12: Generalized Linear Models (Linear Regression)

Environmental Data Analytics | Kateri Salk

Spring 2020

Objectives

2. Apply special cases of the GLM (linear regression) to real datasets
3. Interpret and report the results of linear regressions in publication-style formats
4. Apply model selection methods to choose model formulations

Set up

```
getwd()
```

```
## [1] "/Users/ks501/Documents/GitHub_Repos/Environmental_Data_Analytics_2020"
```

```
library(tidyverse)
options(scipen = 4)

PeterPaul.chem.nutrients <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Proc

# Set theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
```

Linear Regression

The linear regression, like the t-test and ANOVA, is a special case of the **generalized linear model** (GLM). A linear regression is comprised of a continuous response variable, plus a combination of 1+ continuous response variables (plus the error term). The deterministic portion of the equation describes the response variable as lying on a straight line, with an intercept and a slope term. The equation is thus a typical algebraic expression:

$$y = \alpha + \beta * x + \epsilon$$

The goal for the linear regression is to find a **line of best fit**, which is the line drawn through the bivariate space that minimizes the total distance of points from the line. This is also called a "least squares" regression. The remainder of the variance not explained by the model is called the **residual error.**

epsilon

The linear regression will test the null hypotheses that

1. The intercept (alpha) is equal to zero.
2. The slope (beta) is equal to zero

Whether or not we care about the result of each of these tested hypotheses will depend on our research question. Sometimes, the test for the intercept will be of interest, and sometimes it will not.

Important components of the linear regression are the correlation and the R-squared value. The **correlation** is a number between -1 and 1, describing the relationship between the variables. Correlations close to -1 represent strong negative correlations, correlations close to zero represent weak correlations, and correlations close to 1 represent strong positive correlations. The **R-squared value** is the correlation squared, becoming a number between 0 and 1. The R-squared value describes the percent of variance accounted for by the explanatory variables.

Simple Linear Regression

For the NTL-LTER dataset, can we predict irradiance (light level) from depth?

```
irradiance.regression <- lm(PeterPaul.chem.nutrients$irradianceWater ~ PeterPaul.chem.nutrients$depth)
# another way to format the lm function
irradiance.regression <- lm(data = PeterPaul.chem.nutrients, irradianceWater ~ depth)
summary(irradiance.regression)
```

```
##
## Call:
## lm(formula = irradianceWater ~ depth, data = PeterPaul.chem.nutrients)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -458.9  -144.1   -41.2    90.3 23813.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  486.818      4.063  119.82   <2e-16 ***
## depth        -95.890      1.153  -83.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 303.4 on 15449 degrees of freedom
##   (7557 observations deleted due to missingness)
## Multiple R-squared:  0.3091, Adjusted R-squared:  0.3091  test statistic is Multiple R2
## F-statistic:  6912 on 1 and 15449 DF,  p-value: < 2.2e-16
```

```
# Correlation
cor.test(PeterPaul.chem.nutrients$irradianceWater, PeterPaul.chem.nutrients$depth)
```

```
##
##  Pearson's product-moment correlation
##
## data:  PeterPaul.chem.nutrients$irradianceWater and PeterPaul.chem.nutrients$depth
## t = -83.137, df = 15449, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5667674 -0.5449776
## sample estimates:
##       cor
## -0.555968   (-0.555968)^2=0.3091 ( R-squared)
```

Question: How would you report the results of this test (overall findings and report of statistical output)?

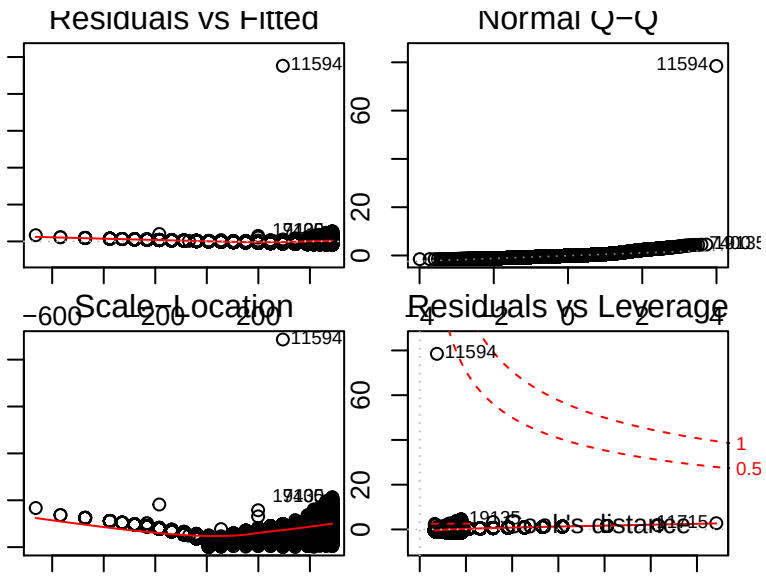
So, we see there is a significant negative correlation between irradiance and depth (lower light levels at greater depths), and that this model explains about 31 % of the total variance in irradiance. Let's visualize this

