CH11: Linear Regression and Correlation

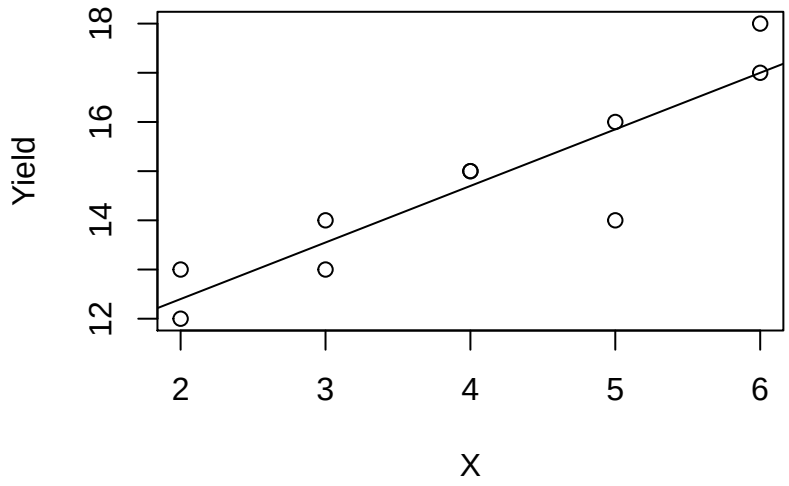
1 Corn Example: Simple Linear Regression

(Simple linear) regression is used to model the linear relationship between a numerical response variable and a single numerical predictor. In this example, corn yield is the response and fertilizer (X) is the predictor.

```
Corn <- read.csv("C:/hess/STAT511_FA11/RData/CH10_Corn.csv")
str(Corn)
```

```
## 'data.frame':    10 obs. of  2 variables:
##  $ Yield: int  12 13 13 14 15 15 14 16 17 18
##  $ X    : int  2 2 3 3 4 4 5 5 6 6
```

```
#Scatterplot
plot(Yield ~ X, data = Corn)
#Overlay fitted regression line
abline(lm(Yield ~ X, data = Corn))
```



1.1 Regression

```
Fit <- lm(Yield ~ X, data = Corn)
summary(Fit)
```

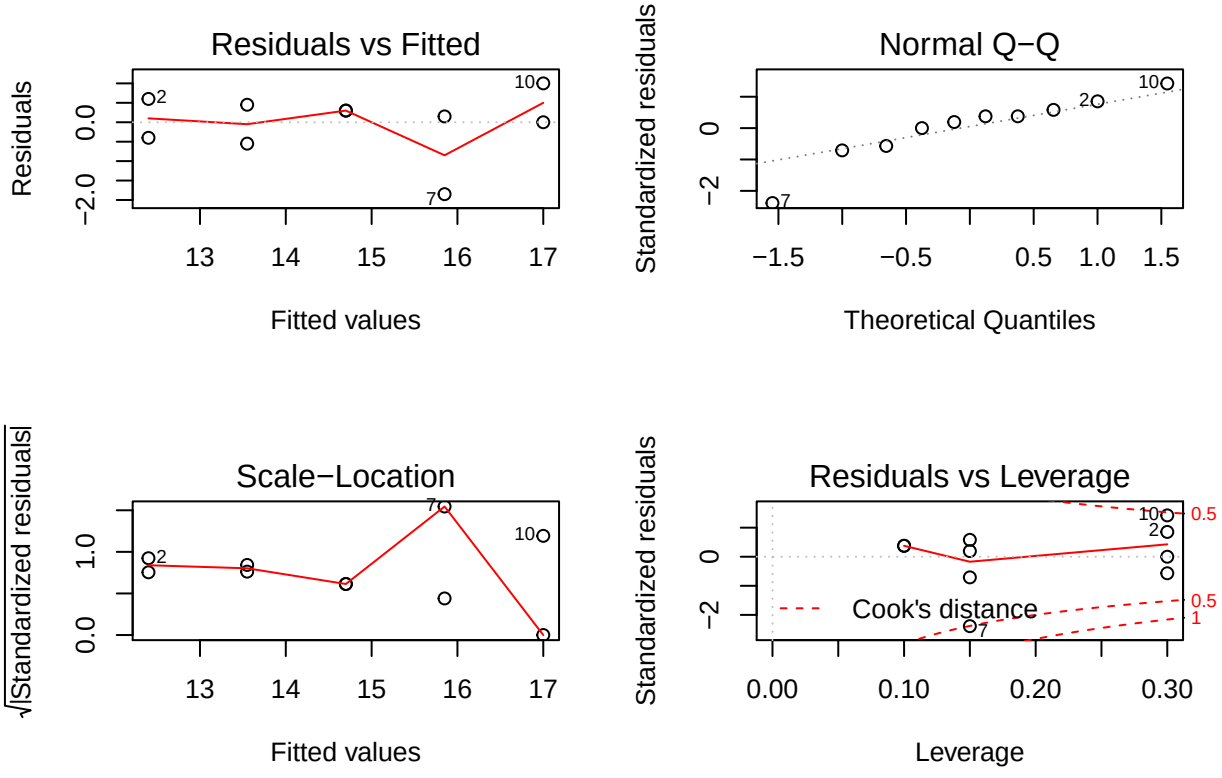
```
##
## Call:
## lm(formula = Yield ~ X, data = Corn)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -1.8500 -0.3000  0.2250  0.4125  1.0000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.1000     0.7973   12.67 1.42e-06 ***
## X             1.1500     0.1879    6.12 0.000283 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8404 on 8 degrees of freedom
## Multiple R-squared:  0.824,  Adjusted R-squared:  0.802
## F-statistic: 37.45 on 1 and 8 DF,  p-value: 0.0002832
```

```
Fit
```

```
##
## Call:
## lm(formula = Yield ~ X, data = Corn)
##
## Coefficients:
## (Intercept)            X
##       10.10         1.15
```

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



1.2 Additional Details and Predicted Values

```
#Confidence Intervals
confint(Fit, level = 0.95)
```

```
##                 2.5 %    97.5 %
## (Intercept) 8.2615130 11.938487
## X           0.7166645  1.583336
```

```
#ANOVA
anova(Fit)
```

```
## Analysis of Variance Table
##
## Response: Yield
##           Df Sum Sq Mean Sq F value    Pr(>F)
## X          1  26.45 26.4500  37.451 0.0002832 ***
## Residuals  8   5.65  0.7062
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Model Predictions
newdata <- data.frame(X = 5.5)
predict(Fit, newdata, interval = "predict",  level = 0.90)
```

