ZNZ Crash Course in Statistics

Adin Ross-Gillespie, QUANTIK

Friday, February 20, 2015

Chapter 3

Read in the data set **piggrowth.csv**

```
Dat <- read.csv("data/piggrowth.csv", header=TRUE)
```

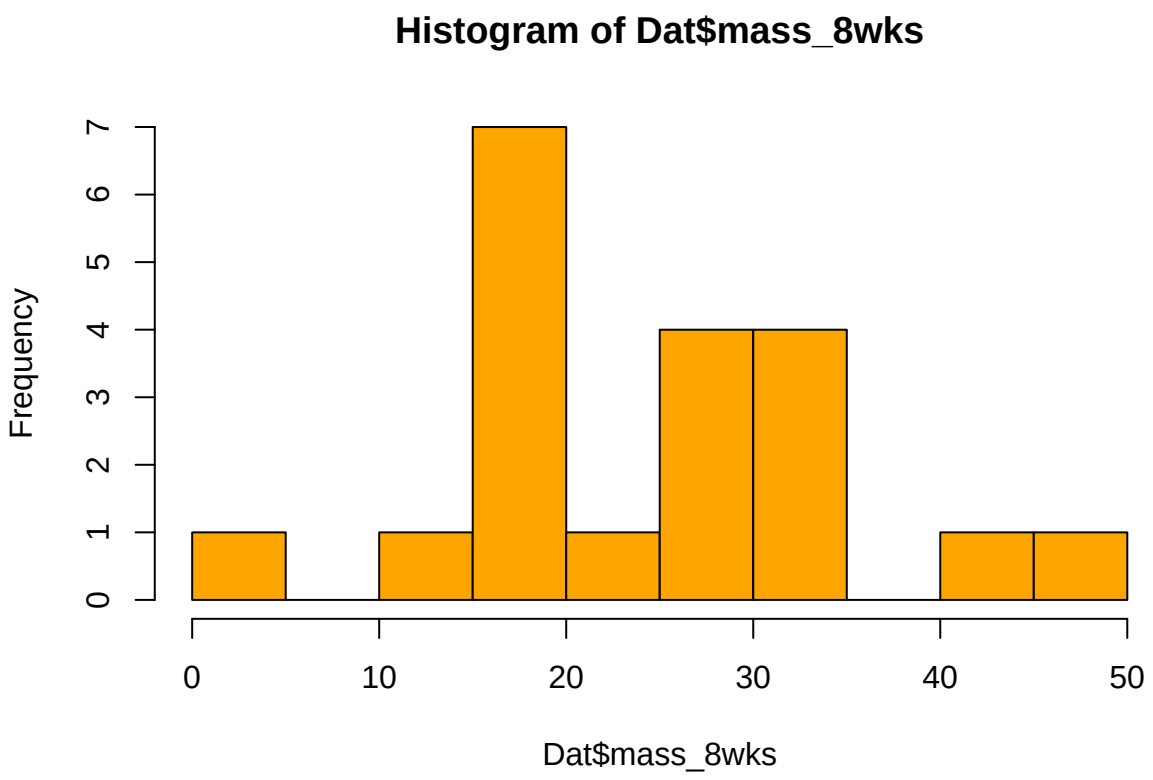
Take a look at the data using the function **summary(...)**

```
summary(Dat)
```

```
##  weekly_intake     mass_8wks
##  Min.   : 9.40   Min.   : 4.30
##  1st Qu.:13.82   1st Qu.:16.10
##  Median :20.20   Median :25.40
##  Mean   :20.91   Mean   :24.34
##  3rd Qu.:25.20   3rd Qu.:31.10
##  Max.   :38.60   Max.   :46.30
```

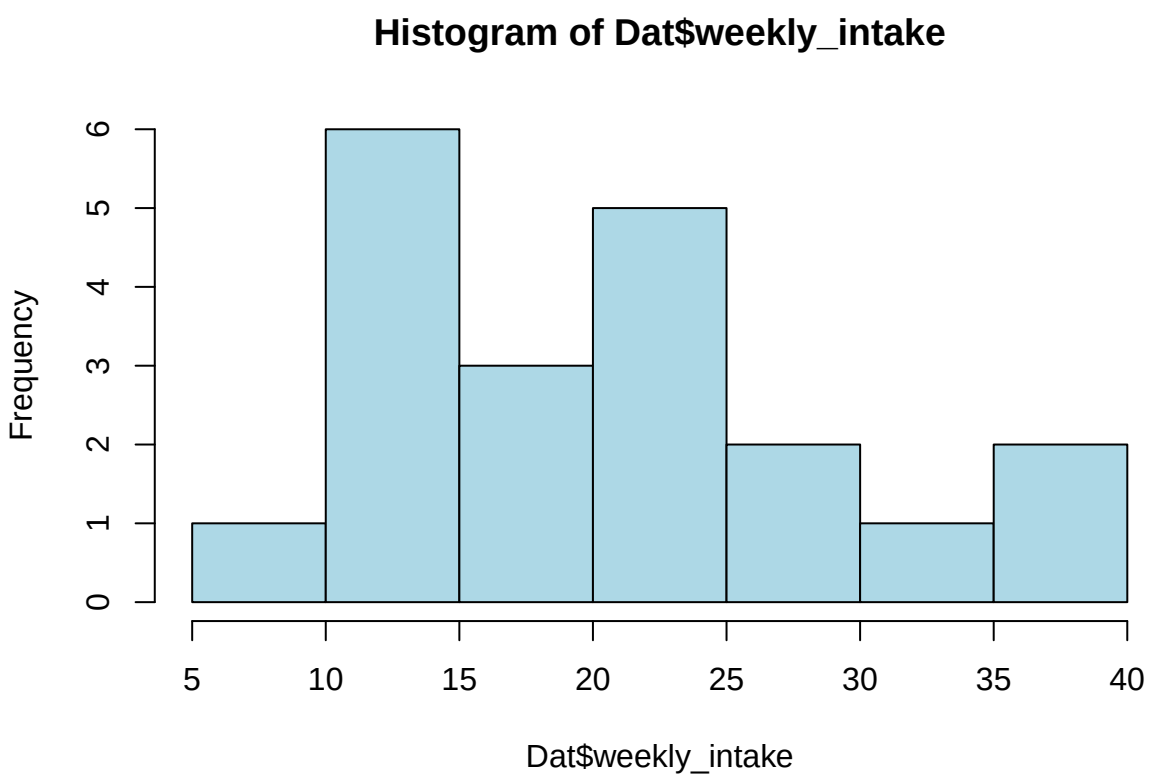
Plot a *histogram* of the masses of piglets after eight weeks...

```
hist(Dat$mass_8wks, col="orange")
```



