4 Statistical Frameworks Study Guide (and other notes?)

EA30 2023 Students and Marc

2023-03-27

# R Study Guide Assignment

## Progress in Six Weeks

The study guide development was completed over a six week. While each topic rotated between groups, it was not always easy to track progress because each team had to ``start from scrach’’, each week. Thus, the idea of a progression wasn’t as successful as expected.

Nevertheless, we have created a very good document as a basic study guide and hope the product is useful.



Calystegia purpurata is a species of morning glory known by the common name Pacific false bindweed. It is endemic to California, where it grows in the seaside scrub of the coastline and the chaparral of the coastal and inland valleys.

# R and Rstudio

## What is R?

R is a language and environment for statistical computing and graphics. R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, etc.) and graphical techniques, and is highly extensible.

One of R’s strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formula where needed.

To get help with R, see [help function](#help) or [help.start function](#help.start)

## Using R Studio Server

We used the college’s RStudio Sever to develop the study guide. However, students can download R and Rstudio (which are free) to continue to have access to the program after the semester.



A deciduous climbing or trailing evergreen shrub. Native blackberry is not aggressive like it’s Himalayan cousin. The berries are edible and tasty. Flowers host beneficial insects. Tolerates salt spray in shoreline and bluff plantings.

# Variables and Statistics

## Defining Variable Types

### Continuous vs. Descrete vs. Categorical

Continuous data includes complex numbers and varying data values measured over a particular time interval. Discrete data is a numerical type of data that includes whole, concrete numbers with specific and fixed data values determined by counting, often these might be referrred to as ``count’’ data

Categorical variables represent types of data which may be divided into groups. Examples of categorical variables are race, sex, age group, and educational level. While the latter two variables may also be considered in a numerical manner by using exact values for age and highest grade completed, it is often more informative to categorize such variables into a relatively small number of groups.

We can interrogate the data if we are unsure using the str() function, which coerces the data.frame into a format to evaluate the structure.

str(pressure)

## 'data.frame': 19 obs. of 2 variables:  
## $ temperature: num 0 20 40 60 80 100 120 140 160 180 ...  
## $ pressure : num 0.0002 0.0012 0.006 0.03 0.09 0.27 0.75 1.85 4.2 8.8 ...

In this case, we find that the object “pressure” is a data.frame that is composed of two variables – and both seem to be numeric and continuous.

### Dependent vs. Independent Data

In some cases, we can manipulate an environmental setting or characteristic and test the impact on another characteristics. We generally call the one that we can manipulate as the independent variable and the measured response as the dependent variable.

Graphically, we often see the independent variable as the x-axis. The independent call might also be referred to as the predictor variable. The response as the dependent (or response) variable.

In our titles, we often refer to the dependent first – “Response as a Function of X” or “Water Temperature vs. Air Temperature”

NOTE: Spatial Data and Time Series data have special concerns. For example, when we take a sample from a creek every day – the creek water might be different, but the processes that contribute to its characteristics are quite similar, compared to water from another creek. Thus, we might note that the samples taken are autocorrelated.

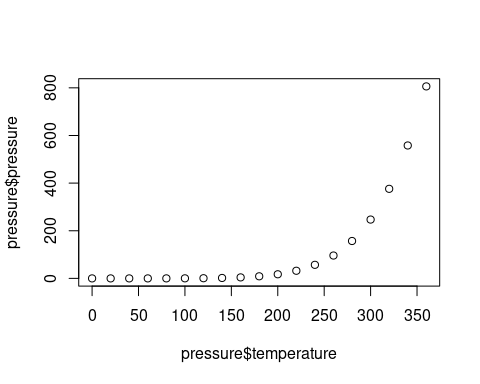
“Auto-correlation” occurs when data are not independent – for example, if we sample the same locations over time, we might expect each location to be self-correlated, i.e. auto-correlated. This violates a statistical assumption of indendence. There are ways around this violation, but that is beyone our study guide’s capacity to address, see [Autocorrelation Section](#autocorrelation) for more information.

#### Graphical Example of Independent and Dependent Variables

Plotting data as a data.frame is pretty straightforward, but there are two main syntax methods.

1. Define x and y in the plot function:

plot(x=pressure$temperature, y=pressure$pressure) # Notice the data.frame is explicitly referenced and then the variable of interest after the dollar symbol.



Notice that the x- and y-axes are labeled that show’s how we used the variables in the plot function.

1. Alternatively, you can use the formula syntax:

plot(pressure ~ temperature, data=pressure) # in this case the data.frame is called by the "data=" argument.



In this case, the x- and y-labels reflect the varible names without the data.frame reference. Plus, the formula syntax, helps us note the ‘y is a function of x’ relationship.

knitr::include\_graphics("https://github.com/marclos/beginnersluck/raw/master/Stats/EA30\_Study\_Guide/Calochortus\_plummerae.jpg")



Calochortus plummerae is a perennial herb (bulb) that is native to California, and endemic (limited) to California.

# Four Frameworks

## Linear Regression

Regression analysis is a set of statistical processes for estimating the relationships between a dependent variable (often called the ‘outcome’ or ‘response’ variable, or a ‘label’ in machine learning parlance) and one or more independent variables (often called ‘predictors’, ‘covariates’, ‘explanatory variables’ or ‘features’).

Linear regression is one of the most commonly used predictive analyses. It answers the question: Does a set of predictor variables do a good job in predicting an outcome (dependent) variable? It’s used to predict the value of a variable (the dependent variable) based on an input value (the independent/predictor variable), [Variable Types Section](#dependindepend). Linear regression can answer questions about the relationship between two variables or the strength of a cause/effect relationship. It also is helpful if you have multiple independent variables because you can find out which variable(s) best predicts the dependent variable.

The most common form of regression analysis is linear regression, in which one finds the line (or a more complex linear combination) that most closely fits the data according to a specific mathematical criterion. For example, the method of ordinary least squares computes the unique line (or hyperplane) that minimizes the sum of squared differences between the true data and that line (or hyperplane).

Remember, correlation is not synonymous with causation, so we need to be careful in how linear regression is interpreted.

The regression is cacl a straight line on the plot (a line of best fit) and will measure the distance between the actual values and the predicted values on the line to determine how well the line predicts the values.

Linear regression can be displayed as scatterplot. Linear regression can be used to estimate the coeffients for a ‘best fit line’, with an slope and intercept.

## Linear Regression Examples

### Example #1: Air and Water Temperatures

We know surface water temperatures depend on the sunlight. But what about surface air temperature? In this example, we compared or regressed the water temperature with air temperature. Thus, we might ask the following question:

Does water temperature depend on air temperature? This implies a cause and effect. Can you think of reasons why the temperatures might be correlated, without a cause and effect?

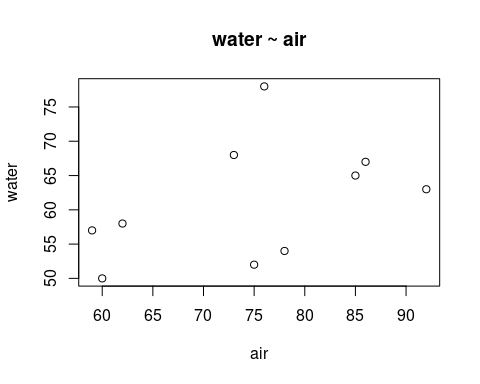
While correlation doesn’t say anything about cause and effect while regression expresses the cause and effect relationship between variables in the form of an equation. But we must be careful because the equation is not a mechanistic explanation, only a mathematical expression. In all cases, we need to have a clear explanation of how the cause and effect works and a good explanation that there are no confounding factors.

