11: Generalized Linear Models

Environmental Data Analytics | Kateri Salk

Spring 2019

LESSON OBJECTIVES

1. Describe the components of the generalized linear model (GLM)
2. Apply special cases of the GLM to real datasets
3. Interpret and report the results of GLMs in publication-style formats

y = alpha + Betha x + error

SET UP YOUR DATA ANALYSIS SESSION

```
getwd()
```

```
## [1] "C:/Users/Felipe/OneDrive - Duke University/1. DUKE/1. Ramos 2 Semestre/EOS-872 Env. Data Analyti
```

```
library(tidyverse)

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

# Set date to date format
PeterPaul.chem.nutrients$sampledate <- as.Date(PeterPaul.chem.nutrients$sampledate,
                                               format = "%Y-%m-%d")

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

SIMPLE AND MULTIPLE LINEAR REGRESSION (line of best fit, continous data)

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.**

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
##   (7921 observations deleted due to missingness)
## Multiple R-squared:  0.3091, Adjusted R-squared:  0.3091
## F-statistic:  6912 on 1 and 15449 DF,  p-value: < 2.2e-16
```

```
#output telling us: irradiance = 486.818(signif) - 95.89*depth + error (high error, only explaining
#30% of the response) R.squared penalises you if you use more explanatory variable

# Correlation (R value. you know the directionality)
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
```

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

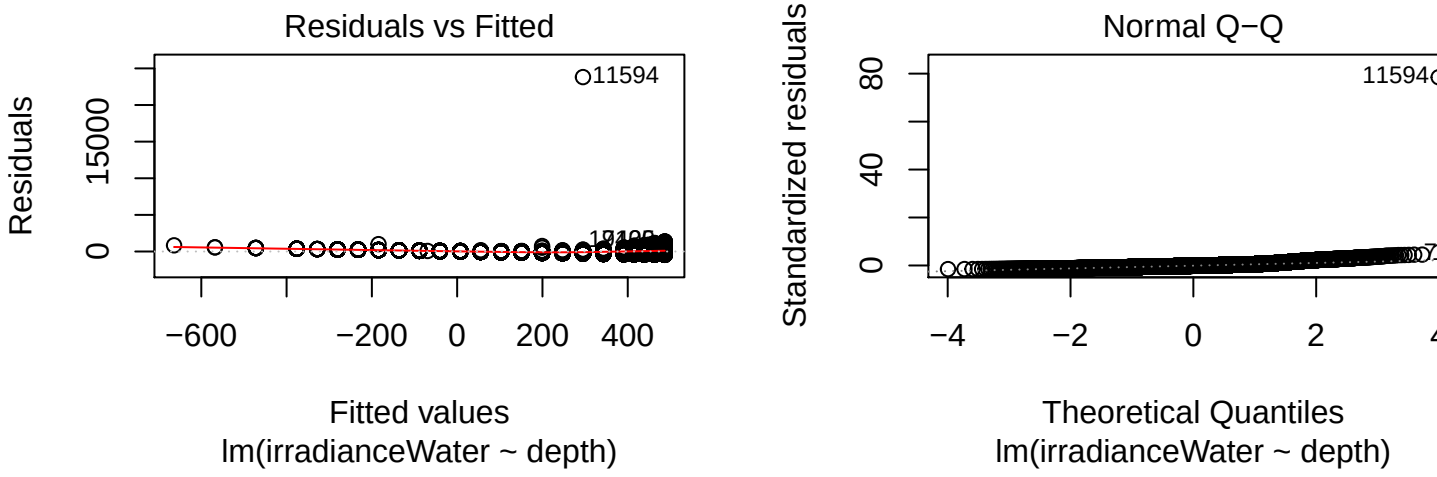
> ANSWER: significant negative correlation between irradiance and depth (lower light levels at greater depths), and that this model explains about 31 % of the total variance in irradiance (linear regression, R2 = 0.31, df = 15449, p < 0.0001(the depth p value in this case)). We dont use the Fstatistic. We use it only for ANOVA. If is part of your question you can report the pvalue of the intersept.

