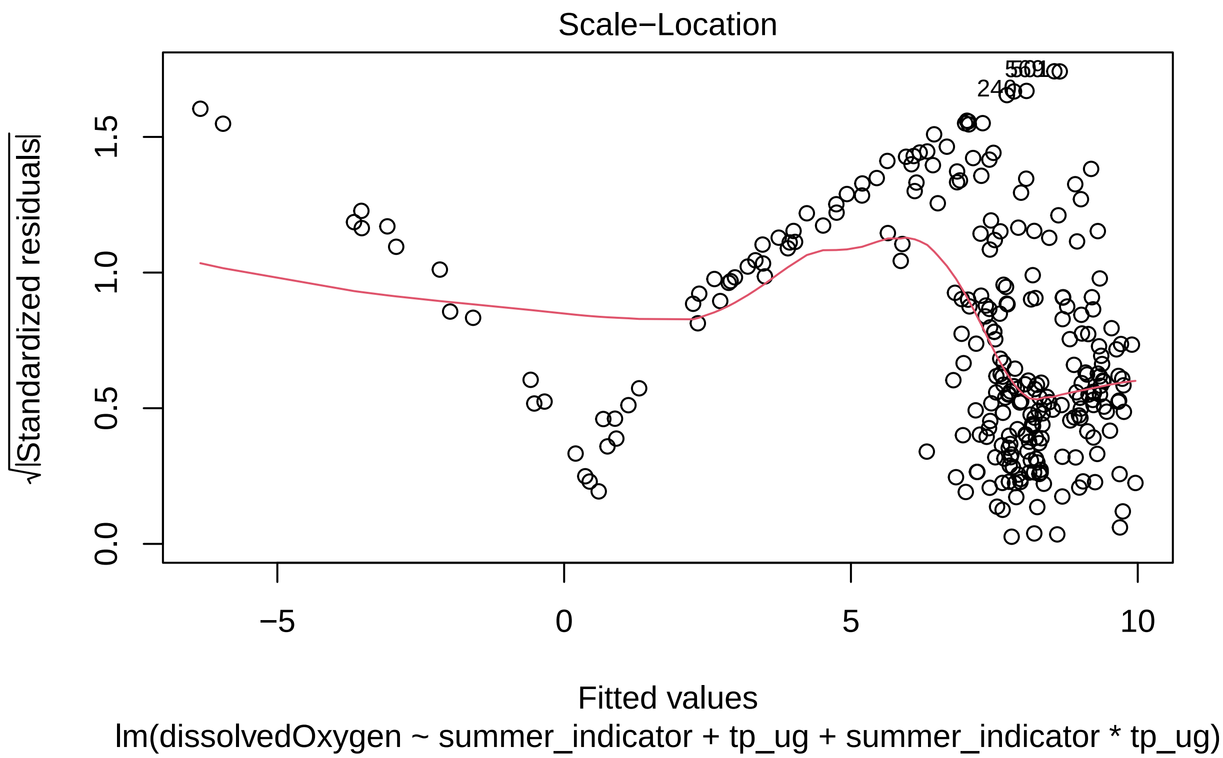


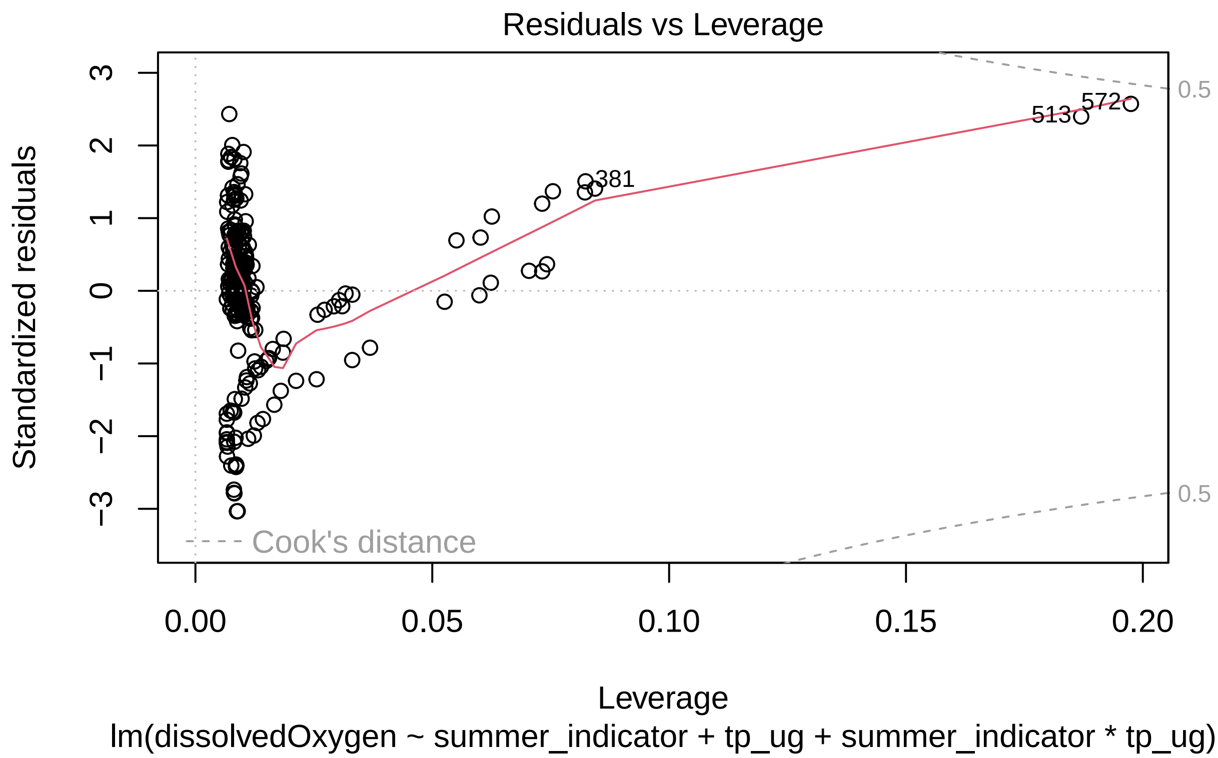

```r
plot(two_season_fit)
```



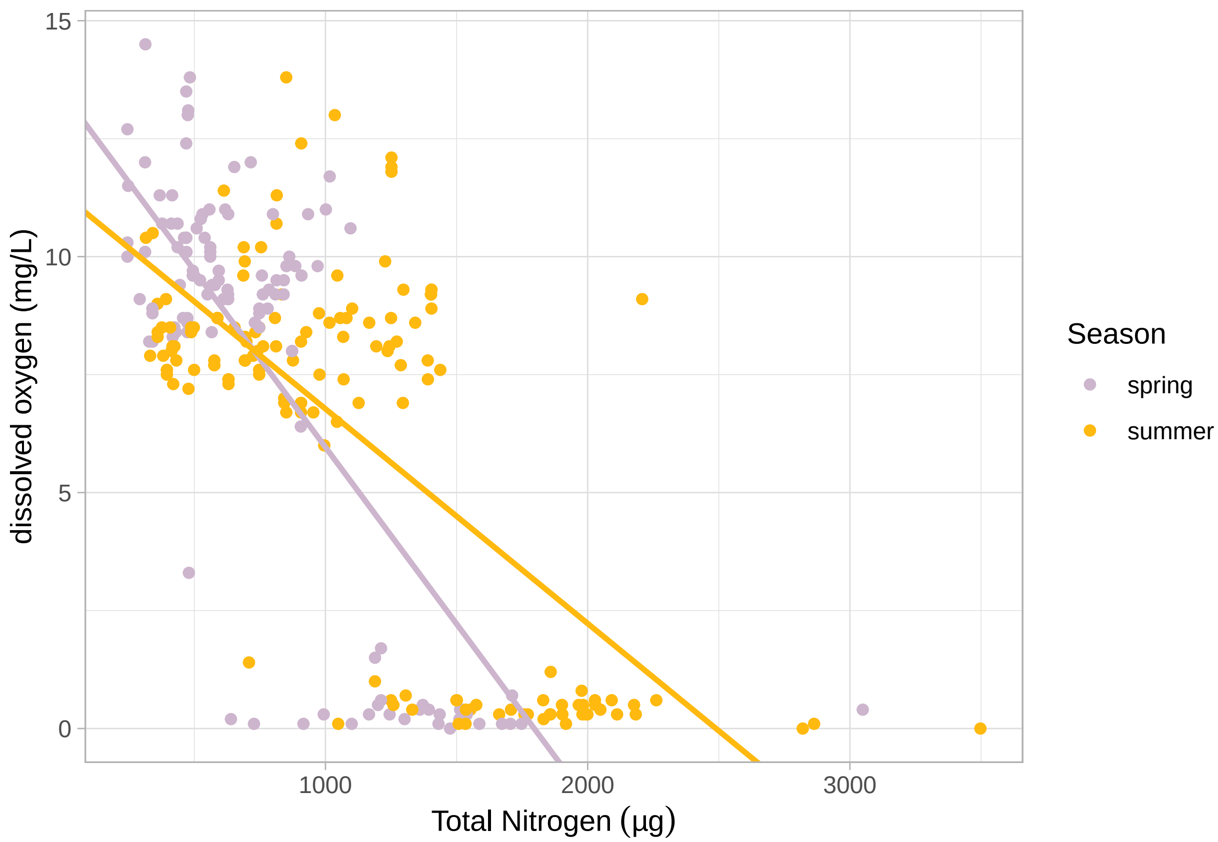
Residuals vs Fitted
Residuals
5
0
−5
−10
−5
0
5
10
240
561
Fitted values
lm(dissolvedOxygen ~ summer_indicator + tp_ug + summer_indicator * tp_ug)


