(Very Basic) Probability and Statistics in R

Ben Augustine

August 28, 2019

Here, we will cover some basic probability and statistics concepts R. This is not (so much) a lesson in probability or statistics, but a demonstration of how to do a few statistical things in R.

Probability Distributions in R

We will start with the "dpqr" type functions for probability distributions. What are these? Let's look at the Binomial distrubtion help file.

```
?dbinom #RMarkdown will not print the help file for us here
```

```
## starting httpd help server ... done
```

dbinom() is the probability density or mass function (PDF or PMF, PMF for discrete distributions), pbinom is the cumulative distribution function (CDF), qbinom() is the quantile function, and rbinom() is the random number generator. Let's look at the random number generator. The Binomial distribution is a distribution for the number of successes for a given number of trials. It has parameters K, the number of trials and p, the success probability per trial. Say we are flipping a fair coin where a success is defined as a heads. Then p can be assumed to be 0.5. If we flip it 10 times, K is 10.

```
rbinom(1,size=10,prob=0.5)
```

```
## [1] 4
```

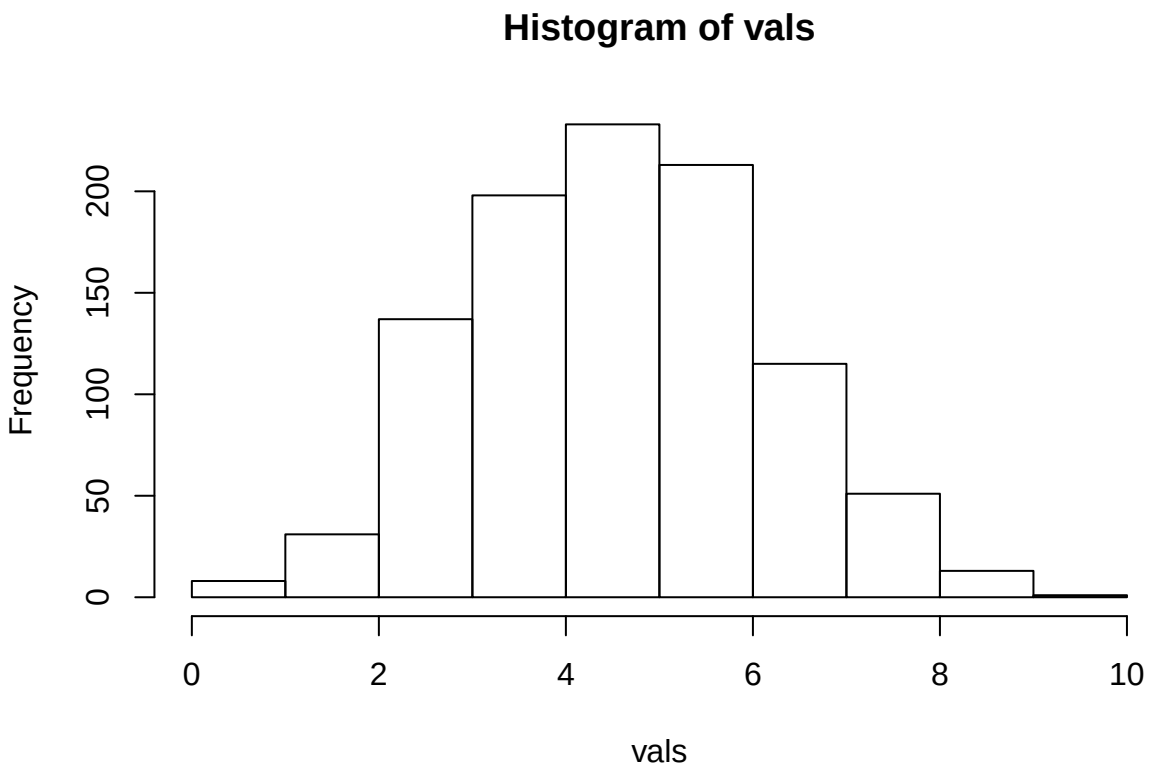
This is a single Binomial random deviate (1 random draw from a Binomial). We can draw multiple at once

```
rbinom(10,size=10,prob=0.5)
```

```
##  [1] 4 3 4 6 6 4 5 2 5 4
```

How many succeses to we expect, on average? The expected number is pK. How much variability do we expect?

```
vals=rbinom(1000,size=10,prob=0.5) #generate 1000 random numbers from a Binomial distribution
hist(vals) #create a histogram of the random numbers
```



How many times do we expect a single heads will be seen when flipping a fair coin 10 times? A good first approximation is to count the proportion of the times this happened in the random data we just simulated.

```
sum(vals==1)/length(vals) #test if vals==1, count them, then divide by the number of numbers
```

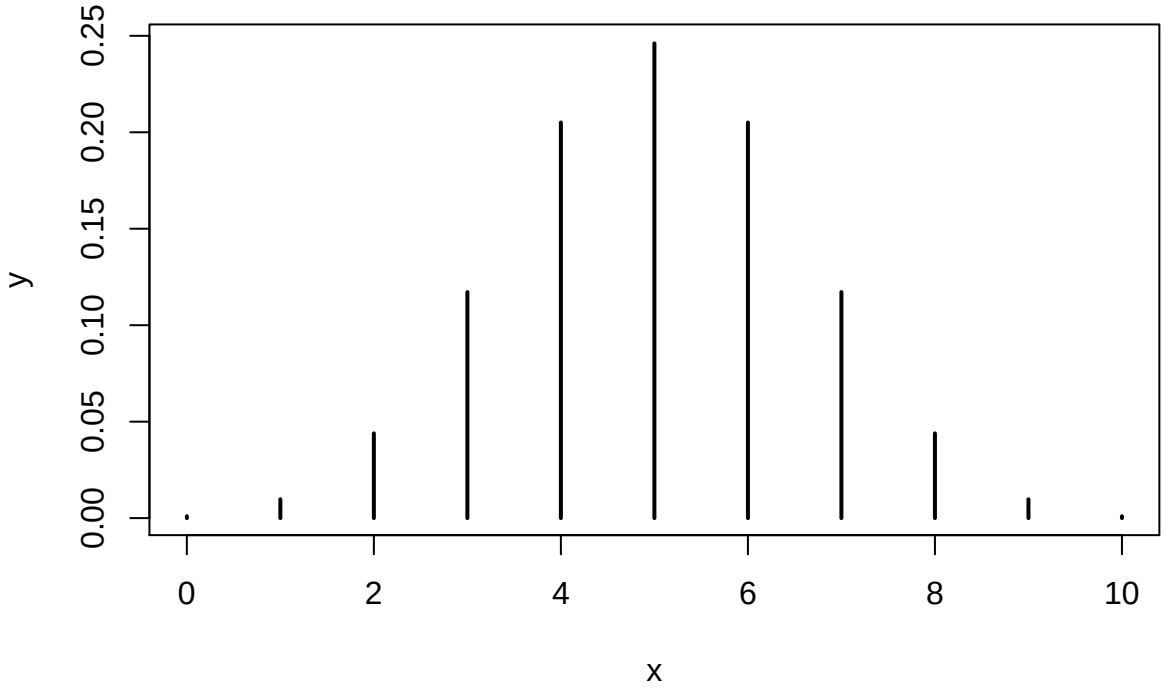
```
## [1] 0.007
```

```
mean(vals==1) #equivalent
```

```
## [1] 0.007
```

This is called Monte Carlo simulation. We can approximate many different types of statistics and probabilities via calculations on simulated random numbers. But we can use the Binomial distribution itself to get the exact answer. We need to use the PMF, or probability mass function. What does this look like?

```
x=0:10
y=dbinom(x,size=10,prob=0.5) #calculate the PMF values for 1 - 10 successes
plot(y~x,type="h",lwd=2)
```



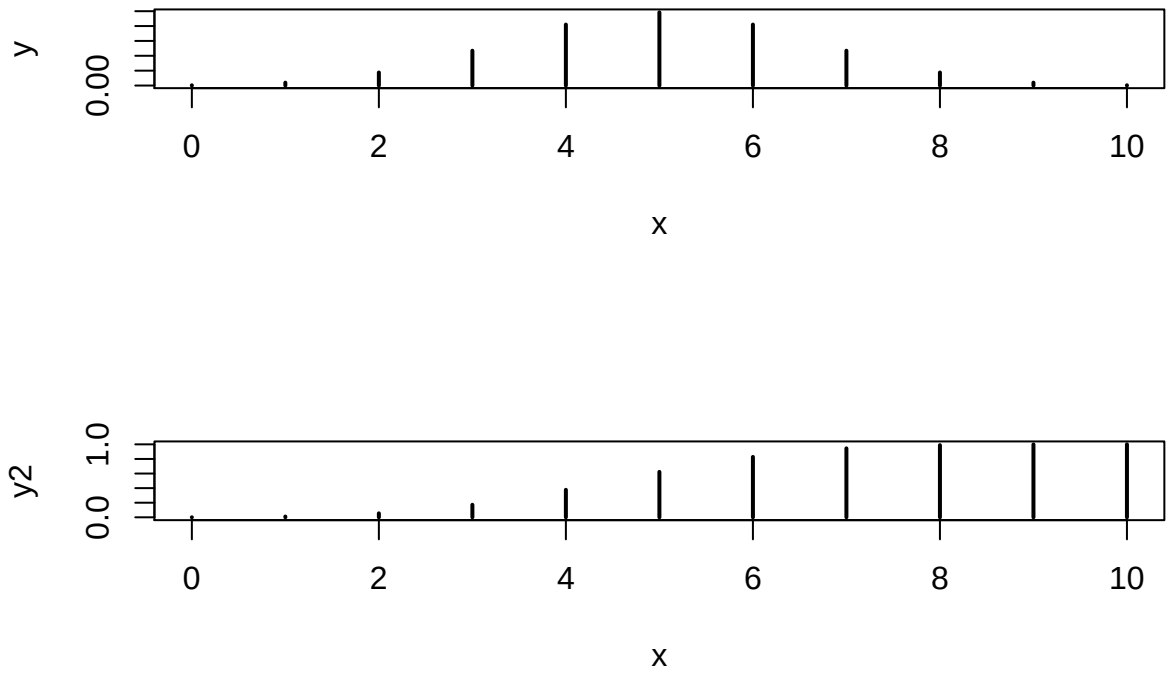
We see the PMF is symmetric around 5 successes. What is the exact probability of 1 success?

```
y[2] #if x is 0 - 10, the 2nd element is 1
```

```
## [1] 0.009765625
```

A *cumulative* mass function is the probability that a random variable is less than or equal to a certain value. Say a random variable X has a Binomial distribution with parameters $p = 0.5$ and $K = 10$. How much probability mass is below value x? More formally, what is $P(X <= x)$. This is perhaps best visualized.

