MATH 3050 – Predictive Analytics

UNC CHARLOTTE

Topic 5 – Linear Modeling

Seven important kinds of regression analysis:

1. Linear regression (the simplest, and much the most frequently used)
2. Polynomial regression (often used to test for non-linearity in a relationship)
3. Piecewise regression (two or more adjacent straight lines)
4. Robust regression (models that are less sensitive to outliers)
5. Multiple regression (where there are numerous explanatory variables)
6. Non-linear regression (to fit a specified non-linear model to data)
7. Non-parametric regression (used when there is no obvious functional form)

UNC CHARLOTTE 1

1

Topic 5: Linear Modeling

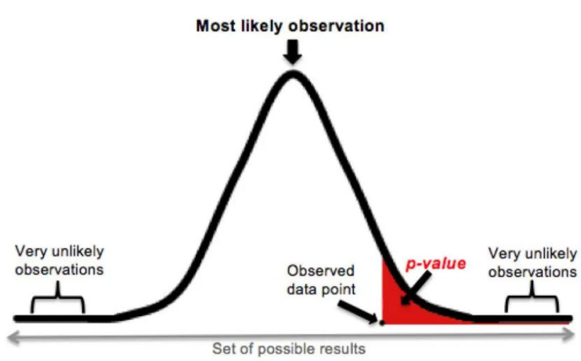
Interpreting P-Values

What is the *P* value?

For a given statistical model when the null hypothesis is true, the *P* – value is the probability the model test statistic is equal to or more extreme than the actual observed results.

For regression analysis, we test

1.) H_0: $\beta_i = 0$
2.) H_0: σ_i are equal



A *p-value* (shaded red area) is the probability of an observed (or more extreme) result arising by chance

UNC CHARLOTTE 2

2

ASA News
AMERICAN STATISTICAL ASSOCIATION
Promoting the Practice and Profession of Statistics

732 North Washington Street, Alexandria, VA 22314 • (703) 684-1221 • Toll Free: (888) 231-3473 • www.amstat.org • www.twitter.com/AmstatNews

AMERICAN STATISTICAL ASSOCIATION RELEASES STATEMENT ON STATISTICAL SIGNIFICANCE AND *P*-VALUES

Provides Principles to Improve the Conduct and Interpretation of Quantitative Science

March 7, 2016

"The increased quantification of scientific research and a proliferation of large, complex data sets has expanded the scope for statistics and the importance of appropriately chosen techniques, properly conducted analyses, and correct interpretation."

Interpreting P-Values

The statement's six principles, which **address many misconceptions and misuse of the p-value**, are the following:

1. P-values can indicate how incompatible the data are with a specified statistical model.
2. P-values do not measure the probability that the studied hypothesis is true, or the probability that the data were produced by random chance alone.
3. Scientific conclusions and business or policy decisions should not be based only on whether a p-value passes a specific threshold.
4. Proper inference requires full reporting and transparency.
5. A p-value, or statistical significance, does not measure the size of an effect or the importance of a result.
6. By itself, a p-value does not provide a good measure of evidence regarding a model or hypothesis.

Interpreting P-Values

The statement's six principles, which **address many misconceptions and misuse of the p-value**, are the following:

1. P-values can indicate how incompatible the data are with a specified statistical model.
2. P-values do not measure the probability that the studied hypothesis is true, or the probability that the data were produced by random chance alone.
3. Scientific conclusions and business or policy decisions should not be based only on whether a p-value passes a specific threshold.
4. Proper inference requires full reporting and transparency.
5. A p-value, or statistical significance, does not measure the size of an effect or the importance of a result.
6. By itself, a p-value does not provide a good measure of evidence regarding a model or hypothesis.

UNC CHARLOTTE 5

5

Interpreting P-Values

H_0: $\beta_0 = \beta_1 = \beta_2 = \beta_3 = 0$

The Regression Equation:

$$y = \underbrace{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 \cdots \beta_k x_k}_{\text{The value } (\hat{y}) \text{ predicted by the variables in the model.}} + \varepsilon.$$

The value ($\hat{y}$)predicted by the variables in the model.

The actual value of "y" we are trying to predict with the model.

The amount of the actual value of "y" we count NOT predict with the model. The residual error.

Note: The predicted value + the error exactly equal the actual values ($y = \hat{y} + \varepsilon$).

UNC CHARLOTTE 6

6

Interpreting P-Values

$H_0: \beta_0 = \beta_1 = \beta_2 = \beta_3 = 0$

Regression Equation:

$$\widehat{\text{Hwy}} = \beta_0 + \beta_1\text{Drv} + \beta_2\text{Cyl} + \beta_3\text{Class} + \varepsilon$$

Where

Hwy = Highway Fuel Economy
Drv = Drivetrain: Front Wheel, Four Wheel, Rear Wheel
Cyl = Number of Cylinders
Class = Class of Vehicle – 2-Seater, Compact, Midsize, Minivan, Pickup, Subcompact, SUV
ε = Residual Error

Interpreting P-Values

$H_0: \beta_0 = \beta_1 = \beta_2 = \beta_3 = 0$

Regression Equation: $\widehat{\text{Hwy}} = \beta_0 + \beta_1\text{Drv} + \beta_2\text{Cyl} + \beta_3\text{Class}$

R Equation: `lm(Hwy ~ Drv + Cyl + Class)`

Regression Output

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     36.4845     1.5876  22.981  < 2e-16 ***
drvf             3.3594     0.5933   5.663 4.55e-08 ***
drvr             0.9405     0.7095   1.326  0.18634
cyl             -1.5781     0.1467 -10.760  < 2e-16 ***
classcompact    -3.4357     1.3608  -2.525  0.01227 *
classmidsize    -3.9144     1.3784  -2.840  0.00493 **
classminivan    -8.2985     1.5289  -5.428 1.48e-07 ***
classpickup     -8.5111     1.3701  -6.212 2.52e-09 ***
classsubcompact -2.7594     1.3110  -2.105  0.03642 *
classsuv        -7.5264     1.2800  -5.880 1.48e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.489 on 224 degrees of freedom
Multiple R-squared:  0.8321,    Adjusted R-squared:  0.8254
F-statistic: 123.3 on 9 and 224 DF,  p-value: < 2.2e-16
```

Observations:

- Pr(>|t|) represents the p-values.
- The number of "*" represents how small that are.
  - " " means > 0.05
  - "*" means < 0.05
  - "**" means < 0.001
  - "***" means close to 0
- Some numbers are so small that they need to be expressed using scientific notation ("e-XX") to avoid printing so many zeros.
- Only "drvr" (rear wheel drive) is insignificant.

Interpreting P-Values

H_0: $\beta_0 = \beta_1 = \beta_2 = \beta_3 = 0$

Regression Equation: $\text{Hwy} = \beta_0 + \beta_1\text{Drv} + \beta_2\text{Cyl} + \beta_3\text{Class}$

R Equation: `lm(Hwy ~ Drv + Cyl + Class)`

Observations:

- Now let's inspect the β_i's they are called "Estimate" in the printout.
- Class Compact and Class Midsize are practically the same value. We might want to collapse these into one category to simplify the model.
- Class Minivan and Class Pickup are also practically the same value. We might also want to collapse these into one category to simplify the model.
- Let's collapse Compact and Midsize first.

Regression Output

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     36.4845     1.5876  22.981  < 2e-16 ***
drvf             3.3594     0.5933   5.663 4.55e-08 ***
drvr             0.9405     0.7095   1.326  0.18634
cyl             -1.5781     0.1467 -10.760  < 2e-16 ***
Classcompact    -3.4357     1.3608  -2.525  0.01227 *
Classmidsize    -3.9144     1.3784  -2.840  0.00493 **
Classminivan    -8.2985     1.5289  -5.428 1.48e-07 ***
Classpickup     -8.5111     1.3701  -6.212 2.52e-09 ***
Classsubcompact -2.7594     1.3110  -2.105  0.03642 *
Classsuv        -7.5264     1.2800  -5.880 1.48e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.489 on 224 degrees of freedom
Multiple R-squared:  0.8321,    Adjusted R-squared:  0.8254
F-statistic: 123.3 on 9 and 224 DF,  p-value: < 2.2e-16
```


Topic 5: Linear Modeling

Interpreting P-Values

H_0: $\beta_0 = \beta_1 = \beta_2 = \beta_3 = 0$

Regression Equation: $\text{Hwy} = \beta_0 + \beta_1\text{Drv} + \beta_2\text{Cyl} + \beta_3\text{Class}$

R Equation: `lm(Hwy ~ Drv + Cyl + Class)`

Observations:

- The new class is CompMidsize.
- All the other values were impacted by this change. This is a natural consequence of simplifying models. Sometimes the model will be improved by the simplification and sometimes it won't. We can determine this by looking at the output below the table. We will discuss shortly.
- Class Minivan and Class Pickup are still similar so we should combine.

Regression Output

```
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        36.6891     1.5689  23.386  < 2e-16 ***
drvf                3.2481     0.5787   5.613 5.84e-08 ***
drvr                0.9522     0.7089   1.343  0.18059
cyl                -1.6052     0.1432 -11.210  < 2e-16 ***
Classcompmidsize   -3.6377     1.3397  -2.715  0.00714 **
Classminivan       -8.2345     1.5262  -5.395 1.73e-07 ***
Classpickup        -8.5255     1.3693  -6.226 2.31e-09 ***
Classsubcompact    -2.7611     1.3102  -2.107  0.03619 *
Classsuv           -7.5447     1.2791  -5.898 1.34e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.487 on 225 degrees of freedom
Multiple R-squared:  0.8315,    Adjusted R-squared:  0.8256
F-statistic: 138.8 on 8 and 225 DF,  p-value: < 2.2e-16
```

Topic 5: Linear Modeling

Interpreting P-Values

H_0: $\beta_0 = \beta_1 = \beta_2 = \beta_3 = 0$

Regression Equation: $Hwy = \beta_0 + \beta_1 Drv + \beta_2 Cyl + \beta_3 Class$

R Equation: `lm(Hwy ~ Drv + Cyl + Class)`

Observations:

- The new class is MiniPickup. With this change each level of class has a distinguishable impact on highway fuel economy.
- All the other values are again impacted by this change.
- The "dvr" level is still insignificant so we can remove it as well.

Regression Output

```
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       36.6877     1.5657  23.433  < 2e-16 ***
drvf               3.3347     0.4864   6.856 6.69e-11 ***
drvr               0.9927     0.6924   1.434  0.15304
cyl               -1.6100     0.1418 -11.353  < 2e-16 ***
Classcompmidsize  -3.6840     1.3266  -2.777  0.00595 **
Classminipickup   -8.4402     1.3315  -6.339 1.24e-09 ***
Classsubcompact   -2.8000     1.3000  -2.154  0.03232 *
Classsuv          -7.5164     1.2725  -5.907 1.27e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.482 on 226 degrees of freedom
Multiple R-squared:  0.8315,	Adjusted R-squared:  0.8263
F-statistic: 159.3 on 7 and 226 DF,  p-value: < 2.2e-16
```

UNC CHARLOTTE 11

11

Topic 5: Linear Modeling

Interpreting P-Values

H_0: $\beta_0 = \beta_1 = \beta_2 = \beta_3 = 0$

Regression Equation: $Hwy = \beta_0 + \beta_1 Drv + \beta_2 Cyl + \beta_3 Class$

R Equation: `lm(Hwy ~ Drv + Cyl + Class)`

Observations:

- This is a pretty good model.
- The betas (β_i's) are all statistically significant.
- This means that are all significantly different from zero.
- We can reject the **Null Hypothesis** above.
- But what about the overall significance of the model?

Regression Output

```
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       37.1137     1.5408  24.088  < 2e-16 ***
drv1f              3.2056     0.4791   6.691 1.71e-10 ***
cyl               -1.5392     0.1332 -11.552  < 2e-16 ***
Classcompmidsize  -4.3522     1.2449  -3.496 0.000568 ***
Classminipickup   -9.3104     1.1879  -7.838 1.77e-13 ***
Classsubcompact   -3.2458     1.2652  -2.565 0.010952 *
Classsuv          -8.2598     1.1647  -7.092 1.66e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.488 on 227 degrees of freedom
Multiple R-squared:  0.8299,	Adjusted R-squared:  0.8255
F-statistic: 184.6 on 6 and 227 DF,  p-value: < 2.2e-16
```

UNC CHARLOTTE 12

12

H_0: σ_i are equal

Under this null hypothesis, we want to test if the model is accounting for a statistically significant amount of the residual error in the data. We can examine the allocation of the residual error between what is accounted for by the model and what is left over using the **Analysis of Variance (ANOVA) Table**.

Interpreting P-Values

H_0: σ_i are equal

R Equation: lm(Hwy ~ Drv + Cyl + Class)

```
Analysis of Variance Table

Response: hwy
           Df  Sum Sq Mean Sq F value    Pr(>F)
drv         2  4384.5 2192.27 354.010 < 2.2e-16 ***
cyl         1  1807.8 1807.84 291.933 < 2.2e-16 ***
class       6   682.1  113.69  18.359 < 2.2e-16 ***
Residuals 224  1387.2    6.19
Total     233  8261.60
```

Sum of Squares (SST)

Sum of Squares Regression (SSR)= 4,384.5 + 1,807.8 + 682.1
= 6,874.40

Mean Squares Regression (MSR) = 6,874.4/9 = 763.82

Sum of Squares Residuals = 1,392.2

Mean Square Error (MSE) = 1,392.2/224 = 6.19

F-Statistic = 763.82/6.19 = 123.3 ***

Conclusion: Reject H_0

Regression Output

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      36.4845     1.5876  22.981  < 2e-16 ***
drvf              3.3594     0.5933   5.663 4.55e-08 ***
drvr              0.9405     0.7095   1.326  0.18634
cyl              -1.5781     0.1467 -10.760  < 2e-16 ***
classcompact     -3.4357     1.3608  -2.525  0.01227 *
classmidsize     -3.9144     1.3784  -2.840  0.00493 **
classminivan     -8.2985     1.5289  -5.428 1.48e-07 ***
classpickup      -8.5111     1.3701  -6.212 2.52e-09 ***
classsubcompact  -2.7594     1.3110  -2.105  0.03642 *
classsuv         -7.5264     1.2800  -5.880 1.48e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.489 on 224 degrees of freedom
Multiple R-squared:  0.8321,    Adjusted R-squared:  0.8254
F-statistic: 123.3 on 9 and 224 DF,  p-value: < 2.2e-16
```

Note: Residual Standard Error = SQRT(MSE)
Multiple R^2 = SSR/SST

Interpreting P-Values

H_0: σ_i are equal