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(dissolvedOxygen ~ summer_indicator + tp_ug + summer_indicator * tp_ug)
569
501
240
3
2
1
0
−1
−2
−3


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
−5
0
5
10
Fitted values
lm(dissolvedOxygen ~ summer_indicator + tp_ug + summer_indicator * tp_ug)
561
240




```r
two_season_nitrogen <- lm(dissolvedOxygen ~ summer_indicator + tn_ug + summer_indicator*tn_ug,
                     data= two_season_clean)

ggplot(two_season_clean, aes(x=tn_ug, y=dissolvedOxygen, color=season))+
  geom_point()+
  geom_abline(intercept = coef(two_season_nitrogen)[1],
              slope = coef(two_season_nitrogen)[3],
              color = "thistle3", size = 1)+
  geom_abline(intercept = coef(two_season_nitrogen)[1] + coef(two_season_nitrogen)[2],
              slope = coef(two_season_nitrogen)[3] + coef(two_season_nitrogen)[4],
              color = "darkgoldenrod1", size = 1)+
  scale_color_manual(values = c("spring" = "thistle3", "summer" = "darkgoldenrod1"))+
  # scale_x_continuous(breaks = seq(0, 300, by = 50))+
  # scale_y_continuous(breaks = seq(0, 15, by = 5))+
  xlab(expression("Total Nitrogen" ~ (µg)))+
  ylab("dissolved oxygen (mg/L)")+
  theme_light()+
  theme(legend.position="right")+
  labs(color = "Season")
```




```r
jenn <- lakes_processed

jenn$subsurface_tp_ug = jenn$tp_ug * jenn$subsurface_indicator
jenn$subsurface_tn_ug = jenn$tn_ug * jenn$subsurface_indicator

jenn_fit_p <- lm(formula = dissolvedOxygen ~ subsurface + tp_ug + subsurface_tp_ug,
                 data = jenn)

print(jenn_fit_p)
```

```
##
## Call:
## lm(formula = dissolvedOxygen ~ subsurface + tp_ug + subsurface_tp_ug,
##     data = jenn)
##
## Coefficients:
##       (Intercept)  subsurfacesurface              tp_ug   subsurface_tp_ug
##         8.0955885          0.5534547         -0.0002088         -0.0462260
```

```r
jenn_fit_n <- lm(formula = dissolvedOxygen ~ subsurface + tn_ug + subsurface_tn_ug,
                 data = jenn)

print(jenn_fit_n)
```

```
##
```

```
## Call:
## lm(formula = dissolvedOxygen ~ subsurface + tn_ug + subsurface_tn_ug,
##     data = jenn)
##
## Coefficients:
##       (Intercept)  subsurfacesurface              tn_ug   subsurface_tn_ug
##        11.8444640         -3.3656422          0.0002059         -0.0061602
```

a. What is the estimate of β_0 and what does it represent in terms of DO?

β_0 for this fit is 8.65, which represents the expected value of DO in surface water with zero total phosphorus.

b. What is the estimate of β_1 and what does it represent in terms of DO?

β_1 for this fit is -0.55, which represents the difference in DO between surface water and subsurface water with zero total phosphorus.

c. What is the estimate of β_2 and what does it represent in terms of DO?