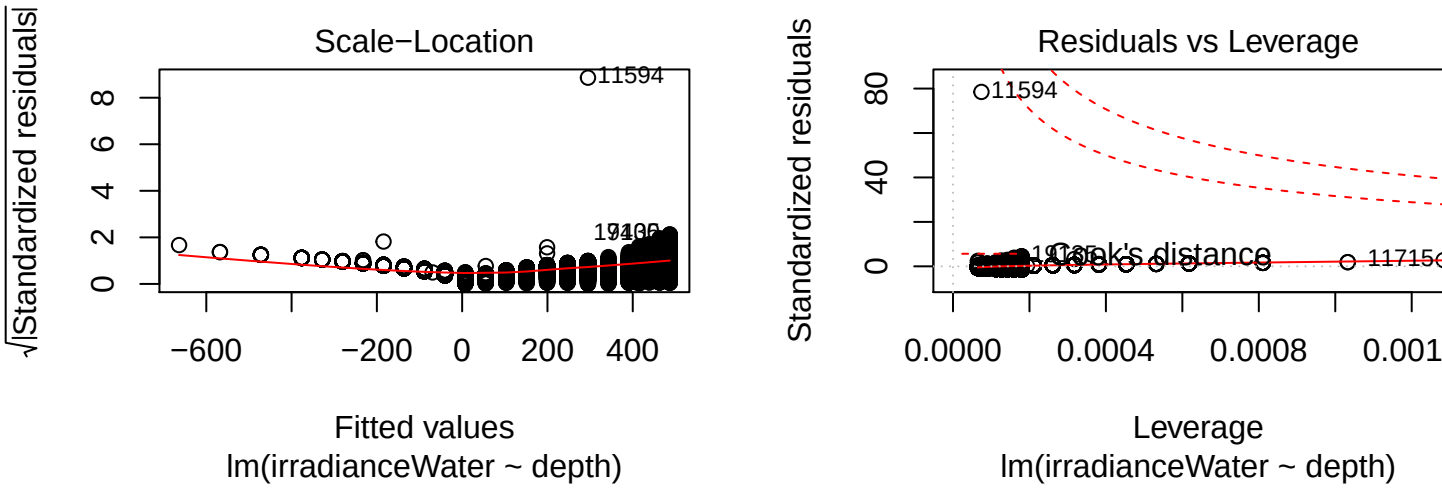
So, we see there is a significant negative correlation between irradiance and depth (lower light levels at greater depths), and that this model explains about 31 % of the total variance in irradiance. Let's visualize this relationship and the model itself.

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.

```
plot(irradiance.regression) #gives diff plots
```

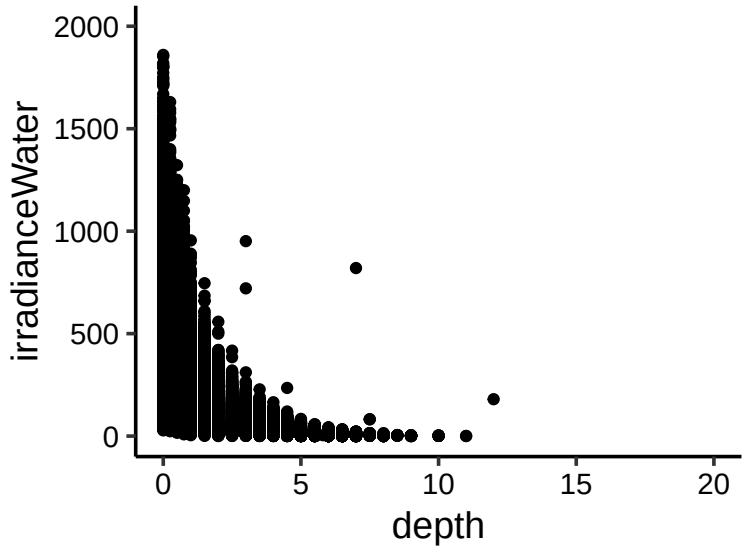




The option best suited for communicating findings is to plot the explanatory and response variables as a scatterplot.

```
# Plot the regression
irradiancebydepth <-
  ggplot(PeterPaul.chem.nutrients, aes(x = depth, y = irradianceWater)) +
  ylim(0, 2000) + #remove the outlier for plotting
  geom_point()
print(irradiancebydepth) # doesnt look linear. not good using linear regression. maybe logtransform.
```

```
## Warning: Removed 7922 rows containing missing values (geom_point).
```