Below we create two vectors (water and air) of tempeature data. What does c() do? See [R Functons in Appendix](#appendix1)

water = c(50, 52, 65, 67, 68, 78, 54, 63, 57, 58)  
air = c(60, 75, 85, 86, 73, 76, 78, 92, 59,62)

Let’s look at the data:

plot(x=air, y=water, main="water ~ air")

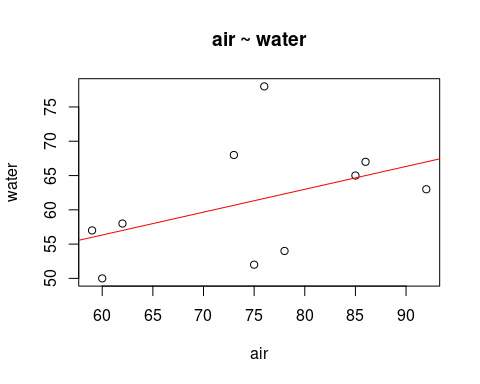


From here, we can coerce the data into a linear model and named object (airwater.lm):

airwater.lm <- lm(formula = water ~ air)

Next we can plot the data and then inlay the best fit line:

plot(x=air, y=water, main="air ~ water")   
abline(coef(airwater.lm), col='red')



We have a best fit line, but now we want to understand the statistical relationship between the two variables. We can extract that relationship using the summary() function:

summary(airwater.lm)

##   
## Call:  
## lm(formula = water ~ air)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.3333 -5.7500 0.6667 1.7500 16.3333   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 36.3333 18.0764 2.01 0.0793 .  
## air 0.3333 0.2398 1.39 0.2020   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 8.211 on 8 degrees of freedom  
## Multiple R-squared: 0.1945, Adjusted R-squared: 0.09386   
## F-statistic: 1.932 on 1 and 8 DF, p-value: 0.202

The null hypothesis is “there is no relationship” and we can reject that null hypothesis if the relationship could not random. So, we compare the results (slope) to see if the result could happen that is beyond the random assortment of measurements. There is lots of probability theory to understand this, which is beyond our course, but for now, let’s just say that the null hypothesis can be rejected if the probability is less than 0.05. Is this true? Be sure to look at the coefficient for predictor and it’s pvalue to make this decision.



Phacelia ramosissima is a perennial herb that is native to California, and also found elsewhere in western North America.

### Example #2: SRP and HABs

Harmful algal blooms (HABs) affect a wide range of ecosystem processes and are a public health threat. In general, HABs are most common in the summer when the water temperatures are highest and daylengths are longests. But for HABs to occur, three must be adequate nutrients.

In this example, we have phosphate concentrationss (as measured as soluable reactive phosphorous – SRP) and the concentration of cyanobacteria – a potential source of HABs.

The independent data are SRP and dependent is cyano:

SRP <- c(.02, .03, 0.4, 0.04, .05, 0.033, 1.4)  
cyano<- c(22, 34, 23, 22, 23, 2, 122)

These data can be analyzed using a linear regression (specifically a simple linear regression) because they are both quantitative forms of data and we are trying to determine if they have a cause and effect relationship, between an independent and dependent variable.

#### Create Linear Model using lm() function

The linear model coerces the data into a new object using a least squares method to calculate the slope and intercept (and other statistical date.)

The rcode is using the lm [linear regression model function] to create a linear model using a least squares approach:

HAB.lm = lm(cyano ~ SRP)  
HAB.lm

##   
## Call:  
## lm(formula = cyano ~ SRP)  
##   
## Coefficients:  
## (Intercept) SRP   
## 15.03 72.37

Using the model allows us to create a best fit line (with the slope and intercept) and illustrate the relationship between between SRP and HABs.

#### Model Summary (Results)

Let’s look at the model results with summary()

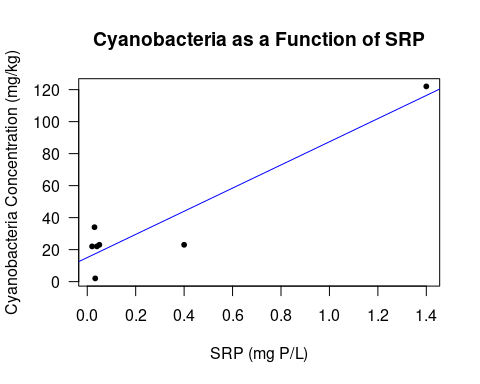
summary(HAB.lm)

##   
## Call:  
## lm(formula = cyano ~ SRP)  
##   
## Residuals:  
## 1 2 3 4 5 6 7   
## 5.523 16.799 -20.979 4.076 4.352 -15.418 5.647   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 15.029 6.398 2.349 0.06563 .   
## SRP 72.374 11.608 6.235 0.00155 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 14.55 on 5 degrees of freedom  
## Multiple R-squared: 0.886, Adjusted R-squared: 0.8632   
## F-statistic: 38.87 on 1 and 5 DF, p-value: 0.001554

he results of this linear regression demonstrate a relationship between SRP and HABs, but it is not a significant one (p-value = 0.06563). There some relationship between the data as demonstrated by the R2 vaue being 0.886, but the p-value demonstrates that it’s not a very significant relationship.

Let’s visualize the results:

plot(cyano ~ SRP, las=1, main="Cyanobacteria as a Function of SRP", pch=20, ylab="Cyanobacteria Concentration (mg/kg)", xlab="SRP (mg P/L)")  
abline(HAB.lm, col="blue")



Note: We added customized x and y labels and title, notice the units. The best fit line is in blue

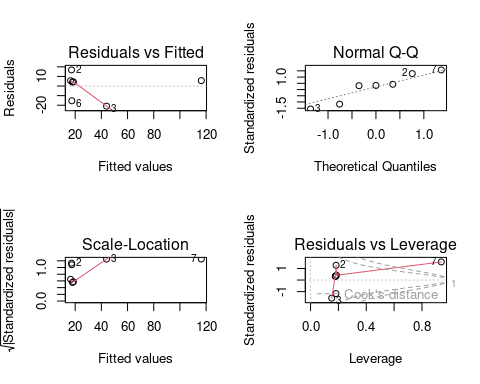
#### Model Interpretation (Discussion)

please add text here…what is the difference between reporting results and interpresting the results?

##### Linear Regression Assumptions

Let’s look at a default plot of model. By default, R produces four diagnostic plots to evaluate the assumptions in a linear regression:

par(mfrow=c(2,2))   
## mfrow allows you to creat a number of plot in one graph, you have to provide a vector of the number of rows and number of columns   
##the par() function creates multiple plots at one  
##Allows you to plot multiple points, the mfrow function allows you to split the screen into panels (basically split your screen)  
plot(HAB.lm)



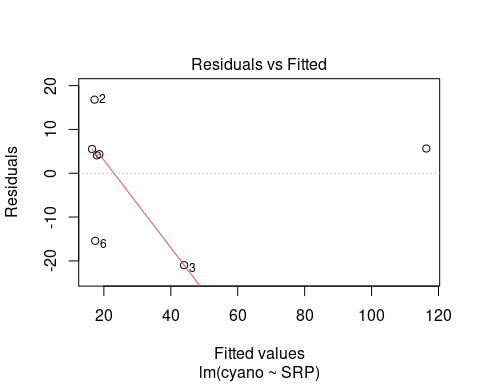
## plot(HAB.lm) is plot diagnostics

Now let’s look at one at a time:

1. Residuals vs Fitted

This plot shows if residuals have non-linear patterns. There could be a non-linear relationship between predictor variables and an outcome variable and the pattern could show up in this plot if the model doesn’t capture the non-linear relationship. If you find equally spread residuals around a horizontal line without distinct patterns, that is a good indication you don’t have non-linear relationships.

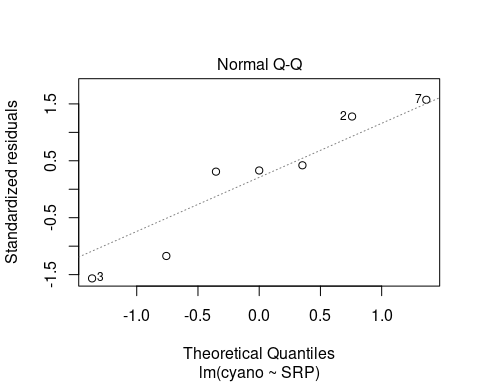
plot(HAB.lm, which=1)



1. Normal Q-Q

This plot shows if residuals are normally distributed. Do residuals follow a straight line well or do they deviate severely? It’s good if residuals are lined well on the straight dashed line.

