HW1

lisa rosenthal

10/13/2017

Question 1

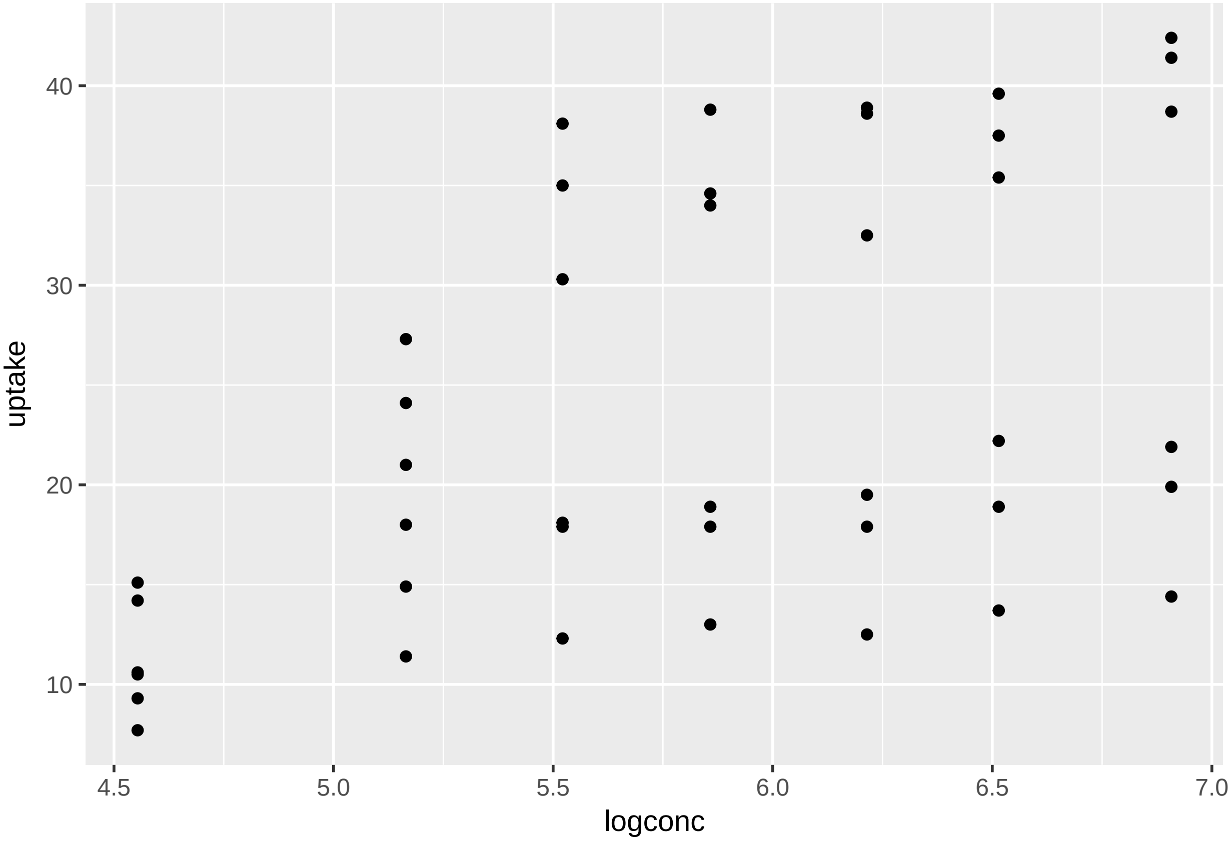
Load the data set "CO2_HW1.txt", which describes the CO2 uptake rates of plants of the grass species Echinochloa crus-galli from Quebec and Mississippi.

```
CO2 <- read.table("~/Desktop/Stat Model class F2017/PLS298_git/Homework1_Lisa/CO2_HW1.txt",header=T)
```

Using a linear model for the analysis, investigate these questions:

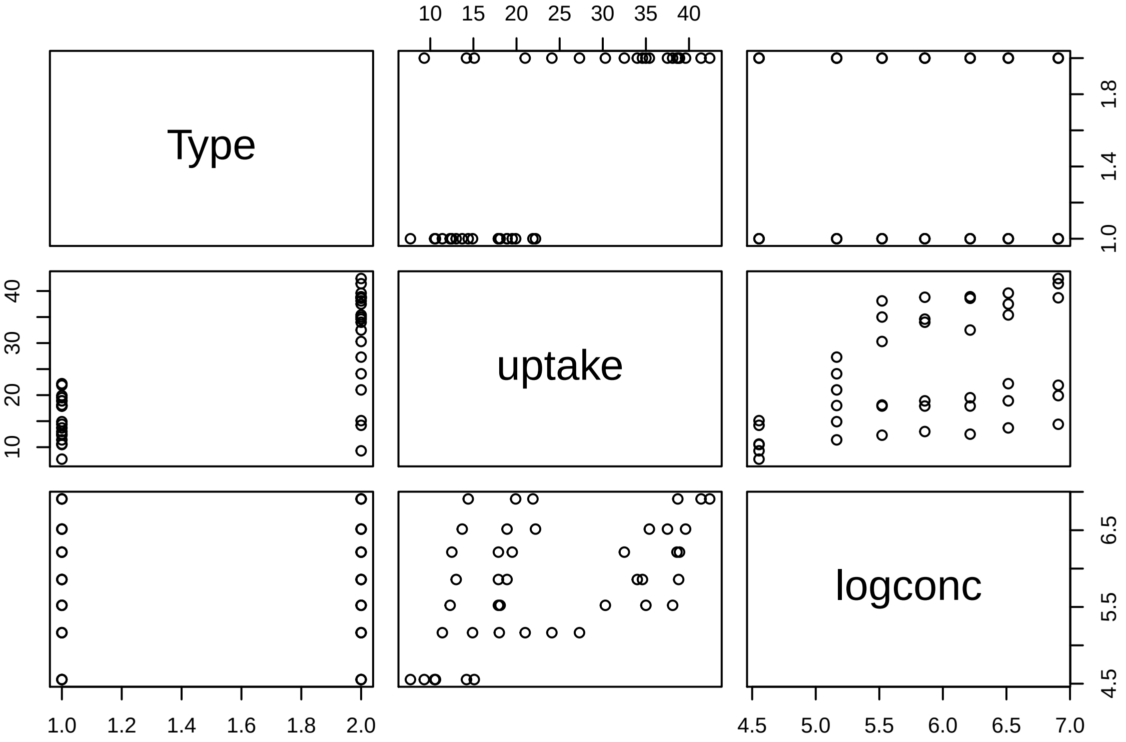
How does the air concentration of CO2 ("logconc") affect a grass plant's CO2 uptake rate ("uptake")?

```
#uptake increases with logconc, but clearly there are 2 groups (type)
ggplot(data = CO2, aes(logconc, uptake))+
        geom_point()
```