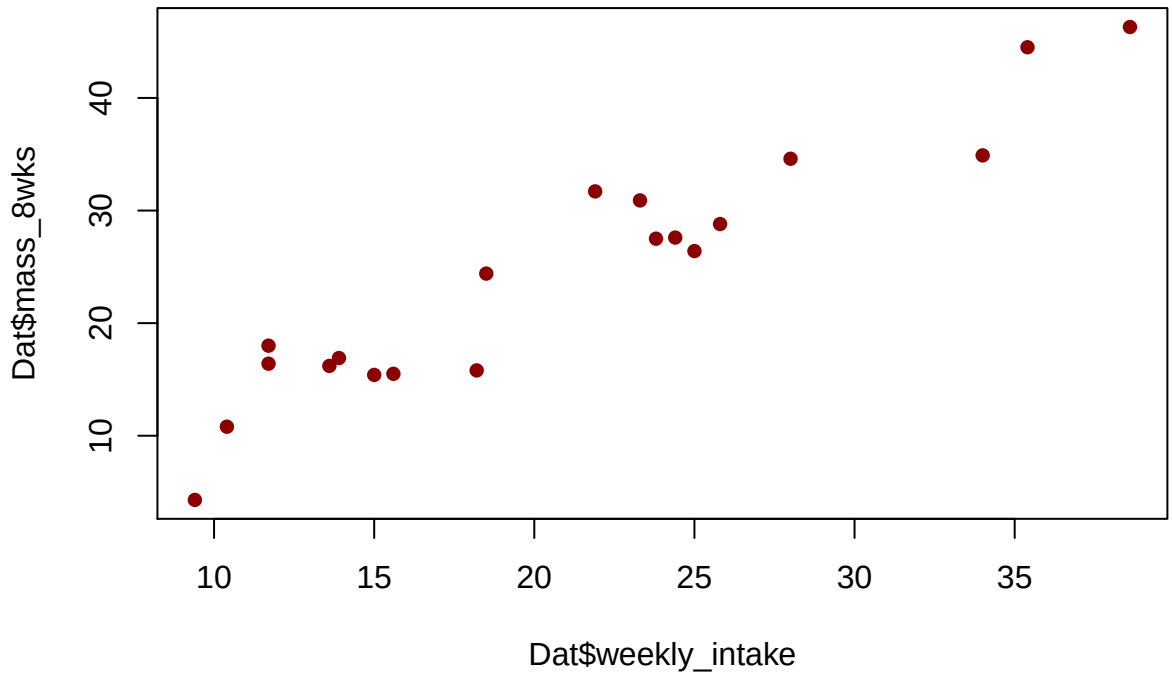
. . . and now look at the distribution of the weekly intakes in the same way.

```
hist(Dat$weekly_intake, col="lightblue")
```



Plot the two variables together to see potential relation:

```
plot(x=Dat$weekly_intake, y=Dat$mass_8wks, pch=16, col="red4")
```



Let's quantify the association between the two variables using Pearson's product moment correlation coefficient:

```
cor.test(x=Dat$weekly_intake, y=Dat$mass_8wks)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Dat$weekly_intake and Dat$mass_8wks
## t = 13.0294, df = 18, p-value = 1.327e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8776216 0.9807189
## sample estimates:
##       cor
## 0.9508606
```

Chapter 7

Read in the data set **BloodPressureChanges.csv**

```
Dat <- read.csv("data/BloodPressureChanges.csv", header=TRUE)
```

Take a look at the data using the function **summary(...)**

```
summary(Dat)
```

```
##      Study1          Study2_Drug      Study2_Placebo
##  Min.   :-15.500   Min.   :-23.100   Min.   :-30.7
##  1st Qu.: -6.300   1st Qu.:-13.600   1st Qu.:-15.9
##  Median : -3.600   Median :-10.500   Median :-10.6
##  Mean   : -2.704   Mean   : -9.694   Mean   :-11.4
##  3rd Qu.:  2.150   3rd Qu.: -4.850   3rd Qu.: -6.2
##  Max.   :  8.500   Max.   :  0.900   Max.   :  2.7
##                    NA's   :20        NA's   :10
```

7.5.3: for a sample $x_1...x_n$ from population $\mathcal{N}(\mu, \sigma^2)$ with σ^2 unknown: is μ equal to some μ_0 (for example, equal to 0) [1 SAMPLE T-TEST]

```
t.test(Dat$Study1, mu = 0)
```

```
##
##  One Sample t-test
##
## data:  Dat$Study1
## t = -3.4508, df = 50, p-value = 0.001145
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -4.277741 -1.130102
## sample estimates:
## mean of x
## -2.703922
```

7.5.4: for independent samples $x_1...x_m$ from population $\mathcal{N}(\mu_1, \sigma^2)$ and $y_1...y_n$ from population $\mathcal{N}(\mu, \sigma^2)$: is $\mu_1 = \mu_2$? [2 SAMPLE T-TEST]

```
t.test(Dat$Study2_Drug, Dat$Study2_Placebo)
```

```
##
##  Welch Two Sample t-test
##
## data:  Dat$Study2_Drug and Dat$Study2_Placebo
## t = 1.0326, df = 67.816, p-value = 0.3055
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.593747  5.011528
## sample estimates:
##  mean of x  mean of y
##  -9.693548 -11.402439
```

Note that when we ask R to perform a 2-sample t-test, by default it performs Welch's modified version of the original t-test. The original t-test assumes that the variances of the two samples to be compared are equal, while Welch's modified version of the test accepts that the variances may be different, and adjusts the estimate of the t-statistic, and the degrees of freedom accordingly to account for this.

In fact, we can test this assumption of equality of the variances in the 'Drug' and 'Placebo' samples as follows:

```
var.test(Dat$Study2_Drug, Dat$Study2_Placebo)
```

```
##
##  F test to compare two variances
##
## data:  Dat$Study2_Drug and Dat$Study2_Placebo
## F = 0.8105, num df = 30, denom df = 40, p-value = 0.5547
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.4171718 1.6282509
## sample estimates:
## ratio of variances
##          0.8105298
```

The variances do not appear to be significantly different. So, let's repeat the t-test, but this time perform the original test, not Welch's conservative modified version.

```
t.test(Dat$Study2_Drug, Dat$Study2_Placebo, var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  Dat$Study2_Drug and Dat$Study2_Placebo
## t = 1.0175, df = 70, p-value = 0.3124
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.640891  5.058673
## sample estimates:
##  mean of x  mean of y
##  -9.693548 -11.402439
```

7.6: An example of a non-parametric test. Pearson's χ^2-test for independence in contingency tables

For this next test, read in the data set **MathsTestResults.csv**

```
Dat <- read.csv("data/MathsTestResults.csv", header=TRUE)
```

This dataset contains qualitative information about the genders and maths-test results for a class of students. Each row of the table contains information on a different student.

How many rows (i.e. students) is that in total?

```
nrow(Dat)
```

```
## [1] 100
```

Data like this can easily be tabulated using the function **table(...)**

```
table(Dat)
```

```
##         Result
## Sex      fail pass
##   female   22   32
##   male     12   34
```

Let's now test for an association between the variable 'Sex' and the variable 'Result' using the χ^2-test.

```
chisq.test(table(Dat))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(Dat)
## X-squared = 1.7688, df = 1, p-value = 0.1835
```

For 2×2 contingency tables, as we have here, R automatically applies Yates' continuity correction when calculating the χ^2. If we want to perform the unmodified χ^2-test, we do so as follows:

```
chisq.test(table(Dat), correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(Dat)
## X-squared = 2.377, df = 1, p-value = 0.1231
```

... although perhaps the more appropriate test for the independence of two variables in a 2×2 contingency table would be Fisher's Exact Test:

```
fisher.test(table(Dat))
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  table(Dat)
## p-value = 0.1424
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.7688327 5.0501270
## sample estimates:
## odds ratio
##   1.934955
```

Let's try these tests again, using another dataset (**DeathSentences.csv**)

```
Dat <- read.csv("data/DeathSentences.csv", header=TRUE)
table(Dat)
```

```
##                   Sentence
## Race               death other
##   african_american    17   149
##   white               19   141
```

```
nrow(Dat)
```

```
## [1] 326
```