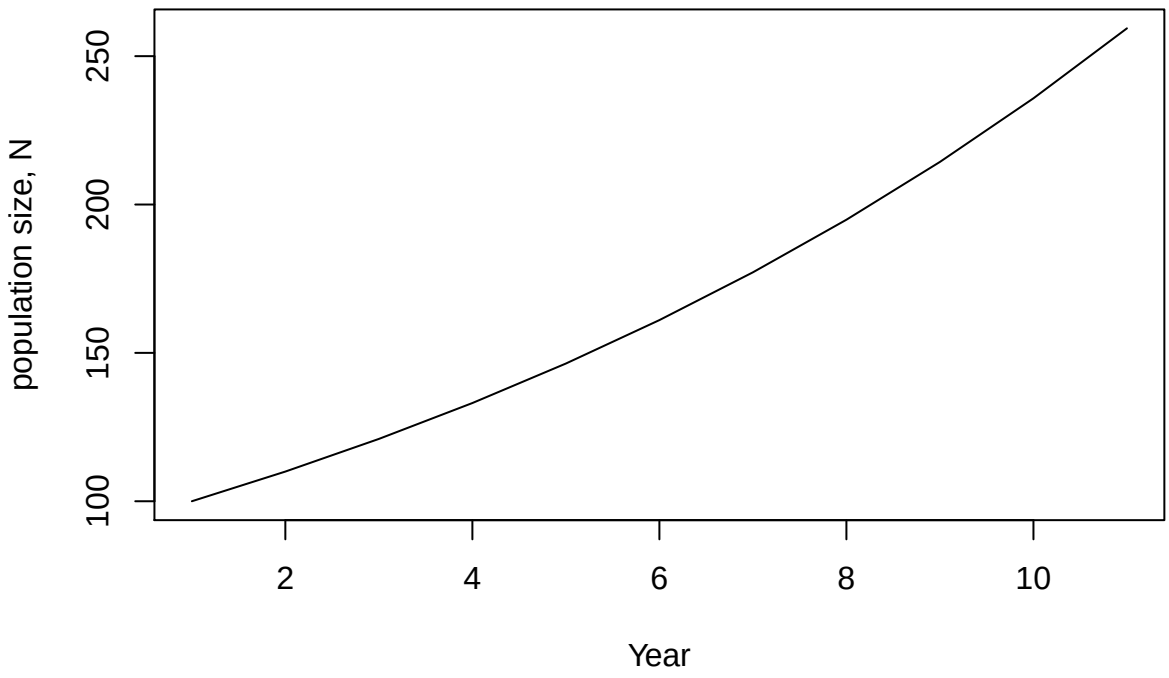
```
y2=pbinom(x,size=10,prob=0.5) #calculate the CMF values for 1 - 10 successes
par(mfrow=c(2,1)) #set plotting environment to plot 2 per screen
plot(y~x,type="h",lwd=2)
plot(y2~x,type="h",lwd=2)
```




```
par(mfrow=c(1,1)) #set plotting environment back to plot 1 per screen
```

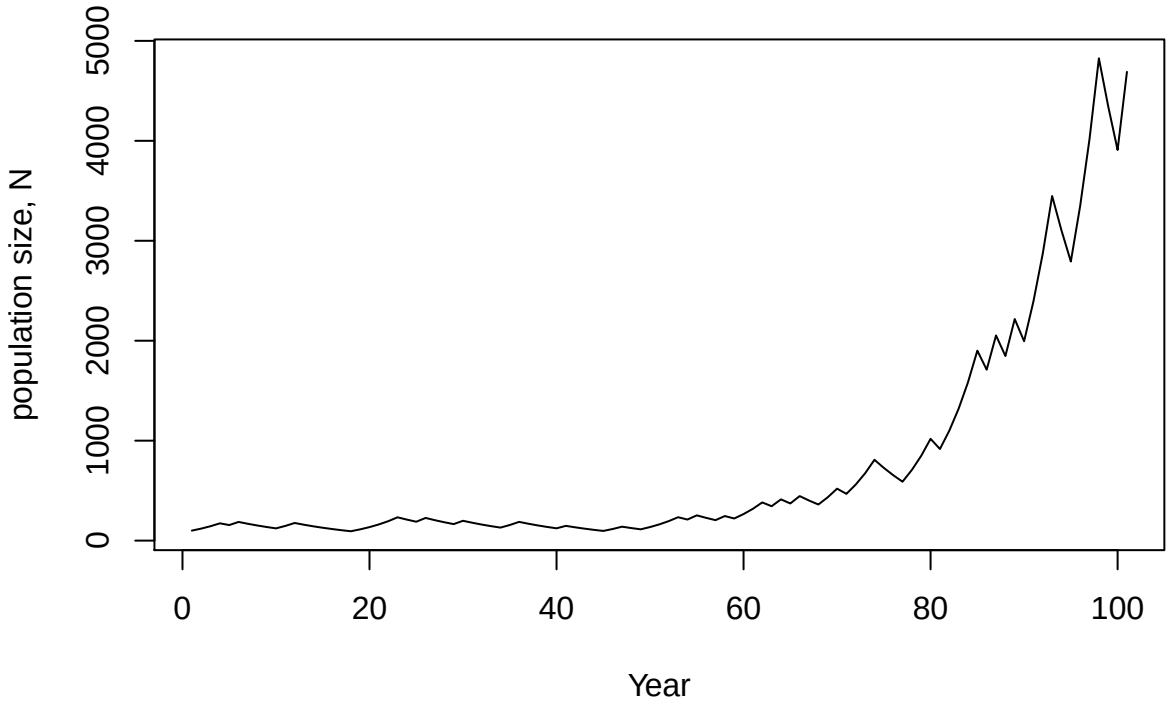
So that's cool, but what are some more interesting things we can do with probability in R? How about modeling population growth in the presence of environmental stochasticity? Say we have a population of size 100 that will undergo exponential population growth following $N_t = \lambda N_{t-1}$. First, let's look back at one of the exercises from Day 2. There, we wanted to predict the population size for 10 years, starting at $N = 100$, with a constant growth rate of $\lambda = 1.1$. How do we do that?

```
Nyears=11 #How many years will there be? Starting year plus 10 more
N=rep(NA,Nyears) #preallocate a vector to store the yearly population sizes
N[1]=100 #fill in the first value
lambda=1.1
for(t in 2:Nyears){
    N[t]=lambda*N[t-1]
}
plot(N,type="l",xlab="Year",ylab="population size, N")
```



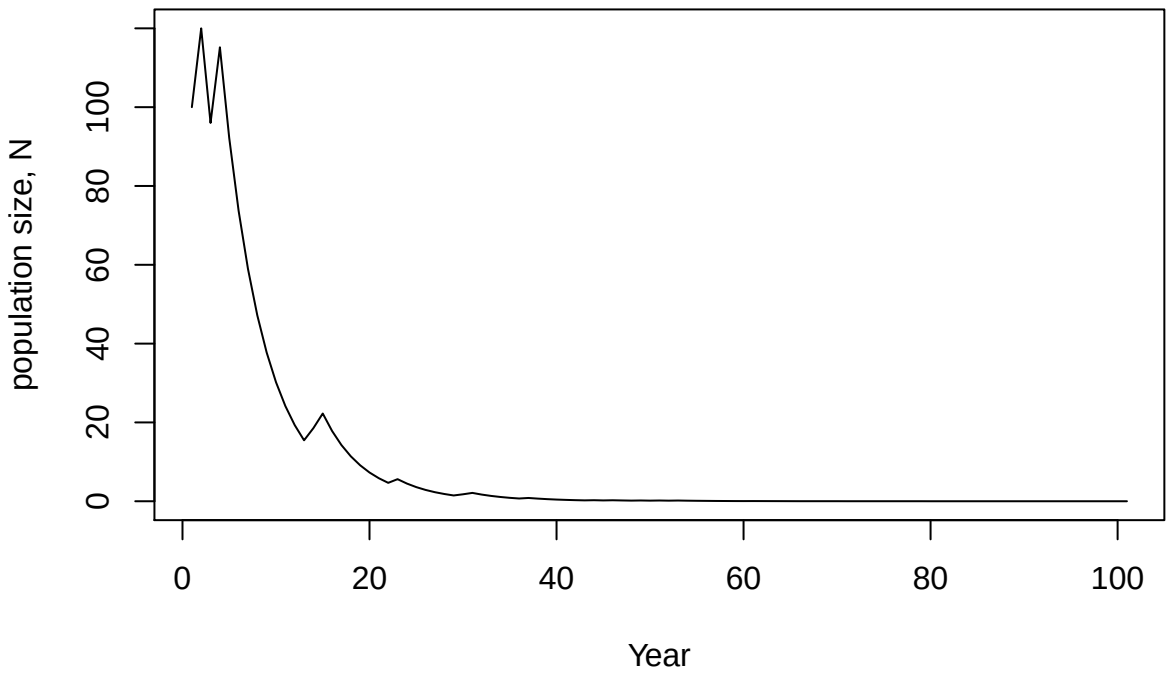
But now, let's consider that there are good and bad years (environmental stochasticity). In good years, $\lambda = 1.2$ and in bad years $\lambda = 0.9$. Good and bad years are equally likely. Let's project this population 100 years into the future.

```
Nyears=101 #How many years will there be? Starting year plus 10 more
N=rep(NA,Nyears) #preallocate a vector to store the yearly population sizes
N[1]=100 #fill in the first value
lambda_bad=0.9
lambda_good=1.2
for(t in 2:Nyears){
  if(runif(1)<0.5){ #randomly apply either condition with probability 0.5
    N[t]=lambda_bad*N[t-1]
  }else{
    N[t]=lambda_good*N[t-1]
  }
}
plot(N,type="l",xlab="Year",ylab="population size, N")
```



Finally, what will happen if the bad years occur with probability 0.75 and good years with probability 0.25? Also, let's say bad years are a little worse with $\lambda = 0.8$

```
Nyears=101 #How many years will there be? Starting year plus 10 more
N=rep(NA,Nyears) #preallocate a vector to store the yearly population sizes
N[1]=100 #fill in the first value
lambda_bad=0.8
lambda_good=1.2
for(t in 2:Nyears){
  if(runif(1)<0.75){ #now bad years happen with probability 0.75
    N[t]=lambda_bad*N[t-1]
  }else{
    N[t]=lambda_good*N[t-1]
  }
}
plot(N,type="l",xlab="Year",ylab="population size, N")
```



