Homework 1 Suglia

Elena Suglia

10/15/2019

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.

```
d = read.table("CO2_HW1.txt", header = TRUE)
# Check the loaded data
head(d)
```

```
##     Type uptake  logconc
## 1 Quebec   14.2 4.553877
## 2 Quebec   24.1 5.164786
## 3 Quebec   30.3 5.521461
## 4 Quebec   34.6 5.857933
## 5 Quebec   32.5 6.214608
## 6 Quebec   35.4 6.514713
```

- Looks like it loaded correctly

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")? 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.

Check structure of data frame

```
class(d) # data.frame
```

```
## [1] "data.frame"
```

```
str(d) # structure of the dataframe and data types in each column
```

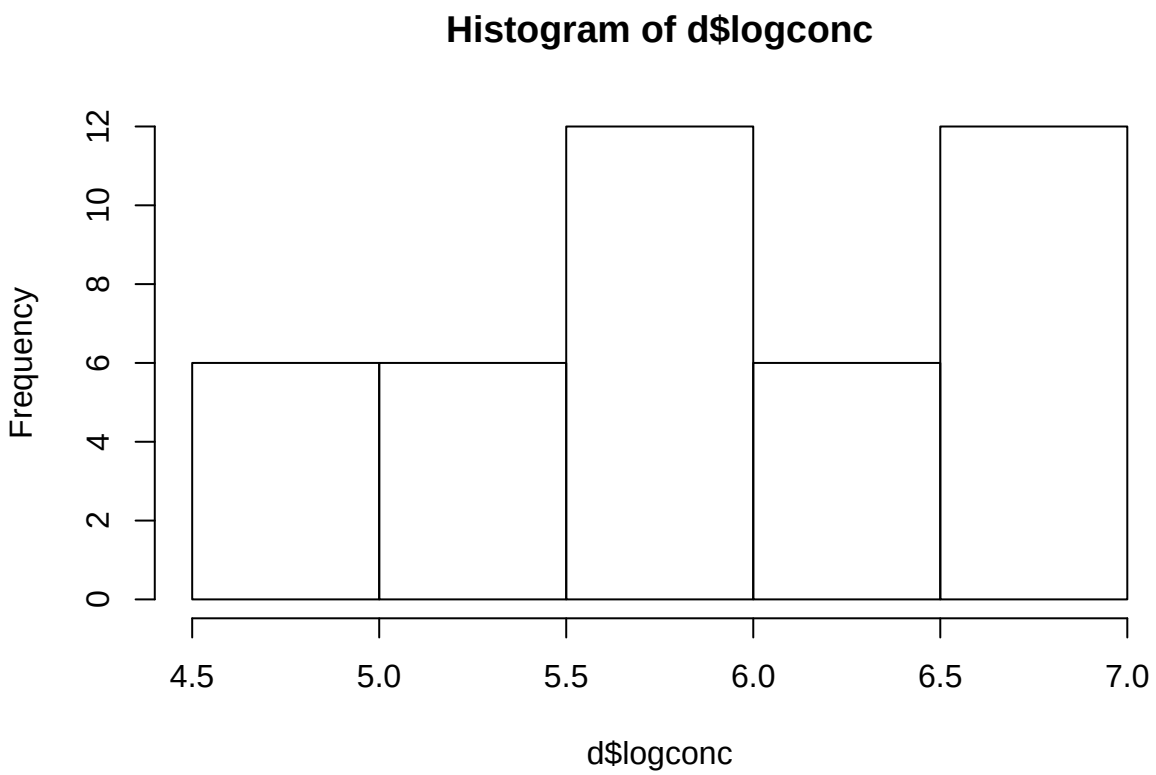
```
## 'data.frame':    42 obs. of  3 variables:
##  $ Type   : Factor w/ 2 levels "Mississippi",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ uptake : num  14.2 24.1 30.3 34.6 32.5 35.4 38.7 9.3 27.3 35 ...
##  $ logconc: num  4.55 5.16 5.52 5.86 6.21 ...
```