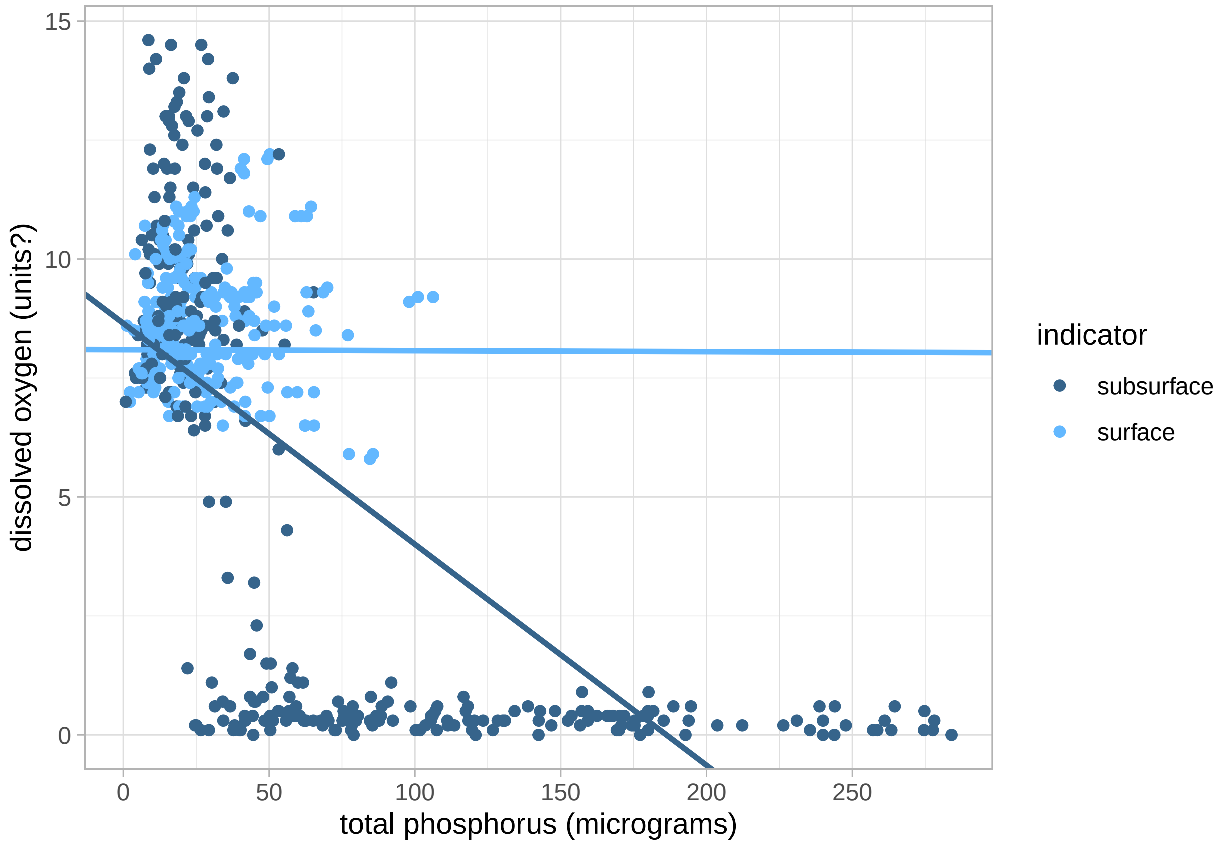
β_2 for this fit is -0.0002, which represents the expected change in DO for each unit increase in total phosphorus in surface water.

d. What is the estimate of β_3 and what does it represent in terms of DO?

β_3 for this fit is -0.046, which represents the adjustment to the slope for each unit increase in total phosphorus in subsurface water.