Summary Statistics

Now, let's switch gears and look at some summary statistics. We will work with the *CO2* data set in R, which we used for plotting.

```
data(CO2) #load the mtcars data set
CO2 #look at it
```

```
##    Plant        Type  Treatment conc uptake
## 1    Qn1      Quebec nonchilled   95   16.0
## 2    Qn1      Quebec nonchilled  175   30.4
## 3    Qn1      Quebec nonchilled  250   34.8
## 4    Qn1      Quebec nonchilled  350   37.2
## 5    Qn1      Quebec nonchilled  500   35.3
## 6    Qn1      Quebec nonchilled  675   39.2
## 7    Qn1      Quebec nonchilled 1000   39.7
## 8    Qn2      Quebec nonchilled   95   13.6
## 9    Qn2      Quebec nonchilled  175   27.3
## 10   Qn2      Quebec nonchilled  250   37.1
## 11   Qn2      Quebec nonchilled  350   41.8
## 12   Qn2      Quebec nonchilled  500   40.6
## 13   Qn2      Quebec nonchilled  675   41.4
## 14   Qn2      Quebec nonchilled 1000   44.3
## 15   Qn3      Quebec nonchilled   95   16.2
## 16   Qn3      Quebec nonchilled  175   32.4
```

```
## 17   Qn3      Quebec nonchilled  250   40.3
## 18   Qn3      Quebec nonchilled  350   42.1
## 19   Qn3      Quebec nonchilled  500   42.9
## 20   Qn3      Quebec nonchilled  675   43.9
## 21   Qn3      Quebec nonchilled 1000   45.5
## 22   Qc1      Quebec    chilled   95   14.2
## 23   Qc1      Quebec    chilled  175   24.1
## 24   Qc1      Quebec    chilled  250   30.3
## 25   Qc1      Quebec    chilled  350   34.6
## 26   Qc1      Quebec    chilled  500   32.5
## 27   Qc1      Quebec    chilled  675   35.4
## 28   Qc1      Quebec    chilled 1000   38.7
## 29   Qc2      Quebec    chilled   95    9.3
## 30   Qc2      Quebec    chilled  175   27.3
## 31   Qc2      Quebec    chilled  250   35.0
## 32   Qc2      Quebec    chilled  350   38.8
## 33   Qc2      Quebec    chilled  500   38.6
## 34   Qc2      Quebec    chilled  675   37.5
## 35   Qc2      Quebec    chilled 1000   42.4
## 36   Qc3      Quebec    chilled   95   15.1
## 37   Qc3      Quebec    chilled  175   21.0
## 38   Qc3      Quebec    chilled  250   38.1
## 39   Qc3      Quebec    chilled  350   34.0
## 40   Qc3      Quebec    chilled  500   38.9
## 41   Qc3      Quebec    chilled  675   39.6
## 42   Qc3      Quebec    chilled 1000   41.4
## 43   Mn1 Mississippi nonchilled   95   10.6
## 44   Mn1 Mississippi nonchilled  175   19.2
## 45   Mn1 Mississippi nonchilled  250   26.2
## 46   Mn1 Mississippi nonchilled  350   30.0
## 47   Mn1 Mississippi nonchilled  500   30.9
## 48   Mn1 Mississippi nonchilled  675   32.4
## 49   Mn1 Mississippi nonchilled 1000   35.5
## 50   Mn2 Mississippi nonchilled   95   12.0
## 51   Mn2 Mississippi nonchilled  175   22.0
## 52   Mn2 Mississippi nonchilled  250   30.6
## 53   Mn2 Mississippi nonchilled  350   31.8
## 54   Mn2 Mississippi nonchilled  500   32.4
## 55   Mn2 Mississippi nonchilled  675   31.1
## 56   Mn2 Mississippi nonchilled 1000   31.5
## 57   Mn3 Mississippi nonchilled   95   11.3
## 58   Mn3 Mississippi nonchilled  175   19.4
## 59   Mn3 Mississippi nonchilled  250   25.8
## 60   Mn3 Mississippi nonchilled  350   27.9
## 61   Mn3 Mississippi nonchilled  500   28.5
## 62   Mn3 Mississippi nonchilled  675   28.1
## 63   Mn3 Mississippi nonchilled 1000   27.8
## 64   Mc1 Mississippi    chilled   95   10.5
## 65   Mc1 Mississippi    chilled  175   14.9
## 66   Mc1 Mississippi    chilled  250   18.1
## 67   Mc1 Mississippi    chilled  350   18.9
## 68   Mc1 Mississippi    chilled  500   19.5
## 69   Mc1 Mississippi    chilled  675   22.2
## 70   Mc1 Mississippi    chilled 1000   21.9
```

```
## 71   Mc2 Mississippi    chilled   95    7.7
## 72   Mc2 Mississippi    chilled  175   11.4
## 73   Mc2 Mississippi    chilled  250   12.3
## 74   Mc2 Mississippi    chilled  350   13.0
## 75   Mc2 Mississippi    chilled  500   12.5
## 76   Mc2 Mississippi    chilled  675   13.7
## 77   Mc2 Mississippi    chilled 1000   14.4
## 78   Mc3 Mississippi    chilled   95   10.6
## 79   Mc3 Mississippi    chilled  175   18.0
## 80   Mc3 Mississippi    chilled  250   17.9
## 81   Mc3 Mississippi    chilled  350   17.9
## 82   Mc3 Mississippi    chilled  500   17.9
## 83   Mc3 Mississippi    chilled  675   18.9
## 84   Mc3 Mississippi    chilled 1000   19.9
```

```
str(CO2) #query its structure
```

```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':   84 obs. of  5 variables
##  $ Plant    : Ord.factor w/ 12 levels "Qn1"<"Qn2"<"Qn3"<..: 1 1 1 1 1 1 1 2 2 2 ...
##  $ Type     : Factor w/ 2 levels "Quebec","Mississippi": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Treatment: Factor w/ 2 levels "nonchilled","chilled": 1 1 1 1 1 1 1 1 1 1 ...
##  $ conc     : num  95 175 250 350 500 675 1000 95 175 250 ...
##  $ uptake   : num  16 30.4 34.8 37.2 35.3 39.2 39.7 13.6 27.3 37.1 ...
##  - attr(*, "formula")=Class 'formula'  language uptake ~ conc | Plant
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "outer")=Class 'formula'  language ~Treatment * Type
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "labels")=List of 2
##   ..$ x: chr "Ambient carbon dioxide concentration"
##   ..$ y: chr "CO2 uptake rate"
##  - attr(*, "units")=List of 2
##   ..$ x: chr "(uL/L)"
##   ..$ y: chr "(umol/m^2 s)"
```

```
?CO2 #look at help file for more description
```

We can calculate basic summary statistics like this:

```
mean(CO2$uptake) #mean CO2 uptake
```

```
## [1] 27.2131
```

```
median(CO2$uptake) #median
```

```
## [1] 28.3
```

```
max(CO2$uptake) #maximum value
```

```
## [1] 45.5
```

```r
min(CO2$uptake) #minimum value
```

```
## [1] 7.7
```

```r
sd(CO2$uptake) #standard deviation
```

```
## [1] 10.81441
```

```r
var(CO2$uptake) #variance
```

```
## [1] 116.9515
```

We can use the *summary* command to look at basic summary statistics for each variable:

```r
summary(CO2)
```

```
##      Plant             Type         Treatment       conc
##  Qn1    : 7   Quebec     :42   nonchilled:42   Min.   :  95
##  Qn2    : 7   Mississippi:42   chilled   :42   1st Qu.: 175
##  Qn3    : 7                                   Median : 350
##  Qc1    : 7                                   Mean   : 435
##  Qc3    : 7                                   3rd Qu.: 675
##  Qc2    : 7                                   Max.   :1000
##  (Other):42
##      uptake
##  Min.   : 7.70
##  1st Qu.:17.90
##  Median :28.30
##  Mean   :27.21
##  3rd Qu.:37.12
##  Max.   :45.50
##
```

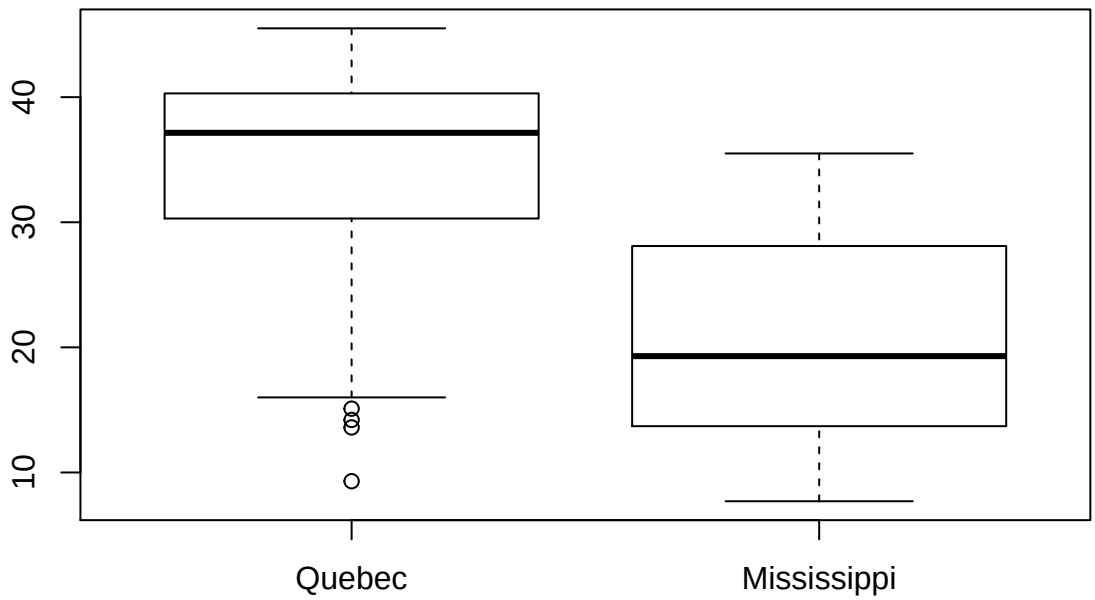
We can calculate all the pairwise correlations between variables:

```r
#In this case, only concentration and uptake are continuous data, the 4th and 5th column
cor(CO2[,4:5])
```