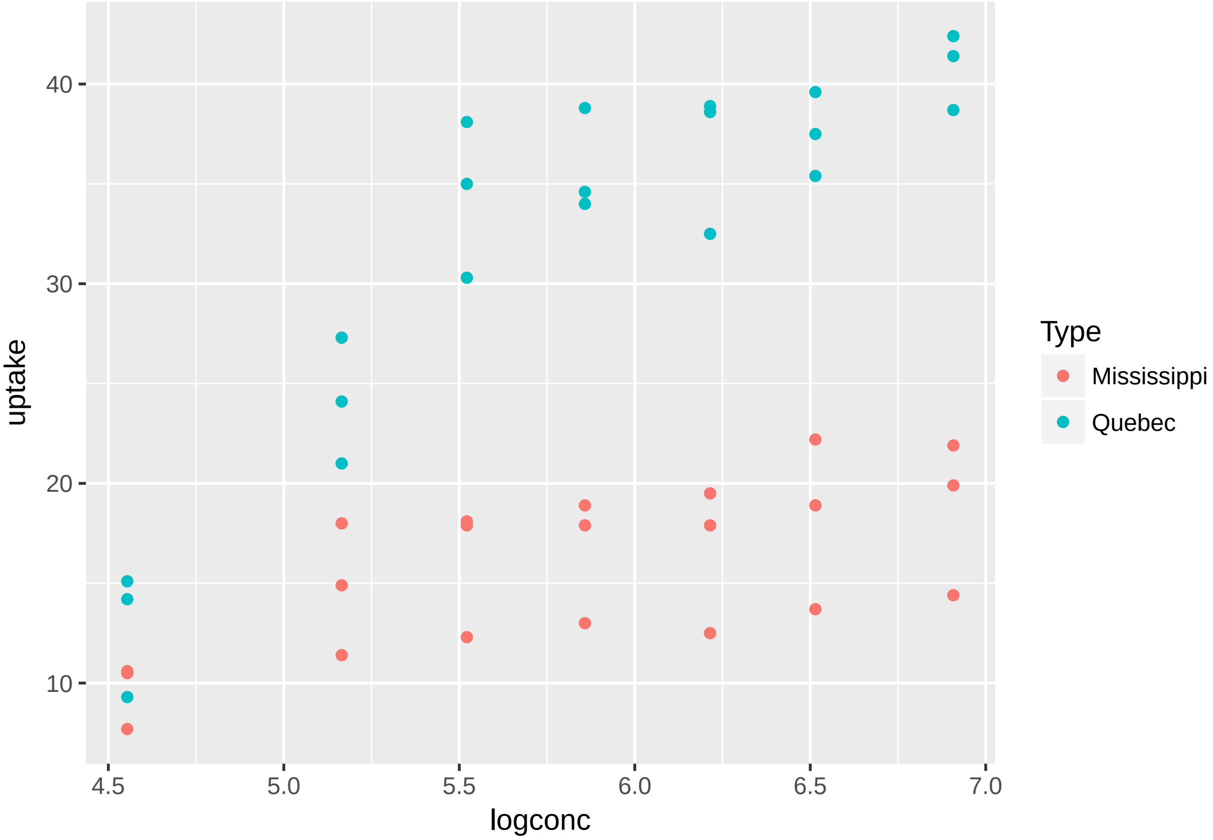


Does this effect depend on the origin of the plant ("Type")? In your answer, include some information on: What transformations if any you made on the data and why. What steps you took to check model assumptions and model performance. What the coefficients of the model are and how you interpret them.

```
pairs(CO2) #shows that the data is different between types
```

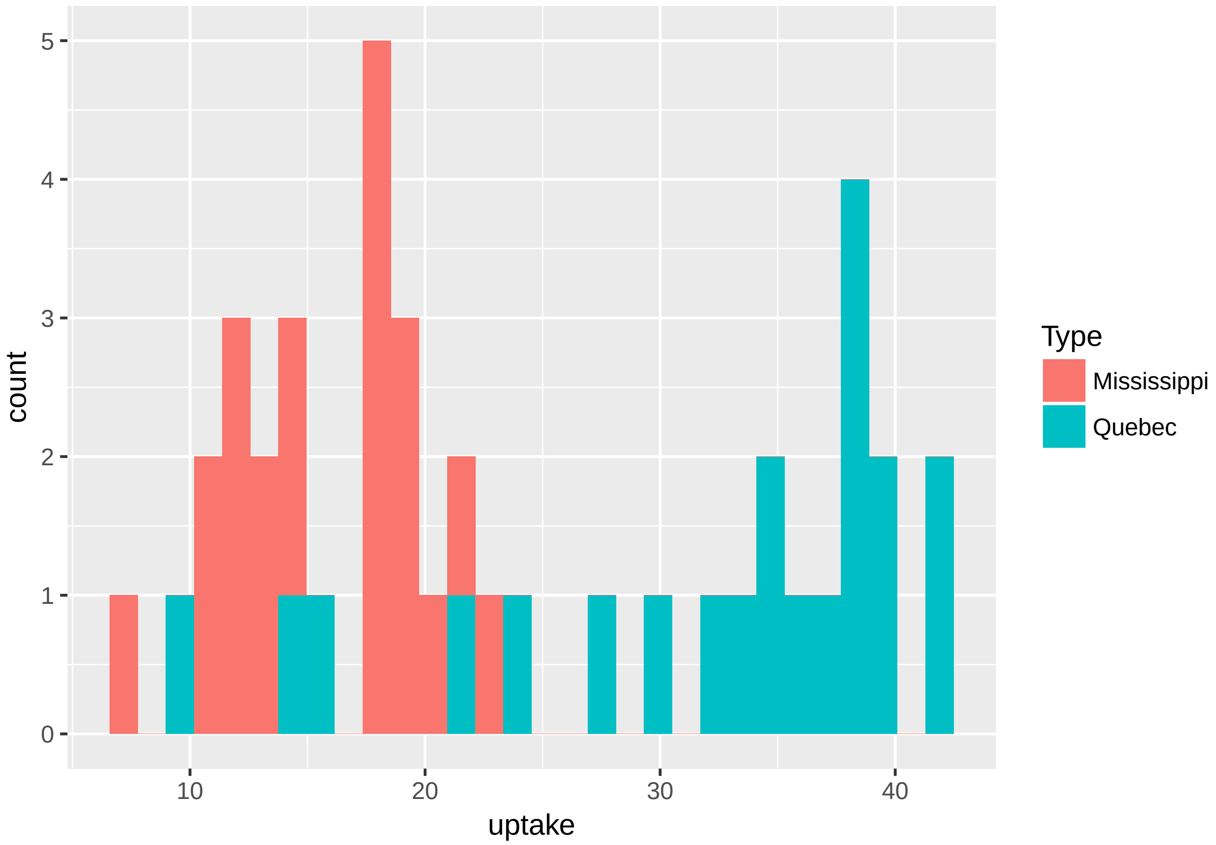


```
ggplot(data = CO2, aes(logconc, uptake, col = Type))+
        geom_point()
```