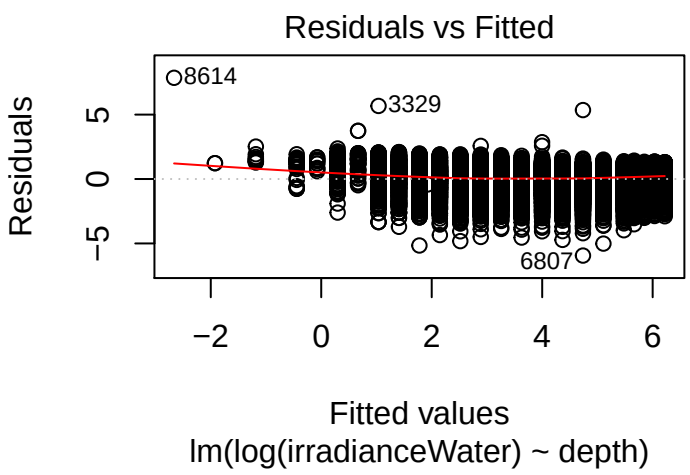
Given the distribution of irradiance values, we don't have a linear relationship between x and y in this case. Let's try log-transforming the irradiance values.

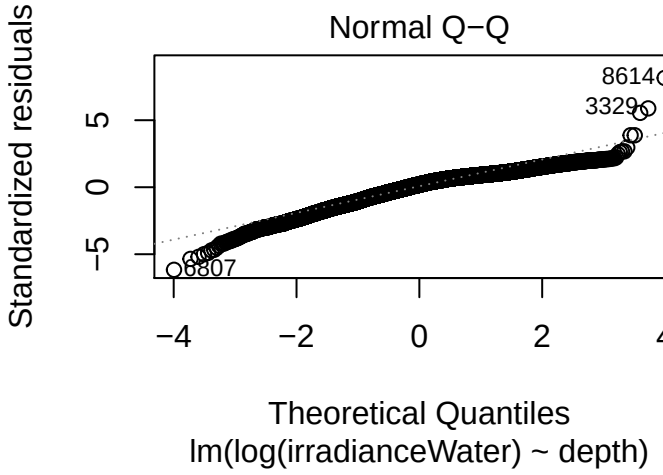
```
PeterPaul.chem.nutrients <- filter(PeterPaul.chem.nutrients, irradianceWater != 0)
irradiance.regression2 <- lm(data = PeterPaul.chem.nutrients, log(irradianceWater) ~ depth)
# we remove the 0 to be able to log  transform. they are only 3 values. not so concerned
```

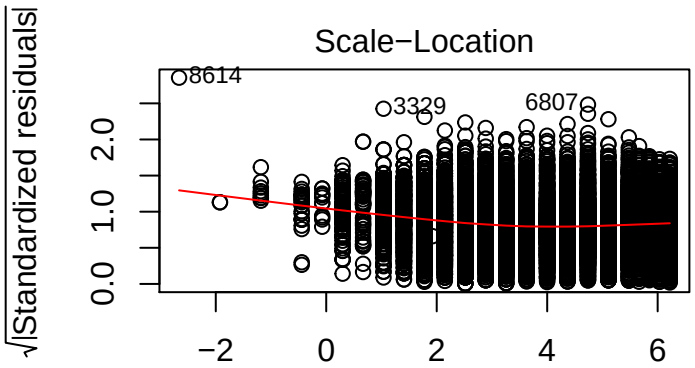
```
# on loosing the data. to few.
summary(irradiance.regression2) #log(irradiance) = 6.2 - 0.74(depth) + error. better r2
```

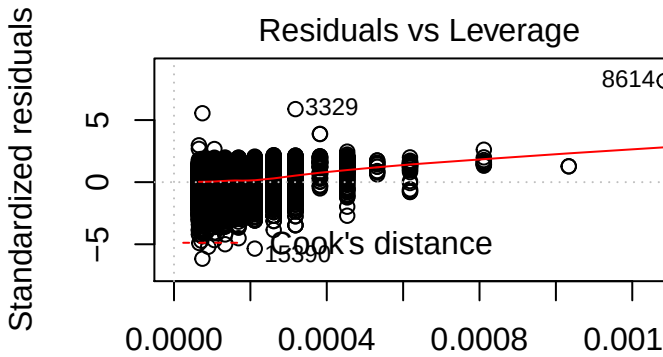
```
##
## Call:
## lm(formula = log(irradianceWater) ~ depth, data = PeterPaul.chem.nutrients)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9425 -0.5745  0.1931  0.7211  7.8571
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.219027   0.012916   481.5   <2e-16 ***
## depth       -0.740261   0.003668  -201.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9643 on 15446 degrees of freedom
## Multiple R-squared:  0.7251, Adjusted R-squared:  0.7251
## F-statistic: 4.074e+04 on 1 and 15446 DF,  p-value: < 2.2e-16
```

```
plot(irradiance.regression2) # better plots. keep an eye on the edges.
```