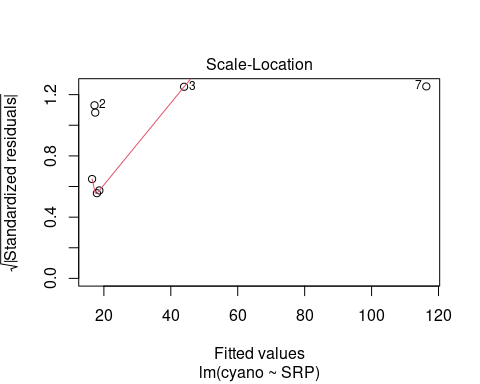
plot(HAB.lm, which=2)



1. Scale-Location

It’s also called Spread-Location plot. This plot shows if residuals are spread equally along the ranges of predictors. This is how you can check the assumption of equal variance (homoscedasticity). It’s good if you see a horizontal line with equally (randomly) spread points.

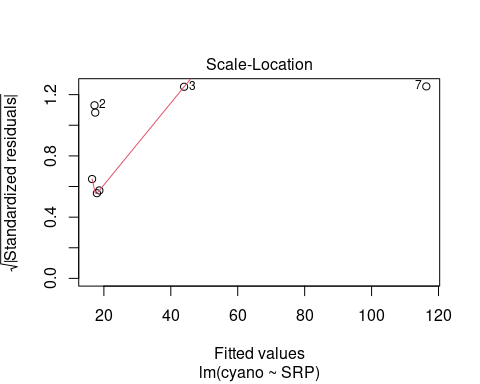
plot(HAB.lm, which=3)



1. Residuals vs Leverage

This plot helps us to find influential cases (i.e., subjects) if any. Not all outliers are influential in linear regression analysis (whatever outliers mean). Even though data have extreme values, they might not be influential to determine a regression line. That means, the results wouldn’t be much different if we either include or exclude them from analysis. They follow the trend in the majority of cases and they don’t really matter; they are not influential. On the other hand, some cases could be very influential even if they look to be within a reasonable range of the values. They could be extreme cases against a regression line and can alter the results if we exclude them from analysis. Another way to put it is that they don’t get along with the trend in the majority of the cases.

plot(HAB.lm, which=3)



## Ascombe Data

This dataset was created to show the importance of visualizing the data and looking at the assumptions

Anscombe = as.data.frame(  
 matrix(c(  
 1, 10, 10, 10, 8, 8.04, 9.14, 7.46, 6.58,   
 2, 8, 8, 8, 8, 6.95, 8.14, 6.77, 5.76,  
 3, 13, 13, 13, 8, 7.58, 8.74, 12.74, 7.71,  
 4, 9, 9, 9, 8, 8.81, 8.77, 7.11, 8.84,  
 5, 11, 11, 11, 8, 8.33, 9.26, 7.81, 8.47,  
 6, 14, 14, 14, 8, 9.96, 8.10, 8.84, 7.04,  
 7, 6, 6, 6, 8, 7.24, 6.13, 6.08, 5.25,  
 8, 4, 4, 4, 19, 4.26, 3.10, 5.39, 12.50,  
 9, 12, 12, 12, 8, 10.84, 9.13, 8.15, 5.56,  
 10, 7, 7, 7, 8, 4.82, 7.26, 6.42, 7.91,  
 11, 5, 5, 5, 8, 5.68, 4.74, 5.73, 6.89),   
 nrow=11, byrow=TRUE))  
  
  
names(Anscombe) = c("Obs.", "x1", "x2", "x3", "x4", "y1", "y2", "y3", "y4")  
str(Anscombe)

## 'data.frame': 11 obs. of 9 variables:  
## $ Obs.: num 1 2 3 4 5 6 7 8 9 10 ...  
## $ x1 : num 10 8 13 9 11 14 6 4 12 7 ...  
## $ x2 : num 10 8 13 9 11 14 6 4 12 7 ...  
## $ x3 : num 10 8 13 9 11 14 6 4 12 7 ...  
## $ x4 : num 8 8 8 8 8 8 8 19 8 8 ...  
## $ y1 : num 8.04 6.95 7.58 8.81 8.33 ...  
## $ y2 : num 9.14 8.14 8.74 8.77 9.26 8.1 6.13 3.1 9.13 7.26 ...  
## $ y3 : num 7.46 6.77 12.74 7.11 7.81 ...  
## $ y4 : num 6.58 5.76 7.71 8.84 8.47 7.04 5.25 12.5 5.56 7.91 ...

## Anscombe Results

Let’s look at the results. NOTE: I added a line, abline(), with the coefficients using coef() from the linear model, .

par(mfrow=c(2,2), las=1)  
plot(y1 ~ x1, Anscombe, main="Dataset A", pch=20)  
 A.lm = lm(y1 ~ x1, Anscombe); abline(coef(A.lm));   
 f <- summary(A.lm)$fstatistic  
 p <- round(pf(f[1],f[2],f[3],lower.tail=F),3)  
 lm.sum = summary(A.lm); str(lm.sum)

## List of 11  
## $ call : language lm(formula = y1 ~ x1, data = Anscombe)  
## $ terms :Classes 'terms', 'formula' language y1 ~ x1  
## .. ..- attr(\*, "variables")= language list(y1, x1)  
## .. ..- attr(\*, "factors")= int [1:2, 1] 0 1  
## .. .. ..- attr(\*, "dimnames")=List of 2  
## .. .. .. ..$ : chr [1:2] "y1" "x1"  
## .. .. .. ..$ : chr "x1"  
## .. ..- attr(\*, "term.labels")= chr "x1"  
## .. ..- attr(\*, "order")= int 1  
## .. ..- attr(\*, "intercept")= int 1  
## .. ..- attr(\*, "response")= int 1  
## .. ..- attr(\*, ".Environment")=<environment: R\_GlobalEnv>   
## .. ..- attr(\*, "predvars")= language list(y1, x1)  
## .. ..- attr(\*, "dataClasses")= Named chr [1:2] "numeric" "numeric"  
## .. .. ..- attr(\*, "names")= chr [1:2] "y1" "x1"  
## $ residuals : Named num [1:11] 0.039 -0.0508 -1.9213 1.3091 -0.1711 ...  
## ..- attr(\*, "names")= chr [1:11] "1" "2" "3" "4" ...  
## $ coefficients : num [1:2, 1:4] 3 0.5 1.125 0.118 2.667 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:2] "(Intercept)" "x1"  
## .. ..$ : chr [1:4] "Estimate" "Std. Error" "t value" "Pr(>|t|)"  
## $ aliased : Named logi [1:2] FALSE FALSE  
## ..- attr(\*, "names")= chr [1:2] "(Intercept)" "x1"  
## $ sigma : num 1.24  
## $ df : int [1:3] 2 9 2  
## $ r.squared : num 0.667  
## $ adj.r.squared: num 0.629  
## $ fstatistic : Named num [1:3] 18 1 9  
## ..- attr(\*, "names")= chr [1:3] "value" "numdf" "dendf"  
## $ cov.unscaled : num [1:2, 1:2] 0.82727 -0.08182 -0.08182 0.00909  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:2] "(Intercept)" "x1"  
## .. ..$ : chr [1:2] "(Intercept)" "x1"  
## - attr(\*, "class")= chr "summary.lm"

sum <- paste0("m= ", round(coef(A.lm)[2], 1), "; r2=", round(lm.sum$r.squared, 3), "; p=", p); sum

## [1] "m= 0.5; r2=0.667; p=0.002"

text(4, 9, sum, adj=0)  
plot(y2 ~ x2, Anscombe, main="Dataset B", pch=20)  
 B.lm = lm(y2 ~ x2, Anscombe); abline(coef(B.lm))  
 f <- summary(B.lm)$fstatistic  
 p <- round(pf(f[1],f[2],f[3],lower.tail=F),3)  
 lm.sum = summary(B.lm); str(lm.sum)