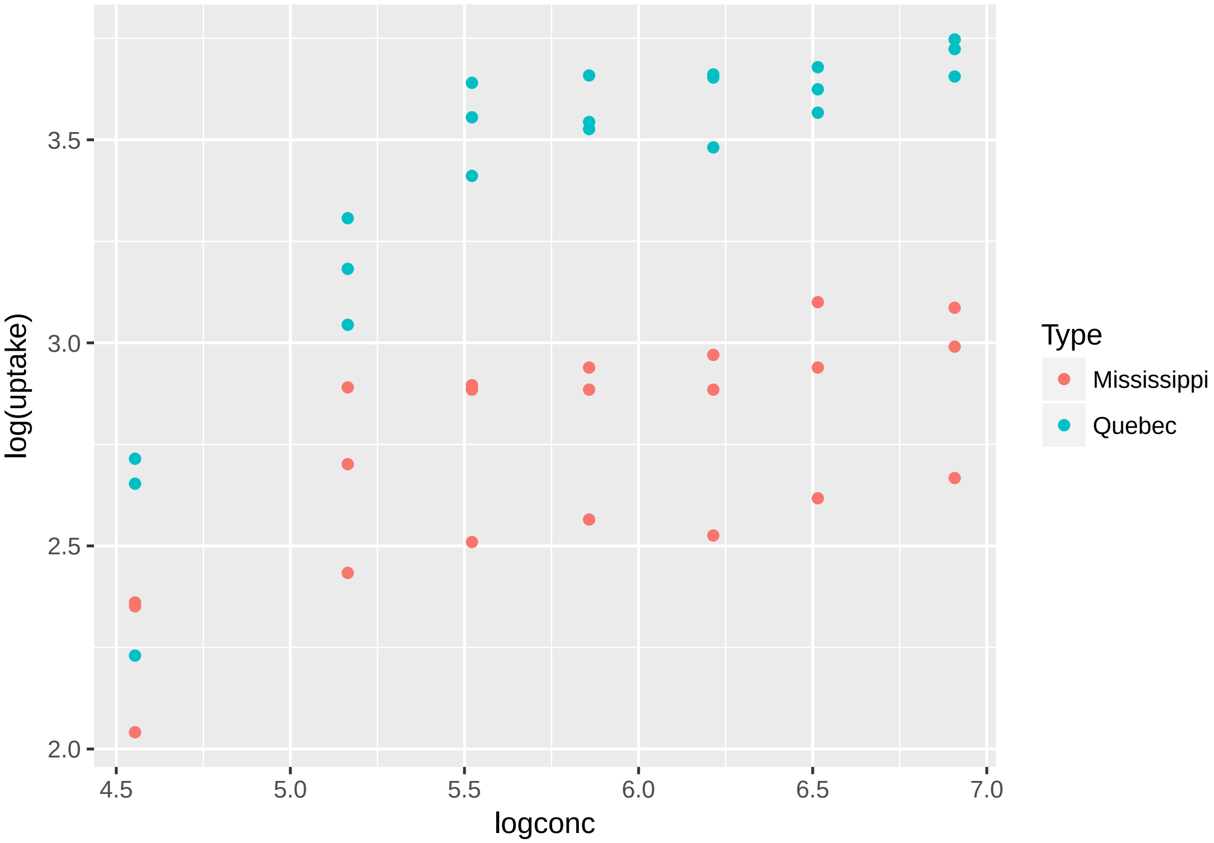


```r
#Does this effect depend on the origin of the plant ("Type")?
#the data clearly depends on type as shown above and when we seperate the data by "type", two distinct
ggplot(data = CO2, aes(uptake, fill = Type))+
        geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

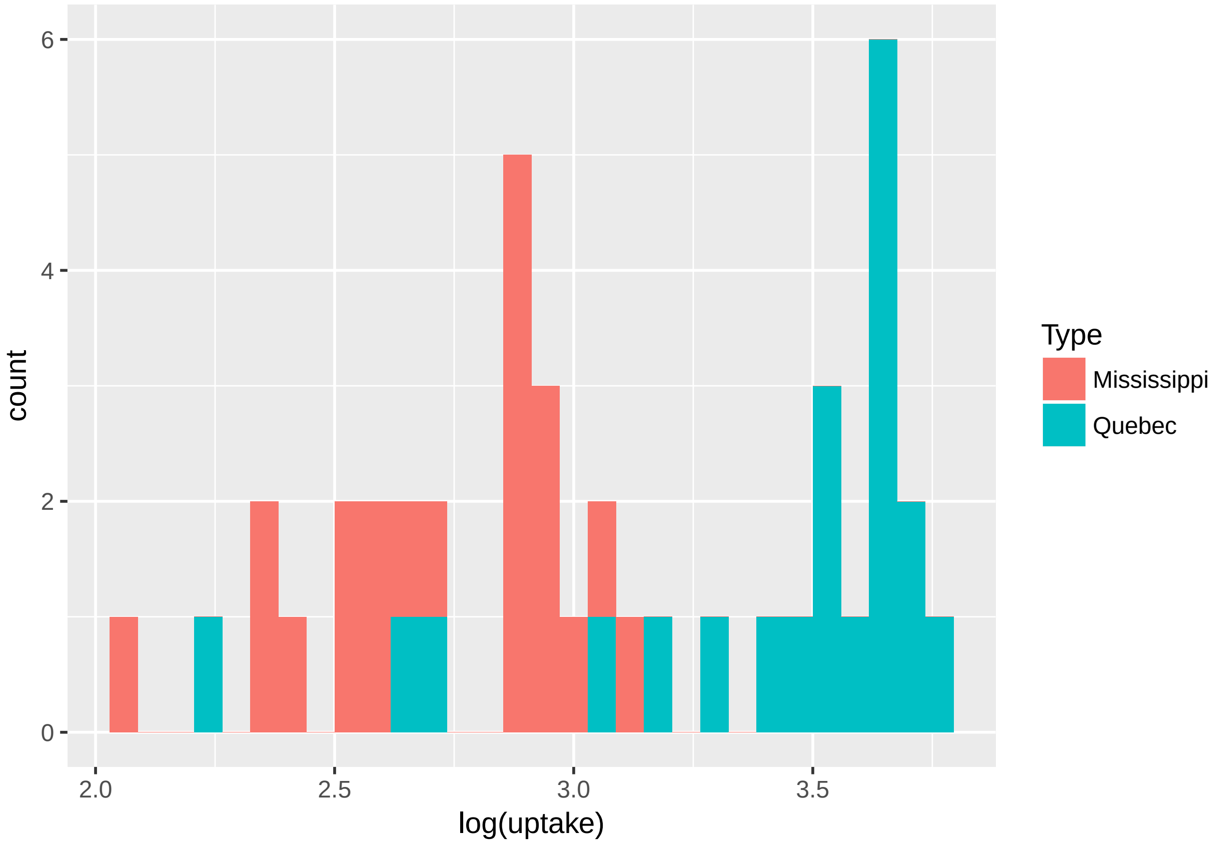