relationship and the model itself.

An exploratory option to visualize the model fit is to use the function `plot`. This function will return four graphs, which are intended only for checking the fit of the model and not for communicating results. The plots that are returned are:

1. **Residuals vs. Fitted.** The value predicted by the line of best fit is the fitted value, and the residual is the distance of that actual value from the predicted value. By definition, there will be a balance of positive and negative residuals. Watch for drastic asymmetry from side to side or a marked departure from zero for the red line - these are signs of a poor model fit.
2. **Normal Q-Q.** The points should fall close to the 1:1 line. We often see departures from 1:1 at the high and low ends of the dataset, which could be outliers.
3. **Scale-Location.** Similar to the residuals vs. fitted graph, this will graph the squared standardized residuals by the fitted values.
4. **Residuals vs. Leverage.** This graph will display potential outliers. The values that fall outside the dashed red lines (Cook's distance) are outliers for the model. Watch for drastic departures of the solid red line from horizontal - this is a sign of a poor model fit.

```
par(mfrow = c(2,2), mar=c(1,1,1,1))
plot(irradiance.regression)
```

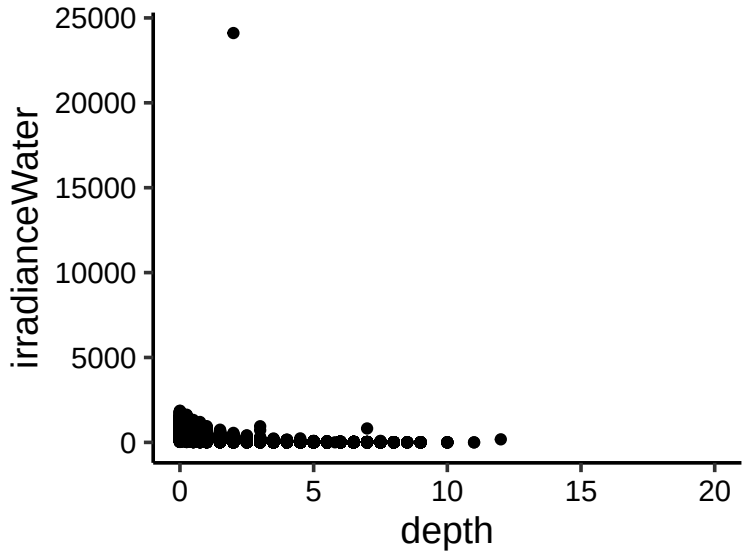



```
par(mfrow = c(1,1))
```

The option best suited for communicating findings is to plot the explanatory and response variables as a scatterplot.

```
# Plot the regression
irradiancebydepth <-
  ggplot(PeterPaul.chem.nutrients, aes(x = depth, y = irradianceWater)) +
  #ylim(0, 2000) +
  geom_point()
print(irradiancebydepth)
```

```
## Warning: Removed 7557 rows containing missing values (geom_point).
```