## List of 11  
## $ call : language lm(formula = y2 ~ x2, data = Anscombe)  
## $ terms :Classes 'terms', 'formula' language y2 ~ x2  
## .. ..- attr(\*, "variables")= language list(y2, x2)  
## .. ..- attr(\*, "factors")= int [1:2, 1] 0 1  
## .. .. ..- attr(\*, "dimnames")=List of 2  
## .. .. .. ..$ : chr [1:2] "y2" "x2"  
## .. .. .. ..$ : chr "x2"  
## .. ..- attr(\*, "term.labels")= chr "x2"  
## .. ..- attr(\*, "order")= int 1  
## .. ..- attr(\*, "intercept")= int 1  
## .. ..- attr(\*, "response")= int 1  
## .. ..- attr(\*, ".Environment")=<environment: R\_GlobalEnv>   
## .. ..- attr(\*, "predvars")= language list(y2, x2)  
## .. ..- attr(\*, "dataClasses")= Named chr [1:2] "numeric" "numeric"  
## .. .. ..- attr(\*, "names")= chr [1:2] "y2" "x2"  
## $ residuals : Named num [1:11] 1.139 1.139 -0.761 1.269 0.759 ...  
## ..- attr(\*, "names")= chr [1:11] "1" "2" "3" "4" ...  
## $ coefficients : num [1:2, 1:4] 3.001 0.5 1.125 0.118 2.667 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:2] "(Intercept)" "x2"  
## .. ..$ : chr [1:4] "Estimate" "Std. Error" "t value" "Pr(>|t|)"  
## $ aliased : Named logi [1:2] FALSE FALSE  
## ..- attr(\*, "names")= chr [1:2] "(Intercept)" "x2"  
## $ sigma : num 1.24  
## $ df : int [1:3] 2 9 2  
## $ r.squared : num 0.666  
## $ adj.r.squared: num 0.629  
## $ fstatistic : Named num [1:3] 18 1 9  
## ..- attr(\*, "names")= chr [1:3] "value" "numdf" "dendf"  
## $ cov.unscaled : num [1:2, 1:2] 0.82727 -0.08182 -0.08182 0.00909  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:2] "(Intercept)" "x2"  
## .. ..$ : chr [1:2] "(Intercept)" "x2"  
## - attr(\*, "class")= chr "summary.lm"

sum <- paste0("m= ", round(coef(B.lm)[2], 1), "; r2=", round(lm.sum$r.squared, 3), "; p=", p); sum

## [1] "m= 0.5; r2=0.666; p=0.002"

text(4, 9, sum, adj=0)  
  
plot(y3 ~ x3, Anscombe, main="Dataset C", pch=20)  
 C.lm = lm(y3 ~ x3, Anscombe); abline(coef(C.lm))  
 f <- summary(C.lm)$fstatistic  
 p <- round(pf(f[1],f[2],f[3],lower.tail=F),3)  
 lm.sum = summary(C.lm); str(lm.sum)

## List of 11  
## $ call : language lm(formula = y3 ~ x3, data = Anscombe)  
## $ terms :Classes 'terms', 'formula' language y3 ~ x3  
## .. ..- attr(\*, "variables")= language list(y3, x3)  
## .. ..- attr(\*, "factors")= int [1:2, 1] 0 1  
## .. .. ..- attr(\*, "dimnames")=List of 2  
## .. .. .. ..$ : chr [1:2] "y3" "x3"  
## .. .. .. ..$ : chr "x3"  
## .. ..- attr(\*, "term.labels")= chr "x3"  
## .. ..- attr(\*, "order")= int 1  
## .. ..- attr(\*, "intercept")= int 1  
## .. ..- attr(\*, "response")= int 1  
## .. ..- attr(\*, ".Environment")=<environment: R\_GlobalEnv>   
## .. ..- attr(\*, "predvars")= language list(y3, x3)  
## .. ..- attr(\*, "dataClasses")= Named chr [1:2] "numeric" "numeric"  
## .. .. ..- attr(\*, "names")= chr [1:2] "y3" "x3"  
## $ residuals : Named num [1:11] -0.54 -0.23 3.241 -0.39 -0.689 ...  
## ..- attr(\*, "names")= chr [1:11] "1" "2" "3" "4" ...  
## $ coefficients : num [1:2, 1:4] 3.002 0.5 1.124 0.118 2.67 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:2] "(Intercept)" "x3"  
## .. ..$ : chr [1:4] "Estimate" "Std. Error" "t value" "Pr(>|t|)"  
## $ aliased : Named logi [1:2] FALSE FALSE  
## ..- attr(\*, "names")= chr [1:2] "(Intercept)" "x3"  
## $ sigma : num 1.24  
## $ df : int [1:3] 2 9 2  
## $ r.squared : num 0.666  
## $ adj.r.squared: num 0.629  
## $ fstatistic : Named num [1:3] 18 1 9  
## ..- attr(\*, "names")= chr [1:3] "value" "numdf" "dendf"  
## $ cov.unscaled : num [1:2, 1:2] 0.82727 -0.08182 -0.08182 0.00909  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:2] "(Intercept)" "x3"  
## .. ..$ : chr [1:2] "(Intercept)" "x3"  
## - attr(\*, "class")= chr "summary.lm"

sum <- paste0("m= ", round(coef(C.lm)[2], 1), "; r2=", round(lm.sum$r.squared, 3), "; p=", p); sum

## [1] "m= 0.5; r2=0.666; p=0.002"

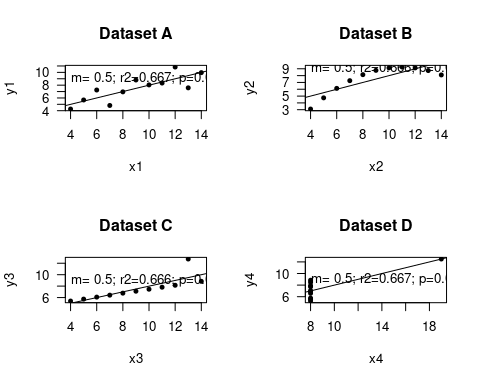
text(4, 9, sum, adj=0)  
  
plot(y4 ~ x4, Anscombe, main="Dataset D", pch=20)  
 D.lm = lm(y4 ~ x4, Anscombe); abline(coef(D.lm))  
 f <- summary(D.lm)$fstatistic  
 p <- round(pf(f[1],f[2],f[3],lower.tail=F),3)  
 lm.sum = summary(D.lm); str(lm.sum)

## List of 11  
## $ call : language lm(formula = y4 ~ x4, data = Anscombe)  
## $ terms :Classes 'terms', 'formula' language y4 ~ x4  
## .. ..- attr(\*, "variables")= language list(y4, x4)  
## .. ..- attr(\*, "factors")= int [1:2, 1] 0 1  
## .. .. ..- attr(\*, "dimnames")=List of 2  
## .. .. .. ..$ : chr [1:2] "y4" "x4"  
## .. .. .. ..$ : chr "x4"  
## .. ..- attr(\*, "term.labels")= chr "x4"  
## .. ..- attr(\*, "order")= int 1  
## .. ..- attr(\*, "intercept")= int 1  
## .. ..- attr(\*, "response")= int 1  
## .. ..- attr(\*, ".Environment")=<environment: R\_GlobalEnv>   
## .. ..- attr(\*, "predvars")= language list(y4, x4)  
## .. ..- attr(\*, "dataClasses")= Named chr [1:2] "numeric" "numeric"  
## .. .. ..- attr(\*, "names")= chr [1:2] "y4" "x4"  
## $ residuals : Named num [1:11] -0.421 -1.241 0.709 1.839 1.469 ...  
## ..- attr(\*, "names")= chr [1:11] "1" "2" "3" "4" ...  
## $ coefficients : num [1:2, 1:4] 3.002 0.5 1.124 0.118 2.671 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:2] "(Intercept)" "x4"  
## .. ..$ : chr [1:4] "Estimate" "Std. Error" "t value" "Pr(>|t|)"  
## $ aliased : Named logi [1:2] FALSE FALSE  
## ..- attr(\*, "names")= chr [1:2] "(Intercept)" "x4"  
## $ sigma : num 1.24  
## $ df : int [1:3] 2 9 2  
## $ r.squared : num 0.667  
## $ adj.r.squared: num 0.63  
## $ fstatistic : Named num [1:3] 18 1 9  
## ..- attr(\*, "names")= chr [1:3] "value" "numdf" "dendf"  
## $ cov.unscaled : num [1:2, 1:2] 0.82727 -0.08182 -0.08182 0.00909  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:2] "(Intercept)" "x4"  
## .. ..$ : chr [1:2] "(Intercept)" "x4"  
## - attr(\*, "class")= chr "summary.lm"

