R Minicourse Workshop, Part 3

Presented to the
Washington State Deptment of Ecology
September 2–3, 2014

Dr. Robin Matthews, Institute for Watershed Studies
Dr. Geoffrey Matthews, Computer Science Department
Western Washington University

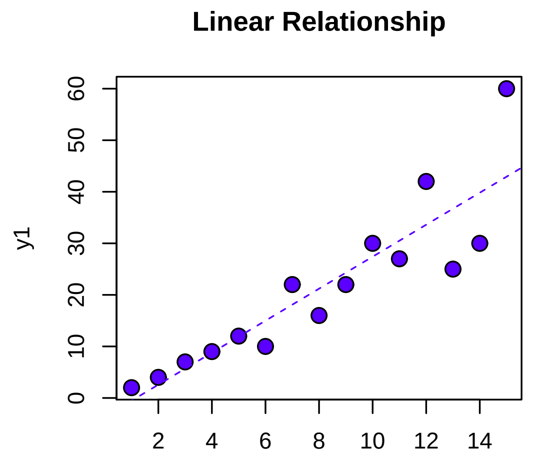
Bivariate Analysis Using R

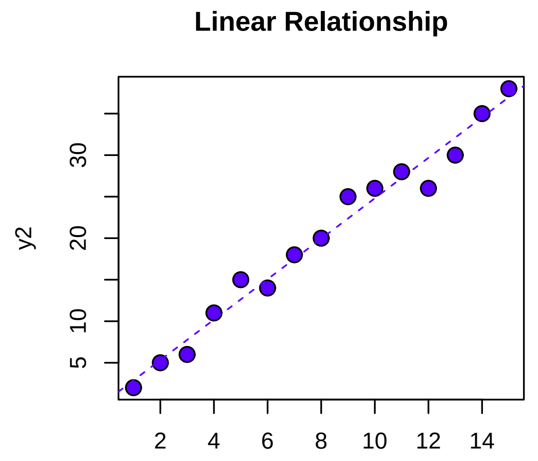
Correlation and Regression

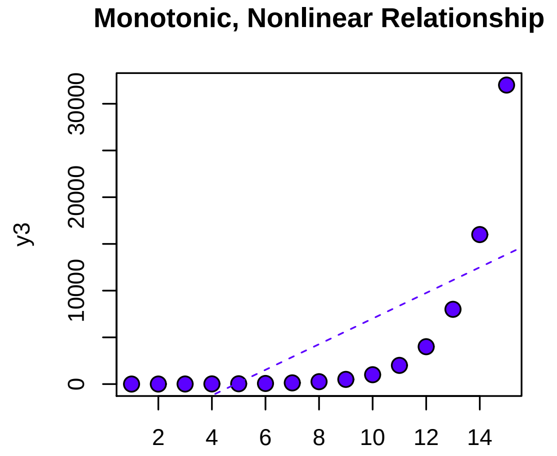
- Regression measures the relationship between an independent variable (x) and one or more dependent variables (y_i)
  - Used to predict (model) unmeasured values of the dependent variable(s)
- Correlation analysis measures the relationship between two variables that are not necessarily functionally dependent
  - Used to explore patterns in measured variables and to identify *indicators* that predict responses in other variables

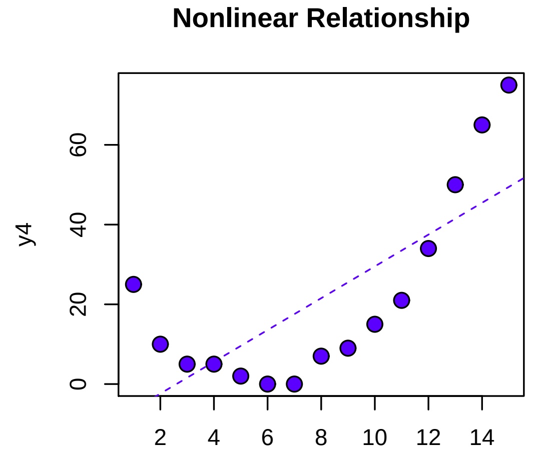
Correlation vs. Regression

Correlation and regression examine *monotonic* relationships between variables









Assumptions for Correlation and Regression

- *Parametric* correlation (r) and simple linear regression (r^2) is based on several important assumptions:
    - The samples were collected randomly
    - The variables or linear residuals are normally distributed
    - The variance is constant (homoscedastic)
    - The relationship is monotonic ($\pm$linear)
    - The data do not contain outliers (or not very many)
- *Nonparametric* correlation (ρ, τ) does not require a linear relationship or homoscedasticity, but it still assumes that the relationship between x and y is monotonic

Correlation Analysis

- The strength of a correlation is measured using a correlation coefficient (Pearson's r, Spearman's ρ, Kendall's τ)

  $$r = \frac{1}{n-1} \sum \left(\frac{x_i - \overline{x}}{s_x} \times \frac{y_i - \overline{y}}{s_y} \right)$$

  ρ is linear correlation computed on ranks

  $$\tau = \frac{\mathrm{S}}{n(n-1)/2}$$

  S = P - M
  P = number of times y increases with x;
  M = number of times y decreases as x increases

- H$_o$: r or ρ or $\tau = 0$ H$_a$: r or ρ or $\tau \neq 0$

- $-1 \leq r$ or ρ or $\tau \leq 1$

Correlation Analysis Using R

Parametric and nonparametric correlation is done using `cor.test`

```
lakes <- read.csv("lakes.csv", T)
attach(lakes)
cor.test(alk, ph)
```

```
 Pearson's product-moment correlation
data:  alk and ph
t = 15.5074, df = 376, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5589116 0.6824434
sample estimates:
      cor
0.6245687
```

```
##### Nonparametric versions (all sig at p-value < 2.2e-16)
cor.test(alk, ph, method="spearman")  #rho=0.6785801
cor.test(alk, ph, method="kendall")   #tau=0.5030349
```

Simple Linear Regression

- Simple linear regression is based on the model:

$$y_i = a + bx_i + \varepsilon_i$$

$$\begin{aligned} a &= \text{intercept} \\ b &= \text{slope} \\ \varepsilon &= \text{residual for the i}^{th}\text{ observation} \end{aligned}$$

- The strength of the regression is measured with the regression statistic or coefficient of determination (r^2)
- H$_o$: $r^2 = 0$ H$_a$: $r^2 \neq 0$
- $-1 \leq r^2 \leq 1$