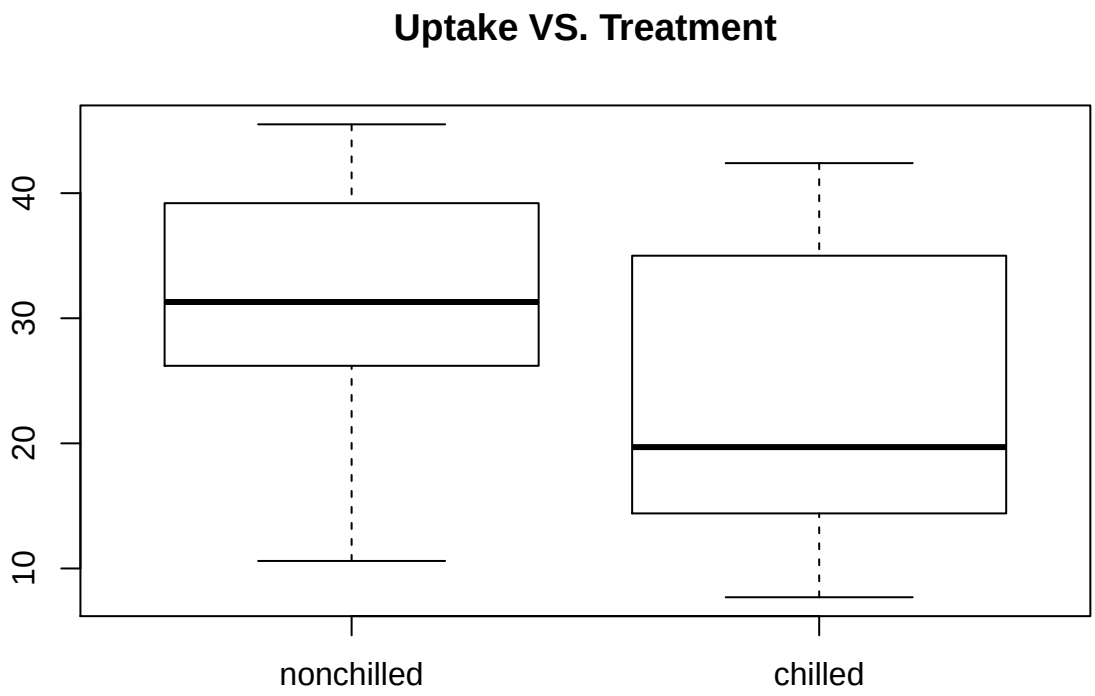
```
##             conc    uptake
## conc   1.0000000 0.4851774
## uptake 0.4851774 1.0000000
```

Yesterday, we looked at some plots for the CO uptake at the two sites by treatment:

```r
boxplot(uptake ~ Type, data = CO2, main = "Uptake VS. Type")
```




```
boxplot(uptake ~ Treatment, data = CO2, main = "Uptake VS. Treatment")
```



Do you think CO2 uptake varies by type? What about by treatment? How would we measure the difference? How about the mean? Let's subset the data into each group and compare the means.

```
#subset out the data for Quebec
Q=CO2[CO2$Type=="Quebec",]
#subset out the data for Mississippi
M=CO2[CO2$Type=="Mississippi",]
#subset out the data for chilled
chill=CO2[CO2$Treatment=="chilled",]
#subset out the data for nonchilled
nonchill=CO2[CO2$Treatment=="nonchilled",]

#compare means of Quebec and Mississippi
mean(Q$uptake)
```

```
## [1] 33.54286
```

```
mean(M$uptake)
```

```
## [1] 20.88333
```

```
#compare means of chilled and nonchilled
mean(nonchill$uptake)
```

```
## [1] 30.64286
```

```r
mean(chill$uptake)
```

```
## [1] 23.78333
```

The means are different, but is this just sampling variation, or is there evidence that the population parameters actually differ?

Basic Inferential Statistics

We can use t-tests to quantify the evidence for a difference.

```r
t.test(Q$uptake,M$uptake)
```

```
##
##  Welch Two Sample t-test
##
## data:  Q$uptake and M$uptake
## t = 6.5969, df = 78.533, p-value = 4.451e-09
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   8.839475 16.479572
## sample estimates:
## mean of x mean of y
##  33.54286  20.88333
```

```r
t.test(chill$uptake,nonchill$uptake)
```

```
##
##  Welch Two Sample t-test
##
## data:  chill$uptake and nonchill$uptake
## t = -3.0485, df = 80.945, p-value = 0.003107
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -11.336682  -2.382366
## sample estimates:
## mean of x mean of y
##  23.78333  30.64286
```

```r
?t.test #let's look at the assumptions we are making for these 2 tests
```

We can also use a linear model to quantify the evidence for a difference. This will do an analysis of variance (ANOVA). We do this using the *lm()* function. We will be estimating the parameters of $uptake_i = \beta_0 + \beta_1 Type_i + \epsilon$ and $uptake_i = \beta_0 + \beta_1 Treatment_i + \epsilon$. We assume $\epsilon \sim Normal(0, \sigma^2)$. We are testing the null hypotheses $H_0 : \beta_1 = 0$. If the p-value is low enough, we reject the null in favor of the alternative $H_a : \beta_1 \neq 0$ (not equal to).

In linear models with categorical predictors, we typically make one level the intercept. This decision is arbitrary. In R, the first level is automatically made the intercept.

```r
levels(CO2$Type)
```

```
## [1] "Quebec"      "Mississippi"
```

```r
levels(CO2$Treatment)
```

```
## [1] "nonchilled" "chilled"
```

```r
#We can reassign levels however we like
CO2$Type=relevel(CO2$Type,ref="Mississippi") #Make Mississippi the reference category
```

Now, let's fit these two linear models and compare the results to the t-test.

```r
mod1=lm(uptake~Type,data=CO2)
mod2=lm(uptake~Treatment,data=CO2)
summary(mod1)
```

```
##
## Call:
## lm(formula = uptake ~ Type, data = CO2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.243  -6.243   1.187   7.027  14.617
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.883      1.357  15.390  < 2e-16 ***
## TypeQuebec    12.660      1.919   6.597 3.83e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.794 on 82 degrees of freedom
## Multiple R-squared:  0.3467, Adjusted R-squared:  0.3387
## F-statistic: 43.52 on 1 and 82 DF,  p-value: 3.835e-09
```

```r
summary(mod2)
```

```
##
## Call:
## lm(formula = uptake ~ Treatment, data = CO2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.0429  -8.6530  -0.4429   9.7321  18.6167
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        30.643      1.591  19.259   <2e-16 ***
## Treatmentchilled   -6.860      2.250  -3.048   0.0031 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.31 on 82 degrees of freedom
## Multiple R-squared:  0.1018, Adjusted R-squared:  0.09084
## F-statistic: 9.293 on 1 and 82 DF,  p-value: 0.003096
```

```
#How do the p-values compare to the t-tests? Why do they differ?
t.test(Q$uptake,M$uptake,var.equal=TRUE) #set var.equal to TRUE
```

```
##
##  Two Sample t-test
##
## data:  Q$uptake and M$uptake
## t = 6.5969, df = 82, p-value = 3.835e-09
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   8.84200 16.47705
## sample estimates:
## mean of x mean of y
##  33.54286  20.88333
```

```
t.test(chill$uptake,nonchill$uptake,var.equal=TRUE) #set var.equal to TRUE
```

```
##
##  Two Sample t-test
##
## data:  chill$uptake and nonchill$uptake
## t = -3.0485, df = 82, p-value = 0.003096
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -11.33581  -2.38324
## sample estimates:
## mean of x mean of y
##  23.78333  30.64286
```

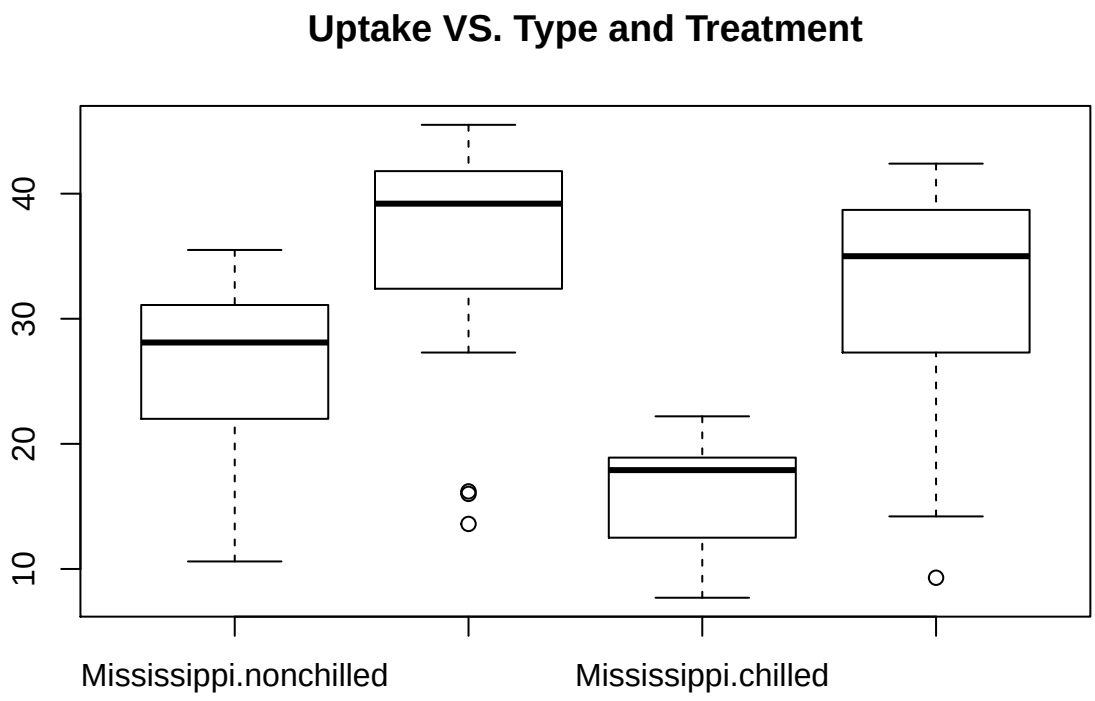
```
#A t-test with equal variances is equivalent to an ANOVA, which assumes
#equal variances by default

#But treatment could be confounded by type. Let's put both in the model.
#Here, we can test for a treatment effect controlling for type
mod3=lm(uptake~Treatment+Type,data=CO2)
summary(mod3)
```

```
##
## Call:
## lm(formula = uptake ~ Treatment + Type, data = CO2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.373  -4.658   1.967   5.747  12.287
##
```

```
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         24.313      1.536  15.825  < 2e-16 ***
## Treatmentchilled    -6.860      1.774  -3.867 0.000222 ***
## TypeQuebec          12.660      1.774   7.136 3.68e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.13 on 81 degrees of freedom
## Multiple R-squared:  0.4485, Adjusted R-squared:  0.4349
## F-statistic: 32.94 on 2 and 81 DF,  p-value: 3.407e-11
```

So we have evidence that CO2 uptake varies by both type and treatment. But does it vary by Treatment the same for each type? Let's look at a boxplot:

```
boxplot(uptake ~ Type+Treatment, data = CO2, main = "Uptake VS. Type and Treatment")
```