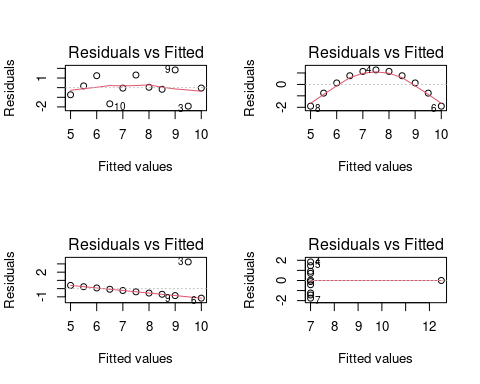
sum <- paste0("m= ", round(coef(D.lm)[2], 1), "; r2=", round(lm.sum$r.squared, 3), "; p=", p); sum

## [1] "m= 0.5; r2=0.667; p=0.002"

text(8, 9, sum, adj=0)

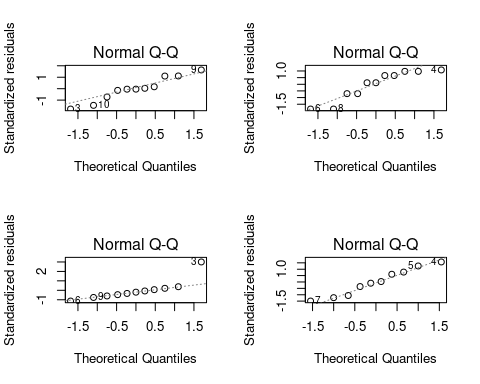
 ## Comparing the Diagonstics

par(mfrow=c(2,2))  
plot(A.lm, which =1)   
plot(B.lm, which =1)   
plot(C.lm, which =1)   
plot(D.lm, which =1)



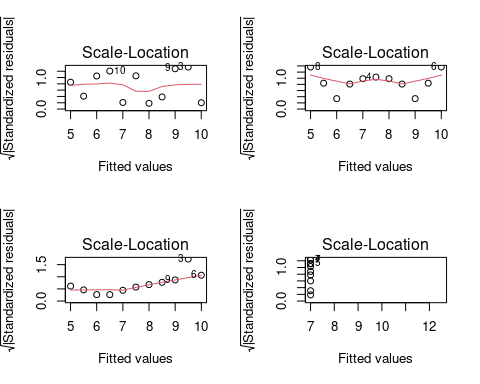
par(mfrow=c(2,2))  
plot(A.lm, which =2)   
plot(B.lm, which =2)   
plot(C.lm, which =2)   
plot(D.lm, which =2)

## Warning: not plotting observations with leverage one:  
## 8



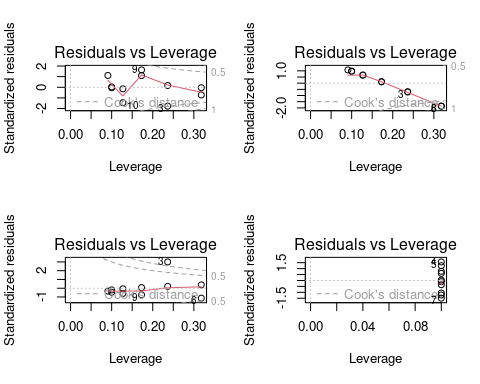
par(mfrow=c(2,2))  
plot(A.lm, which =3)   
plot(B.lm, which =3)   
plot(C.lm, which =3)   
plot(D.lm, which =3)

## Warning: not plotting observations with leverage one:  
## 8



par(mfrow=c(2,2))  
plot(A.lm, which =5)   
plot(B.lm, which =5)   
plot(C.lm, which =5)   
plot(D.lm, which =5)

## Warning: not plotting observations with leverage one:  
## 8





Desert Trumpet – Eriogonum inflatum.

## Logistic Regression

Logistic regression is a regression analysis to conduct when the dependent variable is dichotomous (binary).Like other regression analyses, the logistic regression is a predictive analysis.

Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

### Example #1: Algae Concentration and Toxicity

Toxicity is a measure of toxicity based on organism responses, often measured as LD50s, where 50% of the populations dies at a certain concentration.

Here we use the results of different concentrations of a toxin and the number of surviving Daphnia – a very common freshwater test organism.

In this case, we’ll coerce the data into a data frame.

Survived.1 = data.frame(Dose = rep(Dose, Survived), Result = 1)  
Died.1 = data.frame(Dose = rep(Dose, Died), Result = 0)  
  
Results.df = rbind(Survived.1, Died.1) # row bind   
str(Results.df)

## 'data.frame': 175 obs. of 2 variables:  
## $ Dose : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Result: num 1 1 1 1 1 1 1 1 1 1 ...

Results.df <- Results.df[order(Results.df$Dose),] # re-ordered observations by Dose

Here’s a reasonably nice table of results

table(Results.df)

## Result  
## Dose 0 1  
## 0 7 18  
## 0.5 6 19  
## 1 13 12  
## 1.5 20 5  
## 2 19 6  
## 2.5 23 2  
## 3 24 1

# Results.df$Result <- as.factor(Results.df$Result)  
Results.logit <- glm(Result ~ Dose, data = Results.df, family = "binomial")

### Results

We can extract the results using summary():

summary(Results.logit)

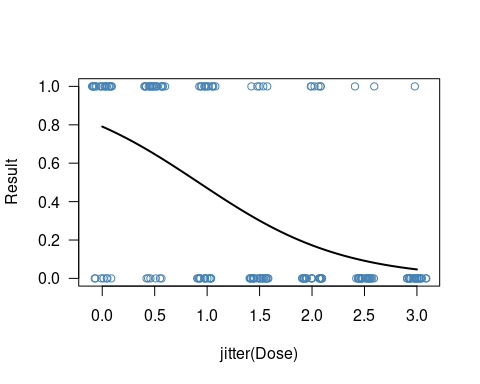
##   
## Call:  
## glm(formula = Result ~ Dose, family = "binomial", data = Results.df)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.7688 -0.8461 -0.4389 0.9334 2.4753   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.3296 0.3275 4.06 4.90e-05 \*\*\*  
## Dose -1.4485 0.2303 -6.29 3.18e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 228.70 on 174 degrees of freedom  
## Residual deviance: 171.04 on 173 degrees of freedom  
## AIC: 175.04  
##   
## Number of Fisher Scoring iterations: 4

Here we find the coefficient and the probability of the null hypothesis. In this case, we have a highly significant results, less then 0.001. Thus, there is a relationship between survival and dose.

#### Interpretation (Discussion)

# define new data frame that contains predictor variable  
newdata <- data.frame(Dose=seq(min(Results.df$Dose), max(Results.df$Dose),len=100))  
  
# use fitted model to predict values of vs  
newdata$Result = predict(Results.logit, newdata, type="response")

# plot logistic regression curve  
plot(Result ~ jitter(Dose), data=Results.df, col="steelblue", las=1)  
lines(Result ~ Dose, newdata, lwd=2)

 LD50 = dose at which Pr(dead | dose) = 0.5.

MLH: found this formula online… don’t know if it works…

Calculating Standard Error… MLH: Haven’t tried this yet

knitr::include\_graphics("https://github.com/marclos/beginnersluck/raw/master/Stats/EA30\_Study\_Guide/Clematis\_lasiantha.jpg")



Chaparral Clematis is a creamy-white flowering vine, belonging to the Ranunculaceae family, of the large Clematis genus. It is found on the Pacific coast of North America, from the San Francisco Bay Area southwards into Baja California. It extends as far east as the western foothills of the Sierra Nevada mountains, at elevations below 1,800 meters. It grows on hillsides, in chaparral, and in open woodland. Its leaves are 3-lobed, and generally grow groups of three to five leaflets, the largest leaves on the plant normally being between 3 and 5 centimeter in size. It can be distinguished from the similar (but much more widely ranging) Virgin’s Bower (Clematis ligusticifolia) by the fact that lasiantha normally only has one flower on each stalk, and at most three, whereas ligusticifolia has multiple flowers on each stem. Virgin’s Bower is more likely to be found along streams or in other wet places, whereas the lasiantha tolerates more open, drier places.

# ANOVA

The one-way analysis of variance (ANOVA) is used to determine whether there are any statistically significant differences between the means of two or more independent (unrelated) groups (although you tend to only see it used when there are a minimum of three, rather than two groups). For example, you could use a one-way ANOVA to understand whether a restoration success (% native plants) differed based on different methods, dividing methods into three independent groups (e.g., hand weeding, herbicide, no-weeding).