```
Analysis of Variance Table

Response: hwy
           Df Sum Sq Mean Sq F value    Pr(>F)    
drv         2 4384.5 2192.27 354.415 < 2.2e-16 ***
cyl         1 1807.8 1807.84 292.266 < 2.2e-16 ***
class       5  677.5  135.51  21.907 < 2.2e-16 ***
Residuals 225 1391.8    6.19
```

R Equation: lm(Hwy ~ Drv + Cyl + Class)

Regression Output

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)      36.6891     1.5689  23.386  < 2e-16 ***
drvf              3.2481     0.5787   5.613 5.84e-08 ***
drvr              0.9522     0.7089   1.343  0.18059    
cyl              -1.6052     0.1432 -11.210  < 2e-16 ***
classcompmidsize -3.6377     1.3397  -2.715  0.00714 ** 
classminivan     -8.2345     1.5262  -5.395 1.73e-07 ***
classpickup      -8.5255     1.3693  -6.226 2.31e-09 ***
classsubcompact  -2.7611     1.3102  -2.107  0.03619 *  
classsuv         -7.5447     1.2791  -5.898 1.34e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.487 on 225 degrees of freedom
Multiple R-squared:  0.8315,	Adjusted R-squared:  0.8256 
F-statistic: 138.8 on 8 and 225 DF,  p-value: < 2.2e-16
```

$$\mathrm{F-Statistic} = \frac{\frac{4{,}384.5+1{,}807.8+677.5}{8}}{\frac{1{,}391.8}{225}} = 138.8\ ***$$

Conclusion: Reject H_0

Interpreting P-Values

Left Intentionally Blank

Topic 5: Linear Modeling

Interpreting P-Values

H_0: σ_i are equal

```
Analysis of Variance Table

Response: hwy
           Df Sum Sq Mean Sq F value    Pr(>F)    
drv         2 4384.5 2192.27 355.868 < 2.2e-16 ***
cyl         1 1807.8 1807.84 293.465 < 2.2e-16 ***
class       4  677.1  169.26  27.476 < 2.2e-16 ***
Residuals 226 1392.2    6.16
```

R Equation: lm(Hwy ~ Drv + Cyl + Class)

Regression Output

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)      36.6877     1.5657  23.433  < 2e-16 ***
drvf              3.3347     0.4864   6.856 6.69e-11 ***
drvr              0.9927     0.6924   1.434  0.15304    
cyl              -1.6100     0.1418 -11.353  < 2e-16 ***
classcompmidsize -3.6840     1.3266  -2.777  0.00595 ** 
classminipickup  -8.4402     1.3315  -6.339 1.24e-09 ***
classsubcompact  -2.8000     1.3000  -2.154  0.03232 *  
classsuv         -7.5164     1.2725  -5.907 1.27e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.482 on 226 degrees of freedom
Multiple R-squared:  0.8315,	Adjusted R-squared:  0.8263 
F-statistic: 159.3 on 7 and 226 DF,  p-value: < 2.2e-16
```

$$\text{F} - \text{Statistic} = \frac{\frac{4{,}384.5+1{,}807.84+677.1}{7}}{\frac{1{,}392.2}{226}} = 159.30^{***}$$

Conclusion: Reject H_0

Topic 5: Linear Modeling

Interpreting P-Values

H_0: σ_i are equal

```
Analysis of Variance Table

Response: hwy
           Df Sum Sq Mean Sq F value    Pr(>F)    
drv1        1 4317.5  4317.5 697.613 < 2.2e-16 ***
cyl         1 1508.6  1508.6 243.750 < 2.2e-16 ***
class       4 1030.7   257.7  41.634 < 2.2e-16 ***
Residuals 227 1404.9     6.2
```

R Equation: lm(Hwy ~ Drv + Cyl + Class)

Regression Output

```
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)       37.1137     1.5408  24.088  < 2e-16 ***
drv1f              3.2056     0.4791   6.691 1.71e-10 ***
cyl               -1.5392     0.1332 -11.552  < 2e-16 ***
classcompmidsize  -4.3522     1.2449  -3.496 0.000568 ***
classminipickup   -9.3104     1.1879  -7.838 1.77e-13 ***
classsubcompact   -3.2458     1.2652  -2.565 0.010952 *  
classsuv          -8.2598     1.1647  -7.092 1.66e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.488 on 227 degrees of freedom
Multiple R-squared:  0.8299,	Adjusted R-squared:  0.8255 
F-statistic: 184.6 on 6 and 227 DF,  p-value: < 2.2e-16
```

$$\text{F} - \text{Statistic} = \frac{\frac{4{,}317.5+1{,}508.6+1030.07}{6}}{\frac{1{,}404.9}{227}} = 184.65^{***}$$

Conclusion: Reject H_0

Interpreting P-Values

Final Observations:

1. The Regression Output let's us test two separate hypotheses:
   - H_0: $\beta_0 = \beta_1 = \beta_2 = \beta_3 = 0$
   - H_0: σ_i are equal
2. When we can reject one but not the other, then there is likely some underlying problem with the data and/or model design that needs to be investigated.
3. The model should be thoroughly inspected to determine where simplifications are possible. Rationales should be sought to understand unnecessary complexity in a model.
4. The ANOVA table can be completely determined from the regression summary.

```
Residual standard error: 2.488 on 227 degrees of freedom
Multiple R-squared:  0.8299,    Adjusted R-squared:  0.8255
F-statistic: 184.6 on 6 and 227 DF,  p-value: < 2.2e-16
```


Linear Regression

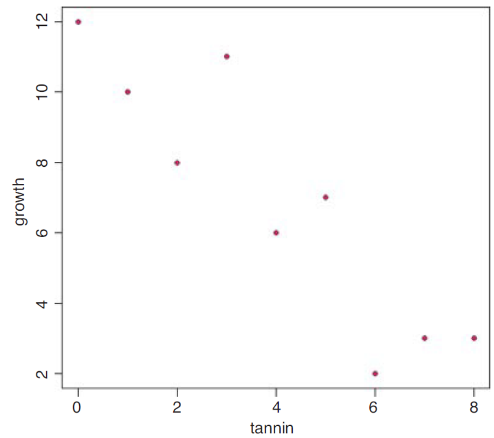
Let us start with an example which shows the growth of caterpillars fed on experimental diets differing in their tannin content:

```
reg.data <- read.table("c:\\temp\\regression.txt",header=T)
attach(reg.data)
names(reg.data)
[1] "growth" "tannin"
```

Linear Regression

```
plot(tannin,growth,pch=21,col="blue",bg="red")
```

The higher the percentage of tannin in the diet, the more slowly the caterpillars grew. Tannins in oak leaves inhibit gypsy moth growth. Gypsy moths are harmful to trees. They defoliate them.



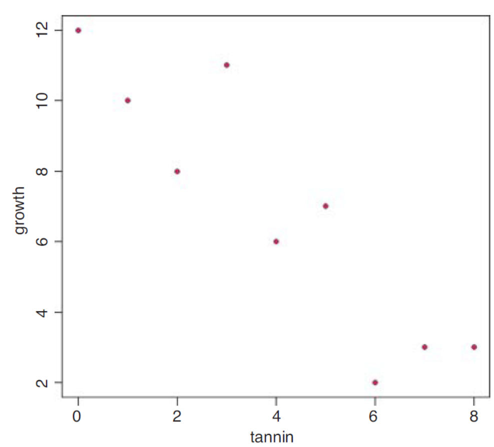
21

Linear Regression

```
plot(tannin,growth,pch=21,col="blue",bg="red")
```

You can get a crude estimate of the parameter values by eye. Tannin content increased by 8 units, in response to which growth declined from about 12 units to about 2 units, a change of –10 units of growth. The slope, b, is the change in y divided by the change in x, so

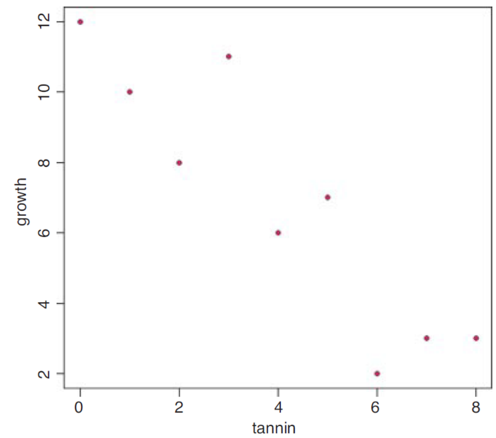
$$b \approx \frac{-10}{8} = -1.25$$



22

Linear Regression

```
plot(tannin,growth,pch=21,col="blue",bg="red")
```



The intercept, a, is the value of y when $x = 0$, and we see by inspection of the scatterplot that growth was close to 12 units when tannin was zero. Thus, our rough parameter estimates allow us to write the regression equation as

$$y \approx 12.0 - 1.25x$$

Of course, different people would get different parameter estimates by eye. What we want is an objective method of computing parameter estimates from the data that are in some sense the 'best' estimates of the parameters for these data and this particular model.

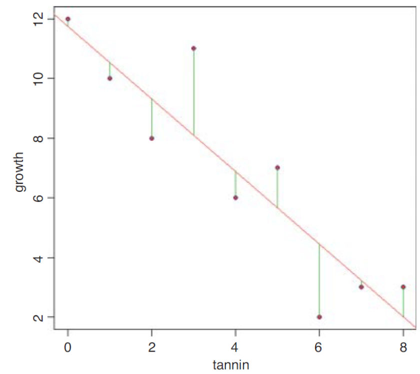
Linear Regression

The convention in modern statistics is to use the **maximum likelihood estimates** of the parameters as providing the 'best' estimates. That is to say that, given the data, and having selected a linear model, we want to find the values of the slope and intercept that make the data most likely.

Linear Regression

Important Assumptions

1. The variance in y is constant (i.e. the variance does not change as y gets bigger). The explanatory variable, x, is measured without error.
2. The difference between a measured value of y and the value predicted by the model for the same value of x is called a residual.
3. Residuals are measured on the scale of y (i.e. parallel to the y axis).
4. The residuals are normally distributed.



Linear Regression

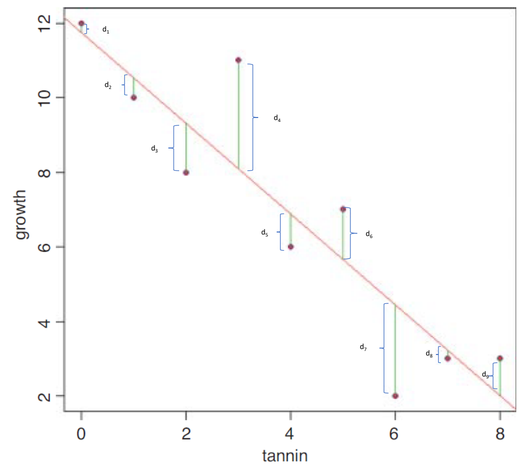
```
model <- lm(growth~tannin)#R Function for Linear Model
abline(model,col="red")
yhat <- predict(model,tannin=tannin)
join <- function(i)
lines(c(tannin[i],tannin[i]),c(growth[i],yhat[i]),col="green")
sapply(1:9,join)
```

Linear Regression

Residuals d_i

Under these assumptions, the maximum likelihood is given by the **method of least squares**. The phrase 'least squares' refers to the residuals, as shown in the figure. The residuals are the vertical differences between the data (solid circles) and the fitted model (the straight line). Each of the residuals is a distance, *d*, between a data point, *y*, and the value predicted by the fitted model, $\hat{y}$, evaluated at the appropriate value of the explanatory variable, *x*:

$$d = y - \hat{y}$$



Topic 5: Linear Modeling

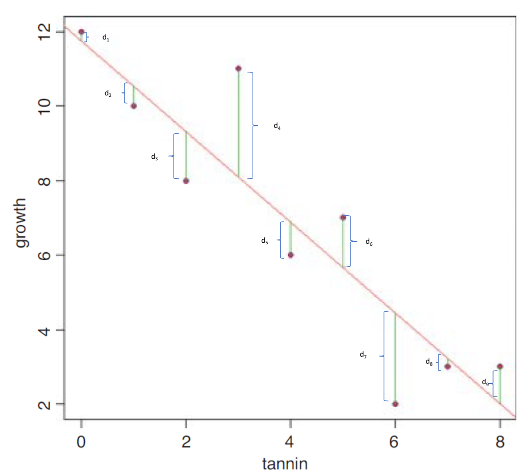
Linear Regression

Residuals d_i

Now we replace the predicted value $\hat{y}$ by its formula $\hat{y} = a + bx$, noting the change in sign

$$d = y - a - bx$$

$$\sum d^2 = \underbrace{\sum (y - a - bx)^2}_{\text{Sum of Squares Errors aka Residuals}}$$



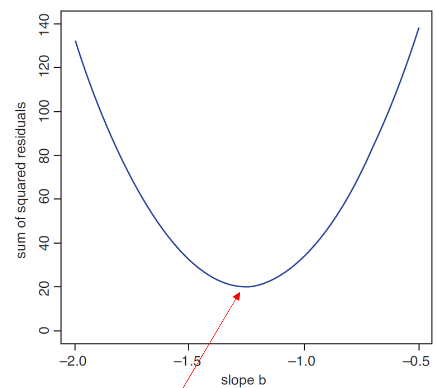
Linear Regression

```
lm(growth~tannin)

Coefficients:
(Intercept)   tannin
11.756        -1.217
```

We can now write the maximum likelihood equation like this:

growth = 11.755 56 – 1.216 667 × tannin.

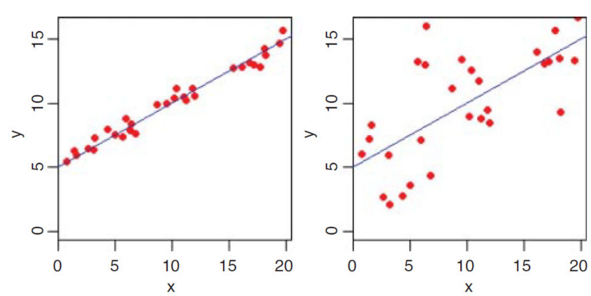
```
bs <- seq(-2,-0.5,0.01)
SSE <- function(i) sum((growth - 12 - bs[i]*tannin)^2)
plot(bs,sapply(1:length(bs),SSE),type="l",ylim=c(0,140),
xlab="slope b",ylab="sum of squared residuals",col="blue")
```



The slope that maximizes the likelihood because it minimizes the sum of squares error

Degree of Scatter

There is another very important issue that needs to be considered, because two data sets with exactly the same slope and intercept could look quite different:



We need a way to quantify the degree of fit, so that the graph on the left has a high value and the graph on the right has a low value.

Sums of Squares Total (SST)

SST = SSR + SSE

df(SST) = df(SSR) + df(SSE) = N - 1

N = The total number of obsetrvations

df = Degrees of Fredom

One (1) degree of freedom is lost because the regression calculates the mean. This leaves N-1 degrees of freedom for the Sums of Squares Total (SST). The mean is one calculated metric that describes the data.

Sums of Squares Total (SST)

Sum of Squares Total (SST) = Sum of Squares Regression (SSR) + Sum of Squares Errors (SSE)

$$SST = \sum (y_i - \bar{y})^2$$

#The total deviation is the sum of the differences between **actual values** and the mean. This is the numerator of the variance.

$$SSR = \sum (\hat{y}_i - \bar{y})^2$$

#The total deviation is the sum of the differences of **predicted values** and the mean. This is the numerator of the variance.

$$SSE = \sum (y_i - \hat{y}_i)^2$$

#The total deviation is the sum of the differences between actual and predicted values. This is the numerator of the variance. These are the **Squared Residuals** or **Deviance**.

Degree of Scatter

It turns out that we already have the appropriate quantity: it is the sum of squares of the residuals (p. 338). This is referred to as the *error sum of squares*, *SSE*. Here, **error** does not mean ‘mistake’, but refers to residual variation or *unexplained variation*:

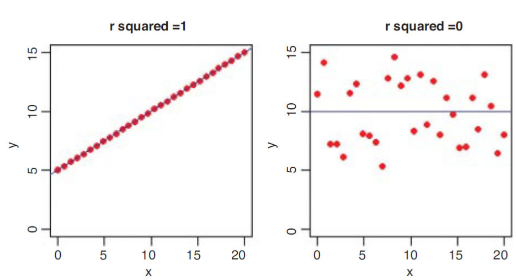
$$SSE = \sum (y - \underbrace{a - bx}_{\text{The Predicted y}})^2$$

The Actual y

Graphically, you can think of *SSE* as the sum of the squares of the lengths of the vertical residuals.

By tradition, however, when talking about the degree of scatter we actually quantify the *lack* of scatter, so the graph on the left, with a perfect fit (zero scatter) gets a value of 1, and the graph on the right, which shows no relationship at all between *y* and *x* (100% scatter), gets a value of 0.

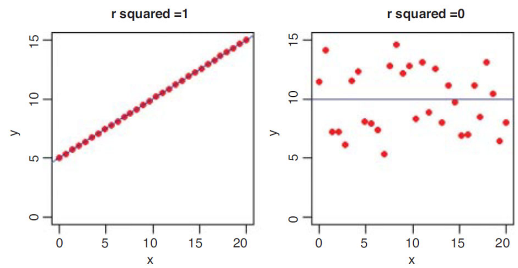
This quantity used to measure the lack of scatter is officially called the coefficient of determination, but everybody refers to it as ‘R squared’.



R squared or R^2

Definition: The fraction of the total variation in *y* that is explained by variation in *x*.

$$R^2 = \frac{SSR}{SST}$$



A value of $r^2 = 1$ means that all of the variation in the response variable is explained by variation in the explanatory variable (the left-hand graph below) while a value of $r^2 = 0$ means none of the variation in the response variable is explained by variation in the explanatory variable (the right-hand graph)