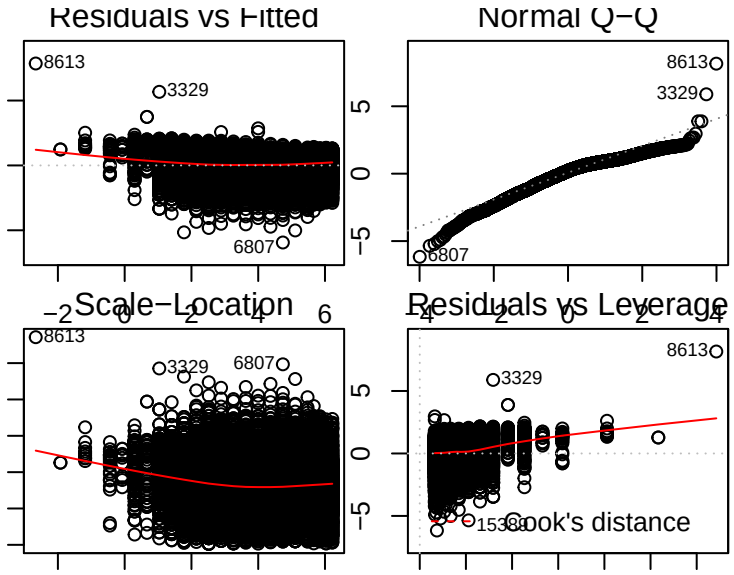


Given the distribution of irradiance values, we don't have a linear relationship between x and y in this case. Let's try log-transforming the irradiance values.

```
PeterPaul.chem.nutrients <- filter(PeterPaul.chem.nutrients,
                                   irradianceWater != 0 & irradianceWater < 5000)
irradiance.regression2 <- lm(data = PeterPaul.chem.nutrients, log(irradianceWater) ~ depth)
summary(irradiance.regression2)
```

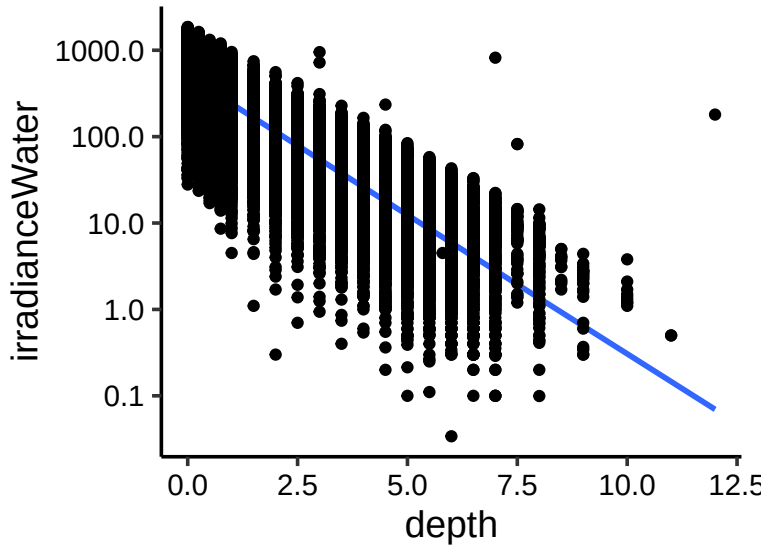
```
##
## Call:
## lm(formula = log(irradianceWater) ~ depth, data = PeterPaul.chem.nutrients)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9421 -0.5745  0.1930  0.7215  7.8568
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.218503   0.012903   481.9   <2e-16 ***
## depth       -0.740198   0.003664  -202.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9634 on 15445 degrees of freedom
## Multiple R-squared:  0.7255, Adjusted R-squared:  0.7254
## F-statistic: 4.081e+04 on 1 and 15445 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2,2), mar=c(1,1,1,1))
plot(irradiance.regression2)
```