```r
#What transformations if any you made on the data and why.
#logtransforming does not make uptake more normal, so I wouldn't try to logtransform the data.
ggplot(data = CO2, aes(logconc, log(uptake), col = Type))+
        geom_point()
```

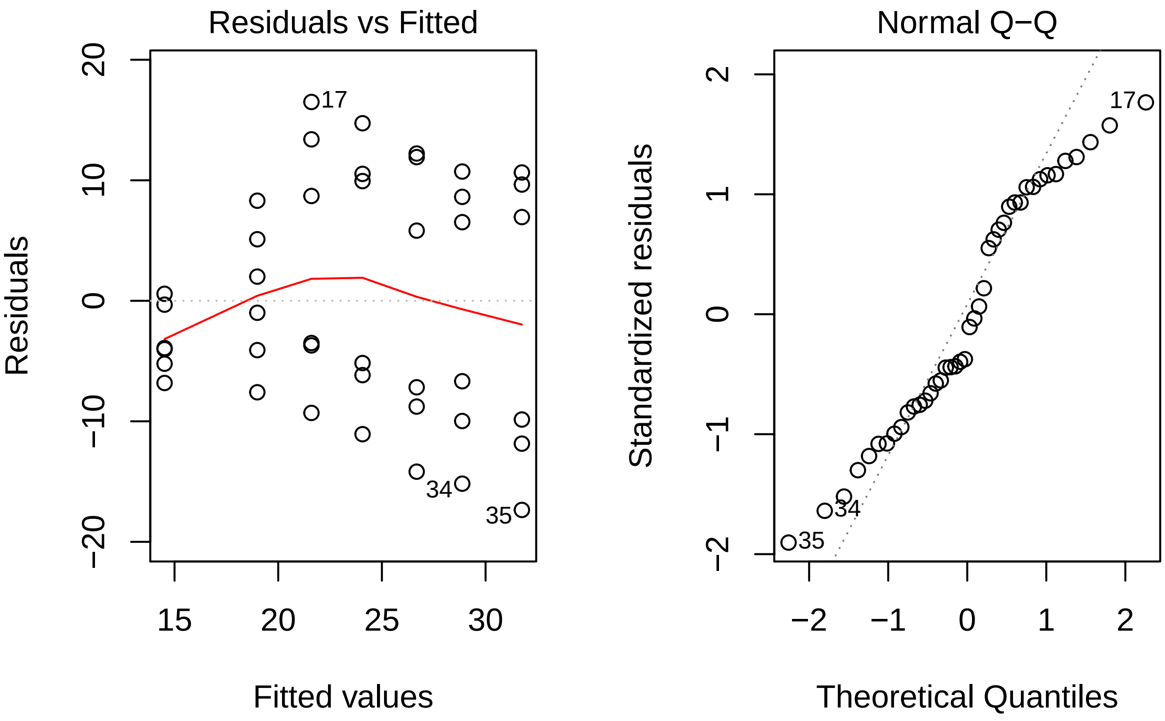


```r
ggplot(data = CO2, aes(log(uptake), fill = Type))+
         geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

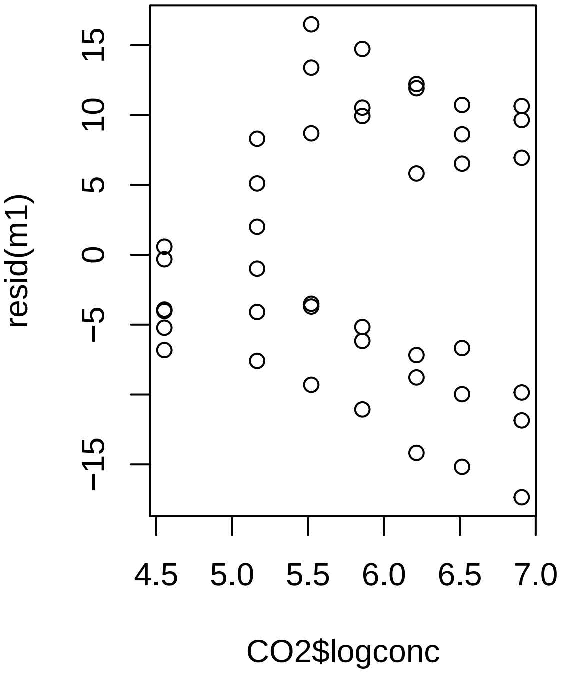


```r
#Here are three models, one that looks at how uptake changes with logconc, one that includes type as an
m1 <- lm(uptake ~ logconc, data = CO2)
par(mfrow = c(1,2))
plot(m1, which=1:2)
```

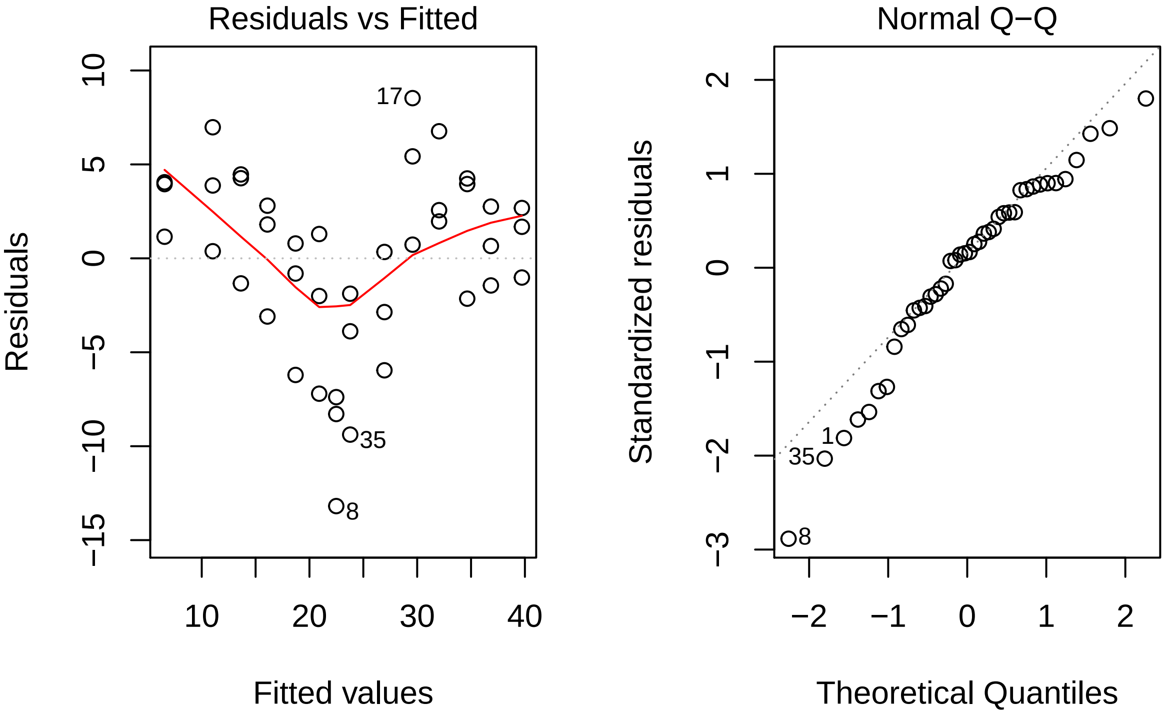


```r
plot(resid(m1)~CO2$logconc)
#This model shows the spread of residuals are not homogenous and clearly forms two groups. They are als

m2 <- lm(uptake ~ logconc+Type, data = CO2)
par(mfrow = c(1,2))
```