- Everything looks good

Visual checks for normality/distribution of the data

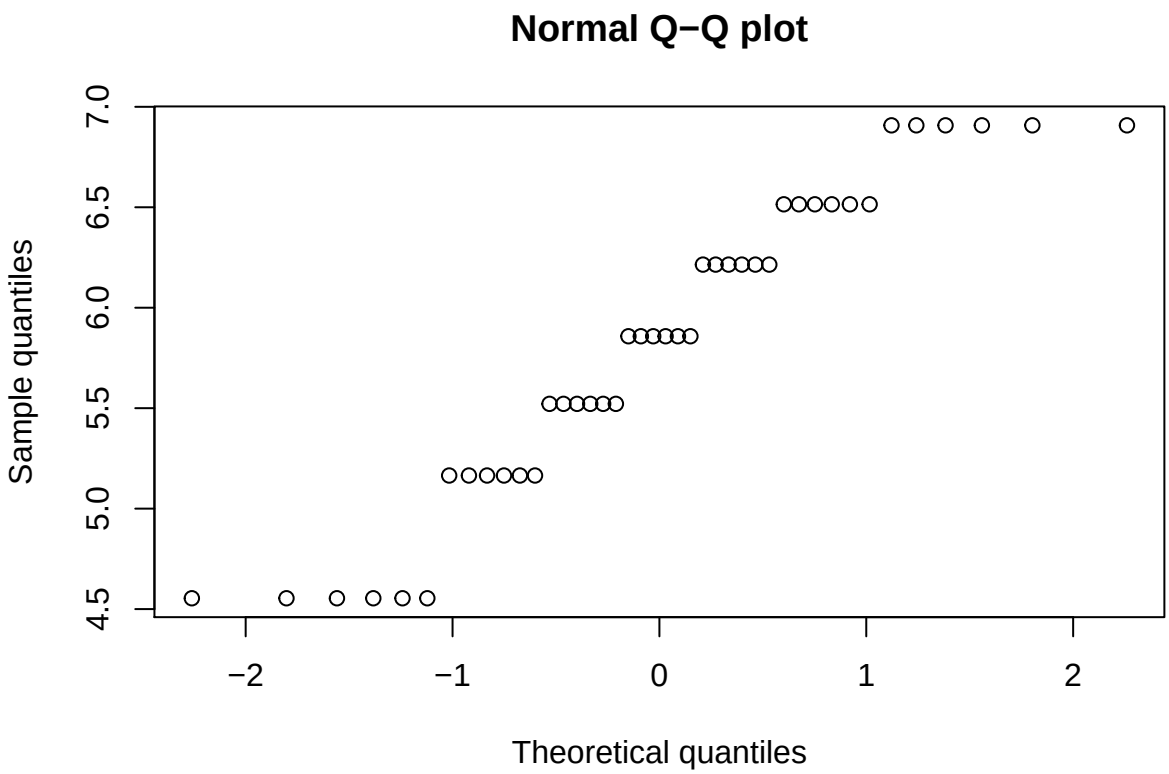
Histogram

```
hist(d$logconc)
```



Normal Q-Q plot

```
qqnorm(d$logconc, main = "Normal Q-Q plot", xlab = "Theoretical quantiles", ylab = "Sample quantiles")
```