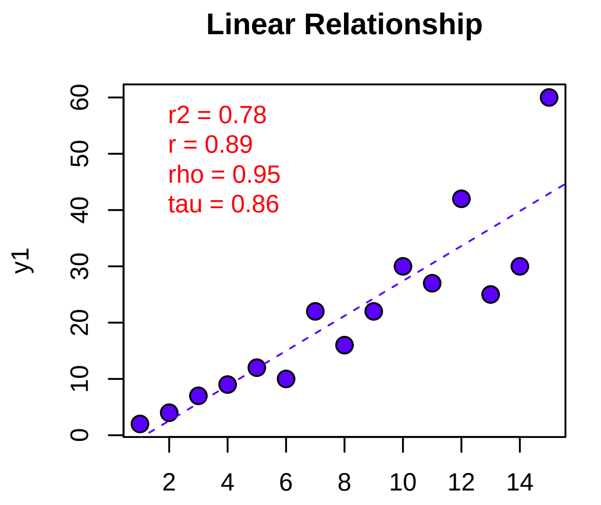
Simple Linear Regressions Using R

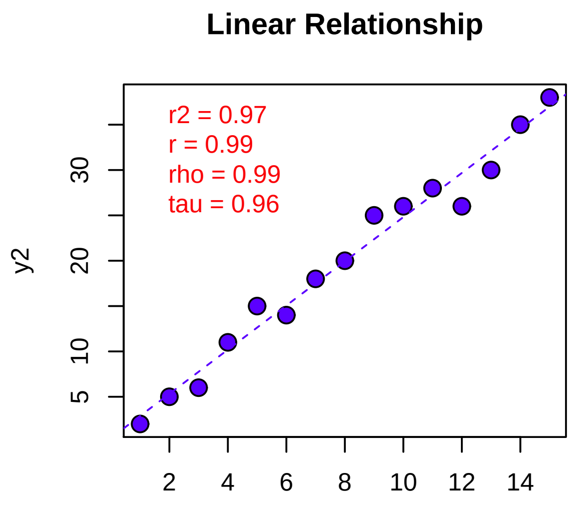
The easiest option for simple linear regressions is `lm`

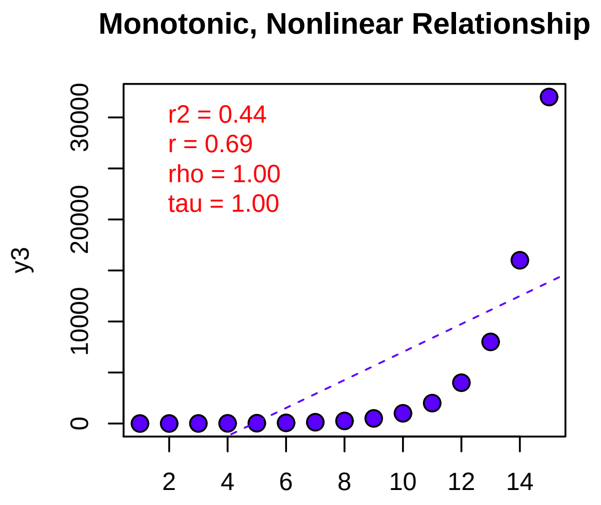
```
X <- c(1:10)
Y <- c(49.8,51.6,53.7,53.9,55.1,56.5,57.4,57.9,58.9,60.8)
##### Y is x+50, with "noise"

lm(Y~X)
Call:
lm(formula = Y ~ X)
Coefficients:
(Intercept)            X
     49.460        1.109

##### X = 1.109 * Y + 49.460

##### Predicting Y for unmeasured values of X:
predict(Yfit, list(X=6.7))
       1
56.89091

##### Predicting Y for all measured values of X:
round(predict(Yfit),2)
    1     2     3     4     5     6     7     8     9    10
50.57 51.68 52.79 53.90 55.01 56.11 57.22 58.33 59.44 60.55
```