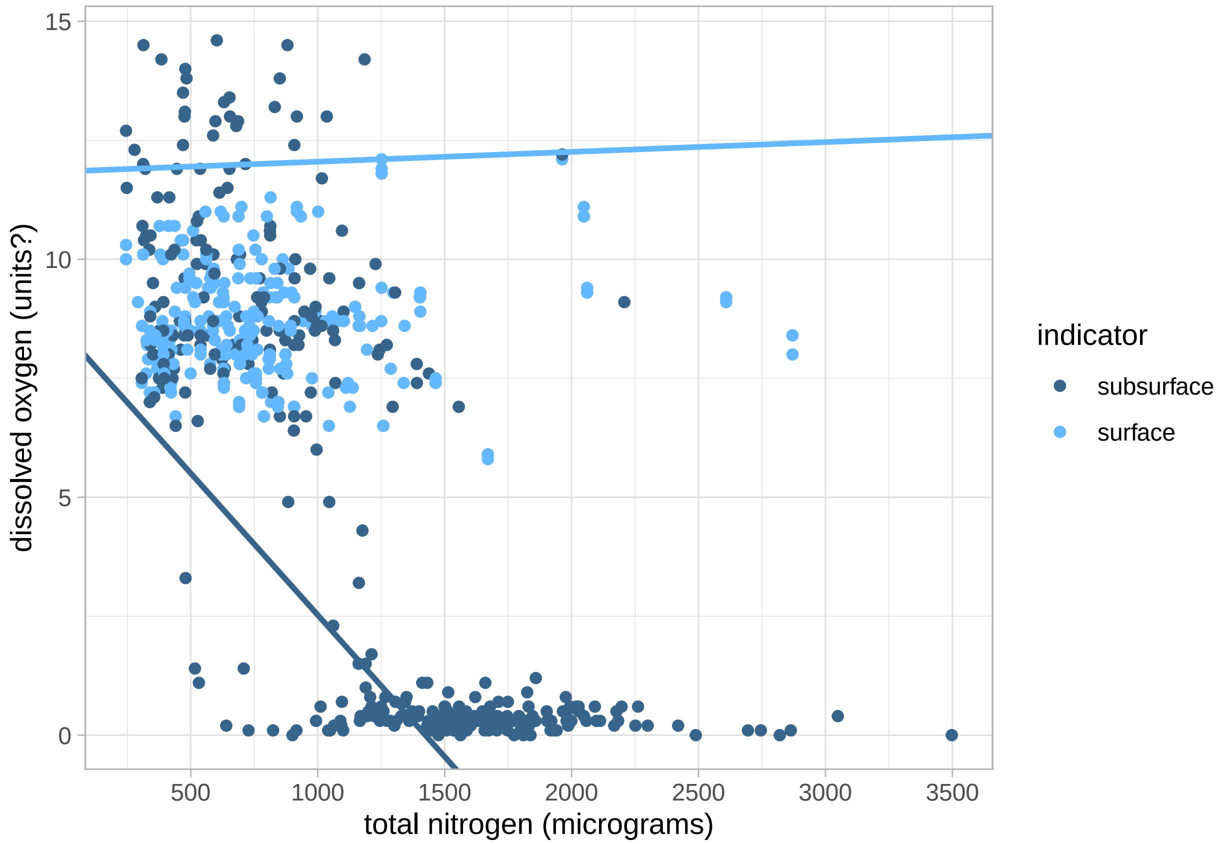
```r
#plot total phosphorus versus dissolved oxygen
ggplot(jenn, aes(x=tp_ug, y=dissolvedOxygen, color=subsurface))+
  geom_point()+
  geom_abline(intercept = coef(jenn_fit_p)[1],
              slope = coef(jenn_fit_p)[3],
              color = "steelblue1", size = 1)+
  geom_abline(intercept = coef(jenn_fit_p)[1] + coef(jenn_fit_p)[2],
              slope = coef(jenn_fit_p)[3] + coef(jenn_fit_p)[4],
              color = "steelblue4", size = 1)+
  scale_color_manual(values = c("surface" = "steelblue1", "subsurface" = "steelblue4"))+
  scale_x_continuous(breaks = seq(0, 300, by = 50))+
  scale_y_continuous(breaks = seq(0, 15, by = 5))+
  xlab("total phosphorus (micrograms)")+
  ylab("dissolved oxygen (units?)")+
  theme_light()+
  theme(legend.position="right")+
  labs(color = "indicator")
```

```
## Warning: Removed 14 rows containing missing values (‘geom_point()‘).
```




```r
#plot total nitrogen versus dissolved oxygen
ggplot(jenn, aes(x=tn_ug, y=dissolvedOxygen, color=subsurface))+
  geom_point()+
  geom_abline(intercept = coef(jenn_fit_n)[1],
              slope = coef(jenn_fit_n)[3],
              color = "steelblue1", size = 1)+
  geom_abline(intercept = coef(jenn_fit_n)[1] + coef(jenn_fit_n)[2],
              slope = coef(jenn_fit_n)[3] + coef(jenn_fit_n)[4],
              color = "steelblue4", size = 1)+
  scale_color_manual(values = c("surface" = "steelblue1", "subsurface" = "steelblue4"))+
  scale_x_continuous(breaks = seq(0, 3500, by = 500))+
  scale_y_continuous(breaks = seq(0, 15, by = 5))+
  xlab("total nitrogen (micrograms)")+
  ylab("dissolved oxygen (units?)")+
  theme_light()+
  theme(legend.position="right")+
  labs(color = "indicator")
```

```
## Warning: Removed 14 rows containing missing values (`geom_point()`).
```




```r
# Fit hierarchical linear mixed-effects model
fit <- lmer(dissolvedOxygen ~ tp_ug + (1|lakename), data = lakes_processed)

# Summary of the model
summary(fit)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: dissolvedOxygen ~ tp_ug + (1 | lakename)
##    Data: lakes_processed
##
## REML criterion at convergence: 3015
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.83766 -0.41123  0.07199  0.53499  2.28062
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  lakename (Intercept) 1.129    1.063
##  Residual             7.938    2.817
## Number of obs: 610, groups:  lakename, 4
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  9.315806   0.553857   16.82
## tp_ug       -0.055933   0.002218  -25.22
```