We perform the tests as before

```
chisq.test(table(Dat))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(Dat)
## X-squared = 0.0863, df = 1, p-value = 0.7689
```

```
chisq.test(table(Dat), correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(Dat)
## X-squared = 0.2214, df = 1, p-value = 0.6379
```

```
fisher.test(table(Dat))
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  table(Dat)
## p-value = 0.7246
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.396142 1.798510
## sample estimates:
## odds ratio
##  0.8471313
```

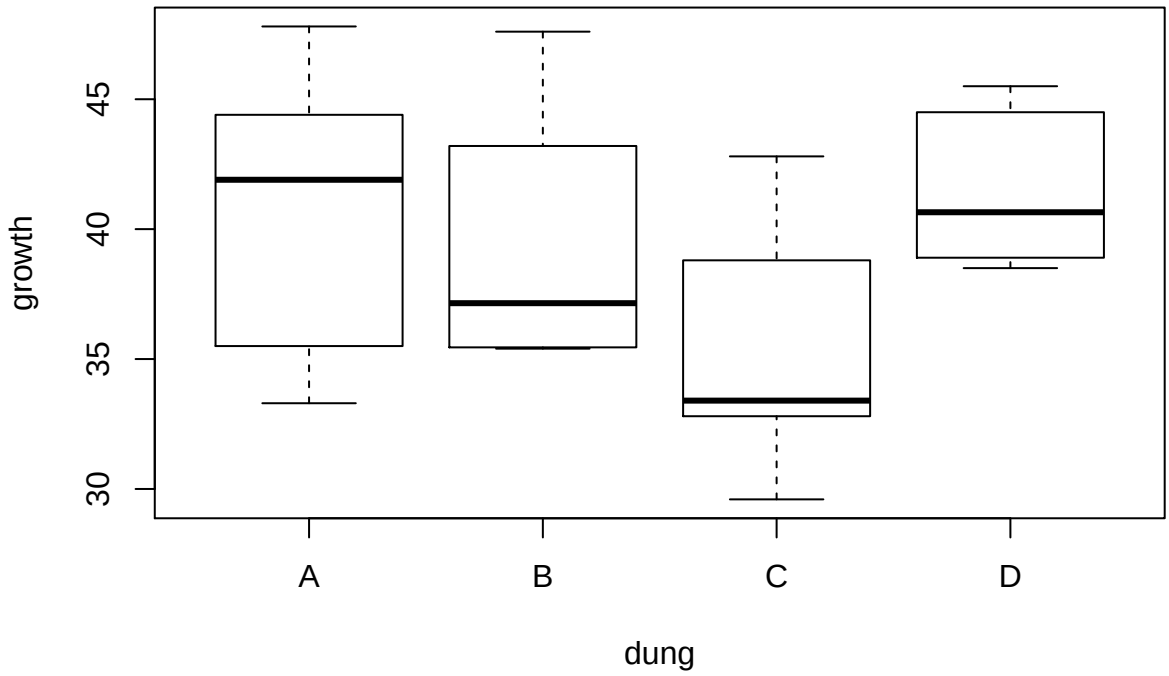
7.7.2: 1-way ANOVA Time to try a one-way ANOVA. Let's begin by reading in and then examining the structure of the **DungGrowth.csv** data set

```
Dat <- read.csv("data/DungGrowth.csv", header=TRUE)
str(Dat)
```

```
## 'data.frame':    21 obs. of  2 variables:
##  $ growth: num  33.3 47.8 44.4 42.9 40.9 35.5 35.5 35.4 47.6 38.8 ...
##  $ dung  : Factor w/ 4 levels "A","B","C","D": 1 1 1 1 1 1 2 2 2 2 ...
```

This dataset presents growth measures for four different dung types, labeled A-D Let's make a quick plot to see how growth differs across the four dung types:

```
plot(growth~dung, data=Dat)
```



To perform an ANOVA, we use the **aov()** function. However, if we want to see the complete ANOVA table, we'll need to ask for a **summary()** of the object produced by the **aov()** call

```
aov(growth~dung, data=Dat)
```

```
## Call:
##    aov(formula = growth ~ dung, data = Dat)
##
## Terms:
##                     dung Residuals
## Sum of Squares  113.7990  408.5105
## Deg. of Freedom        3        17
##
## Residual standard error: 4.902043
## Estimated effects may be unbalanced
```

```
summary(aov(growth~dung, data=Dat))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## dung         3  113.8   37.93   1.579  0.231
## Residuals   17  408.5   24.03
```

Chapter 8

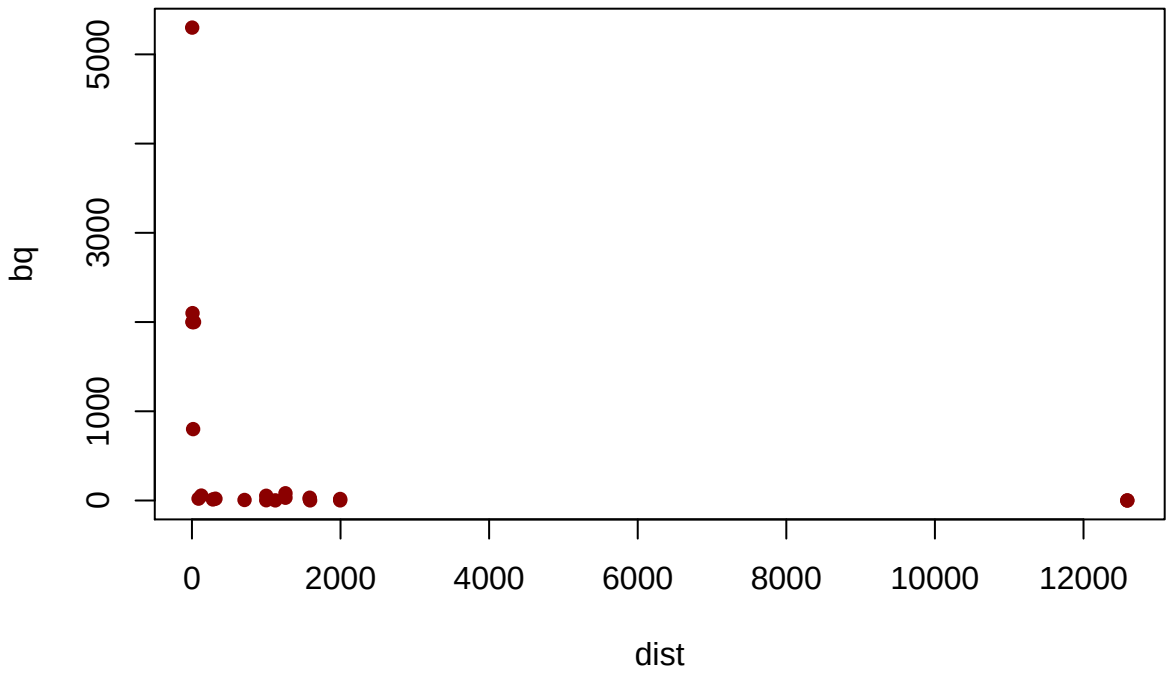
Read in the data set **Tschernobyl.csv**

```
Dat <- read.csv("data/Tschernobyl.csv", header=TRUE)
summary(Dat)
```

```
##           location   rain         dist                bq
##  Chilton       : 1   No :16   Min.   :    3.16   Min.   :   0.15
##  Chistogalovka: 1   Yes: 7   1st Qu.:   58.66   1st Qu.:   4.50
##  Donezk        : 1            Median : 1000.00   Median :  21.00
##  Gaevle        : 1            Mean   : 1789.79   Mean   : 632.67
##  Irland        : 1            3rd Qu.: 1584.89   3rd Qu.: 440.50
##  Japan         : 1            Max.   :12589.25   Max.   :5300.00
##  (Other)       :17
```