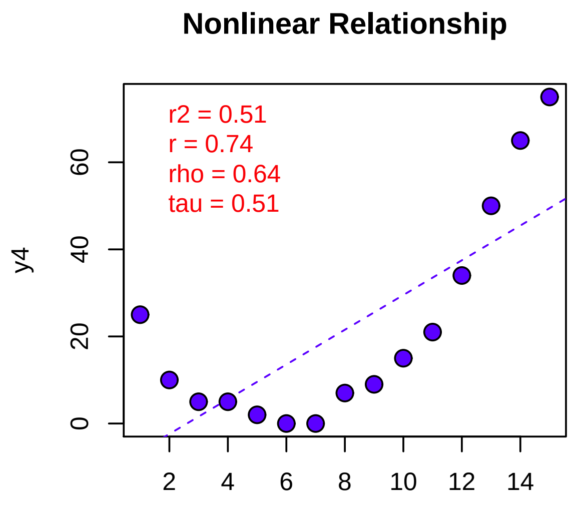
Correlation/Regression

Comparison Using Four Types of Curves

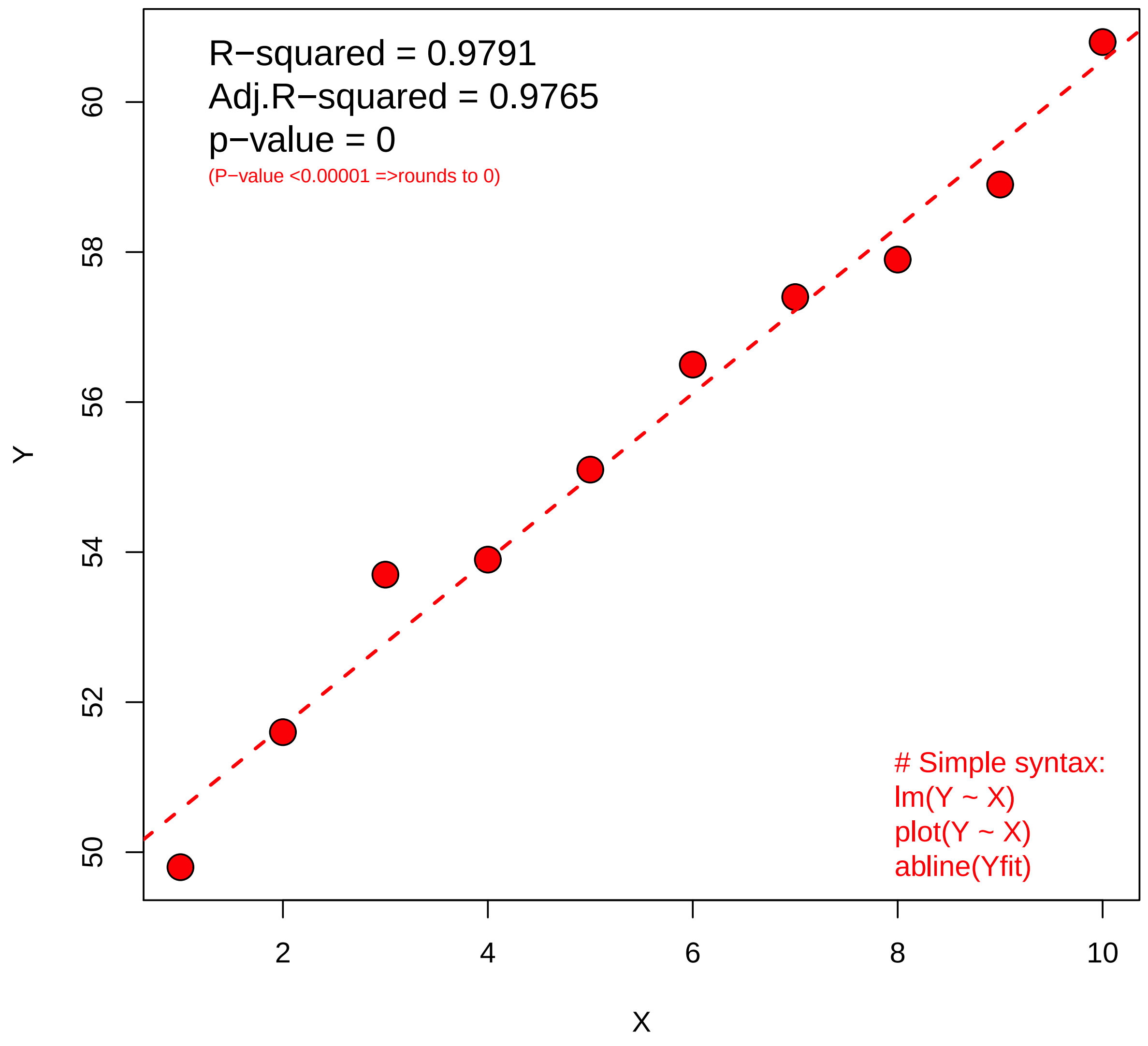






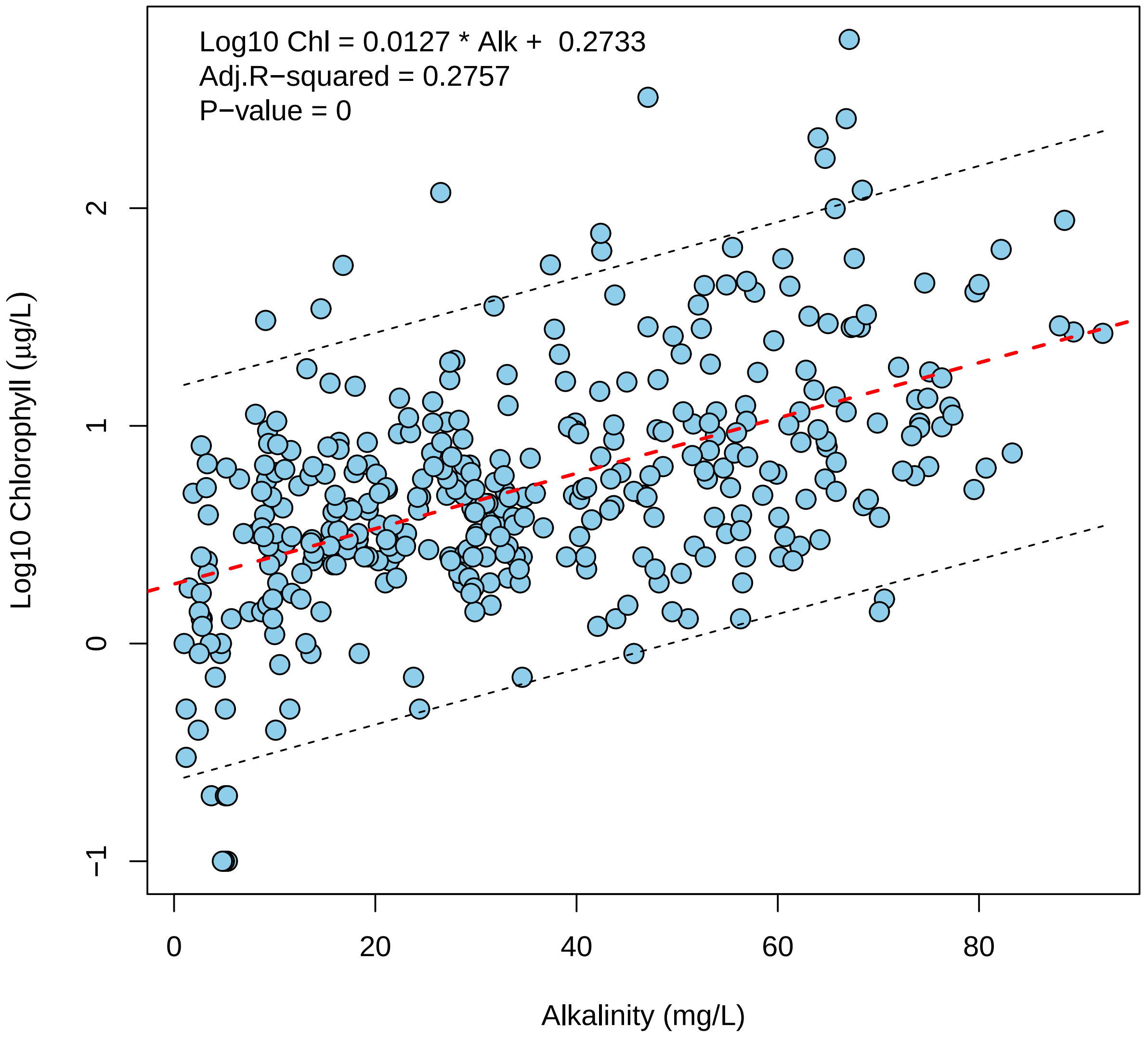


Plotting Simple Linear Regressions Using R



Advanced Plotting Features - IWS Lakes Data

Adding Confidence Intervals



Syntax for Adding Confidence Intervals

```
lakes = read.table("lakes.csv", T, sep=",")
attach(lakes)

### Step 1:  create linear model (chl ~ alk)
alkchl.lm = lm(log10(chl) ~ alk)

### Step 2:  sort the x axis (unique values only)
alk.sort = sort(unique(alk))

### Step 3:  use predict to predict chl ~ tp from linear model
pred.chl = predict(alkchl.lm,
    newdata = data.frame(alk = alk.sort), int="pred")

### Step 4:  plot original data and linear model
plot(log10(chl) ~ alk,
     xlab="Alkalinity (mg/L)",
     ylab=expression(paste("Log10 Chlorophyll " (mu * "g/L"))),
     pch=21, bg="skyblue", cex=1.5)
abline(alkchl.lm, lwd=2, lty=2, col="red")
```

Syntax for Adding Confidence Intervals, continued

```
### Step 5:  add upper and lower CI
lines(alk.sort, pred.chl[,2], lty=2) #lower CI
lines(alk.sort, pred.chl[,3], lty=2) #upper CI

### Step 6:  add a legend with the linear model statistics
legend(x="topleft",
   c(paste("Log10 Chl =", round(alkchl.lm$coef[2],4),
     "* Alk + ", round(alkchl.lm$coeff[1],4)),
    paste("Adj.R-squared =", round(summary(alkchl.lm)$adj.r.squared, 4)),
    paste("P-value =", round(summary(alkchl.lm)$coef[8], 4))),
    bty="n", cex=1)
```

Or, you can use one of the many R packages that adds confidence intervals automatically!