## Example 1: Simulated HAB Data

Five mesocosms have been randomly selected to test the impact of zebra mussels on cyanobacteria concentrations, while five are controls. Water is circulate in each, and the experiment sampled the water column for bacteria counts every day for 10 days.

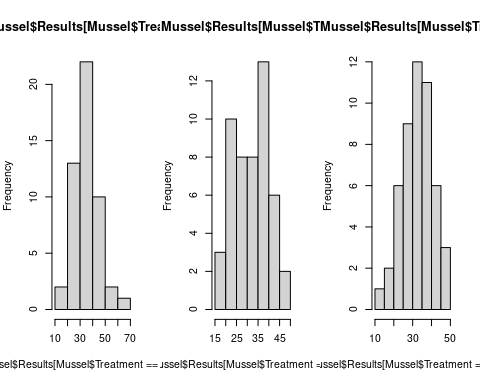
Controls have a mean of 34 (and Standard Deviation of 8), Low Density treatment has a mean of 33 (SD = 8) and high density mean of 31 (SD=8)

### First six observations and histograms

head(Mussel)

## Mesocosm Treatment Day Results  
## 1 1 Control 1 50.7  
## 2 2 Control 1 16.0  
## 3 3 Control 1 40.2  
## 4 4 Control 1 35.7  
## 5 5 Control 1 20.1  
## 6 6 Control 1 34.6

par(mfrow=c(1,3))  
hist(Mussel$Results[Mussel$Treatment=="Control"])  
hist(Mussel$Results[Mussel$Treatment=="LD"])  
hist(Mussel$Results[Mussel$Treatment=="HD"])



### Create AOV Object (Analysis of Variance) Using Least Squares Method

Least-squares means are predictions from a linear model, or averages thereof. They are useful in the analysis of experimental data for summarizing the effects of factors, in this case, i.e. testing linear contrasts among predictions.

Mussel.aov <- aov(results ~ treatment, data=Mussel)

### Results

summary(Mussel.aov)

## Df Sum Sq Mean Sq F value Pr(>F)  
## treatment 2 214 107.01 1.48 0.231  
## Residuals 147 10627 72.29

In this case, we cannot reject the null hypothesis.

### Exploring Chaning Parameters

#### Change Mean

Let’s change the data to see if the results change. In this case, we’ll change the means to 37, 33, and 31, respectively.

set.seed(334)  
results = round(c(rnorm(50, 37, 8), rnorm(50, 33, 8), rnorm(50, 31, 8)), 1)  
Mussel2 = data.frame(Mesocosm = mesocosm, Treatment = treatment, Day=day, Results=results)  
summary(aov(aov(results ~ treatment, data=Mussel2)))

## Df Sum Sq Mean Sq F value Pr(>F)   
## treatment 2 1015 507.4 7.019 0.00123 \*\*  
## Residuals 147 10627 72.3   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Now we should reject the null hypothesis.

#### Change SD

A lower standard deviation meant that the data were more clustered around the mean

set.seed(334)  
results = round(c(rnorm(50, 34, 4), rnorm(50, 33, 4), rnorm(50, 31, 4)), 1)  
Mussel3 = data.frame(Mesocosm = mesocosm, Treatment = treatment, Day=day, Results=results)  
summary(aov(aov(results ~ treatment, data=Mussel3)))

## Df Sum Sq Mean Sq F value Pr(>F)   
## treatment 2 183.7 91.83 5.072 0.00741 \*\*  
## Residuals 147 2661.6 18.11   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Nowwe also have a reason to reject the null hypothesis.

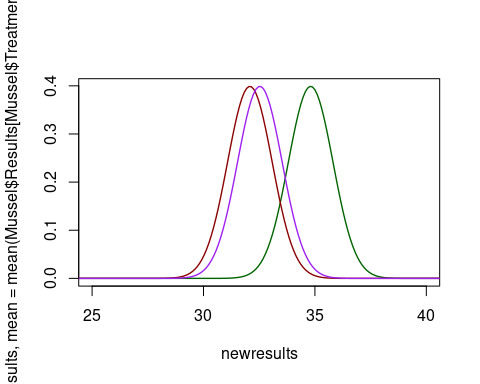
### Some graphical Displays

Using a normal distribution, what do the data look like?

The data is symmetrically distributed with no skew.

MLH: What do these mean?

newresults = seq(min(Mussel$Results), max(Mussel$Results), length=500)  
plot(newresults, dnorm(newresults, mean=mean(Mussel$Results[Mussel$Treatment=="Control"])), ty="l", xlim=c(25,40), col="darkgreen", lwd=1.4)  
lines(newresults, dnorm(newresults, mean=mean(Mussel$Results[Mussel$Treatment=="LD"])), ty="l", col="darkred", lwd=1.4)  
lines(newresults, dnorm(newresults, mean=mean(Mussel$Results[Mussel$Treatment=="HD"])), ty="l", col="purple", lwd=1.4)

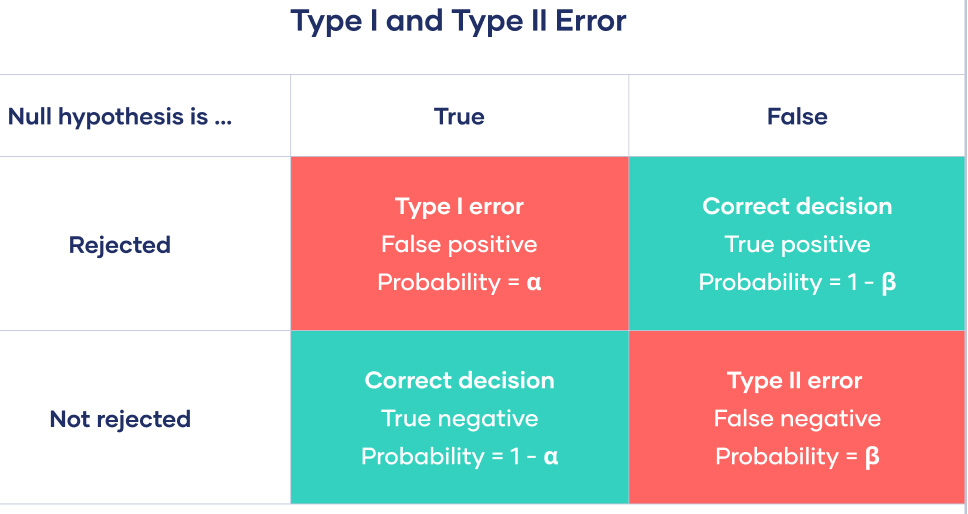


### Power Analaysis – ANOVA and Sample Size

Use this example: <https://stats.oarc.ucla.edu/r/dae/one-way-anova-power-analysis/> and create an “environmental example” and translate the results for class.

First, we need to distinguish between two types of potential mistakes that might be made regarding the rejection of the null hypothesis. Unfortunately, this has been named as Type I and Type II error – but think of these as two types of mistakes.

knitr::include\_graphics("https://github.com/marclos/beginnersluck/raw/master/Stats/EA30\_Study\_Guide/type-i-ii-error.jpg")



Calochortus plummerae is a perennial herb (bulb) that is native to California, and endemic (limited) to California.

### Power Analaysis – ANOVA and Sample Size

A power analysis helps determine an appropriate sample size. To do so, it requires 5 types of data: the number of groups, the between group variance, the within group variance, the alpha level, and the sample size/power.

The number of groups refers to the number of treatments/categories within an analysis of variants. It is often represented as “groups”. The between group variance refers to the difference between group, the within group variance refers to the difference within each individual group, the alpha level relates to the probability of rejecting the null hypothesis when it is true, and the power refers to the probability the test will determine a statistically significant when that difference exists.

Next steps after running the initial Power Anova Test include plotting the results and analyzing effect size.

The between group variance references Use this example: <https://stats.oarc.ucla.edu/r/dae/one-way-anova-power-analysis/> and create an “environmental example” and translate the results for class

groupmeans <- c(550, 598, 598, 646)  
n <- c(seq(2,10,by=1),seq(12,20,by=2),seq(25,50,by=5))  
groups = 4  
 n = NA  
 between.var = 1536  
 within.var = 6400  
 sig.level = 0.05  
 power = .823  
p <- power.anova.test(groups = length(groupmeans),   
between.var = var(groupmeans), within.var = 6400,   
power=0.823,sig.level=0.05,n=NULL)  
p

##   
## Balanced one-way analysis of variance power calculation   
##   
## groups = 4  
## n = 16.98893  
## between.var = 1536  
## within.var = 6400  
## sig.level = 0.05  
## power = 0.823  
##   
## NOTE: n is number in each group

##### Playing with N to see the impact on power:

Let’s continue with 4 treatments – or groups and vary the number of replicates and it’s effect on the power to avoid making a Type II error.