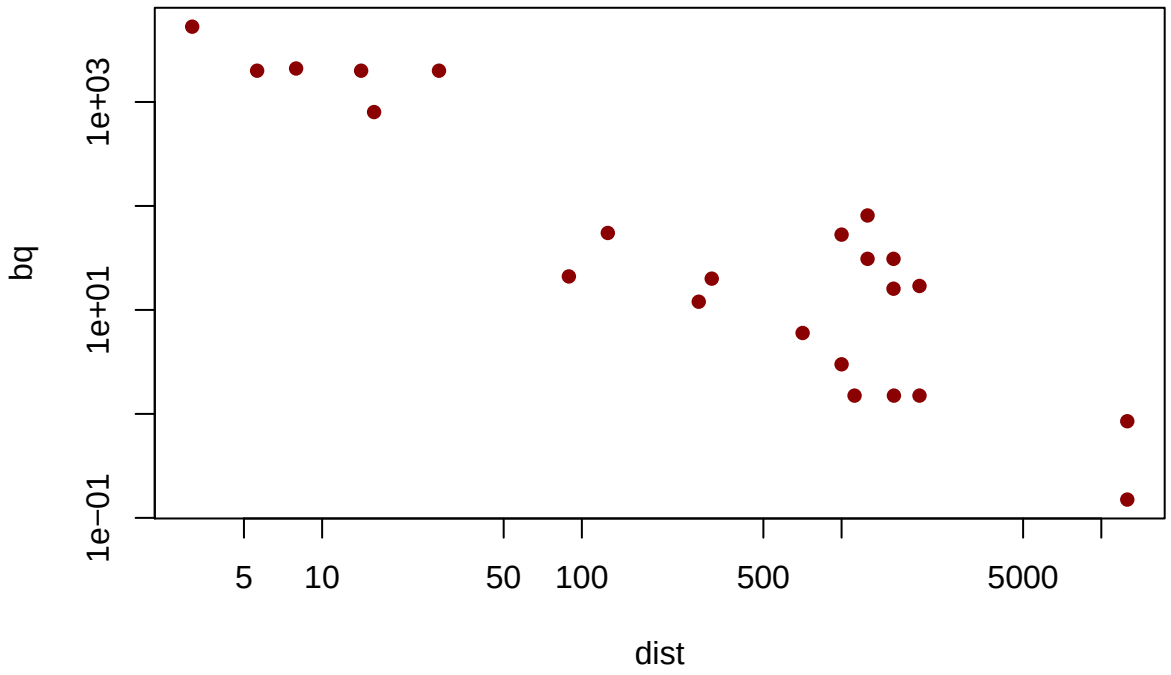
We are interested in how radiactivity ('bq') at various sites is predicted by their distance from Tschernobyl ('dist'). Let's plot this:

```
plot(bq~dist, data=Dat, pch=16, col="red4")
```



Hmm... the values for both 'bq' and 'dist' vary considerably. We should try log-transforming the data. For now, however, let's just convert both axes of the plot to log scale.

```
plot(bq~dist, data=Dat, pch=16, col="red4", log="xy")
```



That's better! From the plot above it looks like the (log-transformed) variables are related. Let's formally test for the correlation:

```
cor.test(log(Dat$bq), log(Dat$dist))
```

```
##
##  Pearson's product-moment correlation
##
## data:  log(Dat$bq) and log(Dat$dist)
## t = -9.6617, df = 21, p-value = 3.527e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9586791 -0.7828950
## sample estimates:
##       cor
## -0.903521
```

This question can also be formulated as a linear model. We'll fit one using the **lm()** function, and call it 'm1'

```
m1 <- lm(log(bq)~log(dist), data=Dat)
summary(m1)
```

```
##
## Call:
## lm(formula = log(bq) ~ log(dist), data = Dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8592 -1.1348 -0.1033  1.1650  2.3798
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.8027     0.7022  13.960 4.26e-12 ***
## log(dist)    -1.0911     0.1129  -9.662 3.53e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.28 on 21 degrees of freedom
## Multiple R-squared:  0.8164, Adjusted R-squared:  0.8076
## F-statistic: 93.35 on 1 and 21 DF,  p-value: 3.527e-09
```

We can also check the anova table from this model using **anova()**

```
anova(m1)
```

```
## Analysis of Variance Table
##
## Response: log(bq)
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## log(dist)  1 152.875 152.875  93.348 3.527e-09 ***
## Residuals 21  34.391   1.638
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

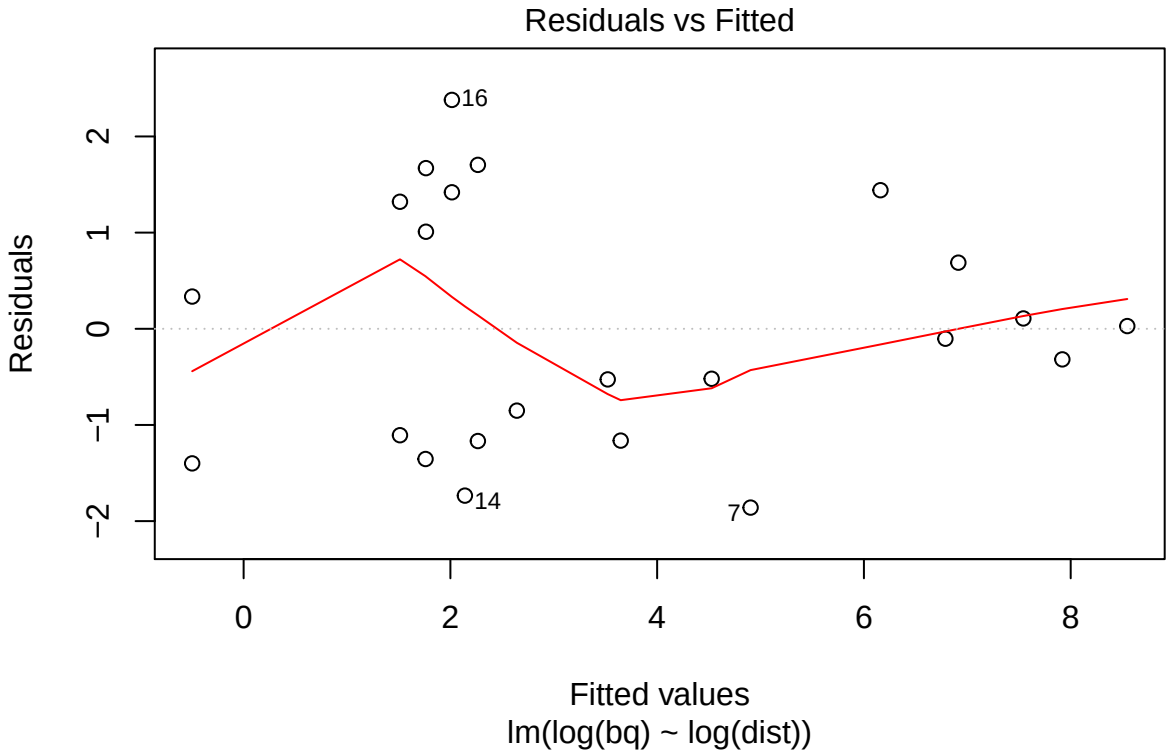
It may be useful to divide each of the continuous variables by their standard deviations before fitting. This rescales the values and centers them around zero. This way, we can interpret the coefficients as units of standard deviation. This is easy in R:

```
m1_scaled <- lm(scale(log(bq))~scale(log(dist)), data=Dat)
summary(m1_scaled)
```

```
##
## Call:
## lm(formula = scale(log(bq)) ~ scale(log(dist)), data = Dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.63725 -0.38895 -0.03541  0.39932  0.81567
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -8.367e-17  9.146e-02   0.000        1
## scale(log(dist)) -9.035e-01  9.352e-02  -9.662 3.53e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4386 on 21 degrees of freedom
## Multiple R-squared:  0.8164, Adjusted R-squared:  0.8076
## F-statistic: 93.35 on 1 and 21 DF,  p-value: 3.527e-09
```

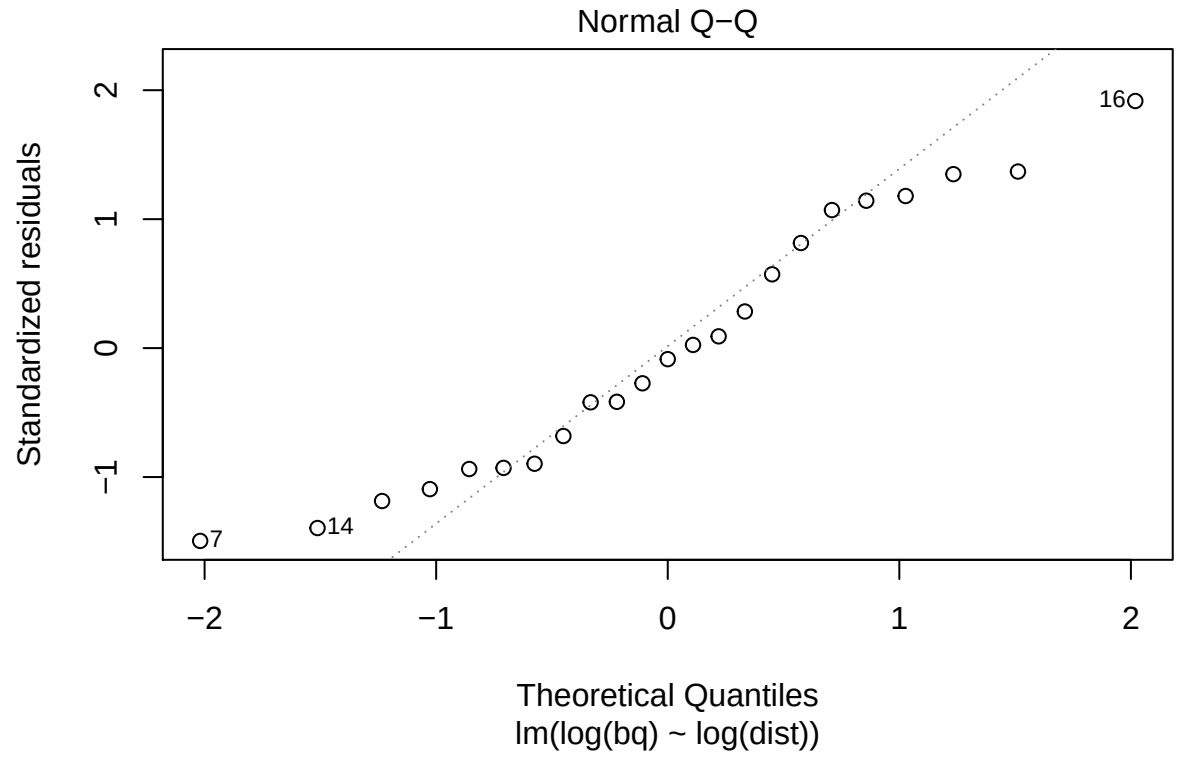
We can access the residuals and fitted values of our models with the **resid()** and **fitted()** functions respectively. Perhaps the most convenient way to inspect them, however, is graphically:

```
plot(m1,which=1)
```



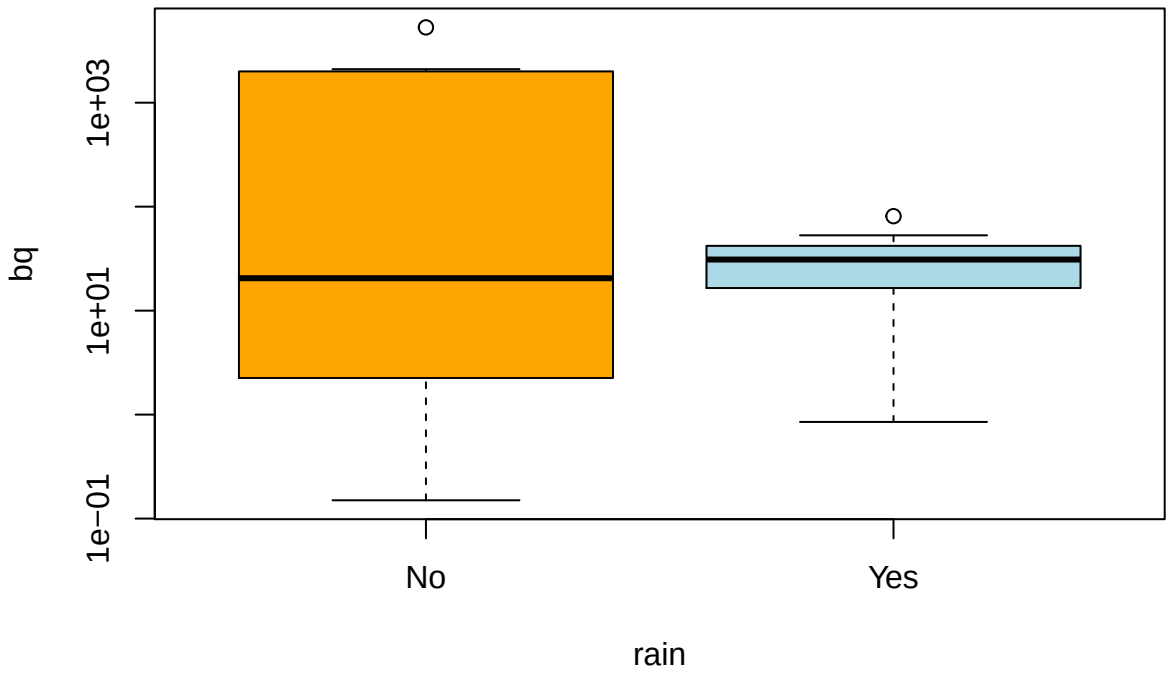
To quickly inspect the normality of the residuals, a QQ-plot can be very useful.

```
plot(m1,which=2)
```



Now... what about that 'rain' variable? Perhaps rain also influences the radioactivity levels at the sites.

```
plot(bq~rain, data=Dat, col=c("orange", "lightblue"), log="y")
```



Time for another model!