Revisiting the Regression Assumptions

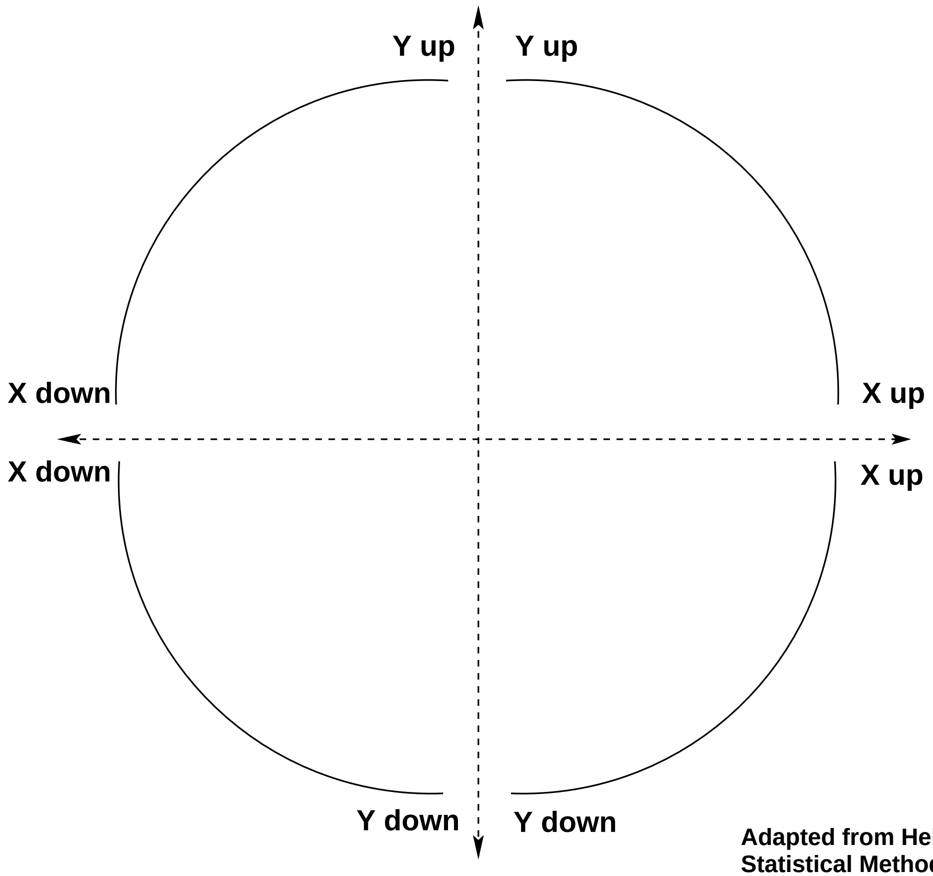
- Linear regression builds a model of the relationship between X and Y, with the assumption that the relationship is monotonic *and* linear
- All of the linear models plotted on page 9 had statistically significant regression statistics (r^2)
- If that relationship between X and Y is *monotonic* but not linear, you can still use simple linear regression if you transpose the variables
- R makes it very easy to insert transformations directly into the regression and plotting syntax

```
##### Untransformed linear model:
lm(Y ~ X)

##### Log (base e) transformed linear model
lm(log(Y) ~ X)

##### Log10 transformed linear model
lm(log10(Y) ~ X)
```

Regression Transformations

Ladder of Power

Match your line shape to one of these curves,
then transform (straighten) by converting x, y, or both



Adapted from Helsel and Hirsch,
Statistical Methods in Water Resources
1992, Elsevier Press

Regression Transformations

Ladder of Power - R Syntax

Description	Transformation	R Code
Negative skewness	Cube	Y∧3
(top of ladder)	Square	Y∧2
	Square root	Y∧(1/2) or sqrt(Y)
	Cube root	Y∧(1/3)
center - start here ⇒	Log	log10(Y)
	Reciprocal root	-1/sqrt(Y)
Positive skewness	Reciprocal	-1/Y
	Reciprocal root	-1/sqrt(Y)
Positive skewness	Reciprocal	-1/Y
(bottom of ladder)	Reciprocal of square	-1/(Y∧2)

Adapted from Helsel and Hirsch, Statistical Methods in Water Resources

Logs other than base 10 can be used
Constants can be added to x to avoid dividing by zero (`log1p(x)` computes log(1+x))
Higher and lower powers can be used

Finding the Best Transformation

Box-Cox Procedure

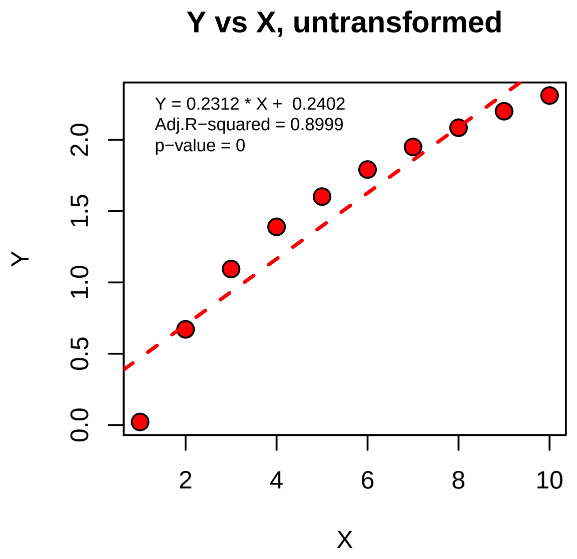
- The Box-Cox procedure in the `MASS` library will estimate the "best" power (lambda) for transforming Y
- The procedure defaults to ± 2 (Y^{-2} to Y^2)

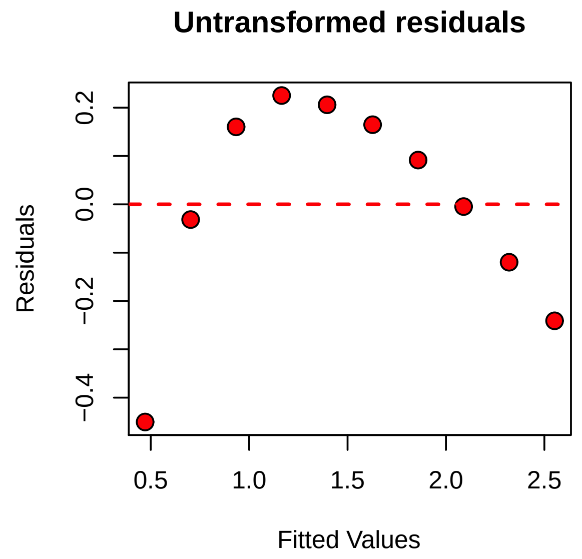
```
X <- c(1:10)
Y <- c(0.021,0.671,1.094,1.390,1.602,1.792,1.950,2.085,2.201,2.311)
### Y is ln(X) plus noise

Yfit <- lm(Y~X)

library(MASS)
Yfit.bc <- boxcox(Yfit)
Yfit.bc$x[which.max(Yfit.bc$y)]
```