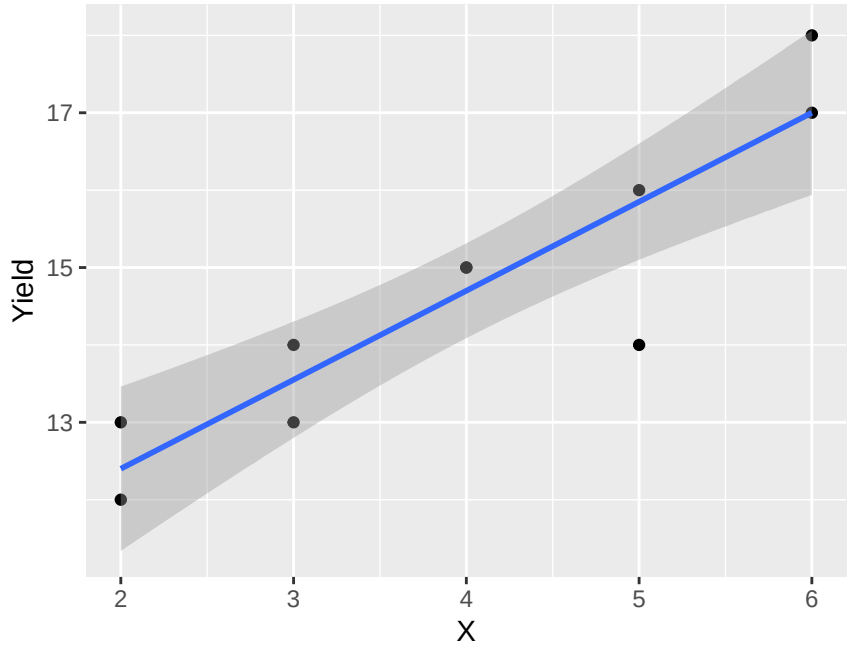
```
##      fit      lwr      upr
## 1 16.425 14.70421 18.14579
```

```
predict(Fit, newdata, interval = "confidence", level = 0.90)
```

```
##      fit      lwr      upr
## 1 16.425 15.70461 17.14539
```

1.3 tidyverse

```
library(tidyverse)
library(broom)
qplot(X, Yield, data = Corn) + geom_smooth(method = lm)
```



As we have seen before, tidy() from the broom package can be used to create "tidy" output.

glance() computes per-model statistics, such as R^2 and AIC.

The augment() method adds fitted values and residuals to the original data.

```
tidy(Fit)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic    p.value
##   <chr>          <dbl>     <dbl>     <dbl>      <dbl>
## 1 (Intercept)    10.1      0.797     12.7  0.00000142
## 2 X               1.15     0.188      6.12 0.000283
```

```
glance(Fit)
```

```
## # A tibble: 1 x 11
##   r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <int>  <dbl> <dbl> <dbl>
## 1     0.824         0.802 0.840      37.5 2.83e-4     2  -11.3  28.7  29.6
## # ... with 2 more variables: deviance <dbl>, df.residual <int>
```

```
head(augment(Fit))
```

```
## # A tibble: 6 x 9
##   Yield     X .fitted .se.fit .resid  .hat .sigma .cooksd .std.resid
##   <int> <int>   <dbl>   <dbl>  <dbl> <dbl>  <dbl>   <dbl>      <dbl>
## 1    12     2    12.4   0.460 -0.400 0.300  0.880 0.0694      -0.569
## 2    13     2    12.4   0.460  0.600 0.3    0.857 0.156        0.853
## 3    13     3    13.6   0.325 -0.55  0.15   0.870 0.0445      -0.710
## 4    14     3    13.6   0.325  0.450 0.15   0.879 0.0298       0.581
## 5    15     4    14.7   0.266  0.300 0.1    0.890 0.00787      0.376
## 6    15     4    14.7   0.266  0.300 0.1    0.890 0.00787      0.376
```

```
rm(Corn, Fit, newdata)
```

2 Intercept Example

In R, we can force the regression line to go through the origin (in other words, force the intercept to be equal to zero) using the -1 option in lm().

In this example, we compare results of the standard regression (allowing the intercept to be estimated) versus regression through the origin. We also look at the effect of a linear scale change for the predictor variable. In this case, we consider temperature on the C and F scales.

```
TempResp <-read.csv("C:/hess/STAT511_FA11/RData/CH10_Intercept.csv")
str(TempResp)
```

```
## 'data.frame':    9 obs. of  2 variables:
##  $ TempF: int  54 56 52 53 53 57 57 56 55
##  $ Resp : num  18 18.1 17.8 17.5 18.2 19.4 18.3 19.1 18.1
```

2.1 Models on F Scale

```
#Standard model with estimated intercept
modelF <- lm(Resp ~ TempF, data = TempResp)
summary(modelF)
```

```
##
## Call:
## lm(formula = Resp ~ TempF, data = TempResp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49032 -0.36774 -0.09839  0.33226  0.60968
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.64355    4.75125   1.188   0.2736
## TempF        0.23065    0.08669   2.660   0.0324 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4551 on 7 degrees of freedom
## Multiple R-squared:  0.5028, Adjusted R-squared:  0.4317
## F-statistic: 7.078 on 1 and 7 DF,  p-value: 0.03244
```