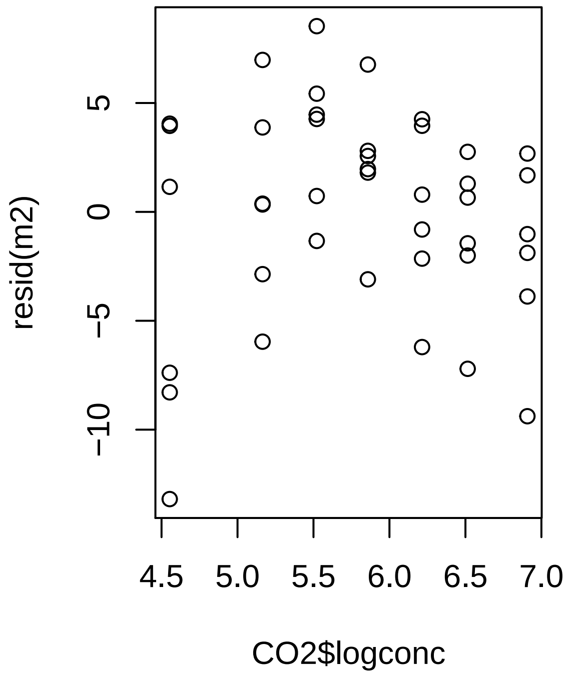
```r
plot(m2, which=1:2)
```



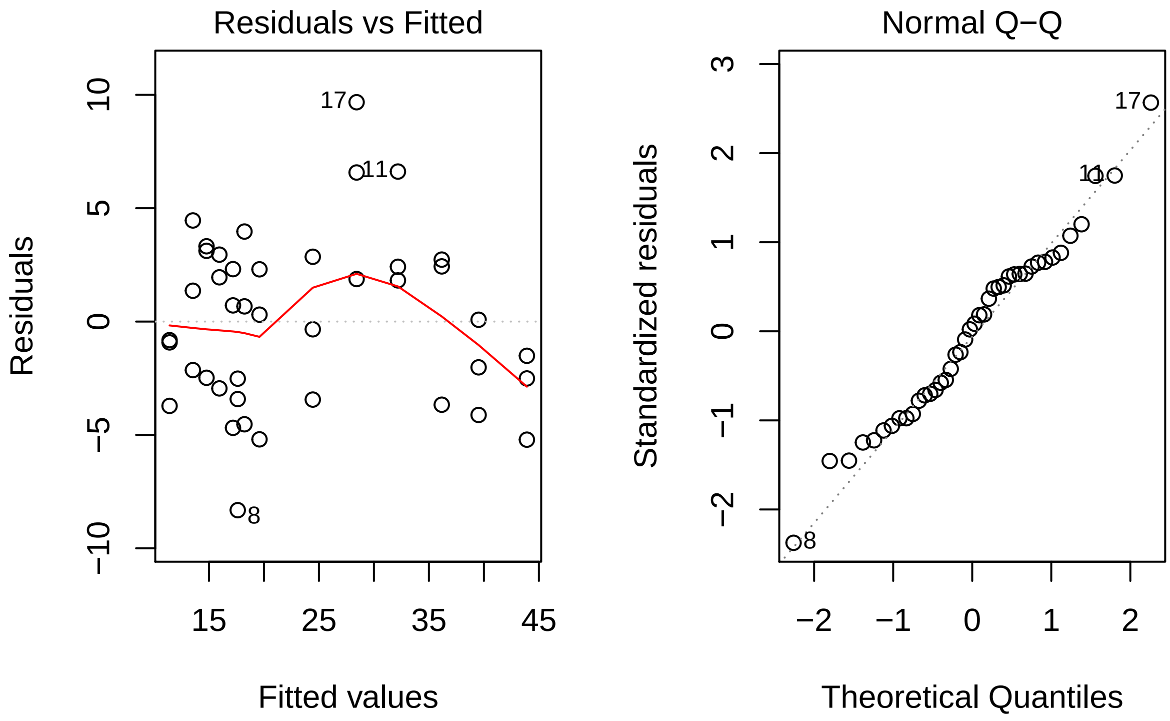
```r
plot(resid(m2)~CO2$logconc)
```

```r
#this model helps a little bit, but the residuals still are not very linear.

m3 <- lm(uptake ~ logconc+Type+logconc*Type, data = CO2)
par(mfrow = c(1,2))
```



```r
plot(m3, which=1:2)
```



```r
plot(resid(m3)~CO2$logconc)
#the interaction term doesn't change things too much, but I'll compare m2 and m3 with AIC to see which
```

```r
AIC(m2, m3)#the difference between AIC is 18. m3 is much better, which includes the interaction term.
```

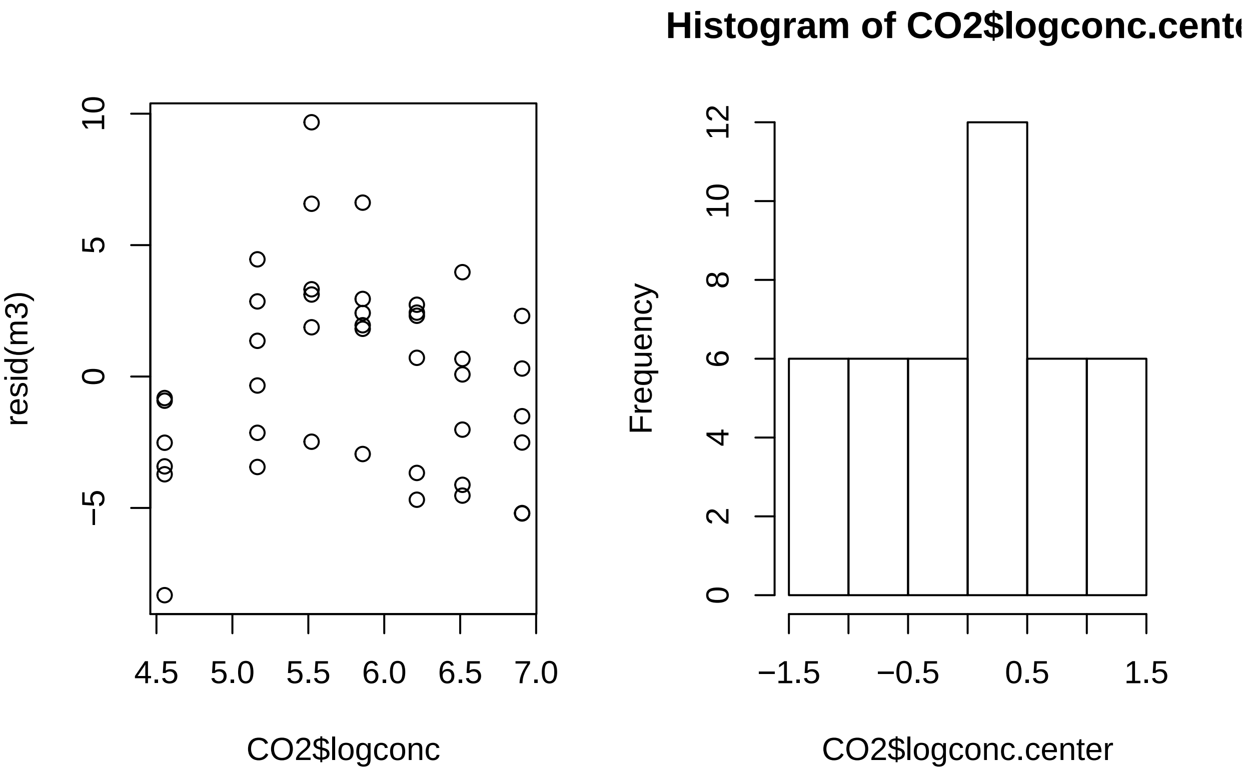
```
##    df      AIC
## m2  4 256.8942
## m3  5 238.7944
```

```r
#now to interpret model3
display(m3)
```