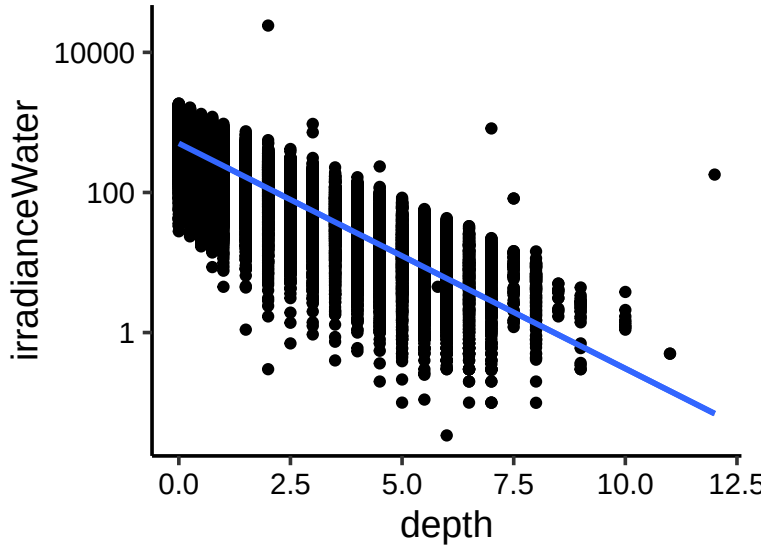




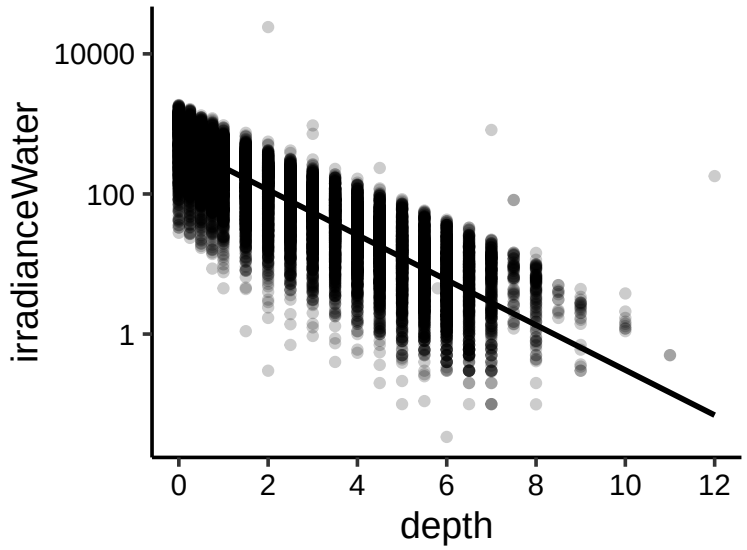





```
# Add a line and standard error for the linear regression
irradiancebydepth2 <-
  ggplot(PeterPaul.chem.nutrients, aes(x = depth, y = irradianceWater)) +
  geom_smooth(method = "lm") +
  scale_y_log10() + #change the scale of the plot
  geom_point() +
  geom_smooth(method = "lm")
print(irradiancebydepth2)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




```
# SE can also be removed
irradiancebydepth2 <-
    ggplot(PeterPaul.chem.nutrients, aes(x = depth, y = irradianceWater)) +
    geom_point(alpha = 0.2) +
    scale_y_log10() +
    geom_smooth(method = 'lm', se = FALSE, color = "black") + #remove the variance from the line
    scale_x_continuous(breaks = c(0,2,4,6,8,10,12))  # number in x axis
  print(irradiancebydepth2)
```




```
# Make the graph attractive
#you can change the transparency (alpha). or an anotation layer.
```

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) # here is how you do it. method
```

```
##
##  Spearman's rank correlation rho
##
## data:  PeterPaul.chem.nutrients$irradianceWater and PeterPaul.chem.nutrients$depth
## S = 1.1474e+12, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.8674103
```