It looks like the treatment *might* have a larger effect for the Mississippi type. We can test this using a linear model with an *interaction* term. We last fit the model $uptake_i = \beta_0 + \beta_1 Treatment_i + \beta_2 Type_i$. The model with an interaction is $uptake_i = \beta_0 + \beta_1 Treatment_i + \beta_2 Type_i + \beta_3 Treatment_i Type_i$. The null hypothesis of no interaction is $H_0 : \beta_3 = 0$ and the alternative is $H_0 : \beta_3 \neq 0$

```
mod4=lm(uptake~Treatment*Type,data=CO2) #multiply instead of add for an interaction
summary(mod4)
```

```
##
```

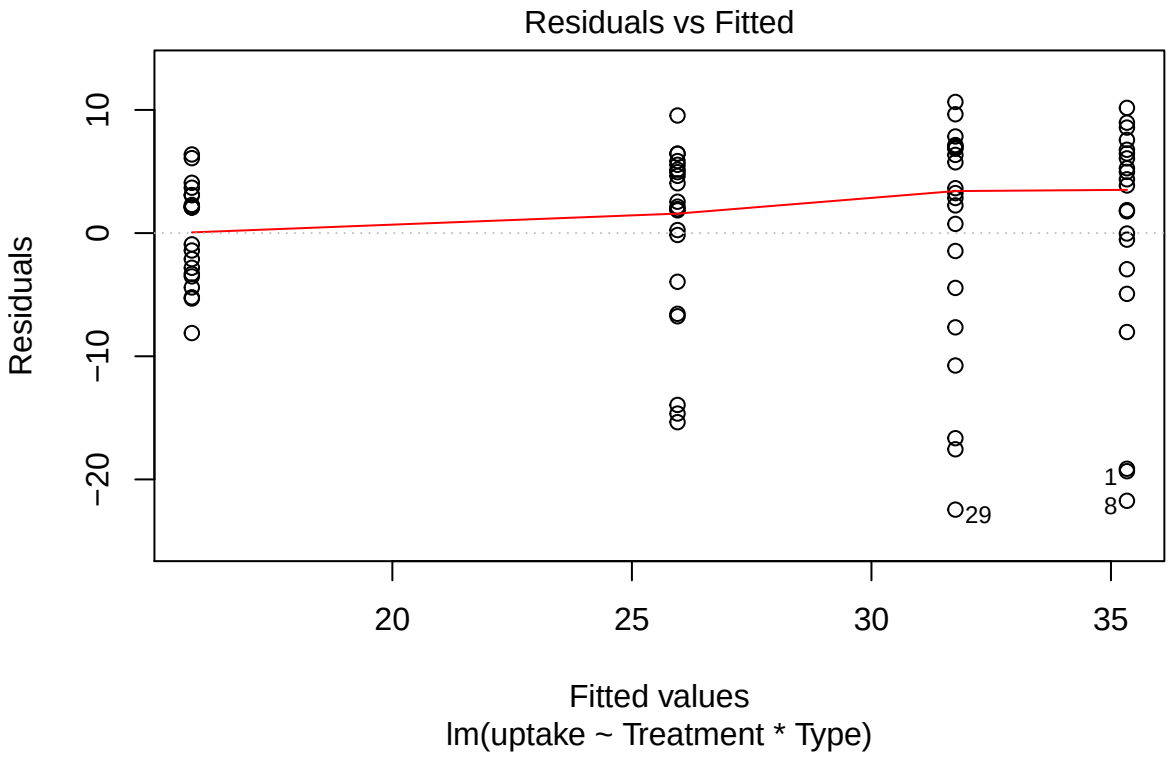
```
## Call:
## lm(formula = uptake ~ Treatment * Type, data = CO2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.452  -3.624   2.167   5.773  10.648
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  25.952      1.747  14.855  < 2e-16 ***
## Treatmentchilled            -10.138      2.471  -4.103 9.74e-05 ***
## TypeQuebec                    9.381      2.471   3.797 0.000284 ***
## Treatmentchilled:TypeQuebec   6.557      3.494   1.877 0.064213 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.006 on 80 degrees of freedom
## Multiple R-squared:  0.4718, Adjusted R-squared:  0.452
## F-statistic: 23.82 on 3 and 80 DF,  p-value: 4.106e-11
```

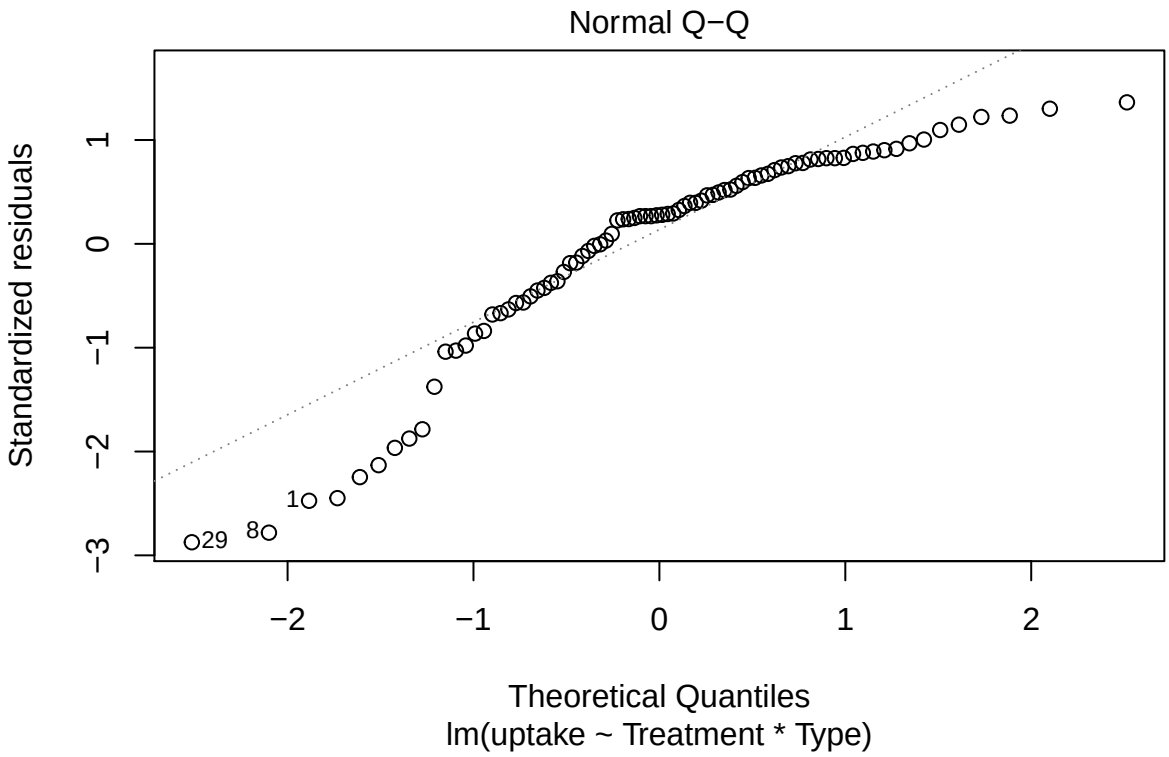
```
AIC(mod1,mod2,mod3,mod4) #We can compare modes via their AIC values, too. Lower is better.
```

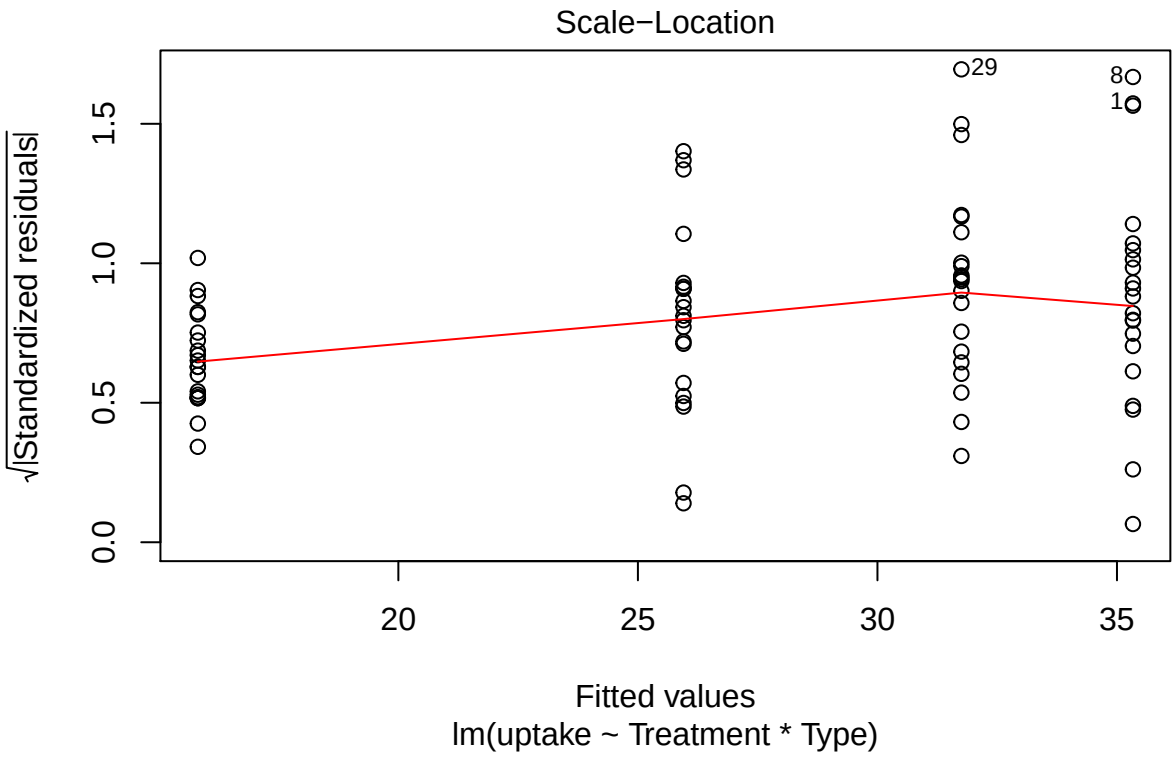
```
##      df      AIC
## mod1  3 607.6014
## mod2  3 634.3456
## mod3  4 595.3728
## mod4  5 593.7540
```