```
y <- 5+0.5*x
plot(x,y,pch=16,xlim=c(0,20),ylim=c(0,15),
col="red",main="r squared = 1")
abline(5,0.5,col="blue")

y <- 5+runif(30)*10
plot(x,y,pch=16,xlim=c(0,20),ylim=c(0,15),
col="red",main="r squared = 0")
abline(h=10,col="blue")
```


Anova Table

```
model <- lm(growth~tannin)
summary(model)
anova(model)
```

```
Analysis of Variance Table

Response: growth
          Df  Sum Sq  Mean Sq  F value  Pr(>F)
tannin     1  88.817  88.817   30.974   0.0008461 ***
Residuals  7  20.072  2.867
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Deviance

Definition: The sum of the squares of the residuals of the model

```
deviance(lm(growth~1))
[1] 108.8889
```

The Intercept Model

```
deviance(lm(growth~tannin))
[1] 20.07222
```

The Intercept Model + Tannin

Notice the huge decrease in deviance!!! This tells us the variable tannin in explaining a huge amount of the variation in growth.

Calculating R^2

Now we can calculate the value of R^2:

$$R^2 = \frac{SST - SSE}{SST} = \frac{108.8889 - 20.0722}{108.8889} = 0.815663$$

You will not be surprised that the value of r_2 can be extracted from the model:

```
summary(lm(growth~tannin))[[8]]
[1] 0.8156633
```

Correlation Coefficient

The **correlation coefficient**, r is given by

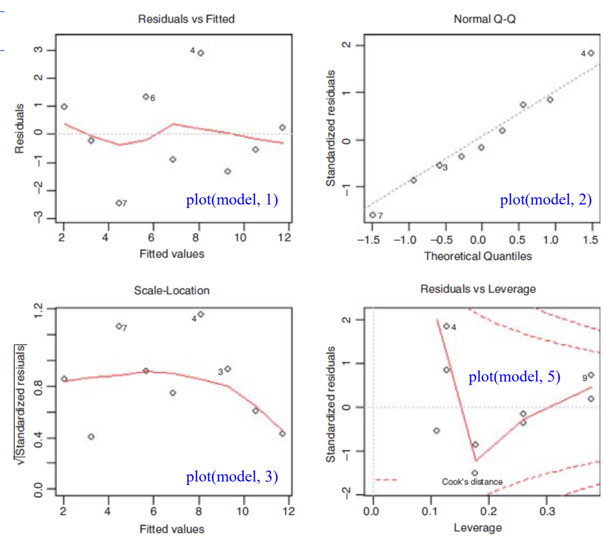
$$r = \frac{SSXY}{\sqrt{SSX \times SSY}}$$

$$r = \frac{-73}{\sqrt{60 \times 108.8889}} = -0.903\ 140\ 7.$$

Model checking

```
windows(7,7)
par(ask = F,mfrow=c(2,2))
plot(model)
```

Note: plot(model) only gives the four plots to the right. They are plot numbers 1, 2, 3, and 5. We can get the other two by asking for them by number.

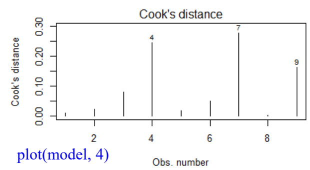


Model Plots

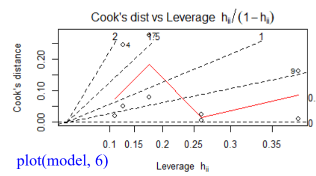
1. plot(model, 1): a plot of residuals against fitted values
2. plot(model, 2): a scale–location plot of √|residuals| against fitted values
3. plot(model, 3): a normal quantile–quantile plot
4. plot(model, 4): a plot of Cook's distances versus observation number
5. plot(model, 5): a plot of residuals against leverages
6. plot(model, 6): a plot of Cook's distances against leverage/(1 – leverage)

Code to generate all 6

```
>par(mfrow=c(2,3))
>plot(model, which=1:6)
```



plot(model, 4)



plot(model, 6)

Updating model without outlier

```
model2 <- update(model,subset=(tannin != 6))
summary(model2)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   11.6892     0.8963  13.042 1.25e-05 ***
tannin        -1.1171     0.1956  -5.712  0.00125 **
```

Changes:

1. We have lost one degree of freedom, because there are now eight values of *y* rather than nine.
2. The estimate of the slope has changed from –1.2167 to –1.1171 (a difference of about 9%)
3. The standard error of the slope has changed from 0.2186 to 0.1956 (a difference of about 12%).

Applying a Natural Log Transformation

A two-parameter model of exponential decay in which the amount of material remaining (y) is a function of time (t):

$$y = y_0 e^{-bt}$$

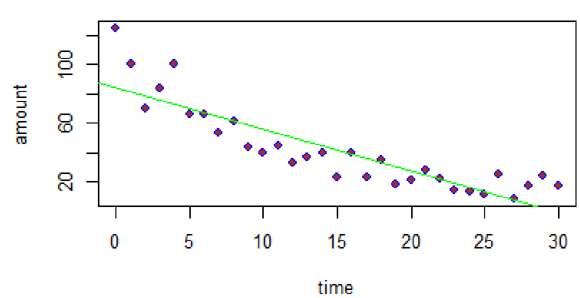
This is NOT a linear model, but we can make it a linear model by applying the Natural Logarithm to both sides:

$$\log(y) = \log(y_0) - bt$$

Now we can apply linear regression techniques!

Applying a Natural Log Transformation

```
data <- read.table("c:\\temp\\Decay.txt",header=T)
names(data)
attach(data)
plot(time,amount,pch=21,col="blue",bg="brown")
abline(lm(amount~time),col="green")
```



Applying a Natural Log Transformation

```
model <- lm(log(amount)~time)
summary(model)
```

```
Call:
lm(formula = log(amount) ~ time)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.5935 -0.2043  0.0067  0.2198  0.6297 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.547386   0.100295   45.34  < 2e-16 ***
time        -0.068528   0.005743  -11.93 1.04e-12 ***

Residual standard error: 0.286 on 29 degrees of freedom
Multiple R-squared: 0.8308,	Adjusted R-squared: 0.825 
F-statistic: 142.4 on 1 and 29 DF,  p-value: 1.038e-12
```

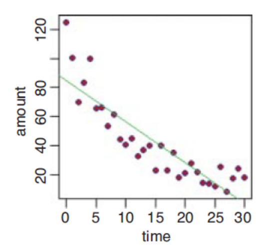

Applying a Natural Log Transformation

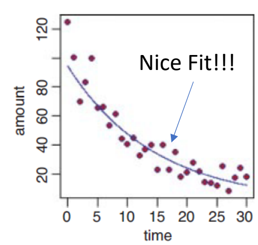
Thus, the **slope is – 0.068 528** and y_0 is the antilog of the intercept: y_0 **= exp(4.547 386) = 94.385 36**. The formula in its original form is:

$$y = 94.385^{-0.0685t}$$

We can draw the fitted line through the data, remembering to take the antilogs of the predicted values (the model predicts `log(amount)` and we want `amount`), like this

```
ts <- seq(0,30,0.02)
left <- exp(predict(model,list(time=ts)))
plot(time,amount,pch=21,col="blue",bg="brown")
lines(ts,left,col="blue")
```





Power Function

$y = ax^b$ } Power Function, but we can still apply a log transformation

Taking the log transformation, we get:

$$\ln(y) = \ln(a) + b\ln(x)$$

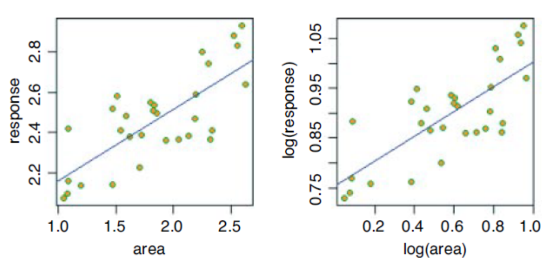
This has a linear form:

$$y' = a' + bx'$$

We can now apply linear regression techniques.

Example

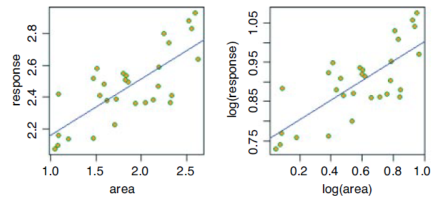
```
power <- read.table("c:\\temp\\power.txt",header=T)
attach(power)
names(power)
plot(area,response,pch=21,col="green",bg="orange")
abline(lm(response~area),col="blue")
plot(log(area),log(response),pch=21,col="green",bg="orange")
abline(lm(log(response)~log(area)),col="blue")
```



Example

The two plots look very similar (this is not always the case), but we need to compare the two models:

```
model1 <- lm(response~area)
model2 <- lm(log(response)~log(area))
summary(model2)
```

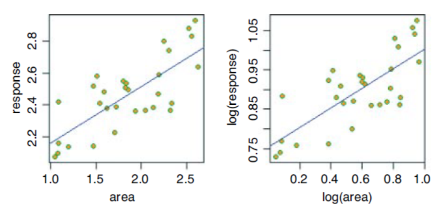



```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.75378    0.02613  28.843  < 2e-16 ***
log(area)    0.24818    0.04083   6.079 1.48e-06 ***
```


Example

We need to do a t test to see whether the estimated shape parameter, b = 0.248 18, is significantly less than b = 1 (a straight line):

$$t = \frac{|0.24818 - 1.0|}{0.04083} = 18.41342.$$

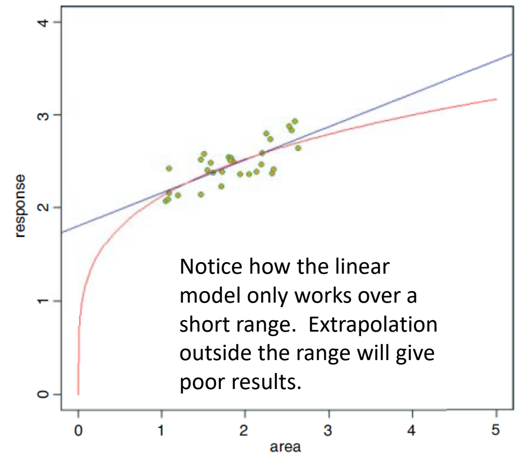


This is highly significant ($p < 0.0001$), so we conclude that there is a non-linear relationship between `response` and `area`.

Example

Let us get a visual comparison of the two models:

```
plot(area,response,pch=21,col="green",
bg="orange")
abline(lm(response~area),col="blue")
xv <- seq(1,2.7,0.01)
yv <- exp(0.75378)*xv^0.24818
lines(xv,yv,col="red")

plot(area,response,xlim=c(0,5),ylim=c(
0,4),pch=21,col="green",bg="orange")
abline(lm(response~area),col="blue")
xv <- seq(0,5,0.01)
yv <- exp(0.75378)*xv^0.24818
lines(xv,yv,col="red")
```



Prediction following regression

There are two kinds of prediction:

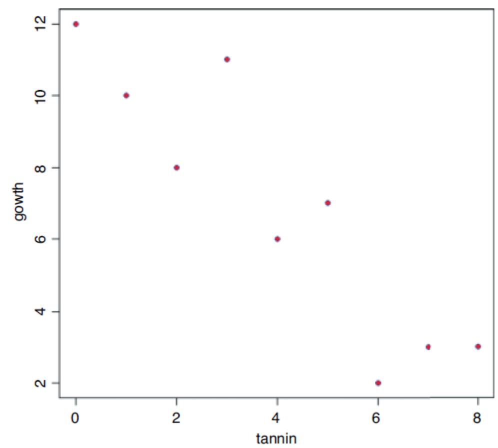
1. **Interpolation**, which is prediction *within* the measured range of the data, can often be very accurate and is not greatly affected by model choice.

2. **Extrapolation**, which is prediction *beyond* the measured range of the data, is far more problematical, and model choice is a major issue.

Prediction following regression

Here are two kinds of plots involved in prediction following regression: the first illustrates uncertainty in the parameter estimates; the second indicates uncertainty about predicted values of the response. We continue with the tannin example:

```
reg.data <-
read.table("c:\\temp\\regression.txt",header=T)
attach(reg.data)
names(reg.data)
plot(tannin,growth,pch=21,col="blue",bg="red")
```

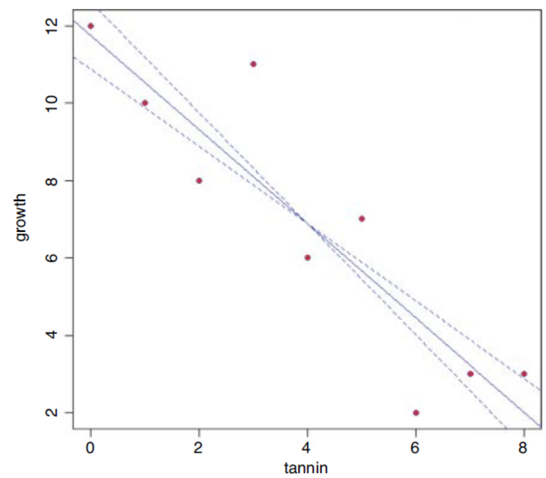


Prediction following regression

```
model <- lm(growth~tannin)
abline(model,col="blue")

The Slope
coef(model)[2]
tannin
-1.216667

The Standard Error
summary(model)[[4]][4]
[1] 0.2186115
```

Prediction following regression

```
se.lines <- function(model){
b1 <- coef(model)[2]+
summary(model)[[4]][4]
b2 <- coef(model)[2]-
summary(model)[[4]][4]
xm <- sapply(model[[12]][2],mean)
ym <- sapply(model[[12]][1],mean)
a1 <- ym-b1*xm
a2 <- ym-b2*xm
abline(a1,b1,lty=2,col="blue")
abline(a2,b2,lty=2,col="blue")
}
se.lines(model)
```



Prediction following regression

We are interested in the uncertainty about predicted values rather than uncertainty of parameter estimates, as above.