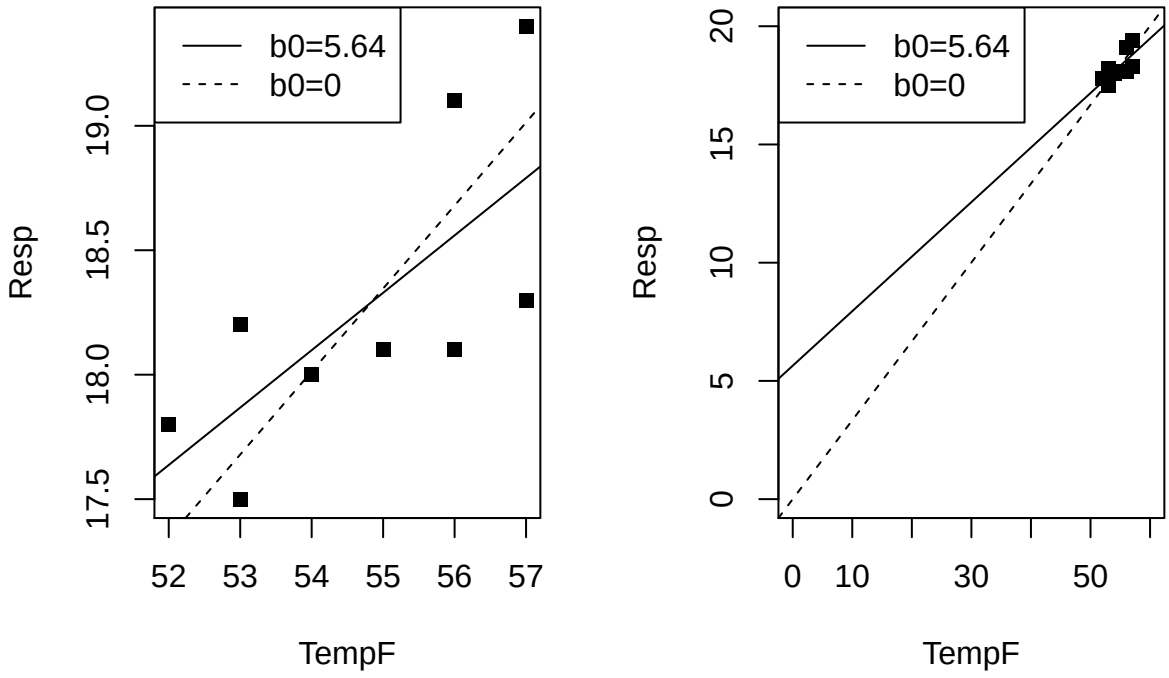
```
#Force intercept to be zero
modelF0 <- lm(Resp ~ TempF - 1, data = TempResp)
summary(modelF0)
```

```
##
## Call:
## lm(formula = Resp ~ TempF - 1, data = TempResp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.71328 -0.24615 -0.01258  0.42028  0.52098
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
```

```
## TempF 0.333566   0.002838   117.5 3.07e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4666 on 8 degrees of freedom
## Multiple R-squared:  0.9994, Adjusted R-squared:  0.9993
## F-statistic: 1.381e+04 on 1 and 8 DF,  p-value: 3.069e-14
```

2.2 Scatterplots with fitted lines overlaid

```
par(mfrow = c(1, 2))
#Plot1
plot(Resp ~ TempF, data = TempResp, pch = 15)
abline(coef(modelF), lty = 1)
abline(0, coef(modelF0), lty = 2)
legend("topleft", lty = c(1, 2), c("b0=5.64", "b0=0"))
#Plot2
plot(Resp ~ TempF, data = TempResp, pch = 15, xlim = c(0, 60), ylim = c(0, 20))
abline(coef(modelF), lty = 1)
abline(0, coef(modelF0), lty = 2)
legend("topleft", lty = c(1, 2), c("b0=5.64", "b0=0"))
```



2.3 Model on C Scale

```
#Add new column with temps in Celsius
TempResp$TempC <- (5/9)*(TempResp$TempF-32)
str(TempResp)
```

```
## 'data.frame':    9 obs. of  3 variables:
##  $ TempF: int  54 56 52 53 53 57 57 56 55
##  $ Resp : num  18 18.1 17.8 17.5 18.2 19.4 18.3 19.1 18.1
##  $ TempC: num  12.2 13.3 11.1 11.7 11.7 ...
```

```
modelC <- lm(Resp ~ TempC, data = TempResp)
summary(modelC)
```

```
##
## Call:
## lm(formula = Resp ~ TempC, data = TempResp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49032 -0.36774 -0.09839  0.33226  0.60968
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.0242     1.9805   6.576 0.000311 ***
## TempC         0.4152     0.1560   2.660 0.032444 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4551 on 7 degrees of freedom
## Multiple R-squared:  0.5028, Adjusted R-squared:  0.4317
## F-statistic: 7.078 on 1 and 7 DF,  p-value: 0.03244
```

2.4 Predictions on F and C scales

```
newF <- data.frame(TempF = 32)
predict(modelF, newF)
```

```
##        1
## 13.02419
```

```
newC <- data.frame(TempC = 0)
predict(modelC, newC)
```

```
##        1
## 13.02419
```

```
rm(TempResp, ModelF, ModelF0, ModelC)
```

3 Stopping Distance Example: Transformations

Transforming data can be used to satisfy model assumptions (linearity, equal variance and normality). In this example, we look at vehicle stopping distance versus speed (prior to braking).

```
Stop <-read.csv("C:/hess/STAT511_FA11/RData/CH11_StopDistance.csv")
str(Stop)
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ Speed: int  4 4 4 7 7 8 9 10 10 10 ...
##  $ Dist : int  2 10 16 4 22 16 10 18 26 34 ...
```