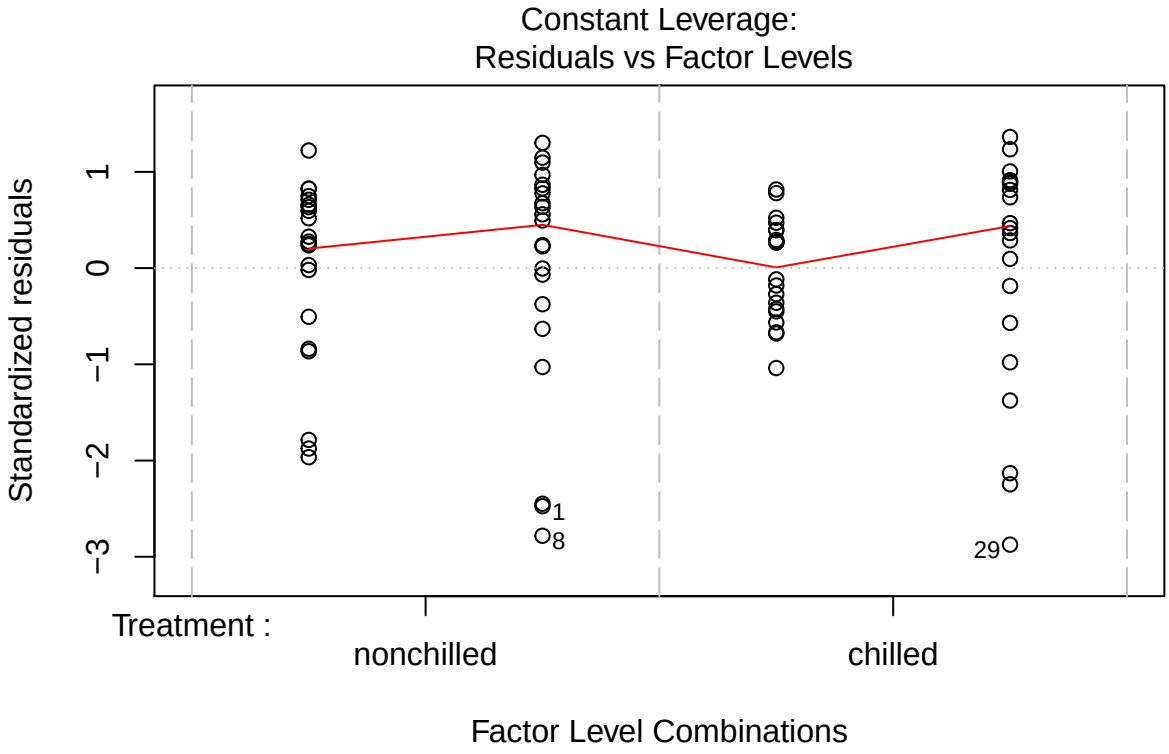
We can also look at residual plots to see if there is evidence that we are violating any assumptions of ANOVA.

```
plot(mod4)
```


Residuals vs Fitted
Residuals
10
0
−10
−20
20
25
30
35
29
1
8
Fitted values
lm(uptake ~ Treatment * Type)

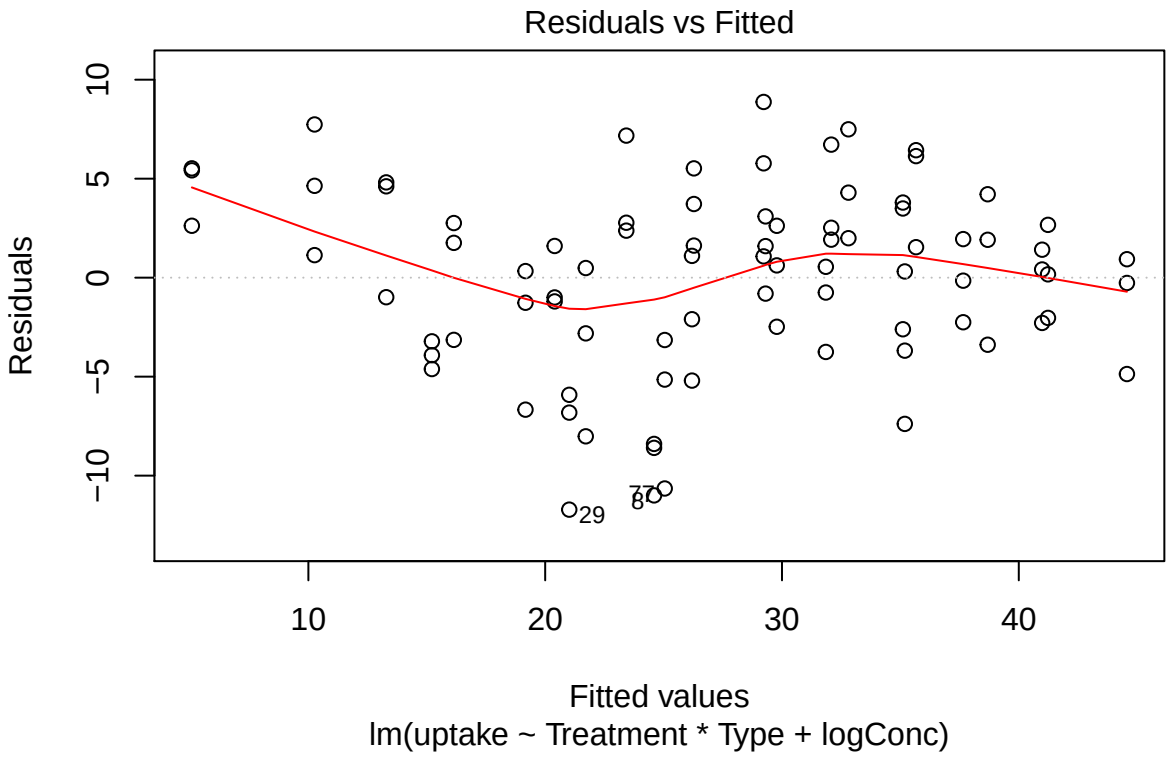

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(uptake ~ Treatment * Type)
29
8
1

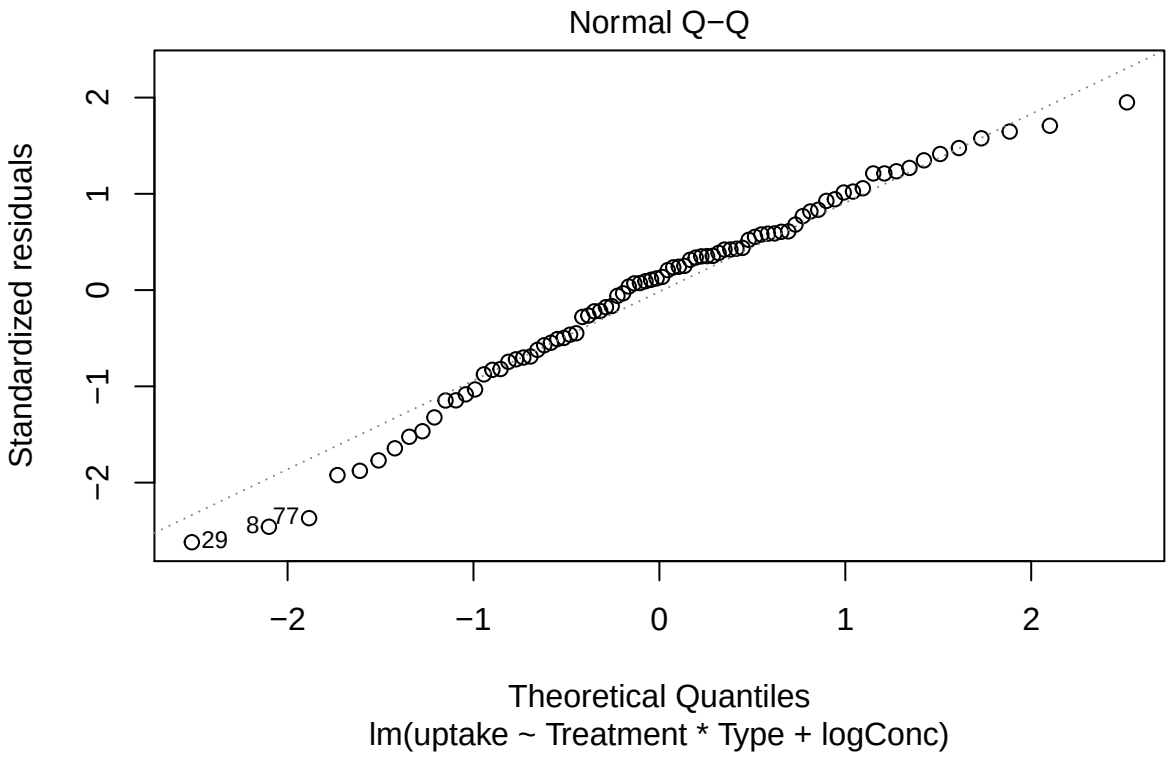

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
20
25
30
35
29
8
1
Fitted values
lm(uptake ~ Treatment * Type)