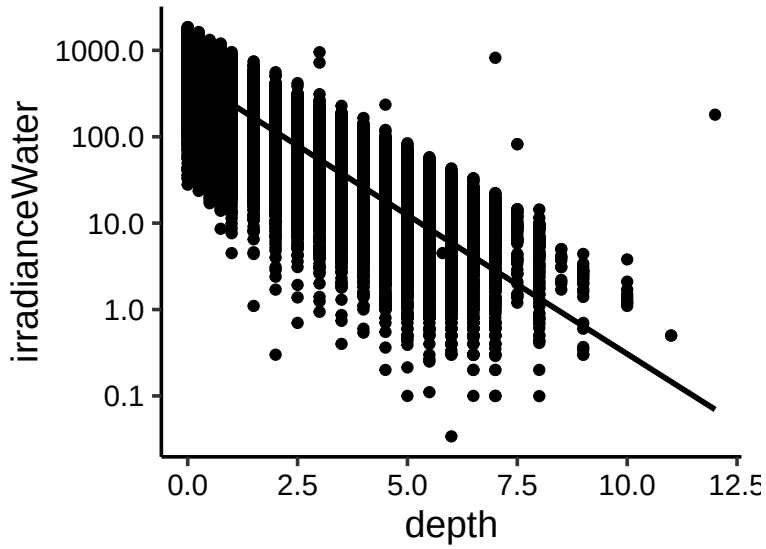



```
par(mfrow = c(1,1))

# Add a line and standard error for the linear regression
irradiancebydepth2 <-
  ggplot(PeterPaul.chem.nutrients, aes(x = depth, y = irradianceWater)) +
  geom_smooth(method = "lm") +
  scale_y_log10() +
  geom_point()
print(irradiancebydepth2)
```




```
# SE can also be removed
irradiancebydepth2 <-
    ggplot(PeterPaul.chem.nutrients, aes(x = depth, y = irradianceWater)) +
    geom_point() +
    scale_y_log10() +
    geom_smooth(method = 'lm', se = FALSE, color = "black")
print(irradiancebydepth2)
```




```
# Make the graph attractive
```

Non-parametric equivalent: Spearman's Rho

As with the t-test and ANOVA, there is a nonparametric variant to the linear regression. The **Spearman's rho** test has the advantage of not depending on the normal distribution, but this test is not as robust as the linear regression.

```
cor.test(PeterPaul.chem.nutrients$irradianceWater, PeterPaul.chem.nutrients$depth,
         method = "spearman", exact = FALSE)
```

```
##
##  Spearman's rank correlation rho
##
## data:  PeterPaul.chem.nutrients$irradianceWater and PeterPaul.chem.nutrients$depth
## S = 1147185713629, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## -0.8674653
```

Multiple Regression

It is possible, and often useful, to consider multiple continuous explanatory variables at a time in a linear regression. For example, total phosphorus concentration in Paul Lake (the unfertilized lake) could be dependent on depth and dissolved oxygen concentration:

```
TPregression <- lm(data = subset(PeterPaul.chem.nutrients, lakename == "Paul Lake"),
                   tp_ug ~ depth + dissolvedOxygen)
summary(TPregression)
```

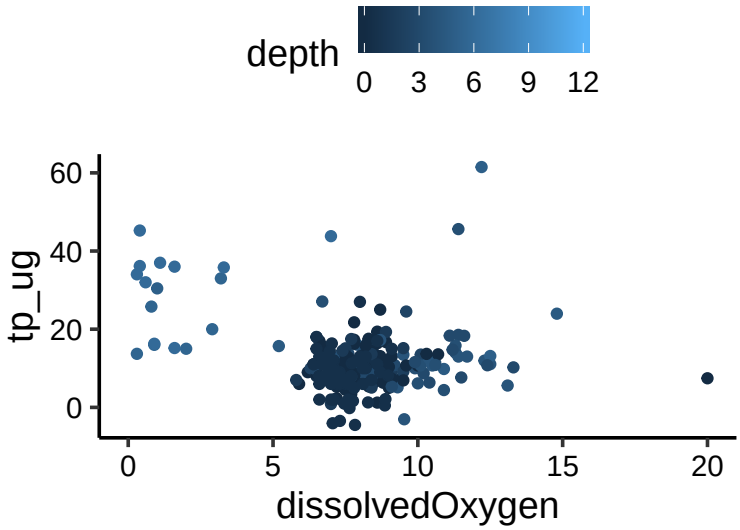
```
##
## Call:
## lm(formula = tp_ug ~ depth + dissolvedOxygen, data = subset(PeterPaul.chem.nutrients,
##     lakename == "Paul Lake"))
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -19.266  -3.436  -0.534   2.425  44.559
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      12.7869     1.4985   8.533 8.27e-16 ***
## depth             2.1691     0.2222   9.762  < 2e-16 ***
## dissolvedOxygen  -0.5494     0.1726  -3.184  0.00161 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.405 on 288 degrees of freedom
##   (7283 observations deleted due to missingness)
## Multiple R-squared:  0.2985, Adjusted R-squared:  0.2936
## F-statistic: 61.28 on 2 and 288 DF,  p-value: < 2.2e-16
```