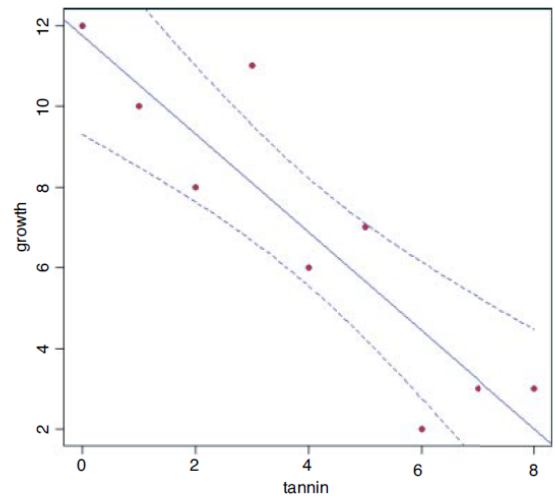
```
ci.lines <- function(model){
xm <- sapply(model[[12]][2],mean)
n <- sapply(model[[12]][2],length)
ssx <- sum(model[[12]][2]^2)-sum(model[[12]][2])^2/n
s.t <- qt(0.975,(n-2))
xv <- seq(min(model[[12]][2]),max(model[[12]][2]),length=100)
yv <- coef(model)[1]+coef(model)[2]*xv
se <- sqrt(summary(model)[[6]]^2*(1/n+(xv-xm)^2/ssx))
ci <- s.t*se
uyv <- yv+ci
lyv <- yv-ci
lines(xv,uyv,lty=2,col="blue")
lines(xv,lyv,lty=2,col="blue")
}
plot(tannin,growth,pch=21,col="blue",bg="red")
abline(model, col= "blue")
```

This code plots the confidence lines around the regression line.

Prediction following regression

```
ci.lines(model)
```

Points at `tannin` = 3 and `tannin` = 6 that fall outside the 95% confidence limits of our fitted values.



Testing for lack of fit in a regression

We want

1. To make the error variance as small as possible.
2. We want to make *SSX* as large as possible, by placing as many points as possible at the extreme ends of the *x* axis.

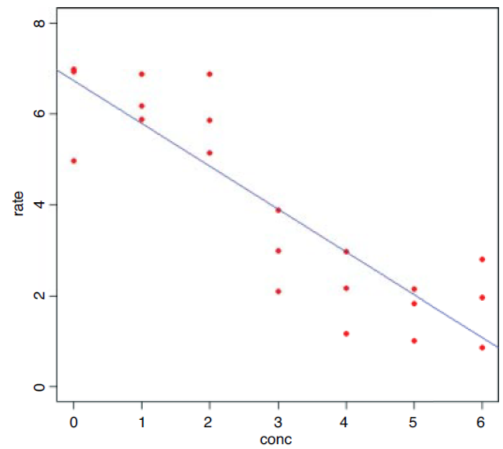
Efficient regression designs allow for:

1. replication of least some of the levels of *x*;
2. a preponderance of replicates at the extremes (to maximize *SSX*);
3. sufficient levels of *x* to allow testing for non-linearity;
4. sufficient different values of *x* to allow accurate location of thresholds.

Testing for lack of fit in a regression

Here is an example where replication allows estimation of pure sampling error, and this in turn allows a test of the significance of the data's departure from linearity. As the concentration of an inhibitor is increased, the reaction rate declines:

```
data <- ead.delim("c:\\temp\\lackoffit.txt")
attach(data)
names(data)
plot(conc,jitter(rate),pch=16,col="red",ylim
=c(0,8),ylab="rate")
abline(lm(rate~conc),col="blue")
```



Topic 5: Linear Modeling

Testing for lack of fit in a regression

The linear regression does not look too bad, and the slope is highly significantly different from zero:

```
model.reg <- lm(rate~conc)
summary(model.reg)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.7262     0.4559  14.755 7.35e-12 ***
conc         -0.9405     0.1264  -7.439 4.85e-07 ***
Residual standard error: 1.159 on 19 degrees of freedom
Multiple R-squared: 0.7444,     Adjusted R-squared: 0.7309
F-statistic: 55.33 on 1 and 19 DF,  p-value: 4.853e-07
```

Pure Error Variance

Because there is replication at each level of x we can do something extra, compared with a typical regression analysis. We can estimate what is called the **pure error variance**. This is the sum of the squares of the differences between the y values and the *mean* values of y for the relevant level of x. **It is the definition of *SSE* from a one-way analysis of variance.**

Homework: Simple Linear Regression

#1. Use the bmi_data.csv data set for this exercise.

1. Assign the data set to the variable “bmi”
2. Calculate the correlation coefficient between Height & Weight
3. Define the linear model
4. Create exploratory plots of the model
5. Create the plot of Height as a function of Weight and add the smooth regression line.
6. Add confidence bands to the regression plot in #4 above
7. Create the regression output
8. Create the ANOVA table
9. What do the results tell you about the relationship between Height and Weight?
10. Are assumptions of normality violated as evidenced by the residuals?
11. How much of the variation in Height is explained by the variation in years of experience?
12. In your opinion is this a good model.

Homework: Simple Linear Regression

#2. Use the Salary_Data.csv data set for this exercise.

1. Assign the data set to the variable "salary"
2. Calculate the correlation coefficient between salary and years of experience
3. Define the linear model
4. Create exploratory plots of the model
5. Create the plot of Salary as a function of YearsExperience and add the smooth regression line.
6. Add confidence bands to the regression plot in #4 above
7. Create the regression output
8. Create the ANOVA table
9. What do the results tell you about the relationship between salary and years of experience?
10. Are assumptions of normality violated as evidenced by the residuals?
11. How much of the variation in salary is explained by the variation in weight?
12. In your opinion is this a good model.

Multiple Regression

A multiple regression is a statistical model with two or more continuous explanatory variables. Multiple regressions models provide some of the most profound challenges faced by the analyst because of some crucial issues:

1. Over-fitting (we often have more explanatory variables than data points)
2. Parameter proliferation (we might want to fit parameters for curvature and interaction)
3. Correlation between explanatory variables (called collinearity)
4. Choice between contrasting models of roughly equal explanatory power

Multiple Regression

The *principle of parsimony* (Occam's razor is again relevant here. It requires that the model should be as simple as possible. This means that **the model should not contain any redundant parameters**. Ideally, we achieve this by fitting a maximal model and then simplifying it by following one or more of these steps:

1. Remove non-significant interaction terms.
2. Remove non-significant quadratic or other non-linear terms.
3. Remove non-significant explanatory variables.
4. Amalgamate explanatory variables that have similar parameter values.

Topic 5: Linear Modeling

Important Approach to Correlated Variables

It is likely that many of the explanatory variables are correlated with each other, and so *the order in which variables are deleted from the model* will influence the explanatory power attributed to them.

There are no hard-and-fast rules about the best way to proceed, but we shall typically carry out simplification of a complex model by stepwise deletion: non-significant terms are left out, and significant terms are added back.

The multiple regression model

There are several important issues involved in carrying out a multiple regression:

1. Which explanatory variables to include;
2. Curvature in the response to the explanatory variables;
3. Interactions between explanatory variables;
4. Correlation between explanatory variables;
5. The risk of overparameterization.

Assumptions for Multiple Regression

The assumptions about the response variable are the same as with simple linear regression:

1. The errors are normally distributed
2. The errors are confined to the response variable,
3. The variance is constant.

The explanatory variables are assumed to be measured without error.

The model for a multiple regression with two explanatory variables (x_1 and x_2) looks like this:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i$$

Multiple Regression

The model for a multiple regression with k explanatory variables looks like this:

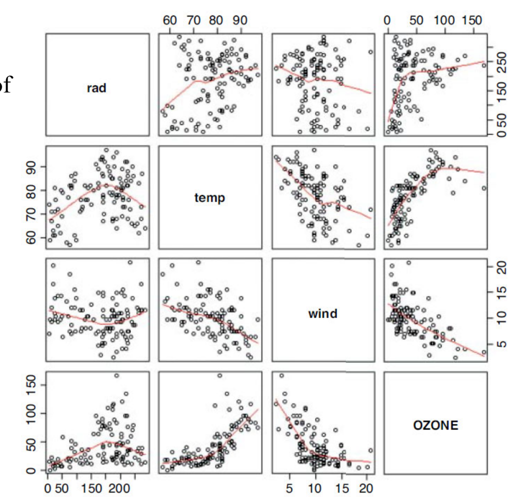
$$y_i = \sum_{j=0}^{k} \beta_j x_{ji} + \varepsilon_i$$

where $x_{0i} = 1$

Multiple Regression Example

Let us begin with an example from air pollution studies. How is ozone concentration related to wind speed, air temperature and the intensity of solar radiation?

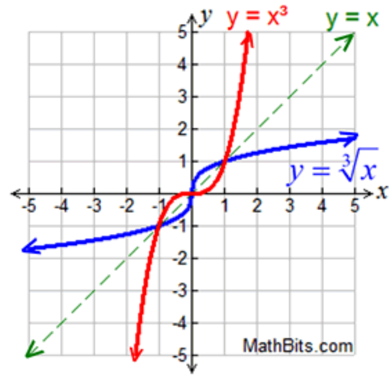
```
ozone.pollution <-
read.table("c:\\temp\\ozone.data.txt",header=T)
attach(ozone.pollution)
names(ozone.pollution)
pairs(ozone.pollution,panel=panel.smooth)
```



Variable Transformations

Cube Root Transformation

- Fairly strong transformation with a substantial effect on distribution shape
- Weaker than the logarithm transformation
- Used for reducing right skewness
- Can be applied to zero and negative values
- Commonly applied to rainfall data.

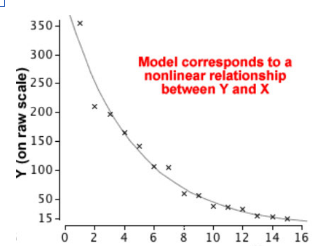


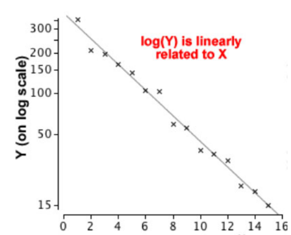
71

Variable Transformations

Logarithm Transformation

- A strong transformation with a major effect on distribution shape.
- Commonly used for reducing right skewness
- Often appropriate for measured variables.
- Can not be applied to zero or negative values.





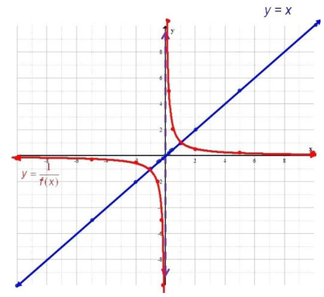
72

Variable Transformations

Reciprocal Transformation

- A very strong transformation with a drastic effect on distribution shape.
- It can not be applied to zero values
- It can be applied to negative values
- It is not useful unless all values are positive
- As easy to interpreted as the ratio itself
- Used to reverse order the values

Example: Population density (people per unit area) becomes area per person

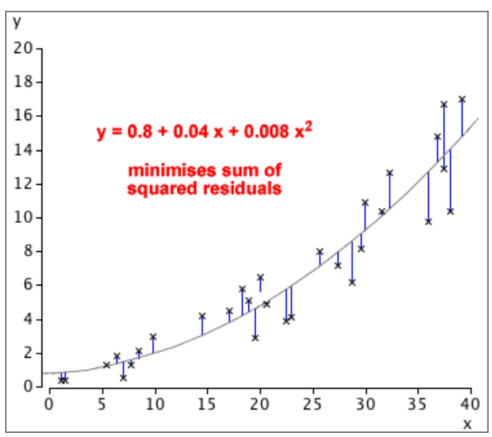


Variable Transformations

$$\sum e_i^2 = \sum(y_i - \hat{y}_i)^2 = \sum(y_i - b_0 - b_1 x_i - b_2 x_i^2)^2$$

Square Transformation

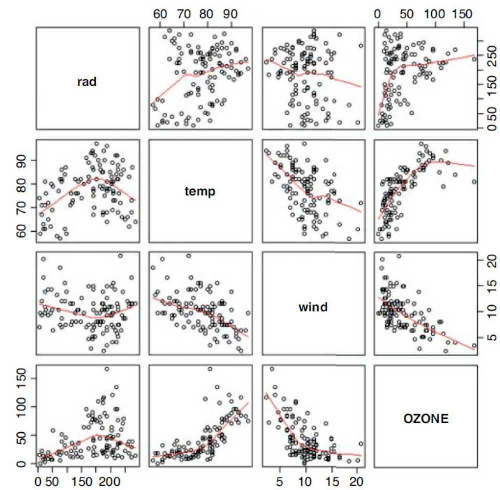
- A moderate effect on distribution shape
- Used to reduce left skewness.



Multiple Regression Example

Observations:

1. The response variable, ozone concentration, is shown on the *y* axis of the bottom row of panels:
   a) there is a strong negative relationship with wind speed
   b) a positive correlation with temperature
   c) a rather unclear, humped relationship with radiation.
2. Wind and temperature are negatively correlated.
3. Wind and radiation are relatively uncorrelated.
4. Temperature and radiation have unclear humped relationship.



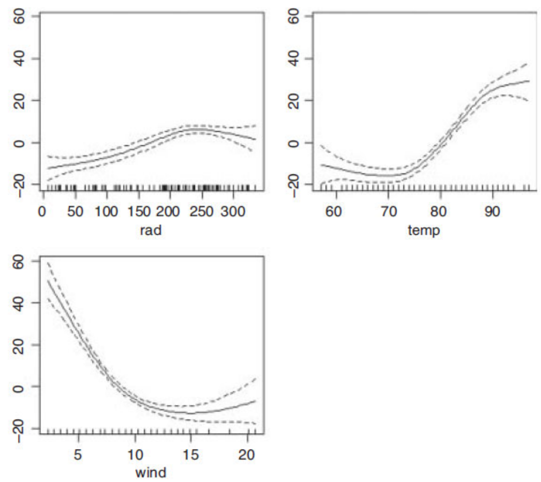
Multiple Regression Example

A good way to tackle a multiple regression problem is using non-parametric smoothers in a generalized additive model like this:

```
library(mgcv)
par(mfrow=c(2,2))
model <- gam(ozone~s(rad)+s(temp)+s(wind))
plot(model)
```

Multiple Regression Example

```
library(mgcv)
par(mfrow=c(2,2))
model <- gam(ozone~s(rad)+s(temp)+s(wind))
plot(model)
```

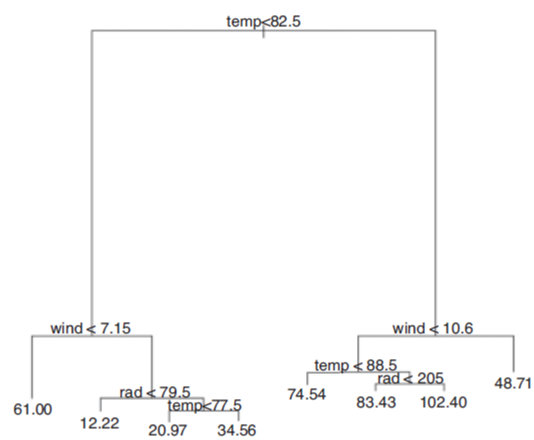


The confidence intervals are sufficiently narrow to suggest that the curvature in the relationships between ozone and temperature and ozone and wind are real, but the curvature of the relationship with solar radiation is marginal. The plots lead us to anticipate that quadratic terms for temperature and wind should be included in our initial model.