```
m2 <- lm(log(bq)~rain, data=Dat)
summary(m2)
```

```
##
## Call:
## lm(formula = log(bq) ~ rain, data = Dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6755 -2.3332 -0.1205  2.1735  4.7970
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7784     0.7399   5.107 4.66e-05 ***
## rainYes      -0.8247     1.3412  -0.615    0.545
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.96 on 21 degrees of freedom
## Multiple R-squared:  0.01769,    Adjusted R-squared:  -0.02909
## F-statistic: 0.3781 on 1 and 21 DF,  p-value: 0.5452
```

```
anova(m2)
```

```
## Analysis of Variance Table
```

```
##
## Response: log(bq)
##           Df  Sum Sq Mean Sq F value Pr(>F)
## rain       1   3.312  3.3119  0.3781 0.5452
## Residuals 21 183.954  8.7597
```

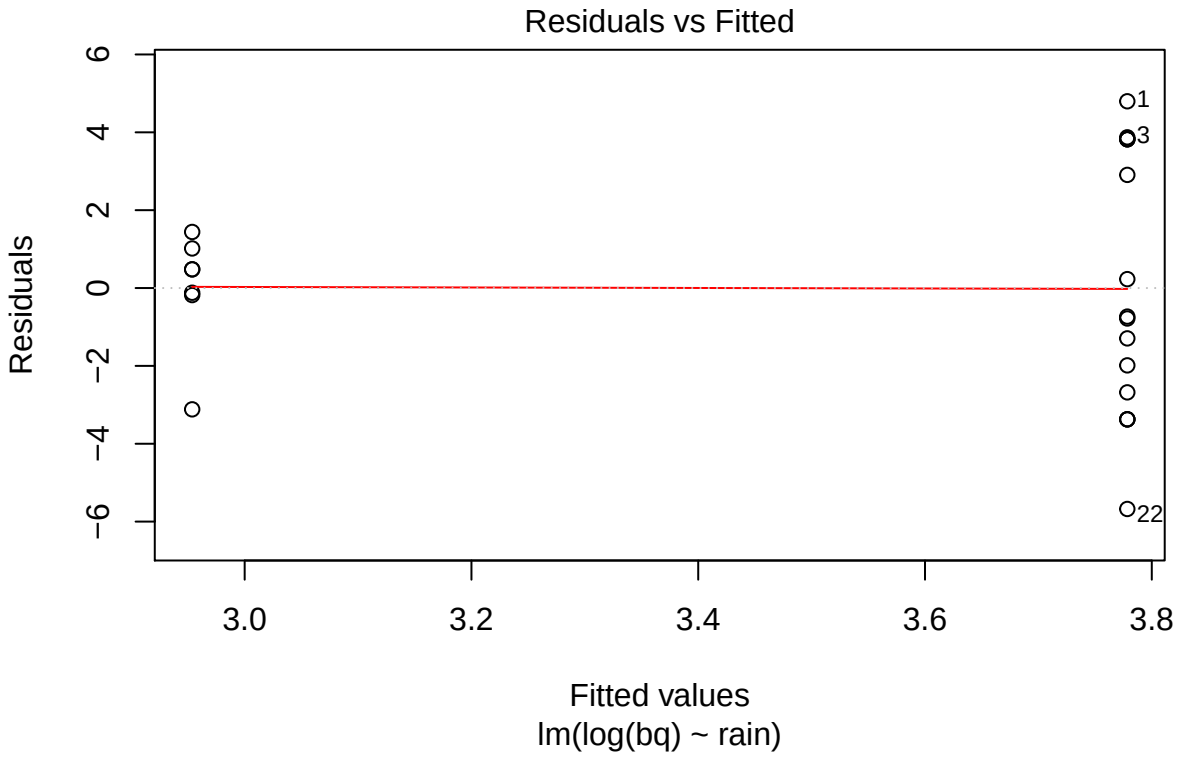
Here, the linear model is equivalent to a one-way ANOVA. As before, we may want to scale the continuous variables by their standard deviations...

```
m2_scaled <- lm(scale(log(bq))~scale(log(dist)), data=Dat)
summary(m2_scaled)
```

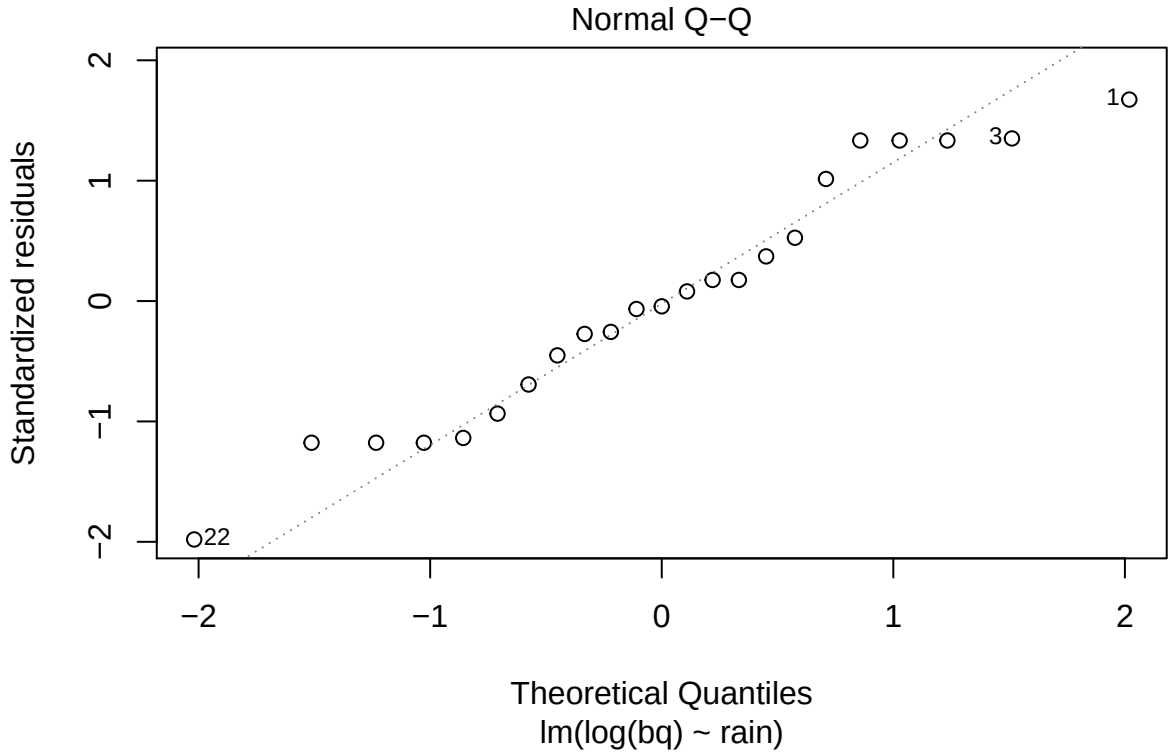
```
##
## Call:
## lm(formula = scale(log(bq)) ~ scale(log(dist)), data = Dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.63725 -0.38895 -0.03541  0.39932  0.81567
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -8.367e-17  9.146e-02   0.000        1
## scale(log(dist)) -9.035e-01  9.352e-02  -9.662 3.53e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4386 on 21 degrees of freedom
## Multiple R-squared:  0.8164, Adjusted R-squared:  0.8076
## F-statistic: 93.35 on 1 and 21 DF,  p-value: 3.527e-09
```

... and we should, as always, inspect the diagnostic plots:

```
plot(m2,which=1)
```

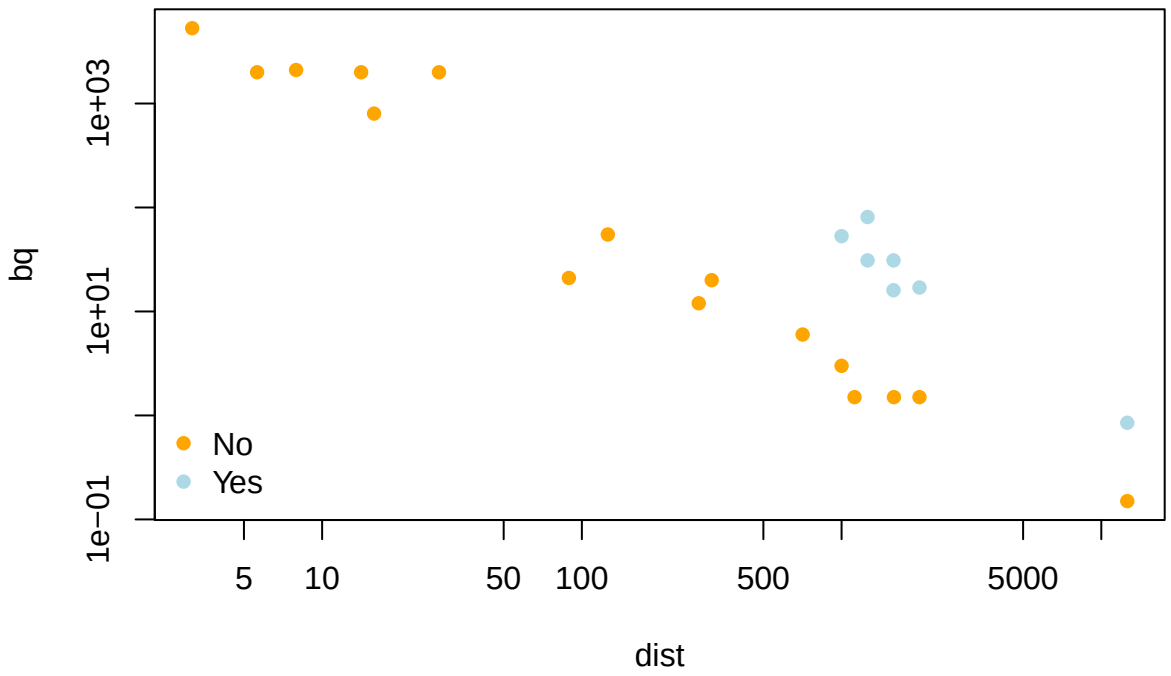



```
plot(m2,which=2)
```



So, we've considered, independently, the effect of 'dist' and 'rain' on radioactivity. Perhaps these two factors have an interactive effect on radioactivity?

```
plot(bq~dist, data=Dat, subset=(rain=="No"), pch=16, col="orange", log="xy")
points(bq~dist, data=Dat, subset=(rain=="Yes"), pch=16, col="lightblue")
legend("bottomleft", bty="n", legend=levels(Dat$rain), pch=16, col=c("orange", "lightblue"))
```



Hmm... previously, it seemed that rain was not an important predictor, but here we see that the points are segregating according to their 'rain' status. Let's examine this with a formal analysis. Here lies the real strength of linear models. When we fit multiple predictor variables together in the same model, we can evaluate the relative influences of the different predictors, independently and in interaction with one another.

```
m3 <- lm(log(bq)~log(dist)+rain, data=Dat)
summary(m3)
```

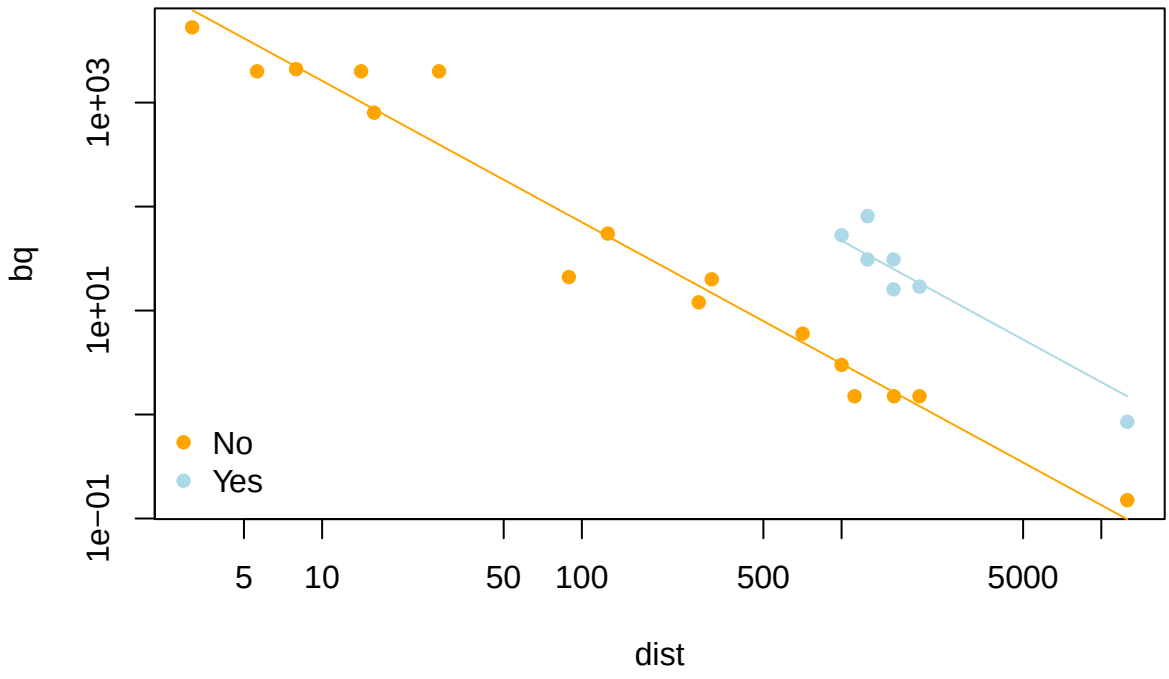
```
##
## Call:
## lm(formula = log(bq) ~ log(dist) + rain, data = Dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.37021 -0.37287 -0.05441  0.21542  1.61984
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.52246    0.34904  30.147  < 2e-16 ***
## log(dist)   -1.36027    0.06316 -21.536 2.62e-15 ***
## rainYes      2.72254    0.32436   8.394 5.51e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6166 on 20 degrees of freedom
## Multiple R-squared:  0.9594, Adjusted R-squared:  0.9553
## F-statistic: 236.3 on 2 and 20 DF,  p-value: 1.219e-14
```

```
anova(m3)
```

```
## Analysis of Variance Table
##
## Response: log(bq)
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## log(dist)  1 152.875 152.875 402.070 1.028e-14 ***
## rain       1  26.787  26.787  70.451 5.508e-08 ***
## Residuals 20   7.604   0.380
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So, once we have controlled for the effects of sites' distance from source, we see that rain also has a significant positive effect on radioactivity levels

```
plot(bq~dist, data=Dat, subset=(rain=="No"), pch=16, col="orange", log="xy")
points(bq~dist, data=Dat, subset=(rain=="Yes"), pch=16, col="lightblue")
legend("bottomleft", bty="n", legend=levels(Dat$rain), pch=16, col=c("orange", "lightblue"))
lines(x = Dat$dist[Dat$rain=="No"], y = exp(fitted(m3))[Dat$rain=="No"], col="orange")
lines(x = Dat$dist[Dat$rain=="Yes"], y = exp(fitted(m3))[Dat$rain=="Yes"], col="lightblue")
```



Once again, we may want to scale the continuous variables by their standard deviations...