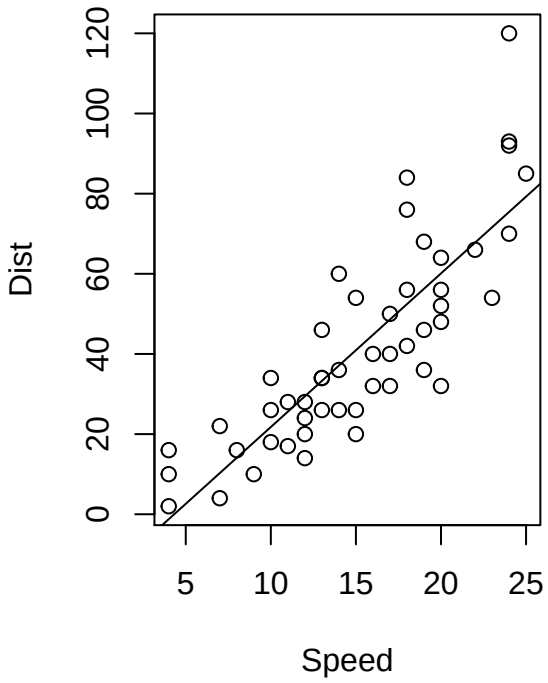
3.1 Model1 Dist ~ Speed

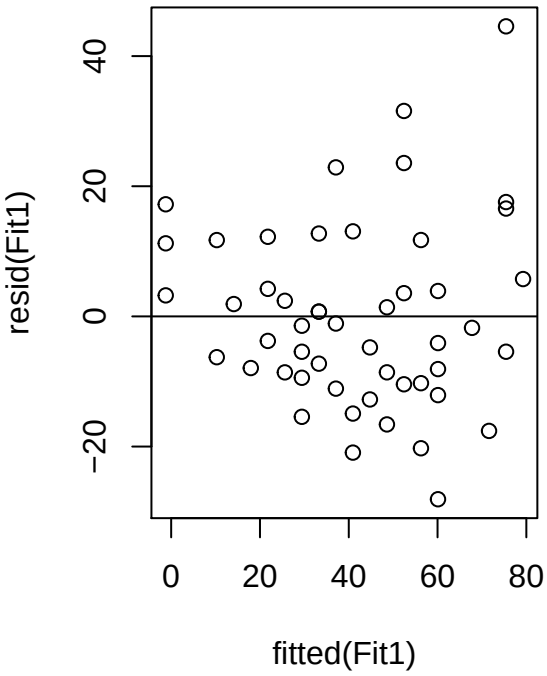
Note the "cornucopia" shape in the plot of resids vs fitted values. This indicates that regression assumptions are NOT satisfied.

```
Fit1 <- lm(Dist ~ Speed, data = Stop)
summary(Fit1)
```

```
##
## Call:
## lm(formula = Dist ~ Speed, data = Stop)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.098  -9.227  -1.599   9.856  44.571
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.5599     5.9790   -2.77  0.00795 **
## Speed         3.8329     0.3701   10.36 7.96e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.32 on 48 degrees of freedom
## Multiple R-squared:  0.6908, Adjusted R-squared:  0.6844
## F-statistic: 107.2 on 1 and 48 DF,  p-value: 7.961e-14
```

```
par(mfrow=c(1,2))
plot(Dist ~ Speed, data = Stop)
abline(coef(Fit1))
plot(resid(Fit1) ~ fitted(Fit1))
abline(h = 0)
```





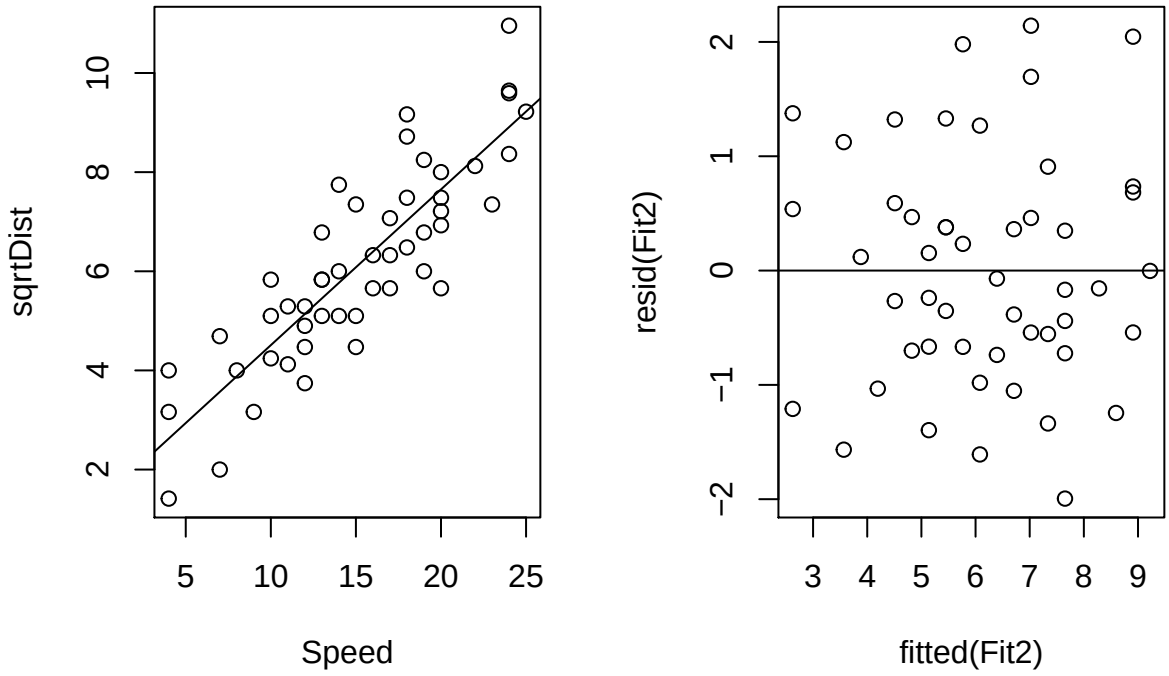
3.2 Model2 sqrt(Dist) ~ Speed

The plot of resids vs fitted for this model suggests that regression assumptions are now satisfied.

```
Stop$sqrtDist <- sqrt(Stop$Dist)
Fit2 <- lm(sqrtDist ~ Speed, data = Stop)
summary(Fit2)
```

```
##
## Call:
## lm(formula = sqrtDist ~ Speed, data = Stop)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9947 -0.6922 -0.1130  0.5767  2.1420
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.36730    0.42748   3.199  0.00245 **
## Speed        0.31421    0.02646  11.874 6.84e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.024 on 48 degrees of freedom
## Multiple R-squared:  0.746,  Adjusted R-squared:  0.7407
## F-statistic:   141 on 1 and 48 DF,  p-value: 6.841e-16
```

```r
par(mfrow=c(1,2))
plot(sqrtDist ~ Speed, data = Stop)
abline(coef(Fit2))
plot(resid(Fit2) ~ fitted(Fit2))
abline(h = 0)
```