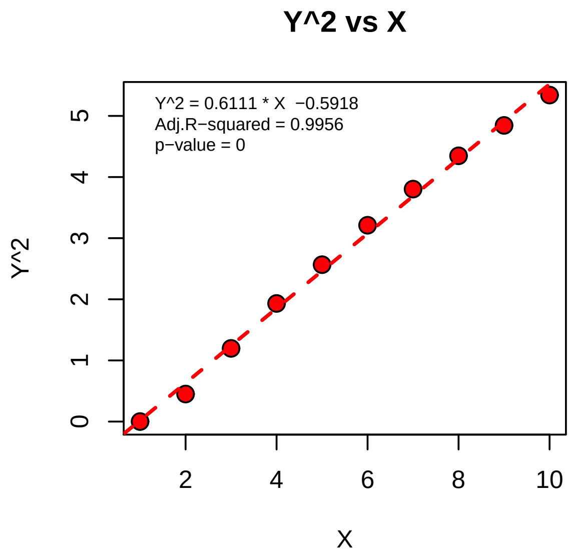
```
[1] 2
```

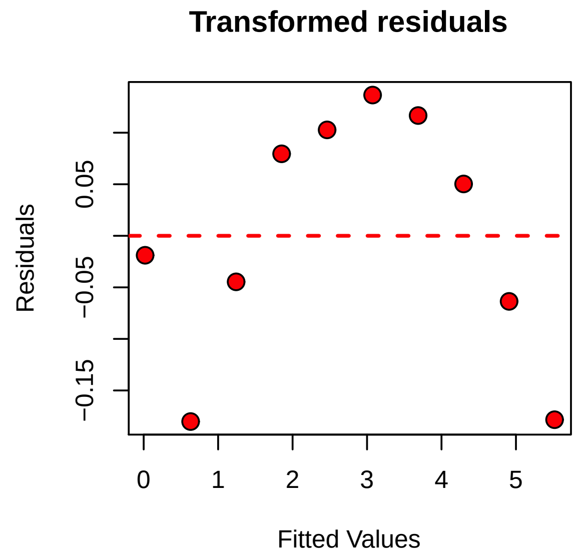
```
##### Best est. for transformation will be Y^2
YfitBC.trans <- lm(Y^2 ~ X)
```

Example of Transforming Y to Y^2









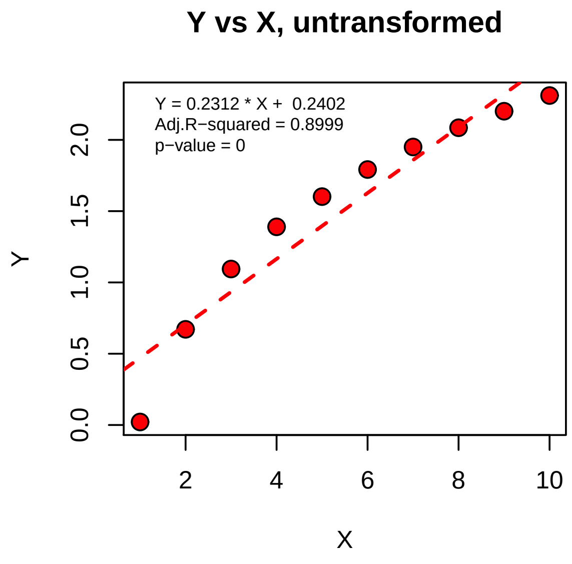
Advanced Plotting Example

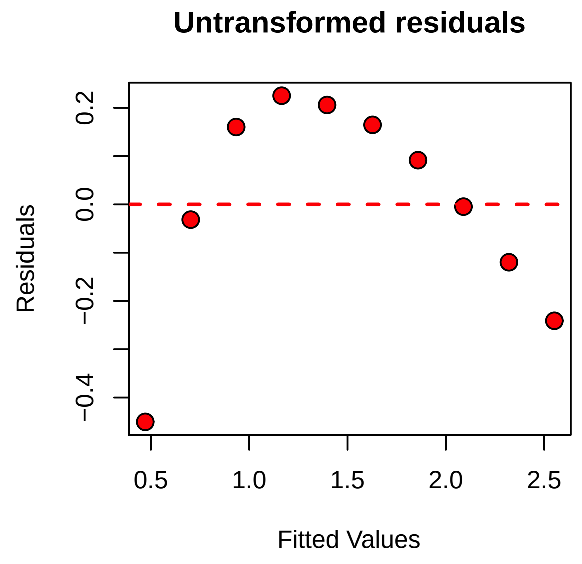
Plotting Linear Models and Residuals With/Without Transformation

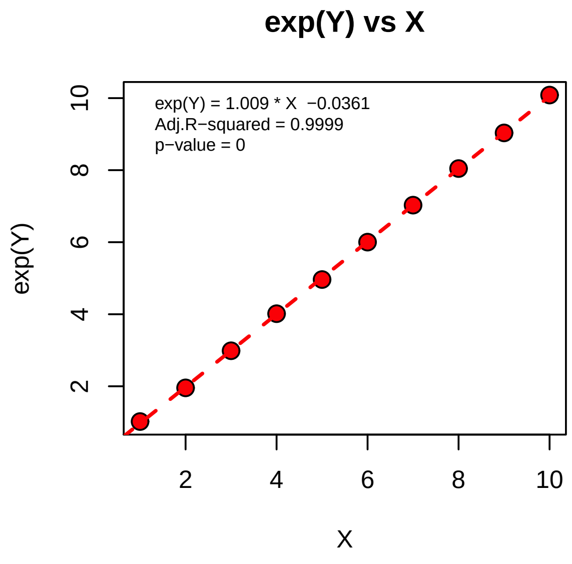
Plotting example uses data and Box-Cox transformation show on page 17