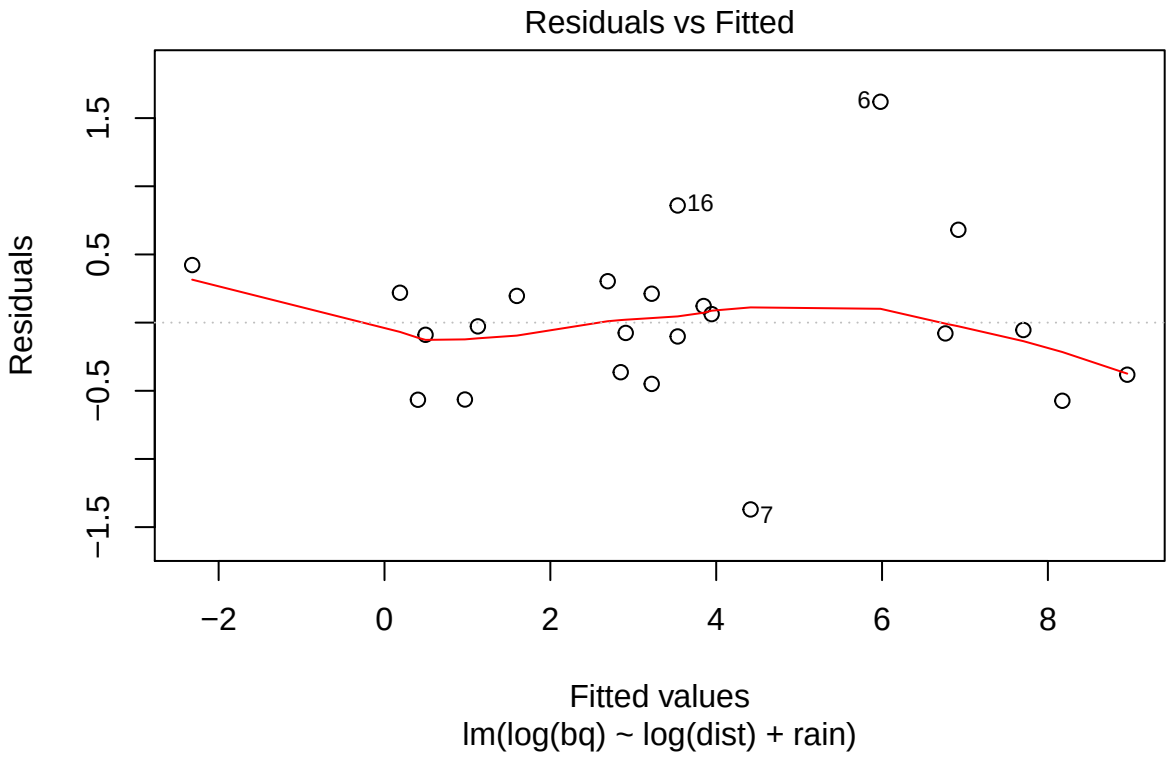
```
m3_scaled <- lm(scale(log(bq))~scale(log(dist))+scale(as.numeric(rain)), data=Dat)
summary(m3_scaled)
```

```
##
## Call:
## lm(formula = scale(log(bq)) ~ scale(log(dist)) + scale(as.numeric(rain)),
##     data = Dat)
##
```

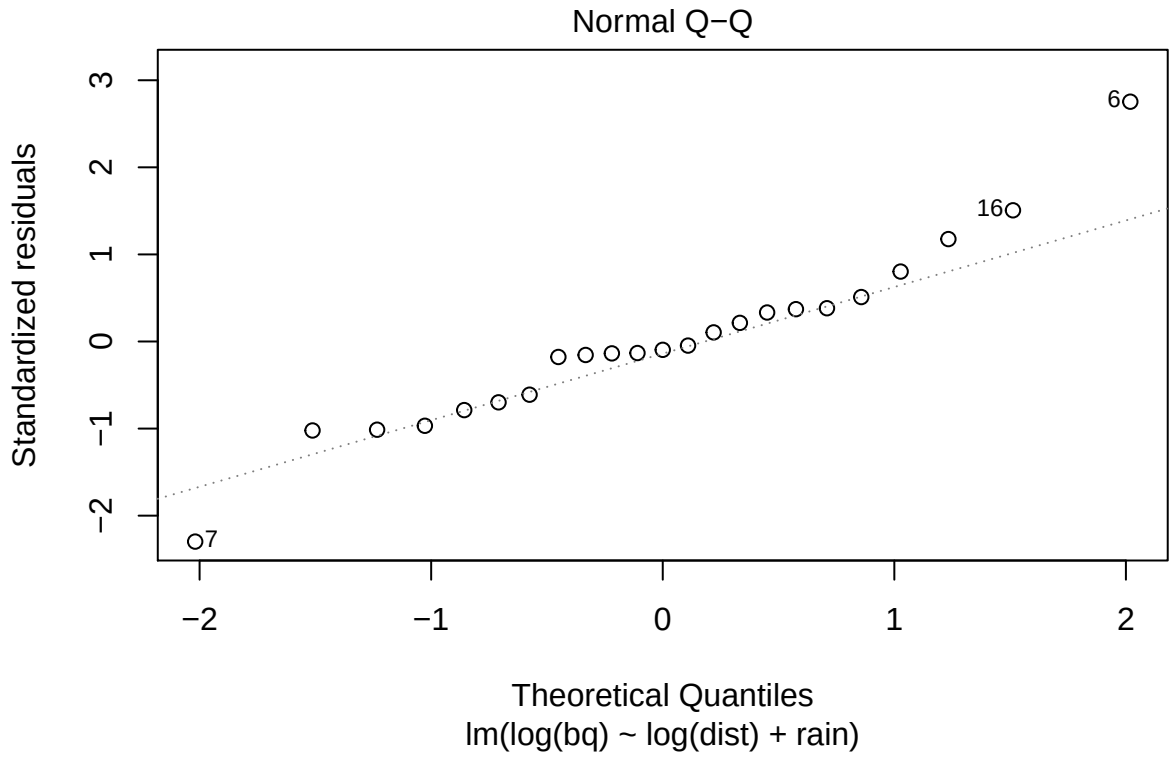
```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46964 -0.12780 -0.01865  0.07383  0.55521
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               7.488e-18  4.407e-02   0.000        1
## scale(log(dist))         -1.126e+00  5.231e-02 -21.536 2.62e-15 ***
## scale(as.numeric(rain))   4.390e-01  5.231e-02   8.394 5.51e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2113 on 20 degrees of freedom
## Multiple R-squared:  0.9594, Adjusted R-squared:  0.9553
## F-statistic: 236.3 on 2 and 20 DF,  p-value: 1.219e-14
```

... and we should, as always, inspect the diagnostic plots:

```
plot(m3,which=1)
```




```
plot(m3,which=2)
```



Finally, let's consider how our various models have reduced the (residual) variation in our data, compared to the null model situation

```
boxplot(list("ln(bq)"=log(Dat$bq), "m1 residuals"=resid(m1), "m3 residuals"=resid(m3)))
```