3.3 Model3 Dist ~ Speed2

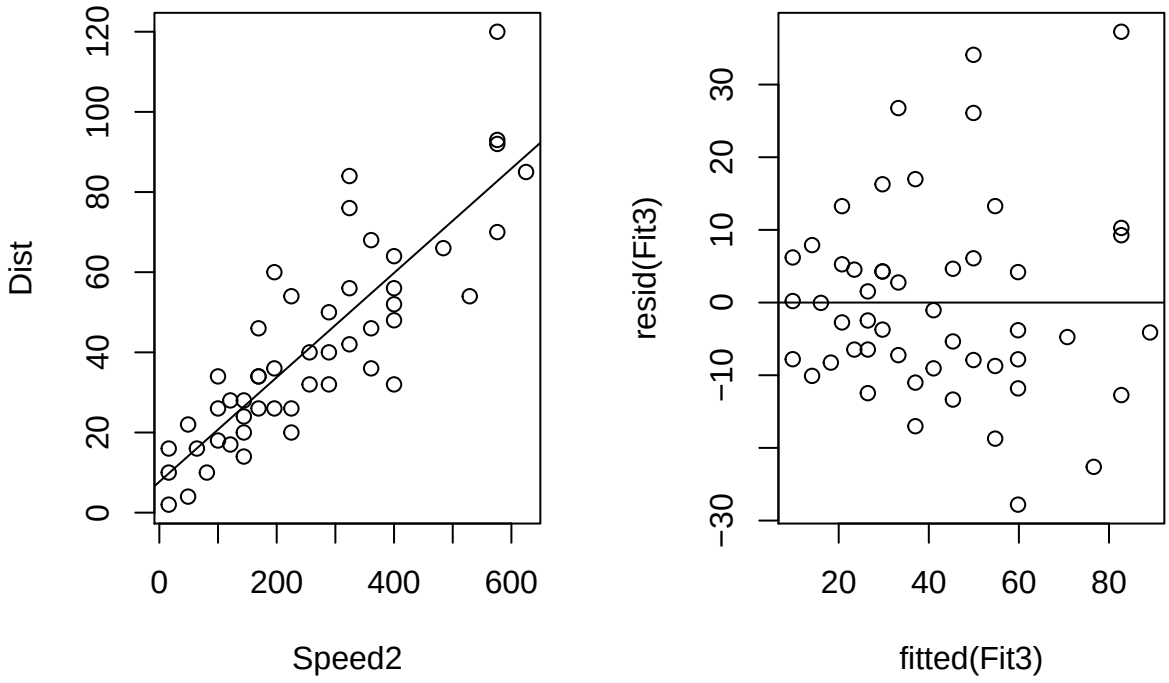
Note the "megaphone" shape in the plot of resids vs fitted values. This indicates that regression assumptions are NOT satisfied.

```r
Stop$Speed2 <- Stop$Speed*Stop$Speed
Fit3 <- lm(Dist ~ Speed2, data = Stop)
summary(Fit3)
```

```
##
## Call:
## lm(formula = Dist ~ Speed2, data = Stop)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.809  -8.173  -2.601   5.883  37.267
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  7.71108    3.57651   2.156   0.0361 *
## Speed2       0.13025    0.01159  11.241 4.79e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.51 on 48 degrees of freedom
## Multiple R-squared:  0.7247, Adjusted R-squared:  0.719
## F-statistic: 126.4 on 1 and 48 DF,  p-value: 4.792e-15
```

```r
par(mfrow=c(1,2))
plot(Dist ~ Speed2, data = Stop)
abline(coef(Fit3))
plot(resid(Fit3) ~ fitted(Fit3))
abline(h = 0)
```




```r
rm(Stop, Fit1, Fit2, Fit3)
```

4 Florida Election Example: Transformations and Outliers

In this example, we look at data from the 2000 presidential election (George W. Bush vs Gore). Specifically we look at vote counts by county. It was alleged that people in Palm Beach county "accidentally" voted for Buchanan (instead of Gore) due to the butterfly ballot design. The response for the model is #votes for Buchanan versus #votes for Bush. The idea is that since both of these candidates are conservative, there should be relationship between their popularity. Even after transformation, Palm Beach county appears to be an outlier (with more votes for Buchanan than predicted based on the model).

```
#In original file, County names are in first column.
#Using row.names will help identify states in the diagnostic plots below.
FL2000 <- read.csv("C:/hess/STAT511_FA11/RData/CH11_FL2000.csv",
                   row.names = 1)
str(FL2000)
```

```
## 'data.frame':    67 obs. of  4 variables:
##  $ Bush    : int  34124 5610 38637 5414 115185 177323 2873 35426 29765 41736 ...
##  $ Gore    : int  47365 2392 18850 3075 97318 386561 2155 29645 25525 14632 ...
##  $ Nader   : int  3226 53 828 84 4470 7101 39 1462 1379 562 ...
##  $ Buchanan: int  263 73 248 65 570 788 90 182 270 186 ...
```

```
head(FL2000)
```

```
##            Bush   Gore Nader Buchanan
## Alachua   34124  47365  3226      263
## Baker      5610   2392    53       73
## Bay       38637  18850   828      248
## Bradford   5414   3075    84       65
## Brevard  115185  97318  4470      570
## Broward  177323 386561  7101      788
```

4.1 Model1 Original Scale

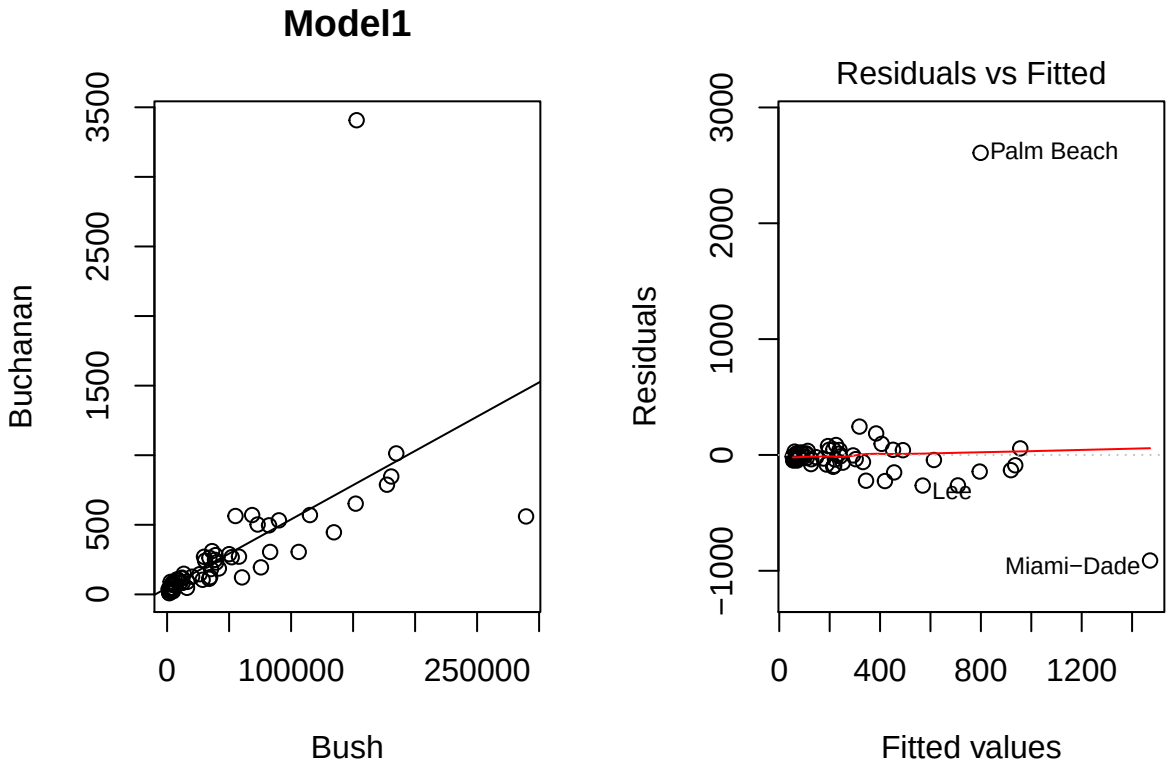
Not shown, but the identify() function can be used to interactively identify points on the scatter plot.

```
Fit1 <- lm(Buchanan ~ Bush, data = FL2000)
summary(Fit1)
```

```
##
## Call:
## lm(formula = Buchanan ~ Bush, data = FL2000)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -911.30  -46.11  -26.05   12.01 2608.01
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.697e+01  5.446e+01   0.863    0.392
## Bush        4.920e-03  7.622e-04   6.455 1.57e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 353.9 on 65 degrees of freedom
## Multiple R-squared:  0.3906, Adjusted R-squared:  0.3813
```

```
## F-statistic: 41.67 on 1 and 65 DF,  p-value: 1.574e-08
```

```
par(mfrow = c(1, 2))
plot(Buchanan ~ Bush, data = FL2000, main = "Model1")
abline(coef(Fit1))
plot(Fit1, which = 1)
```