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 could be dependent on depth and dissolved oxygen concentration:

```
TPregression <- lm(data = PeterPaul.chem.nutrients, tp_ug ~ depth + dissolvedOxygen)
summary(TPregression) #TP = 6 (0ox and 0 depth, not real) +1.5(depth) + 0.94(DO) + error
```

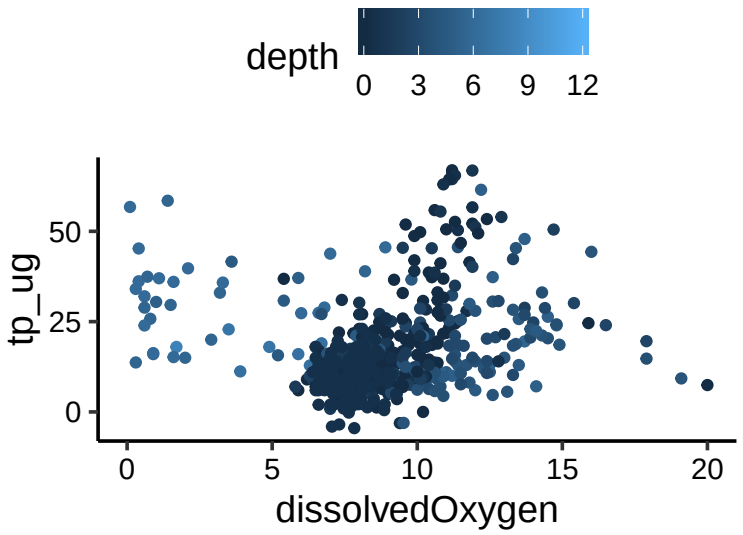
```
##
## Call:
## lm(formula = tp_ug ~ depth + dissolvedOxygen, data = PeterPaul.chem.nutrients)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.004  -6.889  -3.108   3.480  50.377
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.0031     1.7302   3.470 0.000557 ***
## depth             1.5041     0.2580   5.829 8.90e-09 ***
## dissolvedOxygen   0.9386     0.1824   5.147 3.53e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.85 on 633 degrees of freedom
##   (14812 observations deleted due to missingness)
## Multiple R-squared:  0.08214,    Adjusted R-squared:  0.07924
## F-statistic: 28.33 on 2 and 633 DF,  p-value: 1.652e-12
```

```
#low pvalues (significan results) but explaining low variance of the data

TPplot <- ggplot(PeterPaul.chem.nutrients,
                 aes(x = dissolvedOxygen, y = tp_ug, color = depth)) +
  geom_point() +
  xlim(0, 20)
print(TPplot) #showing us that our linear model is not a good predictor
```

```
## Warning: Removed 14812 rows containing missing values (geom_point).
```



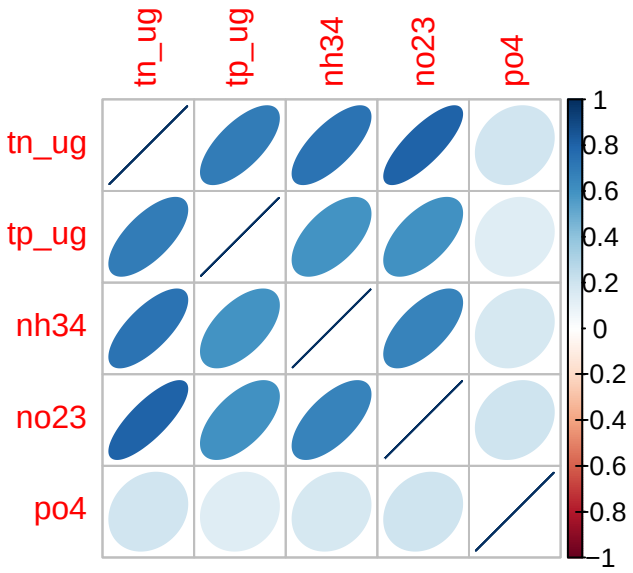
Correlation Plots We can also make exploratory plots of several continuous data points to determine possible relationships, as well as covariance among explanatory variables.