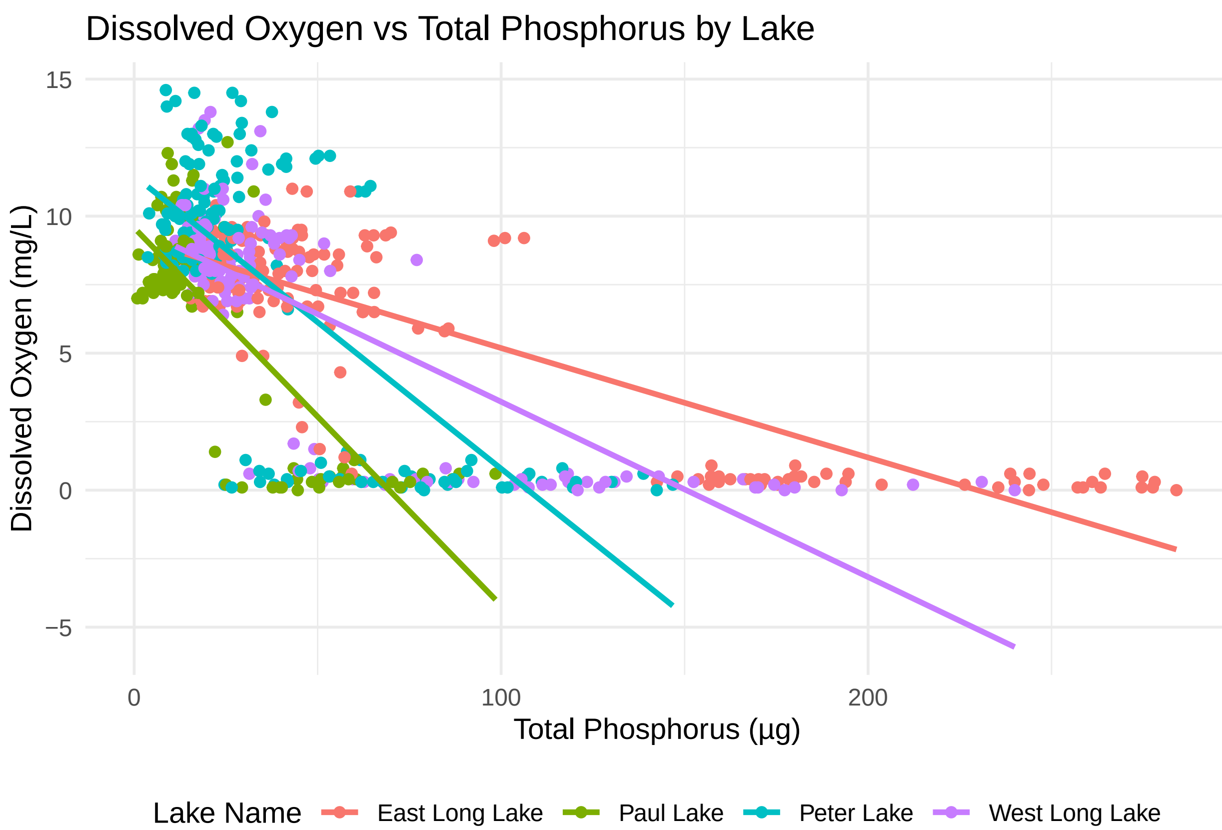
```
##
## Correlation of Fixed Effects:
##       (Intr)
## tp_ug -0.193
```

```r
# Print the model
print(fit)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: dissolvedOxygen ~ tp_ug + (1 | lakename)
##    Data: lakes_processed
## REML criterion at convergence: 3015.029
## Random effects:
##  Groups   Name        Std.Dev.
##  lakename (Intercept) 1.063
##  Residual             2.817
## Number of obs: 610, groups:  lakename, 4
## Fixed Effects:
## (Intercept)        tp_ug
##     9.31581     -0.05593
```

```r
# Extract the random effects
random_effects <- fit@u

# Plotting the relationship between total phosphorus (tp_ug) and dissolved oxygen (dissolvedOxygen) by
ggplot(lakes_processed, aes(x = tp_ug, y = dissolvedOxygen, color = lakename)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +  # Add a linear regression line
  labs(x = "Total Phosphorus (µg)", y = "Dissolved Oxygen (mg/L)", color = "Lake Name") +
  theme_minimal() +
  ggtitle("Dissolved Oxygen vs Total Phosphorus by Lake") +
  theme(legend.position = "bottom")
```

```
## Warning: Removed 14 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 14 rows containing missing values (`geom_point()`).
```