4.2 Model2 Log Scale

```
FL2000$logBush <- log(FL2000$Bush)
FL2000$logBuch <- log(FL2000$Buchanan)
str(FL2000)
```

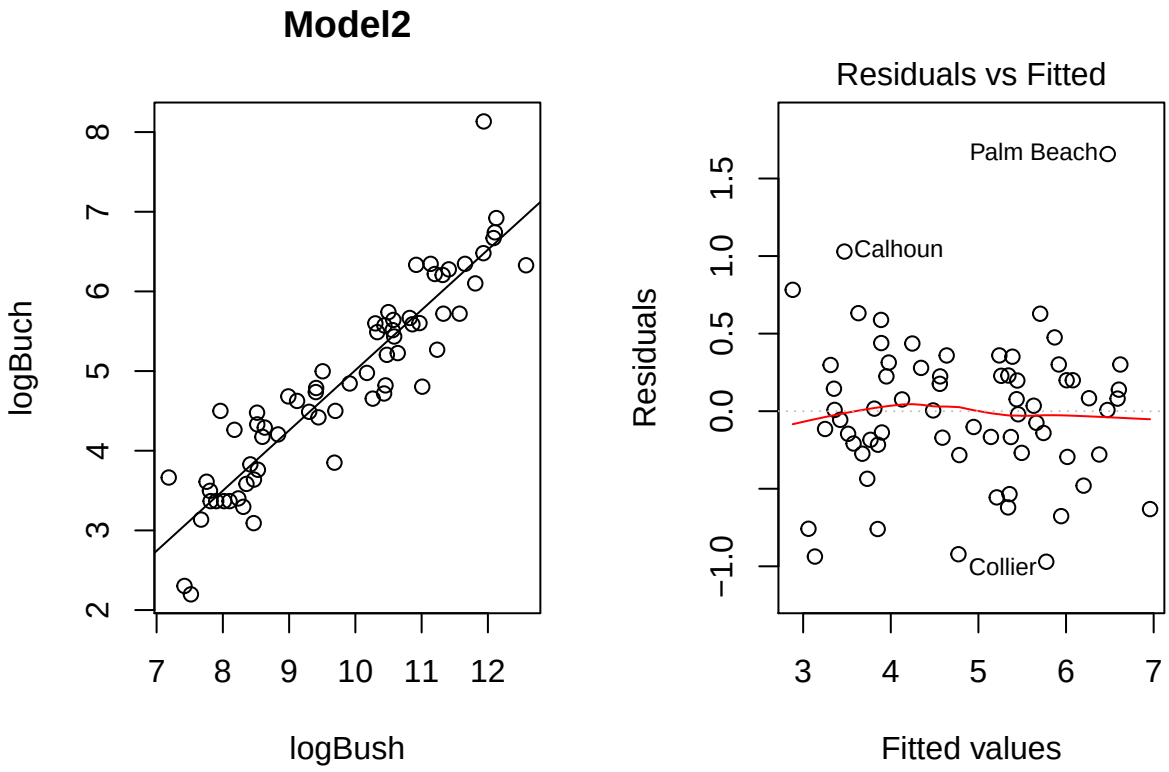
```
## 'data.frame':    67 obs. of  6 variables:
##  $ Bush    : int  34124 5610 38637 5414 115185 177323 2873 35426 29765 41736 ...
##  $ Gore    : int  47365 2392 18850 3075 97318 386561 2155 29645 25525 14632 ...
##  $ Nader   : int  3226 53 828 84 4470 7101 39 1462 1379 562 ...
##  $ Buchanan: int  263 73 248 65 570 788 90 182 270 186 ...
##  $ logBush : num  10.44 8.63 10.56 8.6 11.65 ...
##  $ logBuch : num  5.57 4.29 5.51 4.17 6.35 ...
```

```
Fit2 <- lm(logBuch ~ logBush, data = FL2000)
summary(Fit2)
```

```
##
## Call:
```

```
## lm(formula = logBuch ~ logBush, data = FL2000)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97038 -0.24247  0.00825  0.25452  1.65752
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.55079    0.38903  -6.557 1.04e-08 ***
## logBush      0.75620    0.03934  19.222  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4672 on 65 degrees of freedom
## Multiple R-squared:  0.8504, Adjusted R-squared:  0.8481
## F-statistic: 369.5 on 1 and 65 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(1,2))
plot(logBuch ~ logBush, data = FL2000, main = "Model2")
abline(coef(Fit2))
plot(Fit2, which = 1)
```