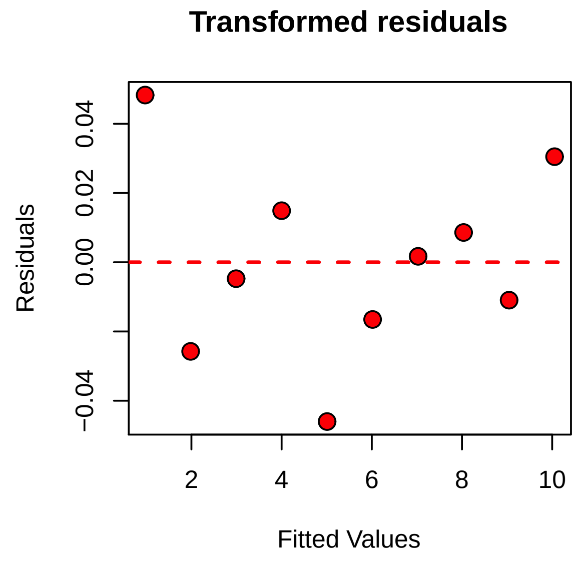
```
### Set plotting output to 4x per page
par(mfrow=c(2,2))

### Plot untransformed Y and residuals from Yfit
plot(Y ~ X, main="Y vs X, untransformed",  pch=21, bg="red", cex=1.5)
abline(Yfit, lwd=2, lty=2, col="red")
legend(x="topleft", c(paste("Y =", round(Yfit$coef[2],4),
         "* X + ", round(Yfit$coef[1], 4)),
         paste("Adj.R-squared =", round(summary(Yfit)$adj.r.squared, 4)),
         paste("p-value =", round(summary(Yfit)$coef[8], 4))),
         bty="n", cex=0.7)
plot(Yfit$fitted.values, resid(Yfit), main="Untransformed residuals",
     pch=21, bg="red", cex=1.5,  xlab="Fitted Values", ylab="Residuals")
abline(h=0, lwd=2, lty=2, col="red")

### Plot transformed Y using Box-Cox estimate and YfitBC residuals
plot(Y^2 ~ X, main="Y^2 vs X", pch=21, bg="red", cex=1.5)
abline(YfitBC.trans, lwd=2, lty=2, col="red")
legend(x="topleft", c(paste("Y^2 =", round(YfitBC.trans$coef[2],4),
         "* X ", round(YfitBC.trans$coef[1], 4)),
         paste("Adj.R-squared =", round(summary(YfitBC.trans)$adj.r.squared, 4)),
         paste("p-value =", round(summary(YfitBC.trans)$coef[8], 4))),
         bty="n", cex=0.7)
plot(YfitBC.trans$fitted.values, resid(YfitBC.trans), main="Transformed residuals",
     pch=21, bg="red", cex=1.5, xlab="Fitted Values", ylab="Residuals")
abline(h=0, lwd=2, lty=2, col="red")
```

Example of Transforming to Y to exp(Y)