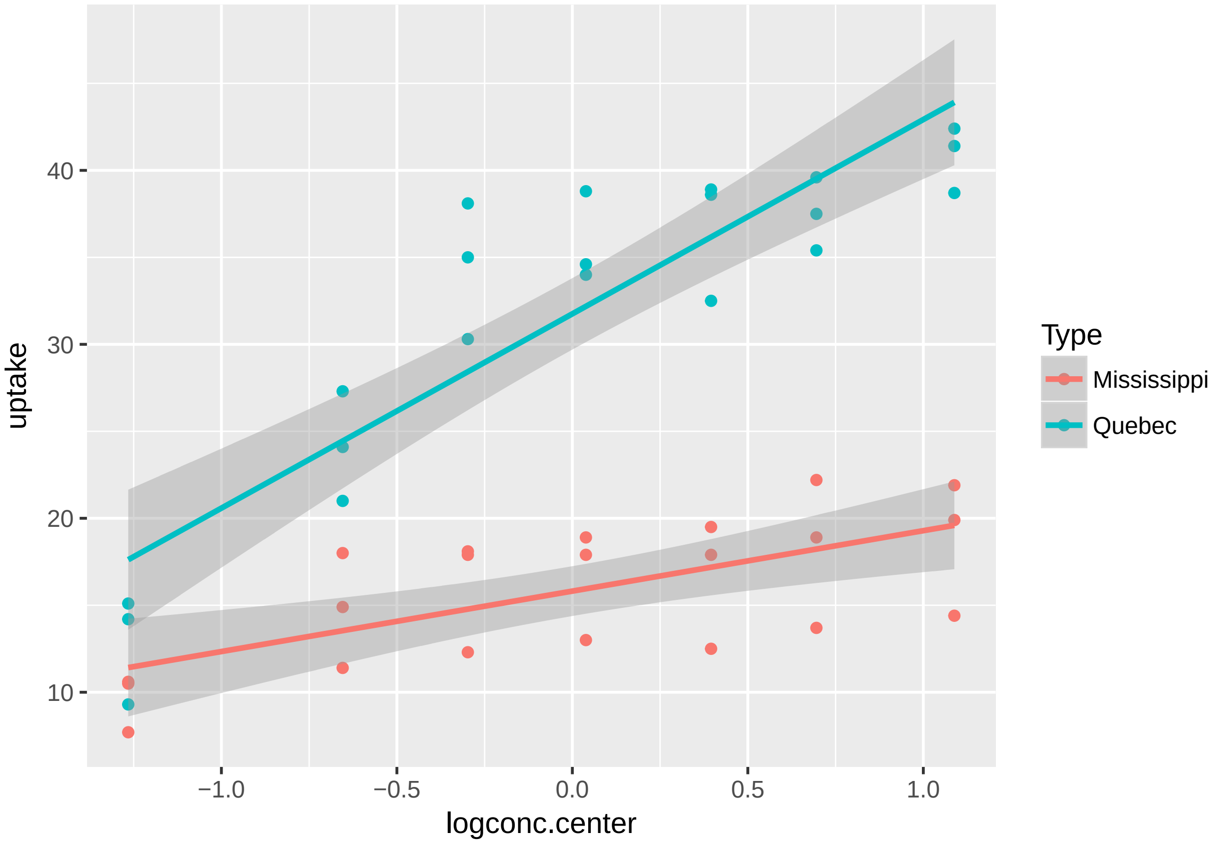
```
## lm(formula = uptake ~ logconc + Type + logconc * Type, data = CO2)
##                      coef.est coef.se
## (Intercept)           -4.40     6.61 
## logconc                3.47     1.13 
## TypeQuebec           -28.85     9.35 
## logconc:TypeQuebec     7.70     1.59 
## ---
## n = 42, k = 4
## residual sd = 3.88, R-Squared = 0.88
```

```r
#this model states that when logconc is 0, uptake is -4.40. This clearly doesn't make sense, so I'll ne

CO2$logconc.center <- CO2$logconc - mean(CO2$logconc)
hist(CO2$logconc.center)
```



```r
m3.centered <- lm(uptake ~ logconc.center+Type+logconc.center*Type, data = CO2)
ggplot(data = CO2, aes(logconc.center, uptake, col = Type))+
  geom_point() +
  geom_smooth(method = "lm")
```



```r
display(m3.centered)
```

```
## lm(formula = uptake ~ logconc.center + Type + logconc.center *
##     Type, data = CO2)
##                             coef.est coef.se
## (Intercept)                 15.81     0.85
## logconc.center               3.47     1.13
## TypeQuebec                  15.94     1.20
## logconc.center:TypeQuebec    7.70     1.59
## ---
## n = 42, k = 4
## residual sd = 3.88, R-Squared = 0.88
```

```r
#now the values are little more interpretable. At the average logconcentration (=0), uptake rate for mi
```

Question 2

Load the data set "ecdata_HW1.txt", which includes some growth and flowering time information on some Erodium cicutarium plants from serpentine and non-serpentine environments. The columns are: sourceSOILTYPE: soil type of source population, 1 = non-serpentine, 2 = serpentine earlylfno: count of leaves early in the plant's growth totallfno: count of total leaves at end of experiment ffdate: date of first flowering in days after germination

```
## null device
##           1
```

Fit a normal distribution to the Erodium ffdate data. Also fit a gamma distribution – does this distribution fit the data better or worse than the normal distribution does? Which is "better" by AIC score, or they both about the same?

Lets first start by plotting the data and superimposing the normal distribution, gamma distribution and density curve