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

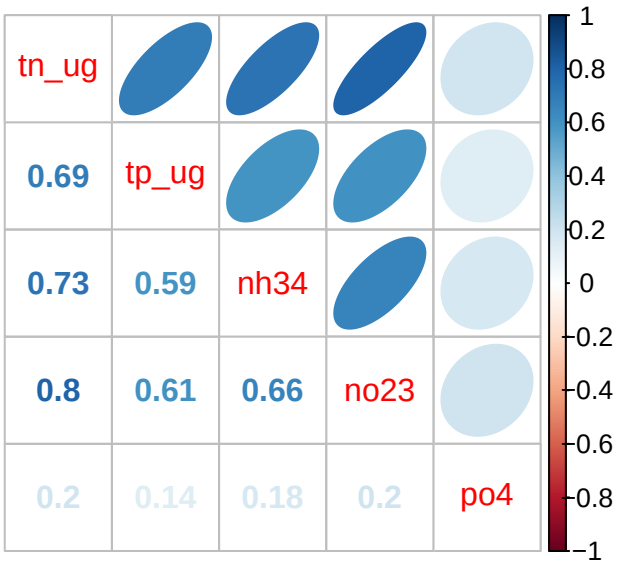
```
## corrplot 0.84 loaded
```

```
PeterPaulnutrients <-
  PeterPaul.chem.nutrients %>%
  select(tn_ug:po4) %>%
```

```
  na.omit() # with NAs you dont get correlations
PeterPaulCorr <- cor(PeterPaulnutrients)
corrplot(PeterPaulCorr, method = "ellipse") # without ellipse it gives you circles
```




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




```
# why not to use too many variables. you will find meaning less correlations.
# think about your questions. Overparametrization of your model. Runing out of degrees of freedom.
```

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. There is a BIC also. Mostly the same. no strong argument for either.

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.

```
PeterPaul.naomit <- na.omit(PeterPaul.chem.nutrients) #here we lose a lot of points
TPAIC <- lm(data = PeterPaul.naomit, tp_ug ~ depth + dissolvedOxygen +
              temperature_C + tn_ug + po4)
step(TPAIC) # the lower AIC value the better
```

```
## Start:  AIC=884.92
## tp_ug ~ depth + dissolvedOxygen + temperature_C + tn_ug + po4
##
##                   Df Sum of Sq     RSS     AIC
## - dissolvedOxygen  1       1.8  8921.6  882.96
## - po4              1      24.9  8944.6  883.59
## - depth            1      48.7  8968.5  884.23
## <none>                          8919.8  884.92
## - temperature_C    1     783.3  9703.1  903.29
## - tn_ug            1    9501.2 18421.0 1058.42
##
## Step:  AIC=882.96
## tp_ug ~ depth + temperature_C + tn_ug + po4
##
##                 Df Sum of Sq     RSS     AIC
## - po4            1      25.5  8947.1  881.65
## - depth          1      54.2  8975.8  882.43
## <none>                        8921.6  882.96
## - temperature_C  1     943.1  9864.7  905.28
## - tn_ug          1    9975.2 18896.8 1062.59
##
## Step:  AIC=881.65
## tp_ug ~ depth + temperature_C + tn_ug
##
##                 Df Sum of Sq     RSS     AIC
## - depth          1      52.3  8999.3  881.06
## <none>                        8947.1  881.65
## - temperature_C  1     921.1  9868.2  903.37
## - tn_ug          1   10251.2 19198.3 1064.42
##
## Step:  AIC=881.06
## tp_ug ~ temperature_C + tn_ug
##
##                 Df Sum of Sq     RSS     AIC
## <none>                        8999.3  881.06
## - temperature_C  1    1117.5 10116.8  907.39
## - tn_ug          1   10334.2 19333.6 1064.12
```

```
##
## Call:
## lm(formula = tp_ug ~ temperature_C + tn_ug, data = PeterPaul.naomit)
##
## Coefficients:
##   (Intercept)  temperature_C          tn_ug
```

```
##      10.06877       -0.46883        0.02794
```

```
TPmodel <- lm(data = PeterPaul.naomit, tp_ug ~ temperature_C + tn_ug)
# you run the best fit model on its own.
summary(TPmodel) #54% of the variance. better!
```

```
##
## Call:
## lm(formula = tp_ug ~ temperature_C + tn_ug, data = PeterPaul.naomit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.462  -3.942  -0.252   3.074  33.220
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   10.068766   1.744314   5.772 2.42e-08 ***
## temperature_C -0.468831   0.086061  -5.448 1.27e-07 ***
## tn_ug          0.027938   0.001686  16.567  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.136 on 239 degrees of freedom
## Multiple R-squared:  0.5418, Adjusted R-squared:  0.538
## F-statistic: 141.3 on 2 and 239 DF,  p-value: < 2.2e-16
```

ANCOVA #adds multiple levels of alphas (categorical variables)

Analysis of Covariance consists of a prediction of a continuous response variable by both continuous and categorical explanatory variables. We set this up in R with the `lm` function, just like prior applications in this lesson.