Multiple Regression Example

What about interactions? This is where tree models can help:

```
library(tree)
model <- tree(ozone~.,data=ozone.pollution)
par(mfrow=c(1,1))
plot(model)
text(model)
```



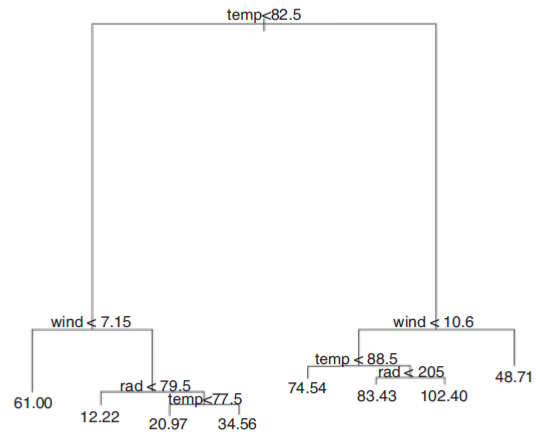
This shows that temperature is by far the most important factor affecting ozone concentration (the longer the branches in the tree, the greater the deviance explained).

Multiple Regression Example

This shows that temperature is by far the most important factor affecting ozone concentration (the longer the branches in the tree, the greater the deviance explained).

Wind speed is important at both high and low temperatures, with still air being associated with higher mean ozone levels (the figures at the ends of the branches).

There is a hint of an interaction between wind and radiation and between wind and temperature, because radiation and temperature change based on changes in wind.



Multiple Regression Example

We could include these in an initial complex model, degrees of freedom permitting:

```
w2 <- wind^2
t2 <- temp^2
r2 <- rad^2
tw <- temp*wind
wr <- wind*rad
tr <- temp*rad
wtr <- wind*temp*rad
```

Multiple Regression Example

Armed with this background information we can begin the linear modelling. We start with the most complicated model: this includes curvature terms for each variable, all three two-way interactions and a three-way interaction:

```
model1 <- lm(ozone~rad+temp+wind+t2+w2+r2+wr+tr+tw+wtr)
summary(model1)
```


Multiple Regression Example

```
model1 <- lm(ozone~rad+temp+wind+t2+w2+r2+wr+tr+tw+wtr)
summary(model1)
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.683e+02  2.073e+02   2.741  0.00725 **
rad         -3.117e-01  5.585e-01  -0.558  0.57799
temp        -1.076e+01  4.303e+00  -2.501  0.01401 *
wind        -3.237e+01  1.173e+01  -2.760  0.00687 **
t2           5.833e-02  2.396e-02   2.435  0.01668 *
w2           6.106e-01  1.469e-01   4.157 6.81e-05 ***
r2          -3.619e-04  2.573e-04  -1.407  0.16265
wr           2.054e-02  4.892e-02   0.420  0.67552
tr           8.403e-03  7.512e-03   1.119  0.26602
tw           2.377e-01  1.367e-01   1.739  0.08519 .
wtr         -4.324e-04  6.595e-04  -0.656  0.51358

Residual standard error: 17.82 on 100 degrees of freedom
Multiple R-squared: 0.7394,     Adjusted R-squared: 0.7133
F-statistic: 28.37 on 10 and 100 DF,  p-value: < 2.2e-16
```

P-values indicate some of the variables are not helpful to the model, although the overall model is statistically significant, p-value < 2.2e-16

Multiple Regression Example

We start by removing the highest-order interaction. An excellent feature of R is that the *p* values are '*p* values on deletion' so we do not have to use `anova` to compare the models produced by stepwise deletions:

```
model2 <- update(model1,~.-wtr)
summary(model2)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.245e+02  1.957e+02   2.680   0.0086 **
rad          2.628e-02  2.142e-01   0.123   0.9026
temp        -1.021e+01  4.209e+00  -2.427   0.0170 *
wind        -2.802e+01  9.645e+00  -2.906   0.0045 **
t2           5.953e-02  2.382e-02   2.499   0.0141 *
w2           6.173e-01  1.461e-01   4.225 5.25e-05 ***
r2          -3.388e-04  2.541e-04  -1.333   0.1855
wr          -1.127e-02  6.277e-03  -1.795   0.0756 .
tr           3.750e-03  2.459e-03   1.525   0.1303
tw           1.734e-01  9.497e-02   1.825   0.0709 .
```

The least significant term is the quadratic term for radiation, so we remove that.

Topic 5: Linear Modeling

Multiple Regression Example

```
model3 <- update(model2,~.-r2)
summary(model3)
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 486.346603 194.333075   2.503  0.01392 *
rad          -0.043163   0.208535  -0.207  0.83644
temp         -9.446780   4.185240  -2.257  0.02613 *
wind        -26.471461   9.610816  -2.754  0.00697 **
t2            0.056966   0.023835   2.390  0.01868 *
w2            0.599709   0.146069   4.106 8.14e-05 ***
wr           -0.011359   0.006300  -1.803  0.07435 .
tr            0.003160   0.002428   1.302  0.19600
tw            0.157637   0.094595   1.666  0.09869 .
```

The temperature by radiation interaction is not significant, so it goes next.

Multiple Regression Example

```
model4 <- update(model3,~.-tr)
summary(model4)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 514.401470 193.783580   2.655  0.00920 **
rad           0.212945   0.069283   3.074  0.00271 **
temp        -10.654041   4.094889  -2.602  0.01064 *
wind        -27.391965   9.616998  -2.848  0.00531 **
t2            0.067805   0.022408   3.026  0.00313 **
w2            0.619396   0.145773   4.249 4.72e-05 ***
wr           -0.013561   0.006089  -2.227  0.02813 *
tw            0.169674   0.094458   1.796  0.07538 .
```

The temperature by wind interaction is the next to go (it is marginally significant, but it should go.

Multiple Regression Example

```
model5 <- update(model4,~.-tw)
summary(model5)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 223.573855 107.618223   2.077 0.040221 *
rad           0.173431   0.066398   2.612 0.010333 *
temp         -5.197139   2.775039  -1.873 0.063902 .
wind        -10.816032   2.736757  -3.952 0.000141 ***
t2            0.043640   0.018112   2.410 0.017731 *
w2            0.430059   0.101767   4.226 5.12e-05 ***
wr           -0.009819   0.005783  -1.698 0.092507 .
```

There is no place for the wind by rain interaction.

Multiple Regression Example

```
model6 <- update(model5,~.-wr)
summary(model6)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 291.16758  100.87723   2.886  0.00473 **
rad           0.06586    0.02005   3.285  0.00139 **
temp         -6.33955    2.71627  -2.334  0.02150 *
wind        -13.39674    2.29623  -5.834 6.05e-08 ***
t2            0.05102    0.01774   2.876  0.00488 **
w2            0.46464    0.10060   4.619 1.10e-05 ***
```

The next job is to subject `model6` to criticism.

Multiple Regression Example

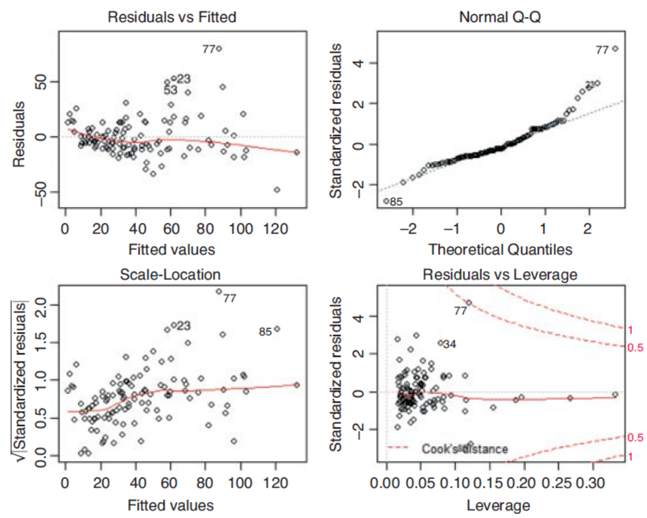
Let's check the AIC of the models:

```
>AIC(model1, model2, model3, model4, model5, model6)

              df      AIC
    model1    12     966.8062
    model2    11     965.2823
    model3    10     965.2184
    model4     9     965.0468
    model5     8     966.4707
    model6     7     967.5059
```

Multiple Regression Example

```
par(mfrow=c(2,2))
plot(model6)
```

This is quite seriously badly behaved. The residuals increase with the fitted values (non-constant variance) and the errors are not normal.



Multiple Regression Example

Let us try transforming the response variable. Having done this we need to start the modelling from scratch with all of the original explanatory variables included. Having transformed the response variable, we should expect that the curvature has been altered:

Multiple Regression Example

Let us try transforming the response variable. Having done this we need to start the modelling from scratch with all of the original explanatory variables included. Having transformed the response variable, we should expect that the curvature has been altered:

```
model7 <- lm(log(ozone) ~ rad+temp+wind+t2+w2+r2+wr+tr+tw+wtr)
```


Multiple Regression Example

```
model7 <- lm(log(ozone)~rad+temp+wind+t2+w2+r2+wr+tr+tw+wtr)
summary(model7)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.803e+00  5.676e+00   0.494   0.6225
rad          2.771e-02  1.529e-02   1.812   0.0729 .
temp        -3.018e-02  1.178e-01  -0.256   0.7983
wind        -9.812e-02  3.211e-01  -0.306   0.7605
t2           6.034e-04  6.559e-04   0.920   0.3598
w2           8.732e-03  4.021e-03   2.172   0.0322 *
r2          -1.489e-05  7.043e-06  -2.114   0.0370 *
wr          -2.001e-03  1.339e-03  -1.494   0.1382
tr          -2.507e-04  2.056e-04  -1.219   0.2256
tw          -1.985e-03  3.742e-03  -0.530   0.5971
wtr          2.535e-05  1.805e-05   1.404   0.1634
```

Multiple Regression Example

```
model7 <- lm(log(ozone)~rad+temp+wind+t2+w2+r2+wr+tr+tw+wtr)
summary(model7)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.803e+00  5.676e+00   0.494   0.6225
rad          2.771e-02  1.529e-02   1.812   0.0729 .
temp        -3.018e-02  1.178e-01  -0.256   0.7983
wind        -9.812e-02  3.211e-01  -0.306   0.7605
t2           6.034e-04  6.559e-04   0.920   0.3598
w2           8.732e-03  4.021e-03   2.172   0.0322 *
r2          -1.489e-05  7.043e-06  -2.114   0.0370 *
wr          -2.001e-03  1.339e-03  -1.494   0.1382
tr          -2.507e-04  2.056e-04  -1.219   0.2256
tw          -1.985e-03  3.742e-03  -0.530   0.5971
wtr          2.535e-05  1.805e-05   1.404   0.1634
```


Multiple Regression Example

```
model8 <- update(model7,~.-wtr)
summary(model8)
model9 <- update(model8,~.-tr)
summary(model9)
model10 <- update(model9,~.-tw)
summary(model10)
model11 <- update(model10,~.-t2)
summary(model11)
model12 <- update(model11,~.-wr)
summary(model12)
```

Multiple Regression Example

```
model12 <-update(model11,~.-wr)
summary(model12)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.724e-01  6.350e-01   1.216 0.226543
rad          7.466e-03  2.323e-03   3.215 0.001736 **
temp         4.193e-02  6.237e-03   6.723 9.52e-10 ***
wind        -2.211e-01  5.874e-02  -3.765 0.000275 ***
w2           7.390e-03  2.585e-03   2.859 0.005126 **
r2          -1.470e-05  6.734e-06  -2.183 0.031246 *

Residual standard error: 0.4851 on 105 degrees of freedom
Multiple R-squared: 0.7004,     Adjusted R-squared: 0.6861
F-statistic:  49.1 on 5 and 105 DF,  p-value: < 2.2e-16
```


Multiple Regression Example

Let's check the AIC of the models:

```
>AIC(model7, model8, model9, model10, model11, model2)
```

	df	AIC
model7	12	168.0206
model8	11	168.1871
model9	10	166.3021
model10	9	164.8488
model11	8	163.3559
model12	7	162.2318

Continually decreasing AIC, unlike AIC for models 1 to 6.

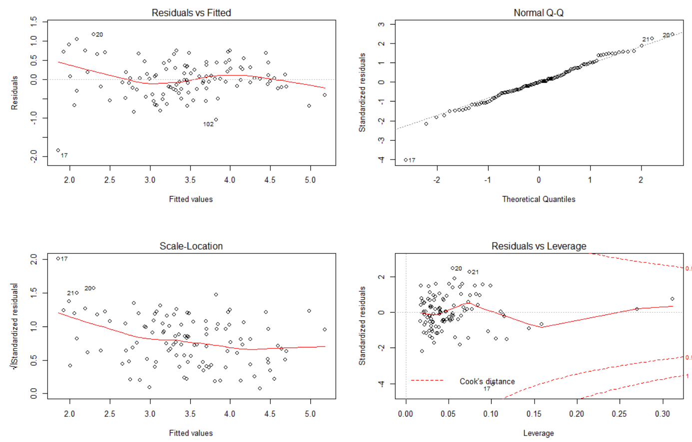
Multiple Regression Example

```
plot(model12)
```

This is the minimum adequate model.

It has five consequential parameters (the intercept of a multiple regression model is usually meaningless; it is the value of the response when every one of the explanatory variables is zero).

As predicted by our initial plots, none of the interactions survived the model simplification.



Common problems arising in multiple regression

The following are some of the problems and difficulties that crop up when we do multiple regression:

1. Differences in the measurement scales of the explanatory variables, leading to large variation in the sums of squares and hence to an ill-conditioned matrix;
2. Multicollinearity, in which there is a near-linear relation between two of the explanatory variables, leading to unstable parameter estimates;
3. Parameter proliferation where quadratic and interaction terms soak up more degrees of freedom than our data can afford;
4. Rounding errors during the fitting procedure;
5. Non-independence of groups of measurements;
6. Temporal or spatial correlation amongst the explanatory variables;
7. Pseudoreplication.

Homework: Multiple Regression

#1. Use the winequality.csv data set for this exercise.