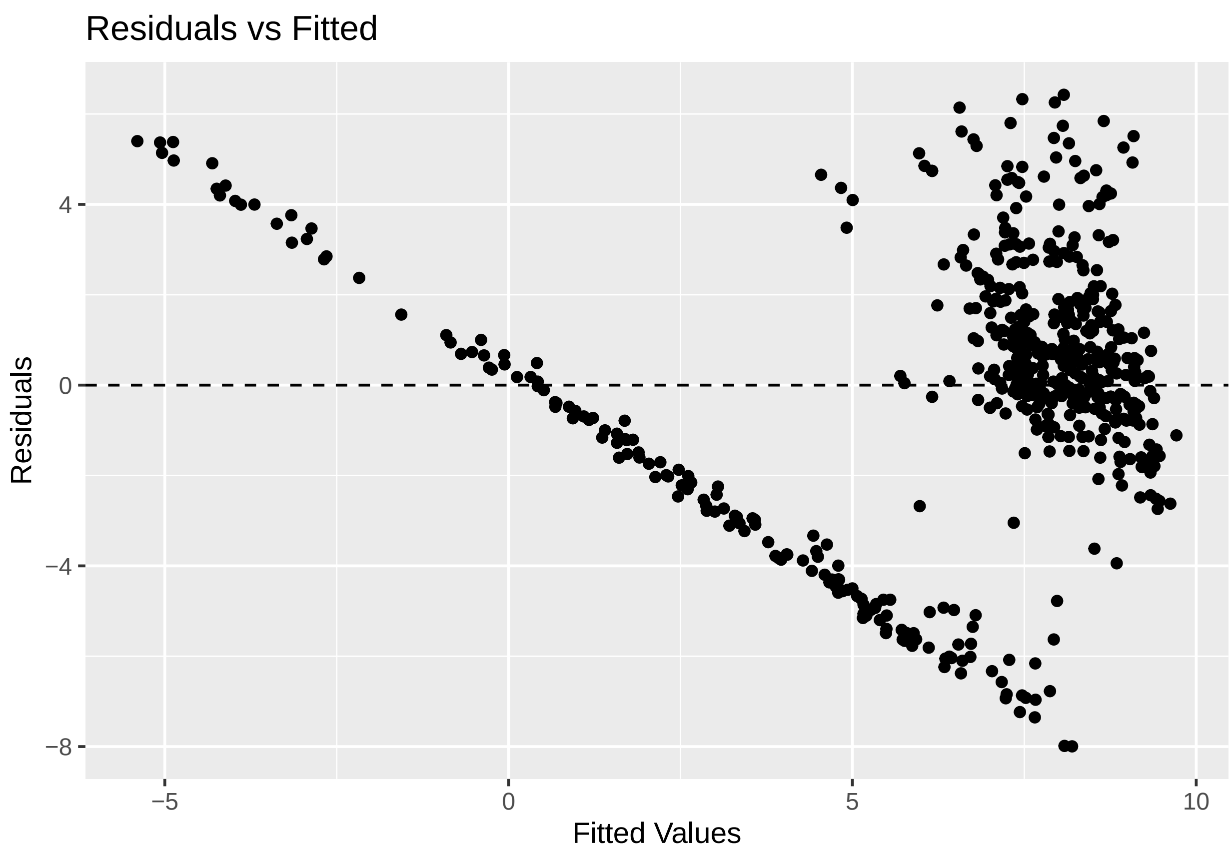


a. β_0 (Intercept): 9.31581

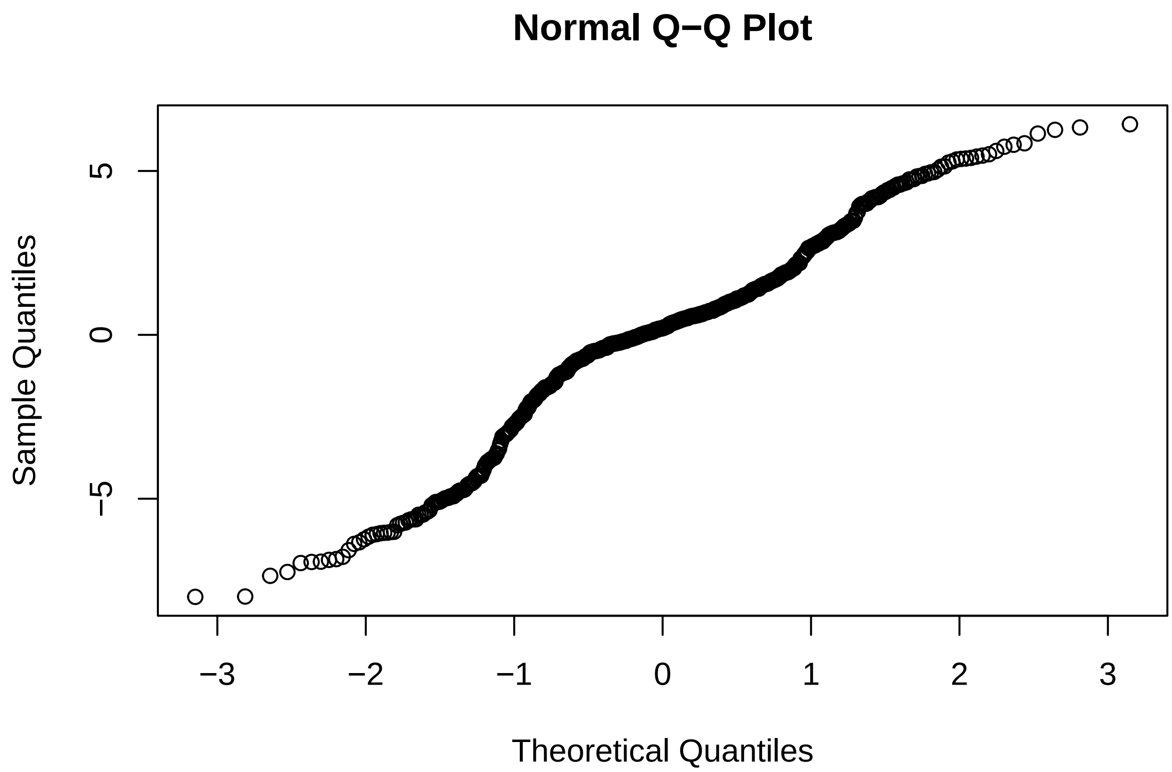
This represents the expected value of dissolved oxygen in surface water with zero total phosphorus. b. β_1 (tp_ug): -0.05593

This represents the change in dissolved oxygen for each unit increase in total phosphorus in surface water.

```r
# Residual Plot (Residuals vs Fitted)
residuals <- resid(fit)
fitted_values <- fitted(fit)

residual_plot <- data.frame(residuals = residuals, fitted_values = fitted_values)

ggplot(residual_plot, aes(x = fitted_values, y = residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = "Fitted Values", y = "Residuals") +
  ggtitle("Residuals vs Fitted")
```




```r
# QQ Plot (Normal Q-Q plot)
qq_plot <- qqnorm(residuals, main = "Normal Q-Q Plot")
```




```
# Random Effects Plot (Dot plot of random effects)
random_effects <- ranef(fit)$lakename

random_effects_plot <- data.frame(lakename = rownames(random_effects), random_effect = random_effects[,
```

APPENDIX