4.3 Outlier Test

The outlierTest() function from the car package will return "raw" and Bonferoni adjusted 2-sided p-value for the most extreme residual

```
library(car)
outlierTest(Fit2)
```

```
##            rstudent unadjusted p-value Bonferroni p
## Palm Beach 4.041946         0.00014474    0.0096977
```

Or we can calculate the outlier test "by hand".

```
FL2000 <- data.frame(FL2000, Fit = fitted(Fit2),
                     Resid = resid(Fit2), StdResid = rstandard(Fit2),
                     RStudent = rstudent(Fit2))
subset(FL2000, Resid > 1.5)
```

```
##              Bush   Gore Nader Buchanan  logBush  logBuch      Fit
## Palm Beach 152846 268945  5564     3407 11.93719 8.133587 6.476071
##               Resid StdResid RStudent
## Palm Beach 1.657517 3.635704 4.041946
```

```
2*67*(1-pt(4.0419, 64))
```

```
## [1] 0.009699211
```

```
rm(FL2000, Fit1, Fit2)
```

5 Corn Example: Lack of Fit Test

```
Corn <- read.csv("C:/hess/STAT511_FA11/RData/CH10_Corn.csv")
str(Corn)
```

```
## 'data.frame':    10 obs. of  2 variables:
##  $ Yield: int  12 13 13 14 15 15 14 16 17 18
##  $ X    : int  2 2 3 3 4 4 5 5 6 6
```

5.1 Regression

```
RegFit <- lm(Yield ~ X, data = Corn)
anova(RegFit)
```

```
## Analysis of Variance Table
##
## Response: Yield
##           Df Sum Sq Mean Sq F value    Pr(>F)
## X          1  26.45 26.4500  37.451 0.0002832 ***
## Residuals  8   5.65  0.7062
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.2 ANOVA

```
ANOVAFit <- lm(Yield ~ as.factor(X), data = Corn)
anova(ANOVAFit)
```

```
## Analysis of Variance Table
##
## Response: Yield
##              Df Sum Sq Mean Sq F value  Pr(>F)
## as.factor(X)  4   28.6    7.15  10.214 0.01267 *
## Residuals     5    3.5    0.70
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.3 Lack of Fit test

```
anova(RegFit, ANOVAFit)
```

```
## Analysis of Variance Table
##
## Model 1: Yield ~ X
## Model 2: Yield ~ as.factor(X)
##   Res.Df  RSS Df Sum of Sq      F Pr(>F)
## 1      8 5.65
## 2      5 3.50  3      2.15 1.0238 0.4564
```

6 Flow Rate: Calibration Example

This data is taken from the 6th Edition of Ott and Longnecker.

An engineer is interested in calibrating a flow meter to be used on a liquid soap production line. For the test, 10 different flow rates (X) are fixed and the corresponding meter readings (Y) are observed.

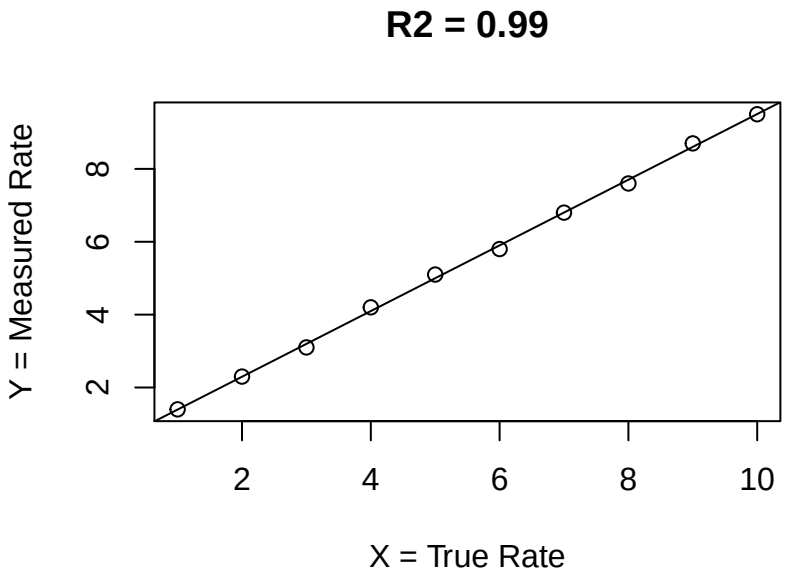
```
FlowRate <- read.csv("C:/hess/STAT511_FA11/RData/CH11_FlowRate.csv")
FlowRate
```

```
##     X   Y
## 1   1 1.4
## 2   2 2.3
## 3   3 3.1
## 4   4 4.2
## 5   5 5.1
## 6   6 5.8
## 7   7 6.8
## 8   8 7.6
## 9   9 8.7
## 10 10 9.5
```

```
Fit1 <- lm(Y ~ X , data = FlowRate)
summary(Fit1)
```

```
##
## Call:
## lm(formula = Y ~ X, data = FlowRate)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.103030 -0.074091  0.001212  0.073182  0.101818
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.493333   0.059101   8.347 3.21e-05 ***
## X           0.901212   0.009525  94.616 1.74e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08652 on 8 degrees of freedom
## Multiple R-squared:  0.9991, Adjusted R-squared:  0.999
## F-statistic:  8952 on 1 and 8 DF,  p-value: 1.738e-13
```

```
plot(Y ~ X, data = FlowRate, main = "R2 = 0.99",
     xlab = "X = True Rate", ylab = "Y = Measured Rate")
abline(Fit1)
```




```
Fit2 <- lm(X ~ Y, data = FlowRate)
coef(Fit2)
```

```
## (Intercept)           Y
##  -0.5420115   1.1086260
```

```
#Predictions using Y = 9 for both models
#Fit1
(9-coef(Fit1)[1])/coef(Fit1)[2]
```

```
## (Intercept)
##    9.439139
```

```
#Fit2
predict(Fit2, newdata = data.frame(Y=9))
```

```
##        1
## 9.435622
```

7 Fat States: Denver Post Article

Faithful states also the fattest, atheist concludes By Electra Draper (DP, 11/25/10)

Praise the Lord, and pass the potatoes.

It might take someone with a slightly jaundiced view of the very religious to notice that the states with the highest obesity rates are mostly the same states - nine out of 10 - professing the greatest religiosity.