Let's say we wanted to predict total nitrogen concentrations by depth and by lake, similarly to what we did with a two-way ANOVA for depth ID and lake.

```
# main effects
TNancova.main <- lm(data = PeterPaul.chem.nutrients, tn_ug ~ lakename + depth)
summary(TNancova.main)
```

```
##
## Call:
## lm(formula = tn_ug ~ lakename + depth, data = PeterPaul.chem.nutrients)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -356.62 -120.28  -32.08   71.53 1564.56
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          353.085     16.005  22.061  < 2e-16 ***
## lakenamePeter Lake   135.361     20.326   6.659 8.52e-11 ***
## depth                 -9.716      6.347  -1.531    0.127
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 209.6 on 425 degrees of freedom
##   (15020 observations deleted due to missingness)
## Multiple R-squared:  0.09734,    Adjusted R-squared:  0.09309
## F-statistic: 22.91 on 2 and 425 DF,  p-value: 3.541e-10
```

```
#TN = 353(paul lake depth 0) + 135.361(peter diff) -9.7depth(not significant). very little variance.

# interaction effects
TNancova.interaction <- lm(data = PeterPaul.chem.nutrients, tn_ug ~ lakename * depth)
summary(TNancova.interaction)
```

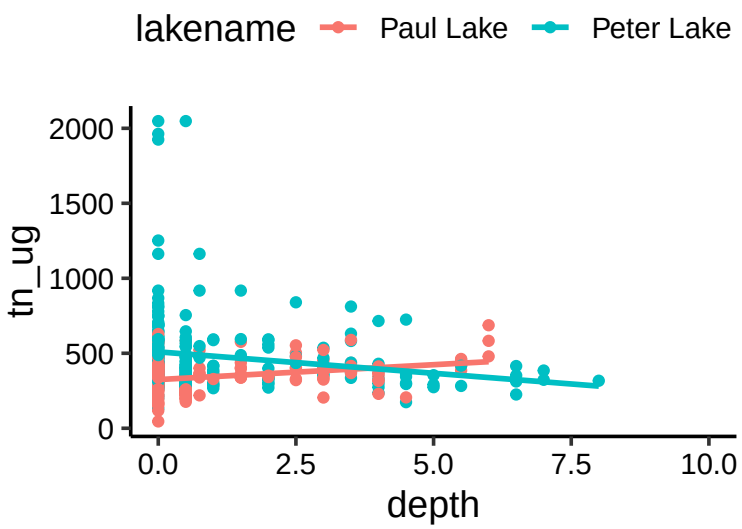
```
##
## Call:
## lm(formula = tn_ug ~ lakename * depth, data = PeterPaul.chem.nutrients)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -377.65 -108.22  -27.51   69.89 1552.95
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 324.53      17.48  18.566  < 2e-16 ***
## lakenamePeter Lake          184.95      23.94   7.726 8.09e-14 ***
## depth                        19.87      10.02   1.982 0.048073 *
## lakenamePeter Lake:depth    -48.42      12.82  -3.777 0.000182 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 206.4 on 424 degrees of freedom
##   (15020 observations deleted due to missingness)
## Multiple R-squared:  0.1267, Adjusted R-squared:  0.1205
## F-statistic: 20.51 on 3 and 424 DF,  p-value: 1.994e-12
```

```
# TN = 325 + 185(Peter) + 20(if paul)depth - 48(if peter)depth + error. Slightly more variance than the

TNplot <- ggplot(PeterPaul.chem.nutrients, aes(x = depth, y = tn_ug, color = lakename)) +
  #color in aes two lines +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  xlim(0, 10)
print(TNplot)
```

```
## Warning: Removed 15020 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 15020 rows containing missing values (geom_point).
```




```
# Make the graph attractive
```