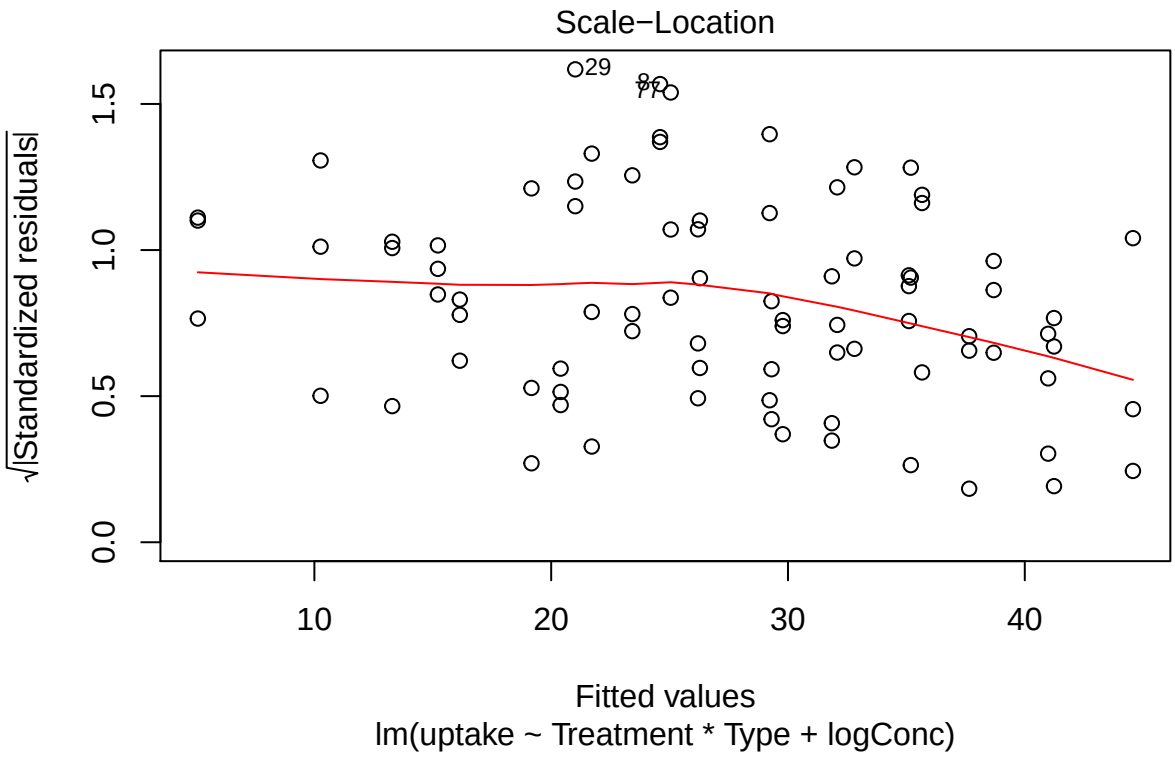


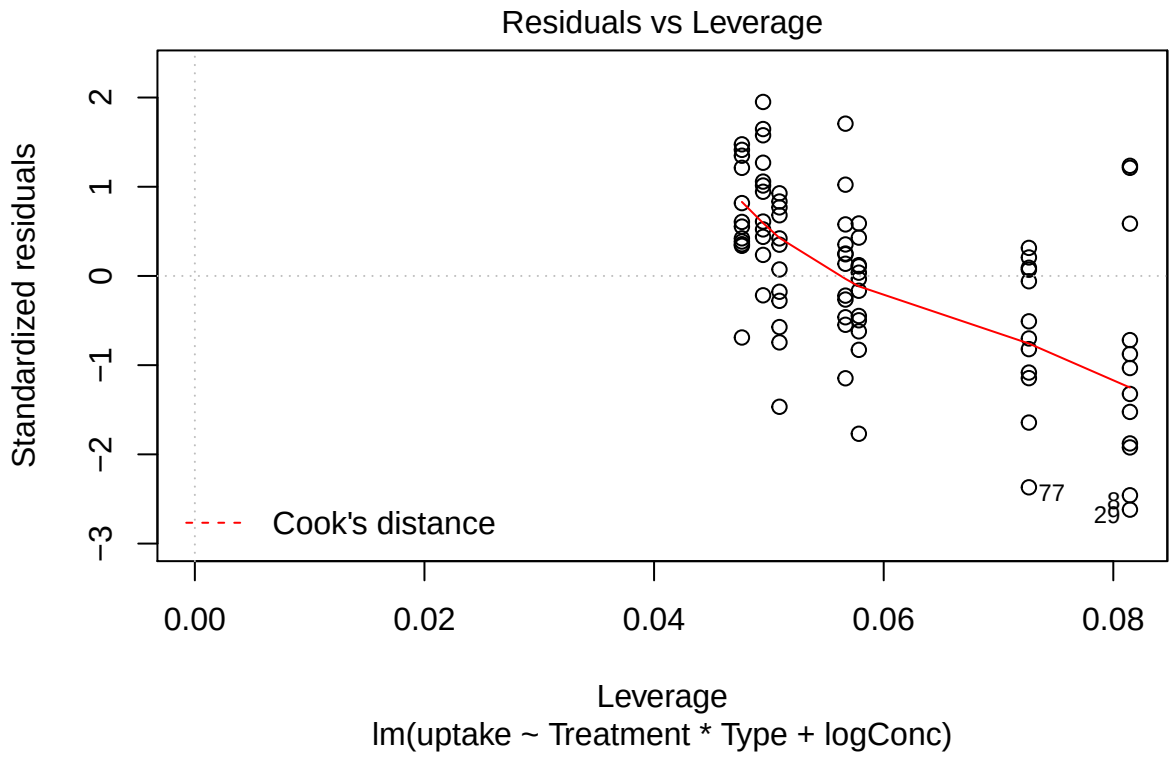
The first 2 plots indicate that the variance increases with the mean. This could be due to a missing covariate. Let's try adding the *concentration* variable.

```
CO2$logConc=log(CO2$conc)
mod5=lm(uptake~Treatment*Type+logConc,data=CO2) #we'll add concentration on the log scale *waives hands
plot(mod5) #still not perfect, but much better.
```


Residuals vs Fitted
Residuals
10
5
0
−5
−10
10
20
30
40
29
77
8
Fitted values
lm(uptake ~ Treatment * Type + logConc)


Normal Q−Q
Standardized residuals
2
1
0
−1
−2
29
8
77
−2
−1
0
1
2
Theoretical Quantiles
lm(uptake ~ Treatment * Type + logConc)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
10
20
30
40
Fitted values
lm(uptake ~ Treatment * Type + logConc)




```
summary(mod5) #The concentration effect is highly significant. Now, so is the interaction between
```

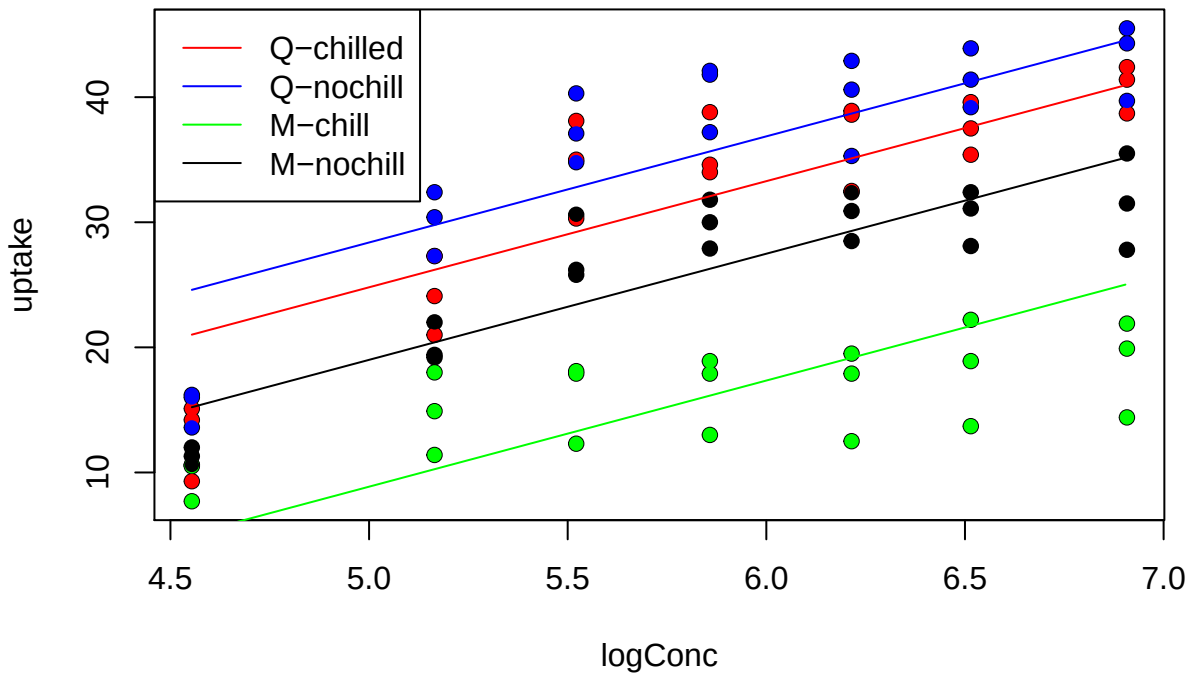
```
##
## Call:
## lm(formula = uptake ~ Treatment * Type + logConc, data = CO2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.7166  -2.8960   0.5837   2.7621   8.8745
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -23.4179     4.0768  -5.744 1.65e-07 ***
## Treatmentchilled           -10.1381     1.4403  -7.039 6.29e-10 ***
## TypeQuebec                   9.3810     1.4403   6.513 6.26e-09 ***
## logConc                      8.4839     0.6783  12.507  < 2e-16 ***
## Treatmentchilled:TypeQuebec  6.5571     2.0368   3.219  0.00187 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.667 on 79 degrees of freedom
## Multiple R-squared:  0.8227, Adjusted R-squared:  0.8138
## F-statistic: 91.67 on 4 and 79 DF,  p-value: < 2.2e-16
```

```
#treatment and uptake.
AIC(mod4,mod5) #The new model is much better as judged by AIC as well.
```

```
##      df      AIC
## mod4  5 593.7540
## mod5  6 504.0333
```

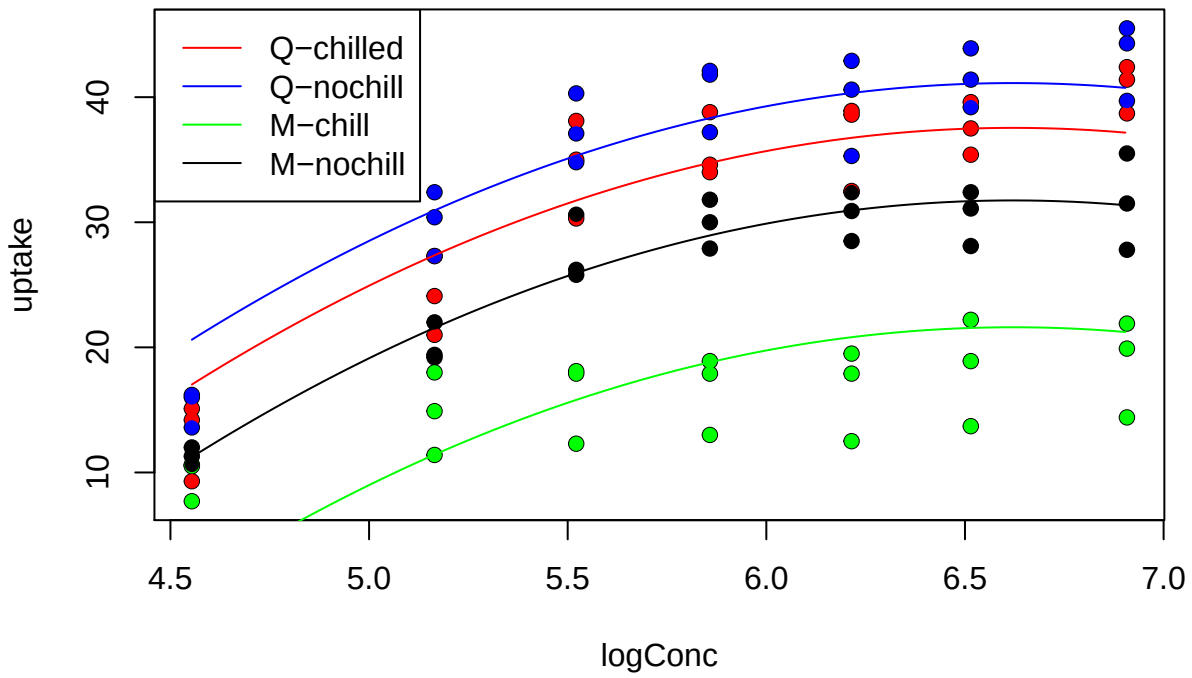
Finally, we can plot the fitted model with a little bit of work. We won't look too deeply at this code now.

```
logConc=seq(min(CO2$logConc),max(CO2$logConc),0.01)
newdata1=data.frame(Type="Quebec",Treatment="chilled",logConc=logConc)
newdata2=data.frame(Type="Quebec",Treatment="nonchilled",logConc=logConc)
newdata3=data.frame(Type="Mississippi",Treatment="chilled",logConc=logConc)
newdata4=data.frame(Type="Mississippi",Treatment="nonchilled",logConc=logConc)
predict1=predict(mod5,newdata=newdata1)
predict2=predict(mod5,newdata=newdata2)
predict3=predict(mod5,newdata=newdata3)
predict4=predict(mod5,newdata=newdata4)

plot(uptake~logConc,data=CO2)
points(uptake~logConc,data=CO2[CO2$Type=="Quebec"&CO2$Treatment=="chilled",],col="red",pch=16)
points(uptake~logConc,data=CO2[CO2$Type=="Quebec"&CO2$Treatment=="nonchilled",],col="blue",pch=16)
points(uptake~logConc,data=CO2[CO2$Type=="Mississippi"&CO2$Treatment=="chilled",],col="green",pch=16)
points(uptake~logConc,data=CO2[CO2$Type=="Mississippi"&CO2$Treatment=="nonchilled",],col="black",pch=16)
lines(predict1~logConc,col="red")
lines(predict2~logConc,col="blue")
lines(predict3~logConc,col="green")
lines(predict4~logConc,col="black")
legend("topleft",legend=c("Q-chilled","Q-nochill","M-chill","M-nochill"),
                          lty=c(1,1,1,1),col=c("red","blue","green","black"))
```