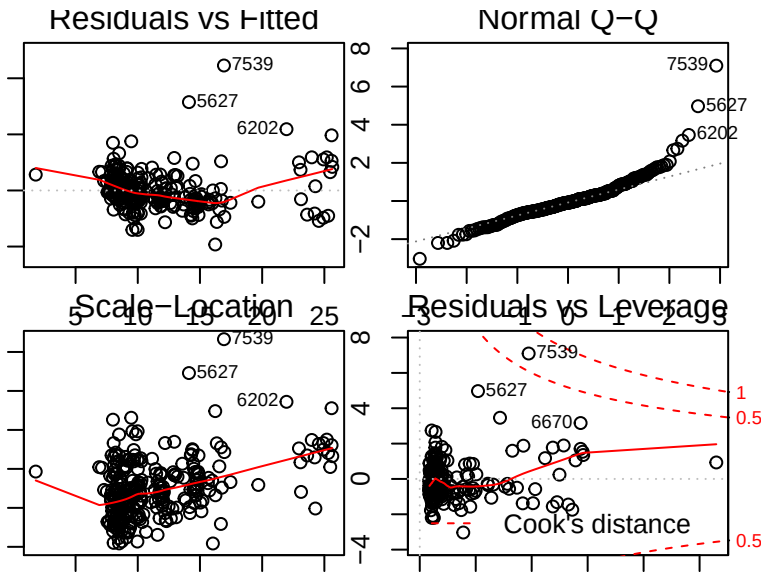
Large T value, small p value, so significant

```
TPplot <- ggplot(subset(PeterPaul.chem.nutrients, lakename == "Paul Lake"),
                 aes(x = dissolvedOxygen, y = tp_ug, color = depth)) +
  geom_point() +
  xlim(0, 20)
print(TPplot)
```

```
## Warning: Removed 7283 rows containing missing values (geom_point).
```




```
par(mfrow = c(2,2), mar=c(1,1,1,1))
plot(TPregression)
```




```
par(mfrow = c(1,1))
```

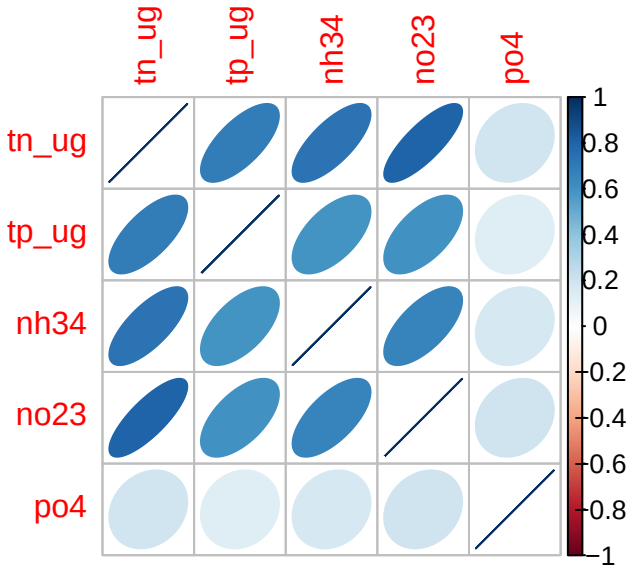
Correlation Plots

We can also make exploratory plots of several continuous data points to determine possible relationships, as well as covariance among explanatory variables.

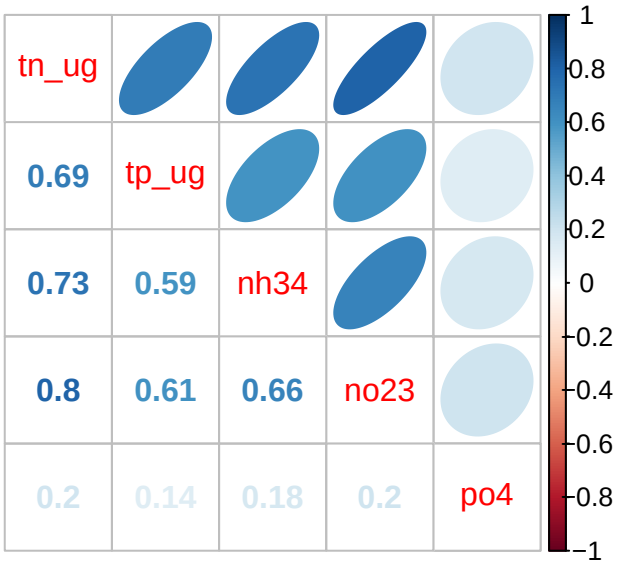
```
#install.packages("corrplot")
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
PeterPaulnutrients <-
  PeterPaul.chem.nutrients %>%
  select(tn_ug:po4) %>%
  na.omit()
PeterPaulCorr <- cor(PeterPaulnutrients)
corrplot(PeterPaulCorr, method = "ellipse")
```




```
corrplot.mixed(PeterPaulCorr, upper = "ellipse")
```