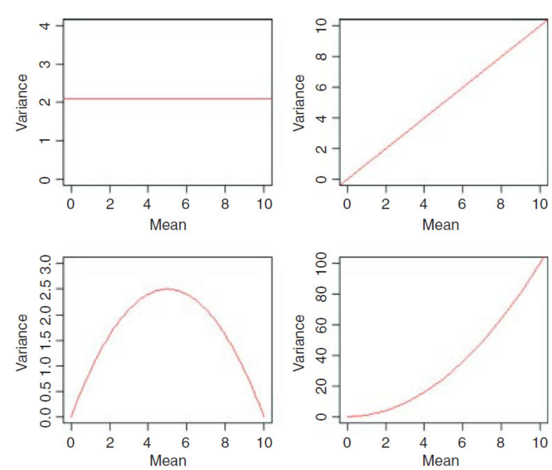
1. Assign the data set to the variable “quality”
2. Calculate the correlation matrix for the data for each wine type
3. Generate the pairs plot of the data for each wine type
4. Define the maximum linear model to include all possible interaction terms
5. Using the p-value approach and define the minimum adequate model eliminating in variable at a time
6. Create the vector of AIC values for each model.
7. Create the regression output for minimum adequate model
8. Create the ANOVA table for minimum adequate model
9. What do the results tell you about the relationship between wine quality and the other exploratory variables/
10. Create the 6 model plots. Describe what they mean.
11. Are assumptions of normality violated as evidenced by the residuals?
12. In your opinion is this a good model.

Generalized Linear Models (GLMs)

Definition: They are an extension of linear regression models that allow the dependent variable to be non-normal.

GLMs are powerful alternatives what two important assumptions of linear modeling have been violated:

1. The variance is not constant,
2. The errors are not normally distributed



Generalized Linear Models

Certain kinds of response variables invariably suffer from these two important contraventions of the standard assumptions, and GLMs are excellent at dealing with them.

Specifically, we might consider using GLMs when the response variable is:

1. Count data expressed as proportions (e.g. logistic regressions);
2. Count data that are not proportions (e.g. log-linear models of counts);
3. Binary response variables (e.g. dead or alive);
4. Data on time to death where the variance increases faster than linearly with the mean (e.g. time data with gamma errors).

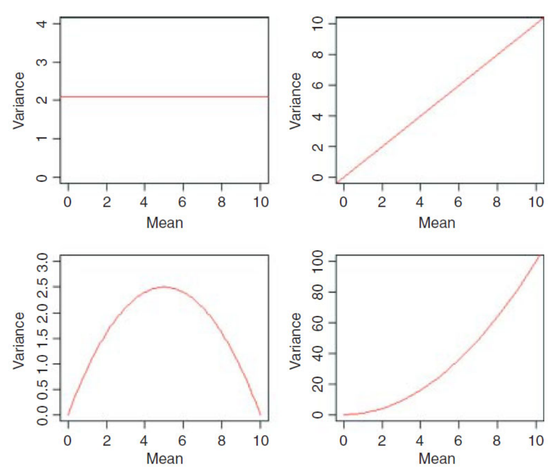
Generalized Linear Models

The central assumption that we have made up to this point is that variance was constant (top left-hand graph).

In count data, however, where the response variable is an integer and there are often lots of zeros in the dataframe, the variance may increase linearly with the mean (top tight).

With proportion data, where we have a count of the number of failures of an event as well as the number of successes, the variance will be an inverted U-shaped function of the mean (bottom left).

Where the response variable follows a gamma distribution (as in time-to-death data) the variance increases faster than linearly with the mean (bottom right).



Generalized Linear Models

Many of the basic statistical methods such as regression and Student's *t* test assume that variance is constant, but in many applications this assumption is untenable. Hence the great utility of GLMs.

A GLM has three important properties:

1. the error structure;
2. the linear predictor;
3. the link function.

Generalized Linear Models: Error Structure

Up to this point, we have dealt with the statistical analysis of data with normal errors. In practice, however, many kinds of data have non-normal errors, for example:

1. errors that are strongly skewed;
2. errors that are kurtotic;
3. errors that are strictly bounded (as in proportions);
4. errors that cannot lead to negative fitted values (as in counts).

In the past, the only tools available to deal with these problems were transformation of the response variable or the adoption of non-parametric methods.

Generalized Linear Models: Error Structure

A GLM allows the specification of a variety of different error distributions:

1. Poisson errors, useful with count data;
2. Binomial errors, useful with data on proportions;
3. Gamma errors, useful with data showing a constant coefficient of variation;
4. Exponential errors, useful with data on time to death (survival analysis).

Generalized Linear Models: Error Structure

The **error structure** is defined by means of the `family` directive, used as part of the model formula.

Examples are

1. `glm(y ~ z, family = poisson)` which means that the response variable *y* has Poisson errors
2. `glm(y ~ z, family = binomial)` which means that the response is binary, and the model has binomial errors.

Generalized Linear Models: Link Functions

Family	Notation	Canonical link	Range of y
Gaussian	$N(\mu, \sigma^2)$	identity: μ	$(-\infty, +\infty)$
Poisson	$\text{Pois}(\mu)$	$\log_e(\mu)$	$0, 1, \ldots, \infty$
Negative-Binomial	$\text{NBin}(\mu, \theta)$	$\log_e(\mu)$	$0, 1, \ldots, \infty$
Binomial	$\text{Bin}(n, \mu)/n$	$\text{logit}(\mu)$	$\{0, 1, \ldots, n\}/n$
Gamma	$G(\mu, \nu)$	μ^{-1}	$(0, +\infty)$
Inverse-Gaussian	$IG(\mu, \nu)$	μ^2	$(0, +\infty)$

A link function that relates the expected value of the response to the linear predictors in the model.

The general form of the link function follows:

$g(\mu_i) = X_i'\beta$

Linear Models

model <- lm(growth~tannin)
summary(model)
anova(model)

```
Call:
lm(formula = growth ~ tannin)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.4556 -0.8889 -0.2389  0.9778  2.8944 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  11.7556     1.0408  11.295 9.54e-06 ***
tannin       -1.2167     0.2186  -5.565 0.000846 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.693 on 7 degrees of freedom
Multiple R-squared:  0.8157,	Adjusted R-squared:  0.7893 
F-statistic: 30.97 on 1 and 7 DF,  p-value: 0.0008461
```

Note: There are 9 observations in the dataset.

Generalized Linear Models (GLMs)

model<-glm(growth~tannin, family = gaussian)
summary(model)
anova(model)

```
Call:
glm(formula = growth ~ tannin, family = gaussian)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.4556  -0.8889  -0.2389   0.9778   2.8944  

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  11.7556     1.0408  11.295 9.54e-06 ***
tannin       -1.2167     0.2186  -5.565 0.000846 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 2.86746)

    Null deviance: 108.889  on 8  degrees of freedom
Residual deviance:  20.072  on 7  degrees of freedom
AIC: 38.76
```

Note: Null Deviance assumes the mean for all observations!

Linear Models

model <- lm(growth~tannin)
summary(model)
anova(model)

```
Analysis of Variance Table

Response: growth
          Df Sum Sq Mean Sq F value    Pr(>F)    
tannin     1 88.817  88.817  30.974 0.0008461 ***
Residuals  7 20.072   2.867                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Generalized Linear Models (GLMs)

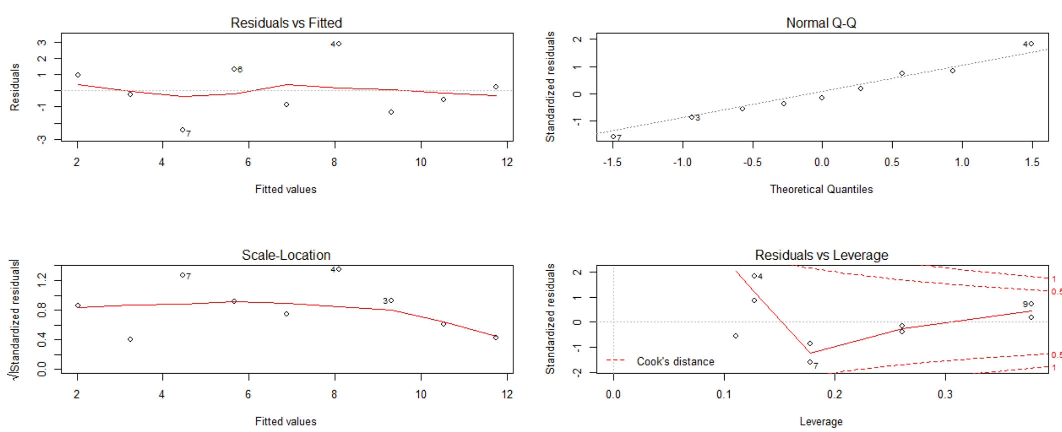
model<-glm(growth~tannin, family = gaussian(link = "identity"))
summary(model)
anova(model)

```
Analysis of Deviance Table

Model: gaussian, link: identity

Response: growth

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev
NULL                       8    108.889
tannin  1   88.817         7     20.072
```

Conclusion: lm() is a special case of glm() where the family is Gaussian & link function is the identity.

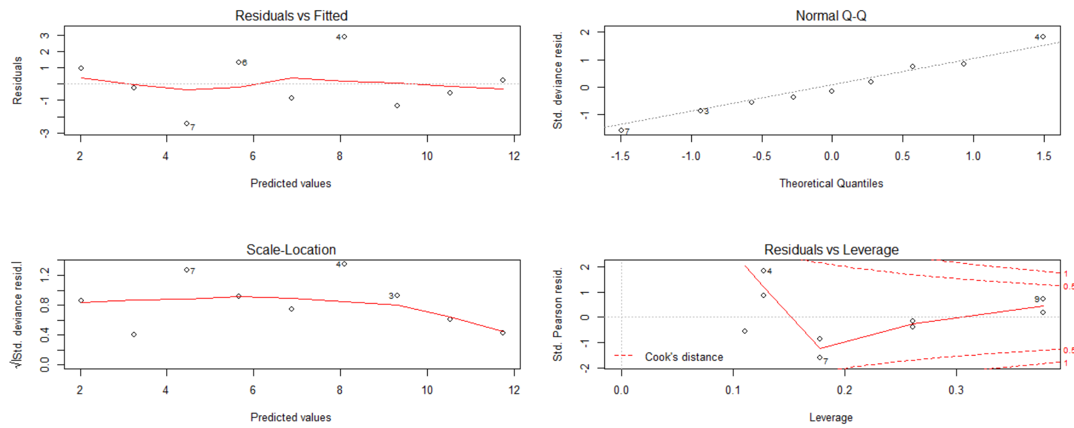
Linear Models

plot(model)



Generalized Linear Models (GLMs)

plot(model)



Generalized Linear Models: Example – Binomial Family

In the mtcars data set, the variable "vs" indicates if a car has a V engine or a straight engine.

We want to create a model that helps us to predict the probability of a vehicle having a V engine or a straight engine given a weight of 2100 lbs. and engine displacement of 180 cubic inches.

First, we fit the model:

We use the glm() function, include the variables in the usual way, and specify a binomial error distribution, as follows:

```
model <- glm(formula= vs ~ wt + disp, data=mtcars, family=binomial)
summary(model)
```

Generalized Linear Models: Example

```
Call:
glm(formula = vs ~ wt + disp, family = binomial, data = mtcars)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.67506  -0.28444  -0.08401   0.57281   2.08234

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.60859    2.43903   0.660    0.510
wt           1.62635    1.49068   1.091    0.275
disp        -0.03443    0.01536  -2.241    0.025 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.86  on 31  degrees of freedom
Residual deviance: 21.40  on 29  degrees of freedom
AIC: 27.4

Number of Fisher Scoring iterations: 6
```

Notice the Deviance measures of fit.

Observations:

- *weight* influences *vs* positively but it is not statistically significant according to the p- value.
- *displacement* has a slightly negative effect.

Generalized Linear Models: Example

We want to calculate a predicted probability of a V engine, for specific values of the predictors: a weight of 2100 lbs. and engine displacement of 180 cubic inches.

```
newdata = data.frame(wt = 2.1, disp = 180)
predict(model, newdata, type="response")
 1
0.2361081
```

The predicted probability is 0.24.

Generalized Linear Models: Example

Deviance is a measure of goodness of fit of a generalized linear model. Or rather, it's a measure of badness of fit–higher numbers indicate worse fit.

R reports two forms of deviance – the null deviance and the residual deviance. The null deviance shows how well the response variable is predicted by a model that includes only the intercept (grand mean).

For our example, we have a value of 43.9 on 31 degrees of freedom. Including the independent variables (weight and displacement) decreased the deviance to 21.4 points on 29 degrees of freedom, a significant reduction in deviance.

The Residual Deviance has reduced by 22.46 with a loss of two degrees of freedom.

```
Call:
glm(formula = vs ~ wt + disp, family = binomial, data = mtcars)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.67506  -0.28444  -0.08401   0.57281   2.08234

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.60859    2.43903   0.660    0.510
wt           1.62635    1.49068   1.091    0.275
disp        -0.03443    0.01536  -2.241    0.025 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.86  on 31  degrees of freedom
Residual deviance: 21.40  on 29  degrees of freedom
AIC: 27.4

Number of Fisher Scoring iterations: 6
```


Topic 5: Linear Modeling

Generalized Linear Models: Example

Fisher Scoring

What about the Fisher scoring algorithm? Fisher's scoring algorithm is a derivative of Newton's method for solving maximum likelihood problems numerically.

For model1 we see that Fisher's Scoring Algorithm needed six iterations to perform the fit.

This doesn't really tell you a lot that you need to know, other than the fact that the model did indeed converge, and had no trouble doing it.


```
Call:
glm(formula = vs ~ wt + disp, family = binomial, data = mtcars)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.67506  -0.28444  -0.08401   0.57281   2.08234

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.60859    2.43903   0.660    0.510
wt           1.62635    1.49068   1.091    0.275
disp        -0.03443    0.01536  -2.241    0.025 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.86  on 31  degrees of freedom
Residual deviance: 21.40  on 29  degrees of freedom
AIC: 27.4

Number of Fisher Scoring iterations: 6
```

Generalized Linear Models: Example

Information Criteria

The Akaike Information Criterion (AIC) provides a method for assessing the quality of your model through comparison of related models.

It's based on the Deviance but penalizes you for making the model more complicated. Much like adjusted R-squared, its intent is to prevent you from including irrelevant predictors.

However, unlike adjusted R-squared, the number itself is not meaningful. If you have more than one similar candidate models (where all of the variables of the simpler model occur in the more complex models), then you should select the model that has the smallest AIC.

So it's useful for comparing models but isn't interpretable on its own.

```
Call:
glm(formula = vs ~ wt + disp, family = binomial, data = mtcars)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.67506  -0.28444  -0.08401   0.57281   2.08234

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.60859    2.43903   0.660    0.510
wt           1.62635    1.49068   1.091    0.275
disp        -0.03443    0.01536  -2.241    0.025 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.86  on 31  degrees of freedom
Residual deviance: 21.40  on 29  degrees of freedom
AIC: 27.4

Number of Fisher Scoring iterations: 6
```


Topic 5: Linear Modeling

Generalized Linear Models: Example

Information Criteria

The Akaike Information Criterion (AIC) provides a method for assessing the quality of your model through comparison of related models.

It's based on the Deviance but penalizes you for making the model more complicated. Much like adjusted R-squared, its intent is to prevent you from including irrelevant predictors.

However, unlike adjusted R-squared, the number itself is not meaningful. If you have more than one similar candidate models (where all of the variables of the simpler model occur in the more complex models), then you should select the model that has the smallest AIC.

So it's useful for comparing models but isn't interpretable on its own.