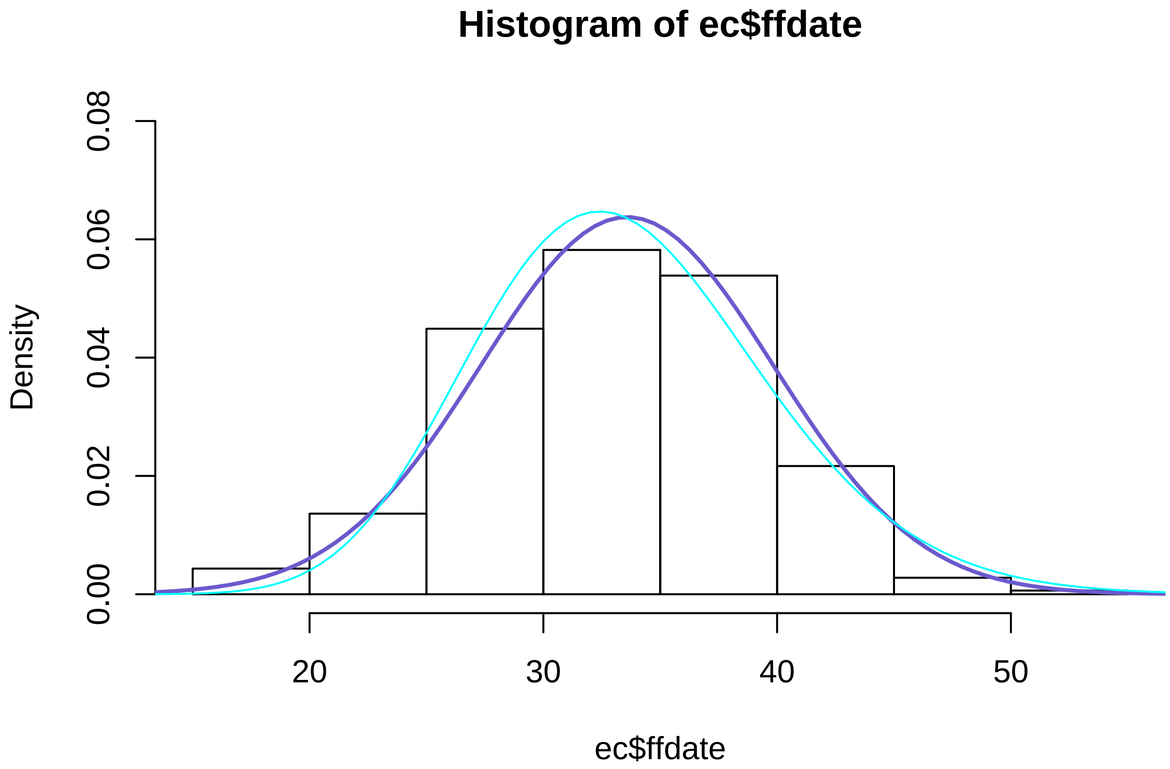
```
hist(ec$ffdate, freq = F, ylim = c(0,.08)) #this is what the raw data look like
# Plot the fitted normal distribution
min(ec$ffdate); max(ec$ffdate)
```

```
## [1] 17
```

```
## [1] 52
```

```
curve(dnorm(x, mean=mean(ec$ffdate), sd=sd(ec$ffdate)), from=10, to=60, n=100, add=T, col="slateblue", l
shape.ec <- mean(ec$ffdate)^2 / var(ec$ffdate)
scale.ec <- var(ec$ffdate) / mean(ec$ffdate)

#plot the gamma distribution
curve(dgamma(x, shape=shape.ec, scale=scale.ec), from=10, to=60, col=5, add=T)
```



Now I need to compare the fits of the two curves.

```
#write out the model for dnorm
model.normal <- mle2(ffdate~dnorm(mean=mu, sd=sigma), data=ec, start=list(mu=mean(ec$ffdate), sigma=sd(e


#i have no clue. start up here again.
model.gamma <- mle2(ffdate~dgamma(shape=shape, scale=scale), data=ec, start=list(shape=shape.ec, scale=s
```

```
## Warning in dgamma(x = c(37L, 37L, 44L, 31L, 40L, 37L, 37L, 33L, 25L, 33L, :
## NaNs produced
```

```
## Warning in dgamma(x = c(37L, 37L, 44L, 31L, 40L, 37L, 37L, 33L, 25L, 33L, :
```

```
## NaNs produced

## Warning in dgamma(x = c(37L, 37L, 44L, 31L, 40L, 37L, 37L, 33L, 25L, 33L, :
## NaNs produced
```

```
AIC(model.normal, model.gamma)
```

```
##          AIC df
## 1 4205.235  2
## 2 4225.783  2
```

Another way to do this is through the function `fitdist`.

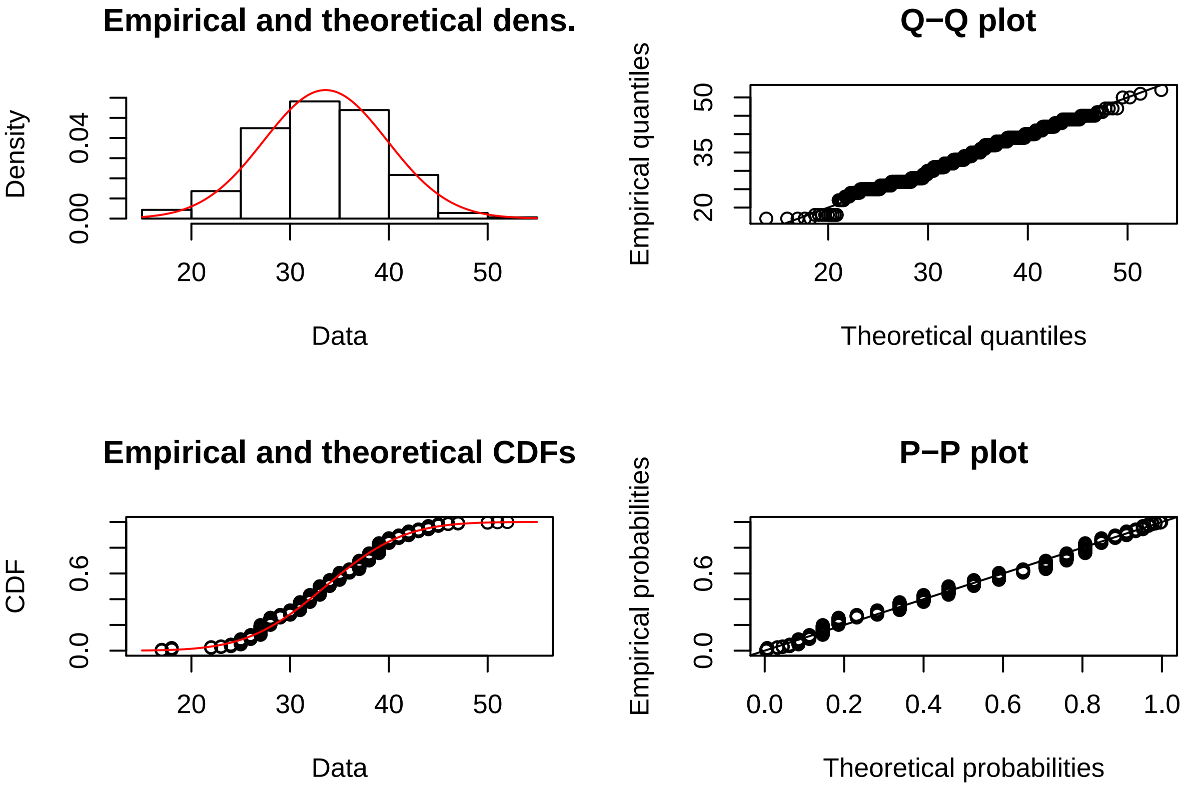
```
## Warning: package 'fitdistrplus' was built under R version 3.2.5

## Loading required package: survival
```

```
#normal distribution
fit.norm <- fitdist(ec$ffdate, "norm", method = "mle")
summary(fit.norm) #Loglikelihood:  -2100.618   AIC:  4205.235   BIC:  4214.177
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters :
##       estimate Std. Error
## mean 33.583591  0.2459547
## sd    6.251314  0.1739162
## Loglikelihood:  -2100.618   AIC:  4205.235   BIC:  4214.177
## Correlation matrix:
##      mean sd
## mean    1  0
## sd      0  1
```

