**Regression using likelihood estimation**

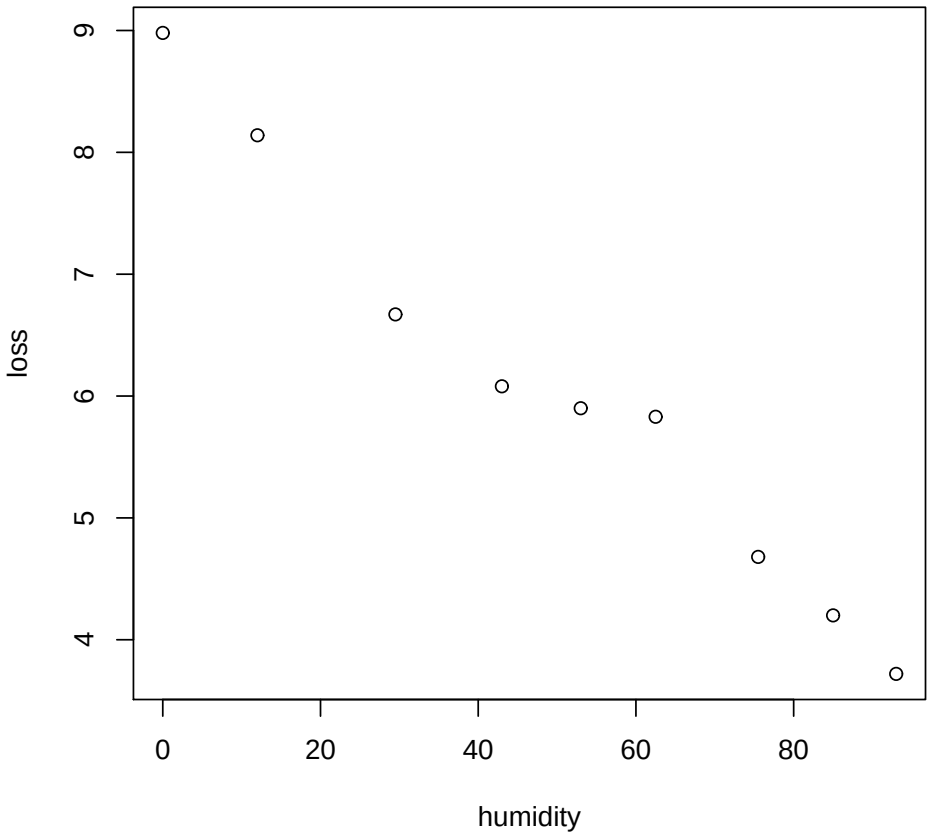
---

The maximum-likelihood estimation procedure can be used to estimate regression equations, by maximizing the likelihood of the residual variation given the parameters of the regression model

***Tribolium* data**

---

```
> humidity <- c(0, 12, 29.5, 43, 53, 62.5, 75.5, 85, 93)
> loss <- c(8.98, 8.14, 6.67, 6.08, 5.9, 5.83, 4.68, 4.2, 3.72)
> plot(loss ~ humidity)
```




**least-squares slope**

---

```
> xdev <- humidity - mean(humidity)
> ydev <- loss - mean(loss)
> b1 <- sum((xdev) * (ydev))/sum(xdev^2)
> b0 <- mean(loss) - mean(humidity) * b1
> c(b1, b0)

[1] -0.05322215  8.70402730
```

### Residuals

---

These residuals depend on the values of b0 and b1, along with the data

```
> rsd <- (b0 + b1 * humidity) - loss
> rsd

[1] -0.27597270 -0.07463851  0.46397383  0.33547479 -0.01674673 -0.45235717
[7]  0.00575486 -0.01985558  0.03436721
```

### Likelihood of residuals

---

The likelihood of the residuals assumes a normal distribution with $\mu = 0$ and $\sigma$ =sd. Taking the log allows addition of terms instead of multiplication. The higher the sum, the higher the probability of seeing all residuals from that distribution

```
> sd <- 1
> sum(dnorm(rsd, mean = 0, sd = sd, log = T))

[1] -8.578478
```

### Optimization

---

It is possible to optimize all of the parameters for the regression equation (b0, b1, sd) using the R function 'optim()'

```
> probf <- function(p, X, Y) {
+     b0 <- p[1]
+     b1 <- p[2]
+     sd <- p[3]
+     rsd <- (b0 + b1 * X) - Y
+     -sum(dnorm(rsd, mean = 0, sd = sd, log = T))
+ }
> optim(c(1, 0, 5), probf, X = humidity, Y = loss)
```

```
$par
[1]  8.70410609 -0.05322245  0.26161153

$value
[1] 0.703108

$counts
function gradient
     168       NA

$convergence
[1] 0

$message
NULL
```

**Likelihood with intercept only**

---

This equation does not have a slope,just an intercept

```
> probf2 <- function(p, X, Y) {
+     b0 <- p[1]
+     sd <- p[2]
+     rsd <- (b0) - Y
+     -sum(dnorm(rsd, mean = 0, sd = sd, log = T))
+ }
> optim(c(1, 5), probf2, X = humidity, Y = loss)

$par
[1] 6.021926 1.637569

$value
[1] 17.20859

$counts
function gradient
      63       NA

$convergence
[1] 0

$message
NULL
```

## Comparing likelihoods

With nested models, it is possible to test if a model is better than another using a Likelihood ratio test. The test statistic is $G$:
$G = 2 * (ln(L(\text{generalmodel})) - ln(L(\text{nestedmodel})))$
In the humidity case:
$G = 2 * (-0.7 - -17.2) = 33$

## Assigning a p-value

The $G$ statistic is distributed as a $\chi^2$ with d.f. = difference in number of parameters between the models.
In this example, $G = 33$, d.f. $= 1$:

```
> 1 - pchisq(33, 1)

[1] 9.215887e-09
```

So there is strong support to include the slope in the analysis

## Same using R functions

The R function 'glm()' is a generalized version of 'lm()' and can be used to compare nested models

```
> fit.gen <- glm(loss ~ humidity)
> fit.nest <- glm(loss ~ 1)
> anova(fit.gen, fit.nest, test = "Chisq")

Analysis of Deviance Table

Model 1: loss ~ humidity
Model 2: loss ~ 1
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1         7     0.6161
2         8    24.1306 -1 -23.5145 4.665e-60
```

## Use AIC

Models fit with the same data but non-nested models can be compared informally with AIC. An improvement of AIC of 3 or more indicates that the term is important.
AIC $= 2k - 2ln(L)$, where $k$ is the number of parameters

```
> fit.gen <- glm(loss ~ humidity)
> fit.nest <- glm(loss ~ 1)
> anova(fit.gen, fit.nest, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: loss ~ humidity
Model 2: loss ~ 1
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1         7     0.6161
2         8    24.1306 -1 -23.5145 4.665e-60
```