Tim Covell, the Albuquerque author of a new book called "Born Atheist," correlated two fairly recent studies not usually correlated to conclude that "religion might make you fat" - except in Colorado. The CalorieLab 2010 obesity report, based on statistics compiled by the Centers for Disease Control and Prevention, indicates the most overweight states, in order, are Mississippi, Alabama, Tennessee, West Virginia, Louisiana, Oklahoma, Kentucky, Arkansas, South Carolina and North Carolina. All the way down at No. 50, the leanest state is Colorado.

Compare this result with a 2008 Gallup poll of more than 355,000 adult Americans on how important religion is in people's lives. The top 10 religious states are Mississippi, Alabama, South Carolina, Tennessee, Louisiana, Arkansas, Georgia, North Carolina, Oklahoma and, finally, Kentucky in a tie with Texas.

Colorado is middle-of-the-pack religious, with 57 percent of this state's respondents saying religion is very important in daily life. It ranked No. 37.

Covell admits he's being deliberately provocative and tongue-in-cheek and that he's no social scientist. One of those might think obesity is more likely correlated with poverty, he said.

Or, maybe, he said, people in poor health have a more immediate interest in the afterlife. Maybe it's just one too many church potlucks.

He simply wants religious people to have something else to chew on this holiday.

"As an atheist, I'm not against a good meal," Covell said, "but Thanksgiving is not a major holiday on the atheist's calendar." He is thankful to family, friends and neighbors, he admits, and for his own efforts to keep food on the table. He isn't going to overindulge because of some ritual crediting a deity for nature's bounty.

He might also want to give thanks for any religious people with a sense of humor.

8 Fat States: Correlation Example

Correlation is used to measure the strength and direction of a linear relationship between two variables. Pearson correlation has the same assumptions as linear regression (normality, equal variances, etc). Spearman correlation is a non-parametric rank-based alternative to Pearson correlation.

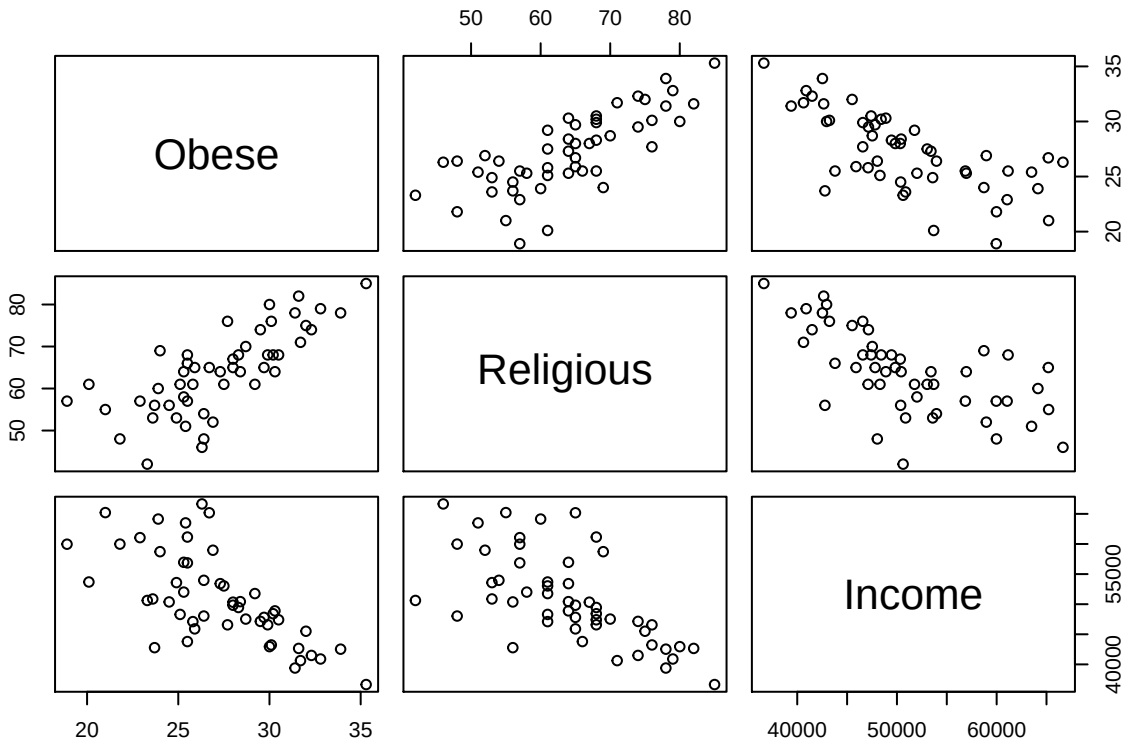
In R, there are two functions to calculate correlation. cor() will calculate pairwise correlations for a number of variables, but does not return p-values. cor.test() only allows two variables, but returns test information including the p-value.

In this example, we look at the correlation between percentage obese adults versus percentage of adults who self-identify as "religious" by state. See news article above! We also consider the correlation between percentage obese adults and median income (by state from census data).

```
InData <-read.csv("C:/hess/STAT511_FA11/RData/CH11_FatStates.csv",  row.names = 1)
head(InData)
```

```
##             Obese Religious Income
## Alabama      31.6        82  42652
## Alaska       25.4        51  63505
## Arizona      25.8        61  47106
## Arkansas     31.4        78  39392
## California   25.5        57  56862
## Colorado     18.9        57  59964
```

```
pairs(InData)
```



```r
cor(InData)
```

```
##                Obese  Religious     Income
## Obese      1.0000000  0.7472844 -0.6903531
## Religious  0.7472844  1.0000000 -0.6502106
## Income    -0.6903531 -0.6502106  1.0000000
```

```r
cor(InData, method = "spearman")
```

```
##                Obese  Religious     Income
## Obese      1.0000000  0.7652479 -0.7050438
## Religious  0.7652479  1.0000000 -0.7018364
## Income    -0.7050438 -0.7018364  1.0000000
```

```r
cor.test(InData$Religious, InData$Obese)
```

```
##
##  Pearson's product-moment correlation
##
## data:  InData$Religious and InData$Obese
## t = 7.872, df = 49, p-value = 2.999e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5940362 0.8481920
## sample estimates:
##       cor
## 0.7472844
```

```r
cor.test(InData$Religious, InData$Obese, method = "spearman")
```

```
## Warning in cor.test.default(InData$Religious, InData$Obese, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  InData$Religious and InData$Obese
## S = 5188, p-value = 6.19e-11
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.7652479
```