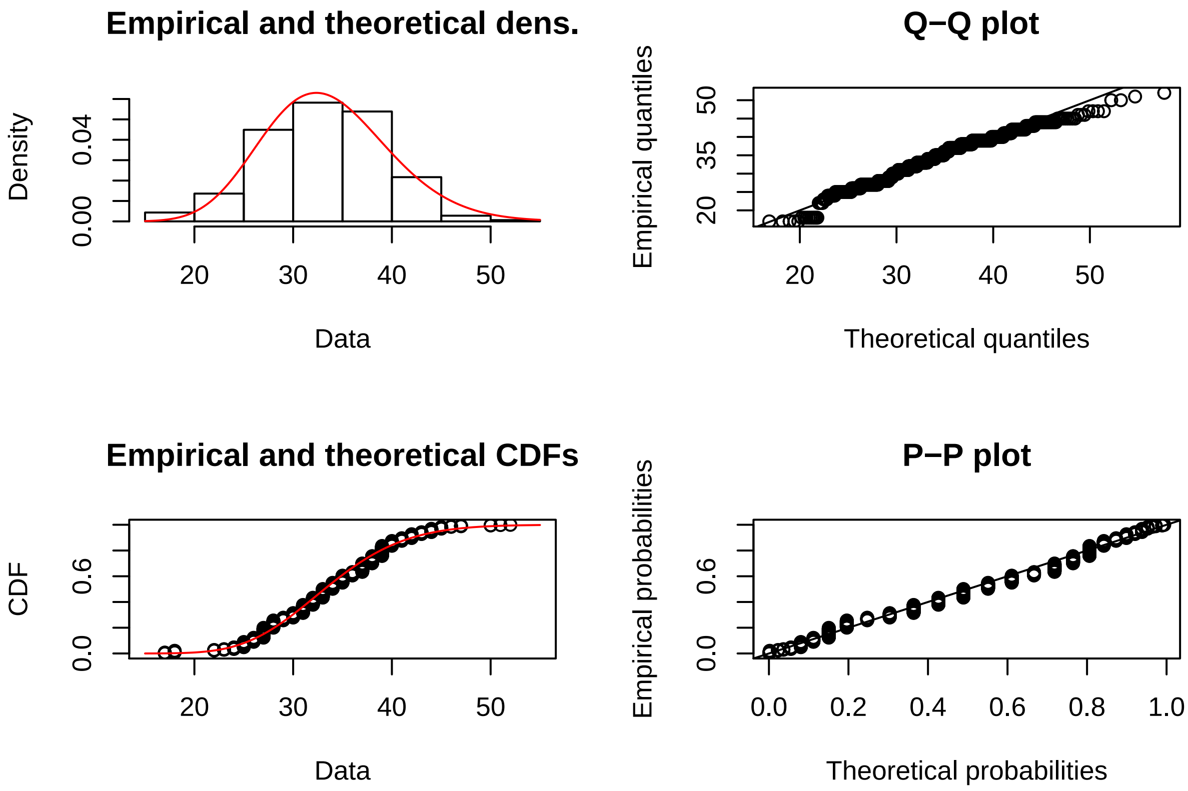
```
plot(fit.norm)
```



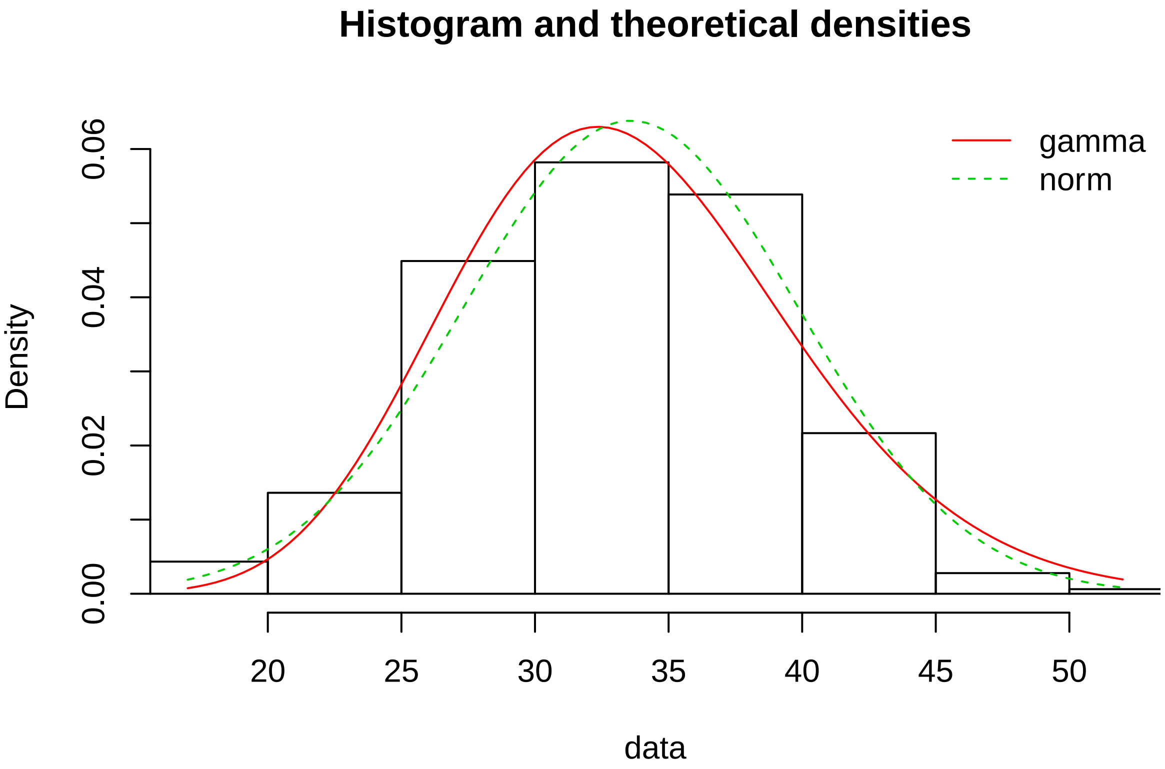
```
#gamma distribution
fit.gamma <- fitdist(ec$ffdate, "gamma")
summary(fit.gamma) #Loglikelihood:  -2110.891   AIC:  4225.783   BIC:  4234.724
```

```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters :
##         estimate Std. Error
## shape 27.2676629 1.50794429
## rate   0.8119317 0.04531581
## Loglikelihood:  -2110.891   AIC:  4225.783   BIC:  4234.724
## Correlation matrix:
##           shape      rate
## shape 1.0000000 0.9908458
## rate  0.9908458 1.0000000
```

```
plot(fit.gamma)
```



```
#comparison
denscomp(list(fit.gamma, fit.norm), legendtext=c("gamma", "norm"))
```



Ultimately, both the original way of fitting the data and through `fitdist` indicate that the normal distribution is the best fit.

Calculate the log-likelihood for the normal distribution at the fitted values of the parameters. Verify graphically (show on some kind of simple plot) that the log-likelihood of the data becomes more negative as the value of the mean moves farther from its maximum-likelihood value.