We see that the relationship between log(concentration) and uptake is not exactly linear. Let's try adding a quadratic term for log(concentration).

```
mod6=lm(uptake~Treatment*Type+poly(logConc,2),data=CO2) #poly(x,2) adds a quadratic term
#same plotting code below
logConc=seq(min(CO2$logConc),max(CO2$logConc),0.01)
newdata1=data.frame(Type="Quebec",Treatment="chilled",logConc=logConc)
newdata2=data.frame(Type="Quebec",Treatment="nonchilled",logConc=logConc)
newdata3=data.frame(Type="Mississippi",Treatment="chilled",logConc=logConc)
newdata4=data.frame(Type="Mississippi",Treatment="nonchilled",logConc=logConc)
predict1=predict(mod6,newdata=newdata1)
predict2=predict(mod6,newdata=newdata2)
predict3=predict(mod6,newdata=newdata3)
predict4=predict(mod6,newdata=newdata4)

plot(uptake~logConc,data=CO2)
points(uptake~logConc,data=CO2[CO2$Type=="Quebec"&CO2$Treatment=="chilled",],col="red",pch=16)
points(uptake~logConc,data=CO2[CO2$Type=="Quebec"&CO2$Treatment=="nonchilled",],col="blue",pch=16)
points(uptake~logConc,data=CO2[CO2$Type=="Mississippi"&CO2$Treatment=="chilled",],col="green",pch=16)
points(uptake~logConc,data=CO2[CO2$Type=="Mississippi"&CO2$Treatment=="nonchilled",],col="black",pch=16)
lines(predict1~logConc,col="red")
lines(predict2~logConc,col="blue")
lines(predict3~logConc,col="green")
lines(predict4~logConc,col="black")
legend("topleft",legend=c("Q-chilled","Q-nochill","M-chill","M-nochill"),
                         lty=c(1,1,1,1),col=c("red","blue","green","black"))
```



This looks better, but still not perfect. Our inferential statistics support this observation:

```
summary(mod6)
```

```
##
## Call:
## lm(formula = uptake ~ Treatment * Type + poly(logConc, 2), data = CO2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3110 -2.2113 -0.0064  2.1886  9.5059
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  25.9524     0.8357  31.055  < 2e-16 ***
## Treatmentchilled            -10.1381     1.1819  -8.578 7.11e-13 ***
## TypeQuebec                    9.3810     1.1819   7.937 1.24e-11 ***
## poly(logConc, 2)1            58.3687     3.8297  15.241  < 2e-16 ***
## poly(logConc, 2)2           -24.0150     3.8297  -6.271 1.85e-08 ***
## Treatmentchilled:TypeQuebec   6.5571     1.6714   3.923 0.000187 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.83 on 78 degrees of freedom
## Multiple R-squared:  0.8822, Adjusted R-squared:  0.8746
## F-statistic: 116.8 on 5 and 78 DF,  p-value: < 2.2e-16
```

```r
AIC(mod5,mod6)
```

```
##      df      AIC
## mod5  6 504.0333
## mod6  7 471.7426
```

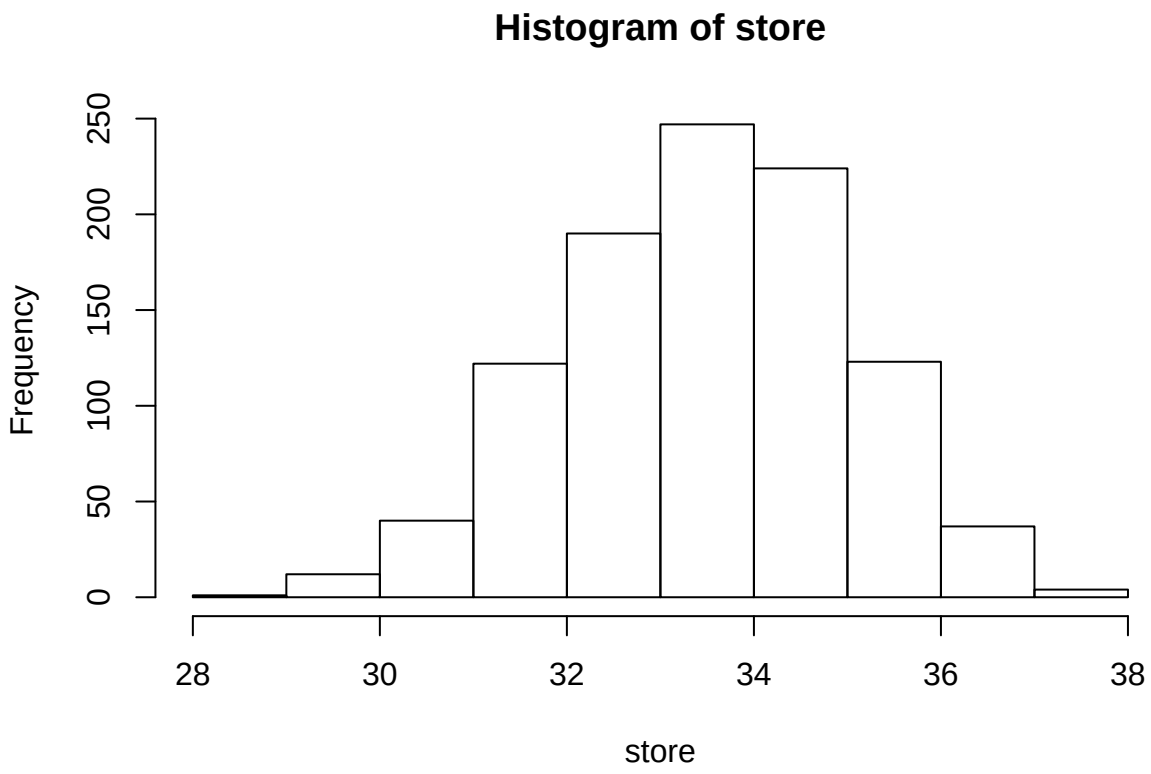
Here is one final thing to look at, if we have time. We can use the *t-test()* command to test if the mean of one variable is different from zero. We can also get a confidence interval. Let's look at the CO2 uptake levels for Quebec.

```r
t.test(Q$uptake)
```

```
##
##  One Sample t-test
##
## data:  Q$uptake
## t = 22.471, df = 41, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  30.52828 36.55743
## sample estimates:
## mean of x
##  33.54286
```

What assumptions does this t-test make? One is that the data are normally distributed. How could we get a p-value and confidence interval without this assumption? One way is to use *randomization methods*, specifically non-parametric *bootstrapping*. The idea is that if we have a random sample of data, we can randomly resample the data to see how *statistics* behave due to random variation. Let's try bootstrapping the sample mean.

```r
Niter=1000 #How many times to resample the data
store=rep(NA,Niter) #preallocate a vector to store the mean
for(i in 1:Niter){
  newdata=sample(Q$uptake,length(Q$uptake),replace=TRUE) #resample the data
  store[i]=mean(newdata) #store the mean of the resampled data
}

hist(store)
```




```
#what is the mean of the resampled data sets?
mean(store)
```

```
## [1] 33.49388
```

```
#this is pretty close to the mean of the actual data
mean(Q$uptake)
```

```
## [1] 33.54286
```

```
#how might we calculate a p-value from the bootstrap distribution of the sample mean?
mean(store<=0)
```

```
## [1] 0
```

```
#how might we get a confidence interval from the bootstrap distribution of the sample mean?
quantile(store,c(0.025,0.975))
```

```
##     2.5%    97.5%
## 30.38524 36.23815
```

```
#how does this compare to the confidence interval from the t-test?
```

Exercises

1. Load the built in data set *PlantGrowth*.

```
data(PlantGrowth)
```

1a. Describe this data set. What is each variable and how many observations are there? How many per treatment group? What is the mean, maximum, and minimum value for each treatment group (use R functions to find these)?

1b. Use the *t.test()* function to see if there is evidence that weight for each treatment group differs from the control group.

1c. Use the *lm()* function to do the same.

2. Load the built in data set *RatPupWeight*. It is in the *nlme* package, so we need to load that first.

```
library(nlme)
data(RatPupWeight)
```

2a. Describe this data set. Make some plots to see if weight appears associated with any predictors.

2b. Use the *lm()* function to see if there is evidence that weight for each treatment group differs from the control group. Do we need to control for other confounders? If so, does the inference about treatment effects change?

3. Load the built in data set *mtcars*.

```
data(mtcars)
```

3a. Describe this data set. Which variable is the response? Make some plots of the predictors vs. the response variable.

3b. Use the *lm()* function to see which predictors are associated with vehicle miles per gallon.

4. Earlier, we looked at population growth that followed the equation $N_t = \lambda N_{t-1}$. We considered that the population growth rate λ varied between good and bad years, which occurred with equal probability. But what if we allow more variation in λ? Modify the previous code (or start from scratch) to vary lambda each year following a Normal random variable with mean 1.1 and standard deviation 0.1. Store the randomly-generated values for λ and plot their histogram. Also, plot a simulated population trajectory.