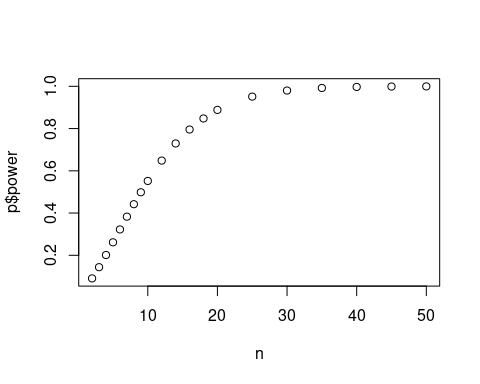
groupmeans <- c(550, 598, 598, 646)  
between.var = var(groupmeans)  
within.var = 6400  
sig.level = 0.05  
  
n <- c(seq(2,10,by=1),seq(12,20,by=2),seq(25,50,by=5))  
  
# power = null  
p <- power.anova.test(groups = length(groupmeans),   
 between.var = var(groupmeans),   
 within.var = 6400,   
 n = n, power= NULL,  
 sig.level=0.05)

## Warning in !is.null(n) && n < 2: 'length(x) = 20 > 1' in coercion to  
## 'logical(1)'

p

##   
## Balanced one-way analysis of variance power calculation   
##   
## groups = 4  
## n = 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 25, 30, 35, 40, 45, 50  
## between.var = 1536  
## within.var = 6400  
## sig.level = 0.05  
## power = 0.09067458, 0.14381195, 0.20139579, 0.26146007, 0.32241923, 0.38293139, 0.44190044, 0.49846996, 0.55200591, 0.64840471, 0.72949120, 0.79552098, 0.84785780, 0.88840023, 0.95127832, 0.98006724, 0.99226927, 0.99713328, 0.99897699, 0.99964689  
##   
## NOTE: n is number in each group

plot(n, p$power)

 You can see that with more replicates, you are more likely to find a reason to reject the null hypothesis.



Marah fabacea is a perennial herb or vine that is native to California, and endemic (limited) to California.

## Tests for Association

This test is used to determine association between two treatments (or differences). The chi-squared test assumes that both variables are are categorical and independent. Assumes that the values are mutually exclusive as well. Data must be frequency data and this test is used when there are larger values. It makes no assumption about the distribution of the data.

An example could be Harmful Algal Blooms (HABs) as a treatment with Hospital Asthma Cases as an output. Could show the relation between HABs and Asthma Cases going to the hospital. Chi-Square tests for independence/association can only be used between categorical variables.

There are several tests for association. One of the more popular is the Chi-squared test ( test). A Chi-Squared test is used to determine the association between two categorical variables. The null hypothesis is no association between the two independent variables.

Strengths of this test is how it might measure how a model compares the actual observed data, tells us if two variables are independent of one another.

The data in the cells should be frequencies, or counts of cases rather than percentages or some other transformation of the data. Categories of the variables are mutually exclusive (one cannot receive the treatment AND not receive the treatment). All observations must be independent.

### Example 1: Asthma and HABs

Hospital asthma cases and HABs in Florida, where respiratory admissions have been recorded for a year and the days associated with coastal harmful algae bloom.

#### Creating Data (as a table)

Creating a data frame is not useful in this test, instead, we create a table with dimensions of 2 x 2.

HAB.tab = c(20, 15, 98, 223)  
sum(HAB.tab)

## [1] 356

dim(HAB.tab) = c(2,2)  
dimnames(HAB.tab) = list(c("Non-HAB", "HAB"), c("No Hospitalizations", "Hospitalizations"))  
  
HAB.tab

## No Hospitalizations Hospitalizations  
## Non-HAB 20 98  
## HAB 15 223

# HAB.mat = matrix(HAB.tab, nrow=2)

table(HAB.tab)

## HAB.tab  
## 15 20 98 223   
## 1 1 1 1

#### Chi Square Test in R

chisq.test(HAB.tab)

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: HAB.tab  
## X-squared = 8.9217, df = 1, p-value = 0.002818

In this case, we can reject the null hypothesis, i.e. there is no association with hospitalizations and HABs.



Chamise or Greasewood, a member of the Rose family, is a flowering plant native to California and northern Baja California. This shrub is one of the most widespread plants of the chaparral biome, sometimes forming monotypic stands. It is an evergreen shrub growing to four meters tall, with dry-looking stick-like branches. The leaves are small, 4-10 millimeters long and one millimeter broad with a pointed tip, and sprout in clusters from the branches. The leaves are shiny with flammable oils, especially in warmer weather. It is said to be highly flammable but can be kept fire-resistant by occasional watering. The branches terminate in bunches of white tubular flowers five millimeters in diameter, with five petals and long stamens. Chamise is one of the best plant for anchoring a slope and resisting erosion due to its wide spreading and deeply penetrating roots. In maturity it develops a large burl from which it will resprout after fire or severe pruning. In the wild it is the host plant of a common root parasite, Chaparral Broomrape.

# Other Types the Might be Important (Marc might discuss if we have time…)

## Count Data

TBD

## Multiple Regression

TBD

## Two-Way and Mult-Way ANOVAs –

we need data that we don’t have to read.csv…

What is a two-way anova?

### Example #1 –Fertilizer and Crop Yield MLH: Better for Multiway ANOVA

## Looked at Scribbr; entered the crop data to R as a csv##  
## Question: How does fertilizer effect crop yield? Testing three different fertilizers##  
## command to read the data##  
crop.data <- read.csv("crop.data.csv", header)  
## command to summarize and make sure everything is working##  
summary(crop.data)  
str(crop.data)  
## use to perform a one.way ANOVA##  
## One way ANOVA has one independent independent variable##  
## This one way ANOVA models crop yield as a function of the type of fertilizer used##  
one.way <- aov(yield~fertilizer,data=crop.data)  
## shows what the data compiles into after a one.way##  
summary(one.way)

## use to perform a two.way ANOVA##  
## Two way ANOVA utilizes two independent variables##  
## This two way models crop yield as a function of type of fertilizer and planting density##  
## Just have to add a plus icon and the extra independent variable##  
two.way <- aov(yield ~ fertilizer + density, data = crop.data)  
## shows what the data compiles into after a two.way##  
summary(two.way)  
## Interaction between variables##  
## Used to test whether two variables have an interaction effect##  
interaction <- aov(yield ~ fertilizer\*density, data = crop.data)  
summary(interaction)

### Logistic Regression – Multiple Predictors

MLH – The example relies on external data, which adds a step that gets in the way. I suggest we developed example two to keep it simpler…

Research question: Which fertilizer produces the most crop yield? ##Independent variable/predictor: Fertilizer A, Fertilizer B, Fertilizer C ##Dependent variable/response: amount of crop yield

Question: Which fertilizer provides the best crop yield?

It is common for factors to be read as quantitative variables when importing a dataset into R. To avoid this, you can use the read.csv() command to read in the data, specifying within the command whether each of the variables should be quantitative (â€œnumericâ€) or categorical (â€œfactorâ€).

##Use the following code, replacing the path/to/your/file text with the actual path to your file:

# install.packages(c(“ggplot2”, “ggpubr”, “tidyverse”, “broom”, “AICcmodavg”))   
  
# Read Data  
# crop.data.csv = file.choose()  
  
crop.data.csv = "/home/mwl04747/beginnersluck/Stats/EA30\_Study\_Guide/ANOVA\_cropdata.csv"  
crop.data <- read.csv(crop.data.csv)  
  
names(crop.data)  
crop.data.aov <- aov(Yield ~ Fertilizer, data=crop.data)  
  
summary(crop.data.aov)

## Time Series Data

### Autocorrelation

TBD

## Count Data

TBD

## Mixed Effects Models

TBD

# Appendix 1: Alphabetic List of R code and explanation

## abline()

* This function adds one or more straight lines through the current plot.
* Syntax: abline(a = NULL, b = NULL, h = NULL, v = NULL, reg = NULL, coef = NULL, untf = FALSE, …)

## aov()

* Fit an analysis of variance (ANOVA) model by a call to lm for each stratum.
* Syntax: aov(formula, data = NULL)

## c()

* Combine Values into a Vector or List.