AIC to select variables

However, it is possible to over-parameterize a linear model. Adding additional explanatory variables takes away degrees of freedom, and if explanatory variables co-vary the interpretation can become overly complicated. Remember, an ideal statistical model balances simplicity and explanatory power! To help with this tradeoff, we can use the **Akaike's Information Criterion (AIC)** to compute a stepwise regression that either adds explanatory variables from the bottom up or removes explanatory variables from a full set of suggested options. The smaller the AIC value, the better.

Let's say we want to know which explanatory variables will allow us to best predict total phosphorus concentrations. Potential explanatory variables from the dataset could include depth, dissolved oxygen, temperature, PAR, total N concentration, and phosphate concentration.

```
Paul.naomit <- PeterPaul.chem.nutrients %>%
  filter(lakename == "Paul Lake") %>%
```

```
  na.omit()

TPAIC <- lm(data = Paul.naomit, tp_ug ~ depth + dissolvedOxygen +
              temperature_C + tn_ug + po4)
step(TPAIC)
```

```
## Start:  AIC=353.95
## tp_ug ~ depth + dissolvedOxygen + temperature_C + tn_ug + po4
##
##                   Df Sum of Sq    RSS    AIC
## - po4              1     1.111 2330.8 352.00
## - depth            1    16.739 2346.4 352.76
## - tn_ug            1    30.674 2360.3 353.43
## <none>                         2329.7 353.95
## - temperature_C    1   178.398 2508.1 360.29
## - dissolvedOxygen  1   232.194 2561.9 362.68
##
## Step:  AIC=352
## tp_ug ~ depth + dissolvedOxygen + temperature_C + tn_ug
##
##                   Df Sum of Sq    RSS    AIC
## - depth            1    15.745 2346.5 350.76
## - tn_ug            1    38.270 2369.1 351.84
## <none>                         2330.8 352.00
## - temperature_C    1   194.143 2524.9 359.04
## - dissolvedOxygen  1   250.981 2581.8 361.56
##
## Step:  AIC=350.76
## tp_ug ~ dissolvedOxygen + temperature_C + tn_ug
##
##                   Df Sum of Sq    RSS    AIC
## <none>                         2346.5 350.76 model has lowest AIC value
## - tn_ug            1     46.09 2392.6 350.96
## - dissolvedOxygen  1    305.67 2652.2 362.60
## - temperature_C    1    562.60 2909.1 373.05
```

```
##
## Call:
## lm(formula = tp_ug ~ dissolvedOxygen + temperature_C + tn_ug,
##     data = Paul.naomit)
##
## Coefficients:
##     (Intercept)  dissolvedOxygen    temperature_C            tn_ug
##       23.905781        -0.834405        -0.487603         0.007002
```

```
TPmodel <- lm(data = Paul.naomit, tp_ug ~ dissolvedOxygen + temperature_C + tn_ug)
summary(TPmodel)
```

```
##
## Call:
## lm(formula = tp_ug ~ dissolvedOxygen + temperature_C + tn_ug,
##     data = Paul.naomit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -12.9443  -2.8674  -0.4834   2.1666  14.4713
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     23.905781   3.586389   6.666 1.12e-09 ***
## dissolvedOxygen -0.834405   0.221438  -3.768 0.000267 ***
## temperature_C   -0.487603   0.095382  -5.112 1.37e-06 ***
## tn_ug            0.007002   0.004786   1.463 0.146304
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.64 on 109 degrees of freedom
## Multiple R-squared:  0.2774, Adjusted R-squared:  0.2575
## F-statistic: 13.95 on 3 and 109 DF,  p-value: 0.00000009208
```