```
Call:
glm(formula = vs ~ wt + disp, family = binomial, data = mtcars)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.67506  -0.28444  -0.08401   0.57281   2.08234

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.60859    2.43903   0.660    0.510
wt           1.62635    1.49068   1.091    0.275
disp        -0.03443    0.01536  -2.241    0.025 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.86  on 31  degrees of freedom
Residual deviance: 21.40  on 29  degrees of freedom
AIC: 27.4

Number of Fisher Scoring iterations: 6
```

Generalized Linear Models: Example

Hosmer-Lemeshow Goodness of Fit

How well our model fits depends on the difference between the model and the observed data. One approach for binary data is to implement a Hosmer Lemeshow goodness of fit test.

To implement this test, first install the ResourceSelection package and load it

```
install.packages("ResourceSelection")
library(ResourceSelection)
```

The test is available through the hoslem.test() function.

Generalized Linear Models: Example

```
hoslem.test(mtcars$vs, fitted(model))

        Hosmer and Lemeshow goodness of fit (GOF) test

data:  mtcars$vs, fitted(model)
X-squared = 6.4717, df = 8, p-value = 0.5945
```

Our model appears to fit well because we have no significant difference between the model and the observed data (i.e. the p-value is above 0.05).

As with all measures of model fit, we'll use this as just one piece of information in deciding how well this model fits. It doesn't work well in very large or very small data sets, but is often useful, nonetheless.

Generalized Linear Models: Example Poisson Family

The Poisson distribution has only one parameter, here μ_i, which is also its expected value. The canonical link function for μ_i is the logarithm, which means I have to apply the exponential function to the linear model to get back to the original scale.

The model form is

$$y_i \sim \text{Poisson}(\mu_i)$$
$$\log(\mu_i) = \alpha + \beta x_i$$
$$\mathbb{E}[y_i] = \exp(\alpha + \beta x_i)$$

The Linear Model

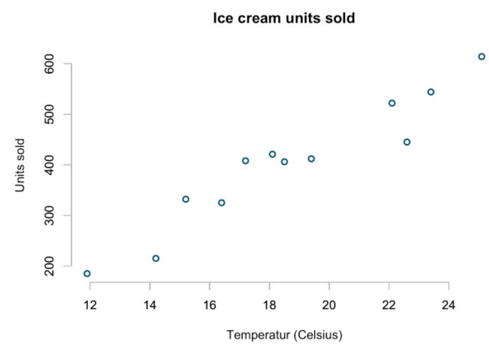
Generalized Linear Models: Example Poisson Family

```
library(arm) # for 'display' function only
icecream <- data.frame(
            temp=c(11.9, 14.2, 15.2, 16.4, 17.2, 18.1,
                        18.5, 19.4, 22.1, 22.6, 23.4, 25.1),
             units=c(185L, 215L, 332L, 325L, 408L, 421L,
                        406L, 412L, 522L, 445L, 544L, 614L)
                        )
```

Note: The “L” after each number explicitly makes the number an integer. This saves memory usage. Program would work fine without the “L”.

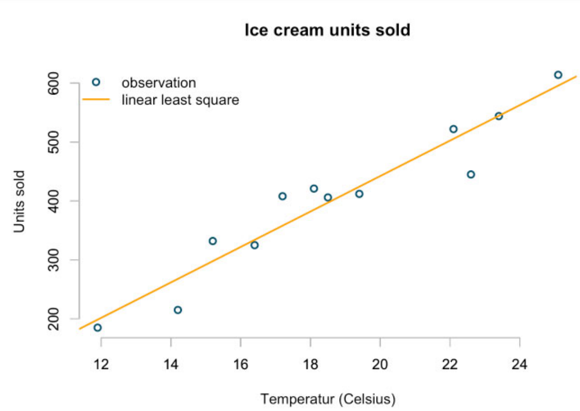
Generalized Linear Models: Example Poisson Family

```
basicPlot <- function(...){
 plot(units ~ temp, data=icecream, bty="n", lwd=2,
    main="Ice cream units sold", col="#00526D",
    xlab="Temperatur (Celsius)",
    ylab="Units sold", ...)
 axis(side = 1, col="grey")
 axis(side = 2, col="grey")
}
basicPlot()
```



Generalized Linear Models: Example Poisson Family

```
lsq.mod <- lsfit(icecream$temp, icecream$units)
abline(lsq.mod, col="orange", lwd=2)
legend(x="topleft", bty="n", lwd=c(2,2), lty=c(NA,1),
     legend=c("observation", "linear least square"),
      col=c("#00526D","orange"),  pch=c(1,NA))
```



Note:
The function lsfit() fits a least squares regression line to the data. It does the same thin as ~.

Generalized Linear Models: Example Poisson Family

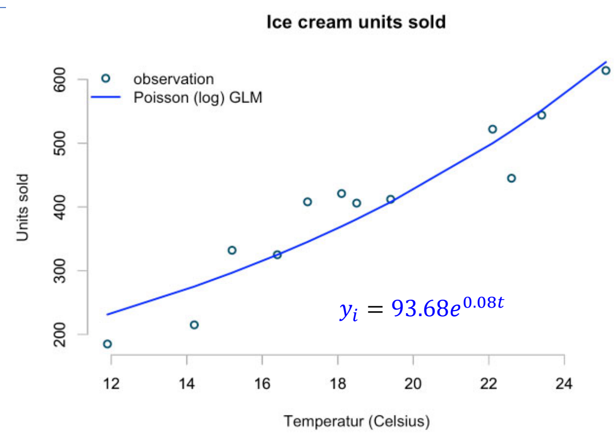
```
pois.mod <- glm(units ~ temp, data=icecream, family=poisson(link="log"))
display(pois.mod)

glm(formula = units ~ temp, family = poisson(link = "log"),
    data = icecream)
            coef.est coef.se
(Intercept) 4.54     0.08
temp        0.08     0.00
---
  n = 12, k = 2
  residual deviance = 60.0, null deviance = 460.1 (difference = 400.1)
```

This means $\propto = 4.54$ *and* $\beta = 0.08$. ⟶ The function is $y_i = e^{4.54}e^{0.08x_i} = 93.68e^{0.08t}$

Generalized Linear Models: Example Poisson Family

```
pois.pred <- predict(pois.mod, type="response")
basicPlot()
lines(icecream$temp, pois.pred, col="blue", lwd=2)
legend(x="topleft", bty="n", lwd=c(2,2), lty=c(NA,1),
     legend=c("observation", "Poisson (log) GLM"),
       col=c("#00526D","blue"), pch=c(1,NA))
```

The curve looks pretty good.



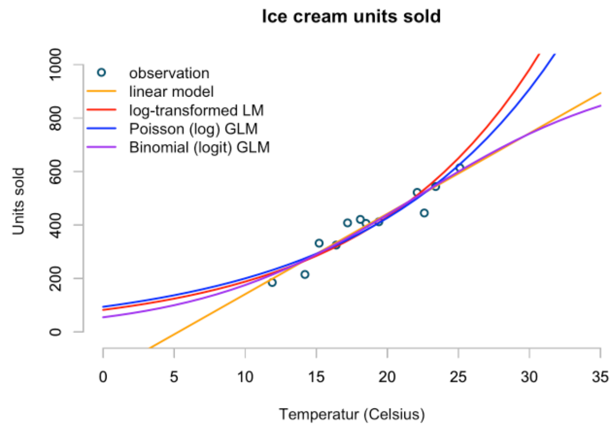
Generalized Linear Models: Example Other Fits

The chart shows the predictions of four models over a temperature range from 0 to 35ºC.

The linear model looks OK between 10 and perhaps 30ºC, it shows clearly its limitation.

The log-transformed linear and Poisson models appear to give similar predictions but will predict an ever-accelerating increase in sales as temperature rise. This makes sense as even the most ice cream loving person can only eat so much ice cream on a really hot day.

The Binomial model to does not seem to suffer from any of the above shortcomings.



Generalized Linear Models: Poisson Example with Deviance

Performing the deviance goodness of fit test in R

Lets now see how to perform the deviance goodness of fit test in R. First, we'll simulate some simple data, with a uniformly distributed covariate x, and Poisson outcome y:

```
set.seed(612312)

n <- 1000
x <- runif(n)
mean <- exp(x)
y <- rpois(n,mean)

mod <- glm(y~x, family=poisson)
summary(mod)
```

Generalized Linear Models: Poisson Example with Deviance

```
Call:
glm(formula = y ~ x, family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3218  -0.7627  -0.1826   0.5154   3.0562

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.01143    0.05485   0.208    0.835
x            1.00283    0.08566  11.708   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1206.9  on 999  degrees of freedom
Residual deviance: 1066.7  on 998  degrees of freedom
AIC: 3149.7

Number of Fisher Scoring iterations: 5
```

To deviance here is labelled as the 'residual deviance' by the glm function, and here is 1066.7. There are 1,000 observations, and our model has two parameters, so the degrees of freedom is 998, given by R as the residual df.

Generalized Linear Models: Poisson Example with Deviance

To calculate the p-value for the deviance goodness of fit test we simply calculate the probability to the right of the deviance value for the chi-squared distribution on 998 degrees of freedom:

```
pchisq(mod$deviance, df=mod$df.residual, lower.tail=FALSE)
[1] 0.0643842
```

The null hypothesis is that our model is correctly specified, and we cannot reject that hypothesis at $\alpha = 0.05$ level of significance.

This is a great model validation statistic for GLMs.

Linear predictor

The linear predictor, η (eta), is a linear sum of the effects of one or more explanatory variables, x_j,

$$\eta_i = \sum_{j=1}^{p} x_{ij}\beta_j$$

The right-hand side of the equation is called the **linear structure**.

To determine the fit of a given model, a GLM evaluates the linear predictor for each value of the response variable, then compares the predicted value with a *transformed* value of y. The transformation to be employed is specified in the link function. The fitted value is computed by applying the inverse of the link function, in order to get back to the original scale of measurement of the response variable.

Link function

One of the difficult things to grasp about GLMs is the relationship between the values of the response variable (as measured in the data and predicted by the model in fitted values) and the linear predictor.

The thing to remember is that the **link function** relates the mean value of y to its linear predictor. In symbols, this means that

$$\eta = g(\mu)$$

Canonical link functions

An important criterion in the choice of link function is to ensure that the fitted values stay within reasonable bounds.

We would want to ensure:

- Counts were all greater than or equal to 0 (negative count data would be nonsense). A log link is appropriate because the fitted values are antilogs of the linear predictor, and all antilogs are greater than or equal to 0.
- If the response variable was the proportion of individuals that died, then the fitted values would have to lie between 0 and 1 (fitted values greater than 1 or less than 0 would be meaningless). The logit link is appropriate because the fitted values are calculated as the antilogs of the log odds, log(p/q).

Family	Notation	Canonical link	Range of y
Gaussian	$N(\mu,\sigma^2)$	identity: μ	$(-\infty,+\infty)$
Poisson	Pois(μ)	$\log_e(\mu)$	$0,1,\ldots,\infty$
Negative-Binomial	NBin(μ,θ)	$\log_e(\mu)$	$0,1,\ldots,\infty$
Binomial	Bin$(n,\mu)/n$	logit(μ)	$\{0,1,\ldots,n\}/n$
Gamma	$G(\mu,\nu)$	μ^{-1}	$(0,+\infty)$
Inverse-Gaussian	$IG(\mu,\nu)$	μ^2	$(0,+\infty)$

The most appropriate link function is the one which produces the minimum residual deviance.

Canonical link functions

Choosing between using a link function (e.g. log link) and transforming the response variable (i.e. having log(y) as the response variable rather than y) takes a certain amount of experience.

The decision is usually based on *whether the variance is constant* on the original scale of measurement.

If the variance was constant, you would use a link function. If the variance increased with the mean, you would be more likely to log-transform the response.

Name	Link function $\eta = g(\mu)$	$\mu = g^{-1}(\eta)$
identity	μ	η
log	$\log \mu$	$\exp(\eta)$
logit	$\log(\mu/(1-\mu))$	$\exp(\eta)/(1+\exp(\eta))$
inverse	$1/\mu$	$1/\eta$
power	μ^k	$\eta^{1/k}$
sqrt	$\sqrt{\mu}$	η^2
probit	$\Phi^{-1}(\mu)$	$\Phi(\eta)$

Proportion data and Binomial Errors

Proportion data have three important properties that affect the way the data should be analyzed:

1. The data are strictly bounded.
2. The variance is non-constant.
3. Errors are non-normal.

Binomial Errors are Bounded

Proportion data and Binomial Errors: Assumption 1

Assumption 1: Data is Unbounded

- You cannot have a proportion greater than 1 or less than 0. This has obvious implications for the kinds of functions fitted and for the distributions of residuals around these fitted functions.
- For example, it makes no sense to have a linear model with a negative slope for proportion data because there would come a point, with high levels of the *x* variable, where negative proportions would be predicted.
- Likewise, it makes no sense to have a linear model with a positive slope for proportion data because there would come a point, with high levels of the *x* variable, where proportions greater than 1 would be predicted.

Proportion data and Binomial Errors: Assumption 2

Assumption 2: Constant Variance

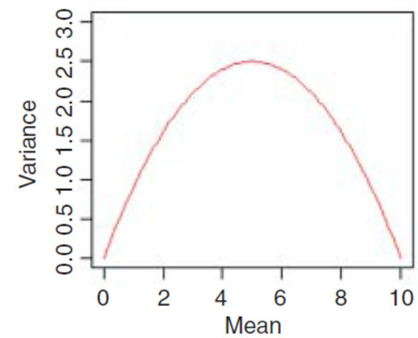
With proportion data, if the probability of success is 0, then there will be no successes in repeated trials, all the data will be zeros and hence the variance will be zero.

Likewise, if the probability of success is 1, then there will be as many successes as there are trials, and again the variance will be 0.

For proportion data, therefore, the variance increases with the mean up to a maximum (when the probability of success is 0.5) then declines again towards zero as the mean approaches 1.

The variance–mean relationship is humped, rather than constant as assumed in the classical tests.

Binomial Errors have Non-Constant Variance



Binomial Distribution:

Mean = pq
Variance = npq

Proportion data and Binomial Errors: Assumption 3

Binomial Errors are NOT Normally Distributed

Assumption 3: Errors are Normally Distributed

The final assumption is that the errors (the differences between the data and the fitted values estimated by the model) are normally distributed.

This cannot be so in proportional data because the data are bounded above and below: no matter how big a negative residual might be at high predicted values, $\hat{y}$, a positive residual cannot be bigger than $1 - \hat{y}$.

Similarly, no matter how big a positive residual might be for low predicted values $\hat{y}$, a negative residual cannot be greater than $\hat{y}$ (because you cannot have negative proportions).

This means that confidence intervals must be asymmetric whenever $\hat{y}$ takes large values (close to 1) or small values (close to 0).

Proportion data and Poisson Errors

Count data have a number of properties that need to be considered during modelling:

1. Count data are bounded below (you cannot have counts less than zero).
2. Variance is not constant (variance increases with the mean).
3. Errors are not normally distributed.
4. The fact that the data are whole numbers (integers) affects the error distribution.

Overdispersion

If, having fitted the minimal adequate model, we discover that the residual deviance is greater than the residual degrees of freedom, then we have contravened an important assumption of the model.

This is called overdispersion, and we can correct for it by specifying `quasipoisson` errors like this:

```
glm(y~x,quasipoisson)
```

It is important to understand that Poisson errors are an assumption, not a fact. Many of the count data you encounter in practice will have variance–mean ratios greater than 1, and in these cases you will need to correct for overdispersion.

Deviance: Measuring the goodness of fit of a GLM

The measure of discrepancy in a GLM to assess the goodness of fit of the model to the data is called the **deviance**. Deviance is defined as –2 times the difference in log-likelihood between the current model and a saturated model (i.e. a model that fits the data perfectly).

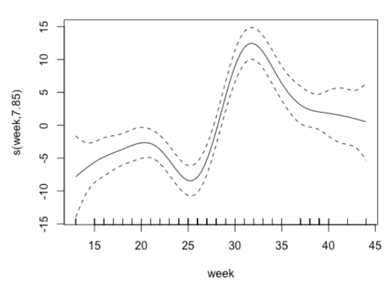
Because the latter does not depend on the parameters of the model, minimizing the deviance is the same as maximizing the likelihood.

Generalized Additive Models (GAMs)

Generalized additive models (GAMs) are like GLMs in that they can have different error structures and different link functions to deal with count data or proportion data.

What makes them different is that the shape of the relationship between *y* and a continuous variable *x* is not specified by some explicit functional form.

They work well with "wiggly" data.



Generalized Additive Models (GAMs)

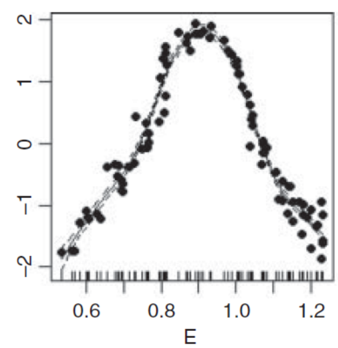
Instead, non-parametric smoothers are used to describe the relationship.

This is especially useful for relationships that exhibit complicated shapes, such as hump-shaped curves.

The model looks just like a GLM, except that the relationships we want to be smoothed are prefixed by `s`.

For example:

```
model <- gam(y~s(w)+s(x)+s(z),poisson)
```

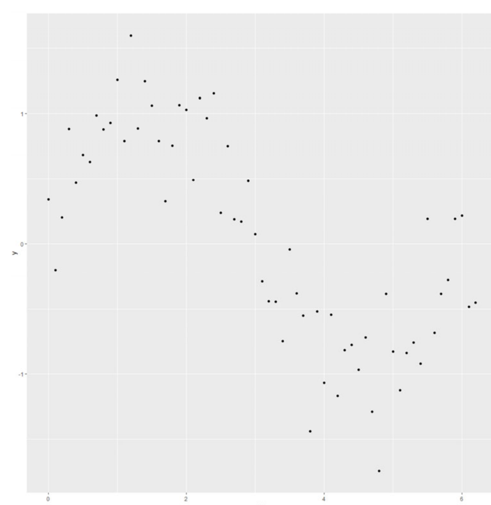


143

Generalized Additive Models (GAMs)

A simple example

```
x <- seq(0, pi * 2, 0.1)
sin_x <- sin(x)
y <- sin_x + rnorm(n = length(x), mean = 0, sd = sd(sin_x / 2))

Sample_data <- data.frame(y,x)
library(ggplot2)
ggplot(Sample_data, aes(x, y)) + geom_point()
```

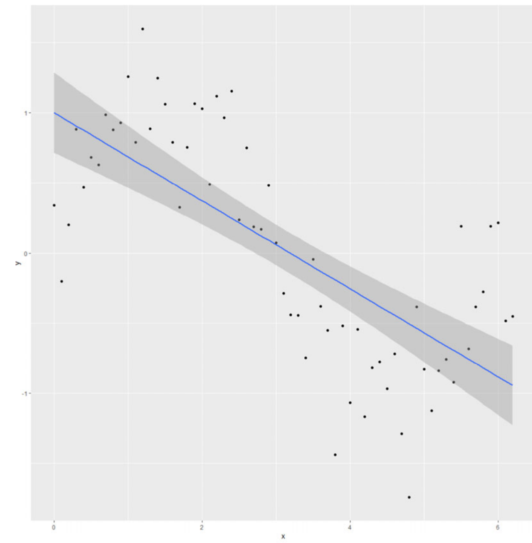


144

Generalized Additive Models (GAMs)

A simple example

lm_y <- lm(y ~ x, data = Sample_data)
ggplot(Sample_data, aes(x, y)) + geom_point() +
geom_smooth(method = lm)

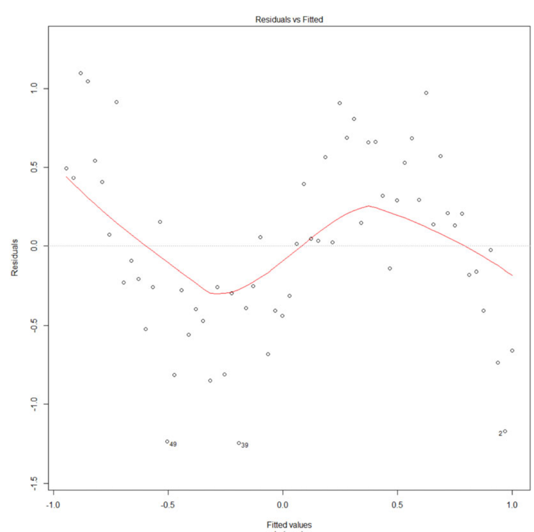


Generalized Additive Models (GAMs)

A simple example

plot(lm_y, which = 1)

Clearly, the residuals are not evenly spread across values of x, and we need to consider a better model.



Generalized Additive Models (GAMs)

A simple example

Before we consider a GAM, we need to load the package mgcv – *the* choice for running GAMs in R.

```
library(mgcv)
gam_y <- gam(y ~ s(x), method = "REML")
```

S stands for spline
REML stands for Residual Maximum Likelihood

To extract the fitted values, we can use predict just like normal:

```
x_new <- seq(0, max(x), length.out = 100)
y_pred <- predict(gam_y, data.frame(x = x_new))
```

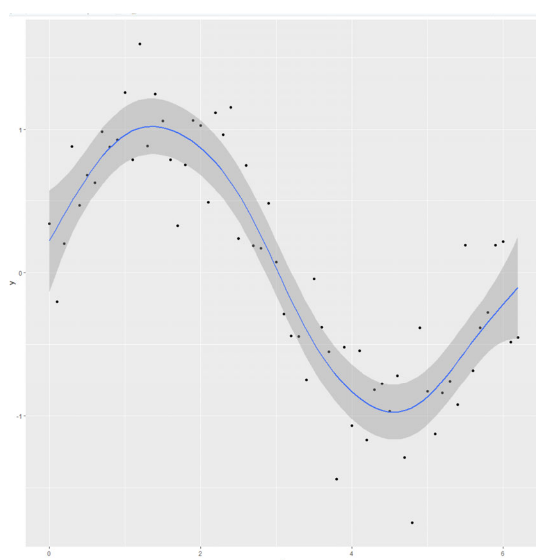
length.out = 100 will create 100 equally spaces numbers from 0 to max(x).

Generalized Additive Models (GAMs)

A simple example

```
ggplot(Sample_data, aes(x, y)) + geom_point() +
geom_smooth(method = "gam", formula = y ~s(x))
```

You can see the model is better fit to the data, but always check the diagnostics.



Generalized Additive Models (GAMs)

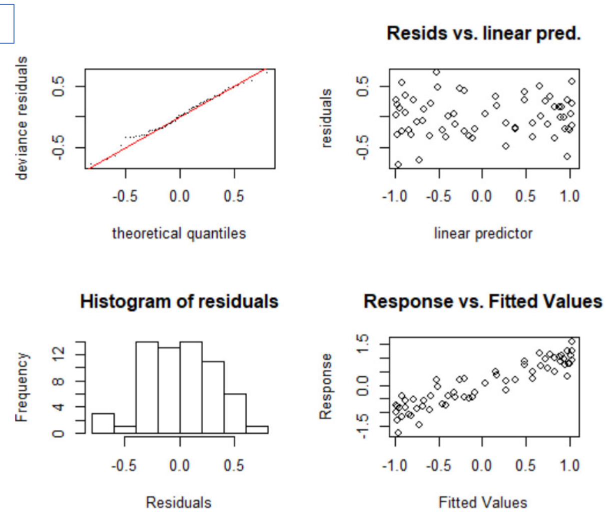
A simple example

```
par(mfrow = c(2,2))
gam.check(gam_y)
```

Method: REML   Optimizer: outer newton
full convergence after 6 iterations.
Gradient range [-1.791982e-09,1.669548e-09]
(score 30.97004 & scale 0.1117725).
Hessian positive definite, eigenvalue range [1.862708,30.71771].
Model rank = 10 / 10

Basis dimension (k) checking results. Low p-value (k-index<1) may indicate that k is too low, especially if edf is close to k'.

```
      k'  edf k-index p-value
s(x) 9.00 5.99    1.21    0.94
```



Generalized Additive Models (GAMs)

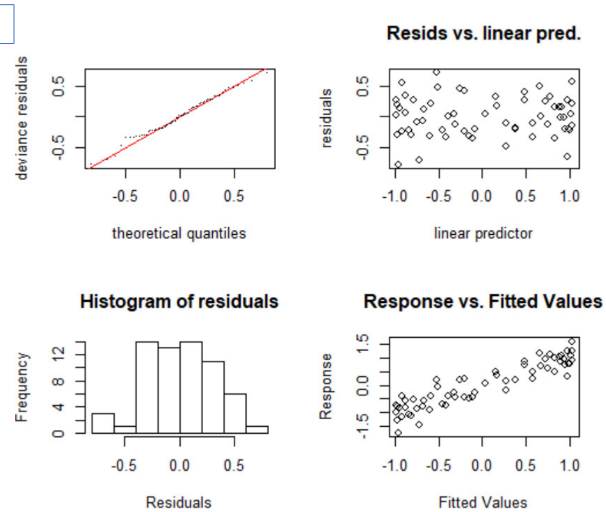
A simple example

Look for departures from normality in each plot.

The QQ plot looks pretty good as does the residuals vs. the linear predictor or fitted values.

The response v. the fitted values show fairly equal variance throughout.

The histogram plot departs from normality.



Homework

Use the trees.txt dataset to build a GAM model as follows:

1. Load library mgcv. The trees dataset in in this library, so you just need to reference it.
2. Volume as a function of Girth and Height
3. Store the results in the variable ct1
4. Print the results
5. Plot the residuals using plot with argument residuals=TRUE
6. Run gam.check
7. Plot fitted ct1 against residuals
8. Plot height against residuals
9. Create a summary of the results
10. Create the anova table
11. Interpret the results

Overdispersion

Overdispersion describes the observation that variation is higher than would be expected.

Some distributions do not have a parameter to fit variability of the observation. For example, the *normal distribution* does that through the parameter σ (i.e. the standard deviation of the model), which is constant in a typical regression.

In contrast, the *Poisson distribution* has no such parameter, and in fact the variance increases with the mean (i.e. the variance and the mean have the same value). In this latter case, for an expected value of $\lambda = 5$, we also expect that the variance of observed data points is $\lambda = 5$.

But what if it is not? What if the observed variance is much higher, i.e. if the data are overdispersed?

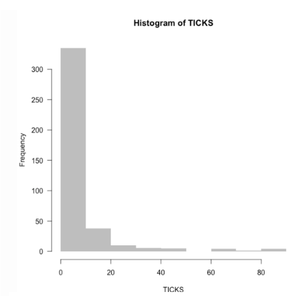
Overdispersion

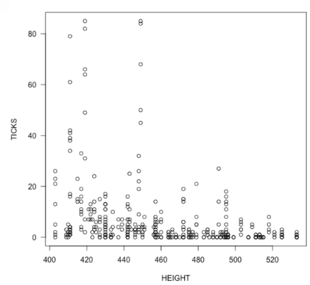
It turns out that the expected residual deviance should equal the degrees of freedom for the Poisson and Binomial distributions for large λ and np.

This means a test for overdispersion is the ratio of residual deviance to degrees of freedom. If the ratio is greater than 1, then there is overdispersion and underdispersion if less than 1.

Overdispersion Example

```
library(lme4)
data(grouseticks)
summary(grouseticks)
head(grouseticks)
attach(grouseticks)
hist(TICKS, col="grey", border=NA, las=1, breaks=0:90)
plot(TICKS ~ HEIGHT, las=1)
summary(fmp <- glm(TICKS ~ HEIGHT*YEAR, family=poisson))
```





Overdispersion Example

$$Dispersion = \frac{3009}{397} = 7.58$$

As you can see this result is much greater than 1, therefore the data is overdispersed, making model estimates unreliable.

```
Call:
glm(formula = TICKS ~ HEIGHT * YEAR, family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-6.0993  -1.7956  -0.8414   0.6453  14.1356

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    27.454732   1.084156   25.32   <2e-16 ***
HEIGHT         -0.058198   0.002539  -22.92   <2e-16 ***
YEAR96        -18.994362   1.140285  -16.66   <2e-16 ***
YEAR97        -19.247450   1.565774  -12.29   <2e-16 ***
HEIGHT:YEAR96   0.044693   0.002662   16.79   <2e-16 ***
HEIGHT:YEAR97   0.040453   0.003590   11.27   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 5847.5  on 402  degrees of freedom
Residual deviance: 3009.0  on 397  degrees of freedom
AIC: 3952

Number of Fisher Scoring iterations: 6
```


Topic 5: Linear Modeling

Homework

Given the results below, what would you conclude regarding overdispersion.

```
Call:
glm.nb(formula = TICKS ~ YEAR * HEIGHT, data = grouseticks, init.theta = 0.9000852793,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3765  -1.0281  -0.5052   0.2408   3.2440

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)   20.030124   1.827525  10.960  < 2e-16 ***
YEAR96       -10.820259   2.188634  -4.944 7.66e-07 ***
YEAR97       -10.599427   2.527652  -4.193 2.75e-05 ***
HEIGHT        -0.041308   0.004033 -10.242  < 2e-16 ***
YEAR96:HEIGHT  0.026132   0.004824   5.418 6.04e-08 ***
YEAR97:HEIGHT  0.020861   0.005571   3.745 0.000181 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(0.9001) family taken to be 1)

    Null deviance: 840.71  on 402  degrees of freedom
Residual deviance: 418.82  on 397  degrees of freedom
AIC: 1912.6

Number of Fisher Scoring iterations: 1
```

Homework

1. Load the car library
2. Create the model: oyster_reg_mod<-lm(Final ~ Initial)
3. Create the anova table
4. Print the summary
5. Interpret the results
6. Create a model for each treatment level
7. Create the anova table
8. Print the summary
9. Interpret the results
10. Create the model: oyster_reg_mod<-lm(Final ~ Trtmt + Initial)
11. Create the anova table
12. Print the summary
13. Interpret the results
14. What is the minimum adequate model? Support your results with p-values

Homework

Fill in the missing values of the Anova Table below

Complete the entries in the ANOVA table below (18 points)

There are 15 observation in the dataset underlying the ANOVA table for a regression.

Source	Degrees of Freedom	Sum of Squares	Mean Square	F- Ratio
Treartment	2	2510		
Error			13	
Total				