## coef()

* generic function which extracts model coefficients from objects returned by modeling functions. coefficients is an alias for it.
* Syntax: coef(object, …)

## chisq.test()

* performs chi-squared contingency table tests and goodness-of-fit tests.
* Syntax: chisq.test(x, y = NULL) (y is ignored if x is a matrix)

## data.frame()

* creates data frames, tightly coupled collections of variables which share many of the properties of matrices and of lists, used as the fundamental data structure by most of R’s modeling software.

## dim()

* Retrieve or set the dimension of an object.
* Syntax: dim(x) <- value

## dimnames()

* Retrieve or set the dimnames of an object.

## file.choose()

* Choose a file interactively.
* Syntax: file.choose(new = FALSE)

## glm()

* used used to fit generalized linear models, specified by giving a description of the linear predictor and a description of the error distribution.
* Syntax: glm(formula, family = gaussian, data, weights, subset, na.action, start = NULL, etastart, mustart, offset, control = list(…), model = TRUE, method = “glm.fit”, x = FALSE, y = TRUE, singular.ok = TRUE, contrasts = NULL, …)

## help()

* the primary interface to the help systems. Can be abreviated as ? or ?? for packages.
* help(topic, package = NULL, lib.loc = NULL, verbose = getOption(“verbose”), try.all.packages = getOption(“help.try.all.packages”), help\_type = getOption(“help\_type”)

## help.start()

* Start the hypertext (currently HTML) version of R’s online documentation.

## jitter()

* adds a small amount of noise to a numerical vector. It can help the data look a bit clearer on the graph.
* Syntax: jitter(x, factor = 1, amount = NULL)

## knitr::include\_graphics()

-When plots are not generated from R code, there is no way for knitr to capture plots automatically. In this case, you may generate the images manually and pass their file paths to this function to include them in the output. The major advantage of using this function is that it is portable in the sense that it works for all document formats that knitr supports, so you do not need to think if you have to use, for example, LaTeX or Markdown syntax, to embed an external image. Chunk options related to graphics output that work for normal R plots also work for these images, such as out.width and out.height.

## list()

* Functions to construct, coerce and check for both kinds of R lists.
* Syntax: list(…)

## lines()

* A generic function taking coordinates given in various ways and joining the corresponding points with line segments.
* Syntax: lines(x, y = NULL, type = “l”, …)

## - lm()

* used to fit linear models, including multivariate ones. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance (although aov may provide a more convenient interface for these) -Syntax: lm(formula, data, subset, weights, na.action, method = “qr”, model = TRUE, x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE, contrasts = NULL, offset, …)

## matrix()

* creates a matrix from the given set of values.
* Syntax: matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)

## par()

* used to set or query graphical parameters. Parameters can be set by specifying them as arguments to par in tag = value form, or by passing them as a list of tagged values.
* Syntax: par(…, no.readonly = FALSE)

## points()

* draw points at specific coordinates and to be plotted in a plotting structure, a two-column matrix, or a time series.
* Syntax: points(x, y = NULL, type = “p”, …)

## plot() – Draw a scatter plot with decorations such as axes and titles in the active graphics window.

## power.anova.test() – Compute power of test or determine parameters to obtain target power.

* Syntax: power.anova.test(groups = NULL, n = NULL, between.var = NULL, within.var = NULL, sig.level = 0.05, power = NULL)

## predict()

* generic function for predictions from the results of various model fitting functions. The function invokes particular methods which depend on the class of the first argument.

## read.csv()

* Reads a comma separated file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.
* Syntax: dataframe.name <- read.csv(“input.csv”)
* To use this, we first need to import the csv file into r. Then use file.choose() to figure out the path. Once the path is figured out, you can use the function to bring the data into R and do fun stuff!

## read()

* to see a preview of the dataset

## str()

* Compactly display the internal structure of an R object

## summary()

-a generic function used to produce result summaries of the results of various model fitting functions.

## table()

* uses cross-classifying factors to build a contingency table of the counts at each combination of factor levels.

# Appendix 2: Questions to Evaluate our Study Guide

## General questions

1. Is all the rcode described and have comments to help the reader?
2. Is all the data within the documented, no need for external data source links that might break?
3. Is each example described and compelling?
4. Are the results described?
5. Are the models interpreted?

## Quiz Like Questions

Add questions here from our quiz or something that you learned about your topic that you can put in here…

### Linear Regression

1. What is a Linear regression?
2. What is the black diagonal line in Figure 1 called?

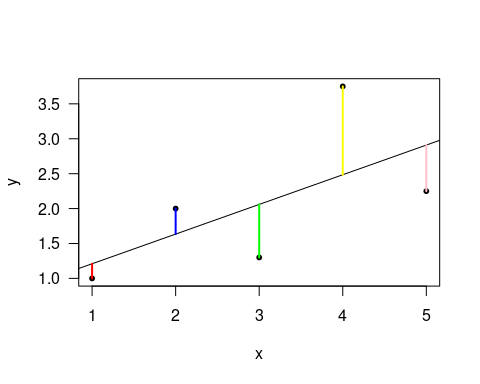
(x <- c(1, 2, 3, 4, 5)) # create a vector of x values

## [1] 1 2 3 4 5

(y <- c(1, 2, 1.3, 3.75, 2.25)) #2. create a vector of y-values

## [1] 1.00 2.00 1.30 3.75 2.25

regression.ex.df = data.frame(X=x, Y=y) #make a data frame… (don’t know the function)  
par(las=1)  
plot(y ~ x, data=regression.ex.df, pch=20)  
regression.ex.lm = lm(y~x, data=regression.ex.df)  
abline(coef(regression.ex.lm))  
regression.ex.df$Residuals = as.vector(resid(regression.ex.lm))  
regression.ex.df$Predicted = as.vector(predict(regression.ex.lm, newdata=data.frame(predicted = x)))  
regression.ex.df$Color = c("red", "blue", "green", "yellow", "pink")  
#lines.mat = mat()  
#(xy.coords.ex = xy.coords(rep(x, times=2), c(regression.ex.df$Predicted, c(regression.ex.df$Predicted + regression.ex.df$Residuals))))  
#xy.coords.ex[order(xy.coords.ex$x)]  
for(i in 1:nrow(regression.ex.df)){  
lines(c(regression.ex.df$X[i], regression.ex.df$X[i]), c(regression.ex.df$Predicted[i], c(regression.ex.df$Predicted[i] + regression.ex.df$Residuals[i])), col=regression.ex.df$Color[i], lwd=2)  
}



#lines(x[1], regression.ex.df$Predicted[1], x, c(regression.ex.df$Predicted[1] + regression.ex.df$Residuals[1]), color=

1. Outline the pseudo R code that might be used to generate this graph. In other words, use your words to describe the steps you might need to use to create this graphic (think about how the colored lines might be made and the labels. If you have guesses of the R code syntax, put this in parenthetical statements.

For example, 1. create a vector of x values (x <- c(1, 2, 3, 4, 5) 2. create a vector of y-values (y <- c(1, 2, 1.3, 1.75, 2.25) 3. make a data frame… (don’t know the function)… 4. …

1. The vertical lines on Figure 1 from the points to the regression line represent the errors of prediction. As you can see, the red point is very near the regression line; its error of prediction is small. By contrast, the yellow point is much higher than the regression line and therefore its error of prediction is large. How can the summary() function be used to access the residuals?
2. The summary() function also reports the r2 and p-values for the model. Describe how to interpret these.

### ANOVA

1. What is the difference between a one-way and two-way ANOVA?
2. Given an experiment with a control and three treatments (Treatment) and some measured response (Response) here’s the R code to create the ANOVA object:

Model1.aov <- aov(Response ~ Treatment)

1. How would you generate a summary of the results? How would you report this in a paper?

### Logistic Regression

1. What kind of data is used in a logistic regression?
2. What distribution family is used in the glm() function for logistic regression? Why is this distribution used?

### Tests for Association