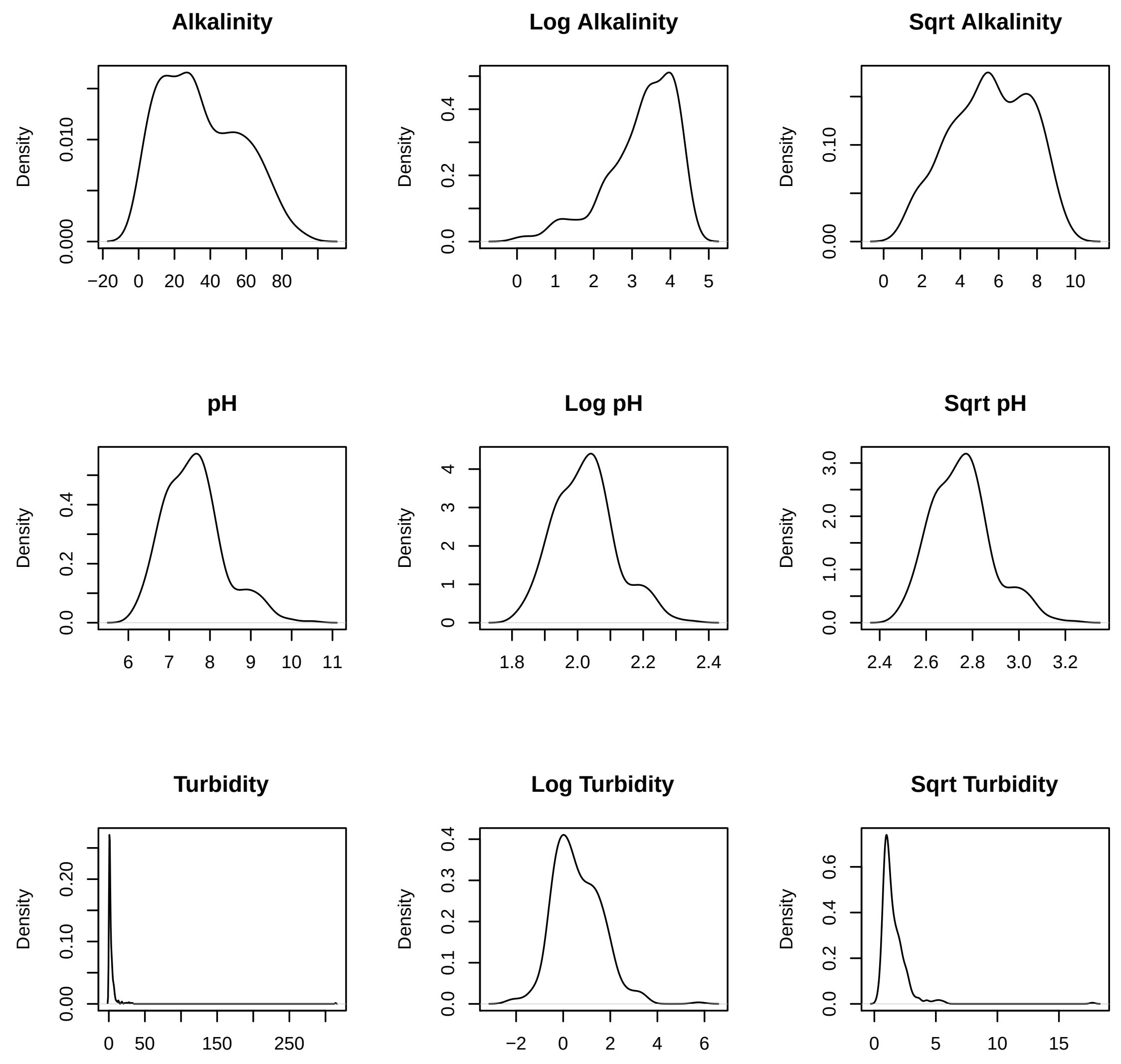






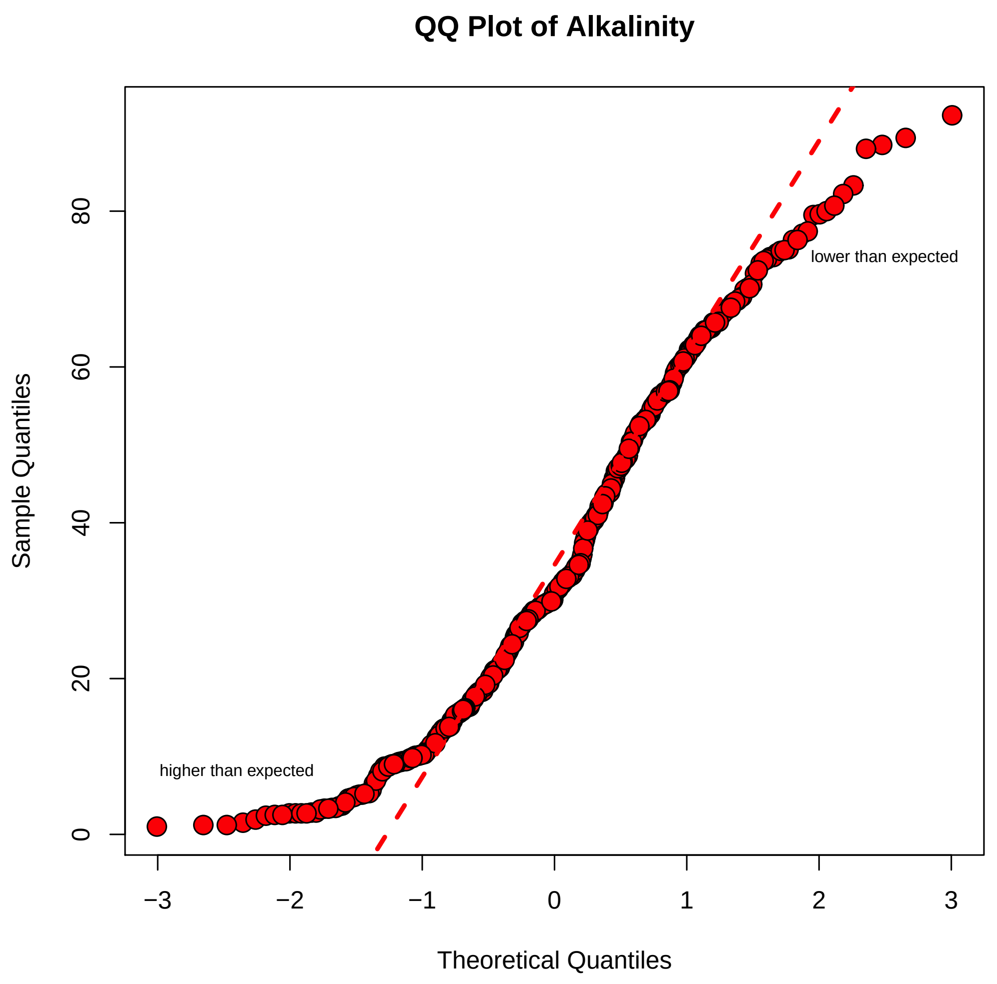


Examining Transformations Using Density Plots

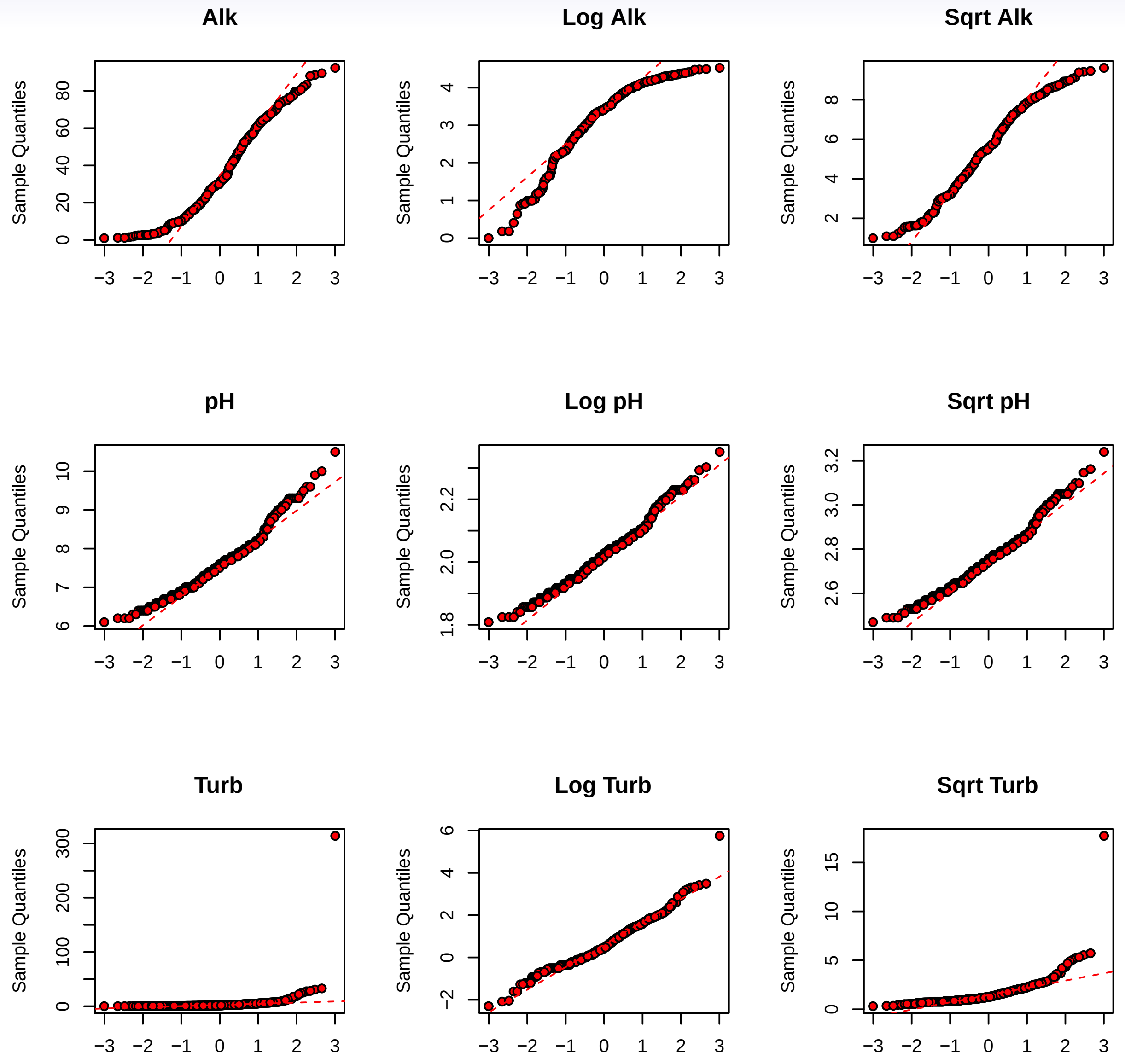


```
##### basic plotting code:
plot(density(alk)); plot(density(log(alk))); plot(density(sqrt(alk)))
```

Examining Transformations Using QQ Plots



QQ plots are a simply way to explore distributions; if the data are normally distributed, the points will be close to the diagonal reference line. Syntax: `qqnorm(alk); qqline(alk)`


Alk
Log Alk
Sqrt Alk
pH
Log pH
Sqrt pH
Turb
Log Turb
Sqrt Turb
Sample Quantiles

Multiple Linear Regression

- The equation for multiple regression is a logical expansion of the basic linear model:

$$y_i = a + b_1 u_i + b_2 v_i + b_3 w_i + \varepsilon_i$$

$$\begin{aligned} a &= \text{intercept} \\ b &= \text{slopes associated with variables } u,\ v, \text{ and } w \\ \varepsilon &= \text{residual for the i}^{th} \text{ observation} \end{aligned}$$

R syntax: `lm(y ~ u + v + w)`

- For linear regressions with interaction terms, the model becomes:

$$y_i = a + b_1 u_i + b_2 v_i + b_3 u_i v_i + \varepsilon_i$$

S syntax: `lm(y ~ u * v)`

- Interpreting the output and selecting the best subset of variables for the final multiple regression model in not simple!

Analysis of Variance

- Analysis of variance procedures (t test, ANOVA, MANOVA) use ratios of within-group variance to total variance to test for significant differences between groups

  H$_o$: $\overline{x_1} = \overline{x_2} = \ldots \overline{x_n}$ $\qquad (n =$ number of groups$)$

  H$_a$: $\overline{x_1} \neq \overline{x_2} \neq \ldots \overline{x_n}$

- The decision to accept or reject the null hypothesis carries a probability (p) of committing a Type I error, which is rejection of the null hypothesis when it is actually true.
- ANOVA tests whether *any* of the groups are significantly different
- ANOVA is often used in conjunction with a multiple range test to determine which groups are different from the others.

ANOVA

Type I and II Errors

- Type I errors become more likely when you repeat ANOVA on subgroups of the data (use p-value adjustment)
- Type II errors usually indicates a small samples size, but can also be caused by extremely large samples sizes (use `sample`)

Decision	H_o is true	H_o is false
Accept H_o (Fail to Reject H_o)	ok Prob $= 1 - \alpha$	Type II error Prob $= \beta$
Reject H_o	Type I error Prob $= \alpha$	ok Prob $= 1 - \beta$
	Significance level	**Power**

ANOVA

Assumptions for Analysis of Variance (ANOVA)

- ANOVA is based on several important assumptions:
  - The samples were collected randomly
  - The measured variables are distributed normally within each group
  - The variances are homogeneous (homoscedastic) across groups $\Rightarrow$ very important assumption!
- Most uses of ANOVA rely heavily on its ability to perform despite departures from normality and homogeneity
- Nonparametric versions of ANOVA are easy to use, powerful, and avoid the issue of normality and homogeneity

ANOVA

Testing Assumption of Normality

- You want to *accept* the hypotheses H_o that the data from each group fits a normal distribution (p-value $>$0.050)
- The Shapiro-Wilks test of normality is simple in R:

```
data(iris); attach(iris)
shapiro.test(Sepal.Length[Species=="setosa"])
        Shapiro-Wilk normality test
data:  Sepal.Length[Species == "setosa"]
W = 0.9777, p-value = 0.4595

##### edited output for remaining iris species
data:  Sepal.Length[Species == "versicolor"]
W = 0.9778, p-value = 0.4647

data:  Sepal.Length[Species == "virginica"]
W = 0.9712, p-value = 0.2583
```

- Alternatively, you could use `qqnorm` to examine the data graphically

ANOVA

Testing Assumption of Variance Homogeneity

- You want to *accept* the H_o that variances are homogeneous
- Use Bartlett's test if the data are normally distributed or Fligner's test if you don't want to make that assumption

```
bartlett.test(Sepal.Length, Species)
        Bartlett test for homogeneity of variances
data:  Sepal.Length and Species
Bartlett's K-squared = 16.0057, df = 2, p-value = 0.0003345
```

```
fligner.test(Sepal.Length, Species)
        Fligner-Killeen test for homogeneity of variances
data:  Sepal.Length and Species
Fligner-Killeen:med chi-squared = 11.618, df = 2, p-value = 0.003000
```

- The sepal length data fail the assumption of homoscedasticity!

  Optional exercise: Modify the syntax to show that a log transformation of the sepal length data corrects the heteroscedasticity problem

- You could also use `boxplots` to examine the data graphically

ANOVA Using R

- The simplest R syntax for ANOVA uses aov

```
##### Untransformed sepal lengths
data(iris); attach(iris)
aov.SL <- aov(Sepal.Length ~ Species)
summary(aov.SL)

             Df Sum Sq Mean Sq F value Pr(>F)
Species       2  63.21  31.606   119.3 <2e-16 ***
Residuals   147  38.96   0.265
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

##### Log-transformed sepal lengths
aov.logSL <- aov(log(Sepal.Length) ~ Species)
summary(aov.logSL)

             Df Sum Sq Mean Sq F value Pr(>F)
Species       2  1.892  0.9459   128.9 <2e-16 ***
Residuals   147  1.079  0.0073
```

- Both tests indicate that there are significant differences between the sepal lengths for the different iris species

ANOVA Using R

Adding Post-Hoc Tests

- The ANOVA doesn't tell us which species are different, so we use a *post-hoc* test
- *Which* post-hoc test to use is beyond the scope of this class . . . there are many choices
- Most R textbooks use `pairwise.t.test`, which does 2-group ANOVAs, correcting the p-value for repeated testing (Type I errors)
- The post-hoc test should use log-transformed data (see page 29)

```
pairwise.t.test(log10(Sepal.Length), Species)
```

```
 Pairwise comparisons using t tests with pooled SD
data:  log10(Sepal.Length) and Species
           setosa  versicolor
versicolor < 2e-16 -
virginica  < 2e-16 1.3e-08
P value adjustment method: holm
```

Nonparametric Alternatives to ANOVA

- The Kruskal-Wallis rank sum test (`kruskal.test`), paired with the Wilcoxon rank sum test (`pairwise.wilcox.test`) will provide a nonparametric alternative to ANOVA
- For *untransformed* heteroscedastic data, the nonparametric results are usually far more useful than anything based on variance

```
kruskal.test(Sepal.Length ~ Species)

 Kruskal-Wallis rank sum test
data:  Sepal.Length by Species
Kruskal-Wallis chi-squared = 96.9374, df = 2, p-value < 2.2e-16

pairwise.wilcox.test(Sepal.Length, Species)
 Pairwise comparisons using Wilcoxon rank sum test
data:  Sepal.Length and Species
           setosa  versicolor
versicolor 1.7e-13 -
virginica  < 2e-16 5.9e-07
P value adjustment method: holm
```

Supplemental References

- Crawley, Michael J. 2013. The R Book. John Wiley & Sons. ISBN 978-0-470-97392-9.
- Faraway, Julian J. 2014. Linear Models with R, 2nd Edition. CRC Press. ISBN 978-1-439-88733-2.
- Lander, Jared P. 2014. R for Everyone, Advanced Analytics and Graphics. Addison Wesley Data & Analytics Series, ISBN 978-0-321-88803-7.
- Teetor, Paul. 2011. The R Cookbook. O'Reilly Publishers. ISBN 978-0-596-880915-7