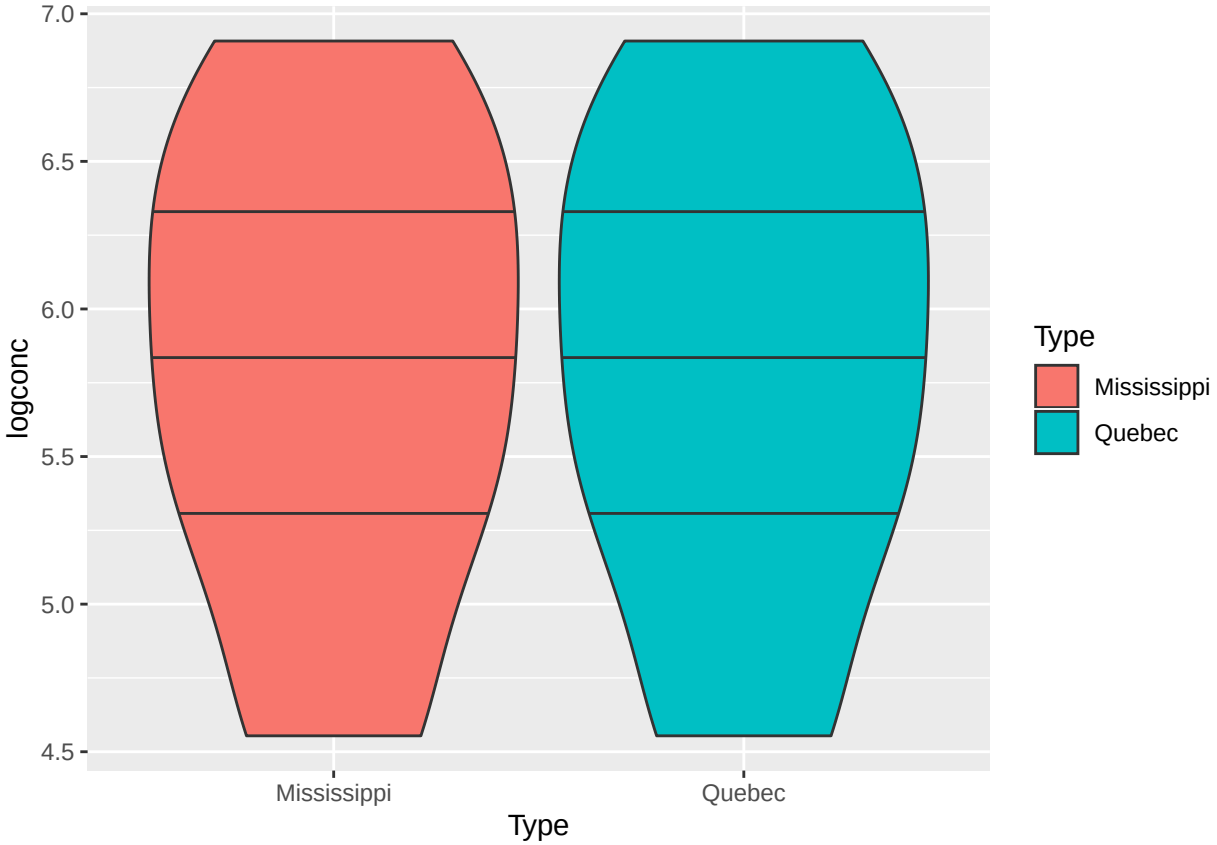


- It's not perfectly normal but it's the best it's going to get; the data has already been transformed and I can't find a different or additional one that would make it any better

Violin plots to look at the spread of the data

First, look at spread of log concentration of CO2 uptake

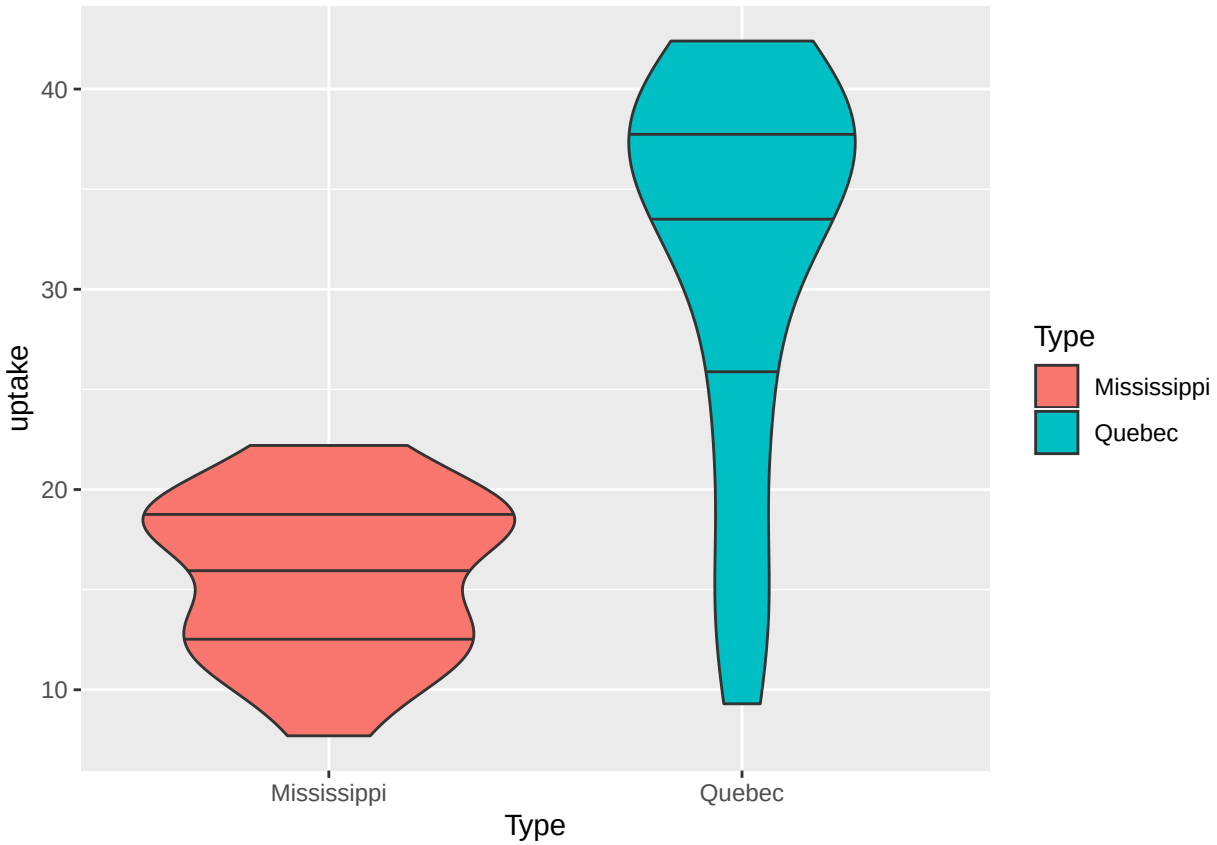
```
ggplot(d, aes(x=Type, y=logconc)) + geom_violin(draw_quantiles = c(0.25, 0.5, 0.75), aes(fill=Type))
```



- These plots tell us that there is virtually no difference in spread of log concentration of CO2 between the 2 sites

Now, let's look at the spread of the response variable, CO2 uptake rate

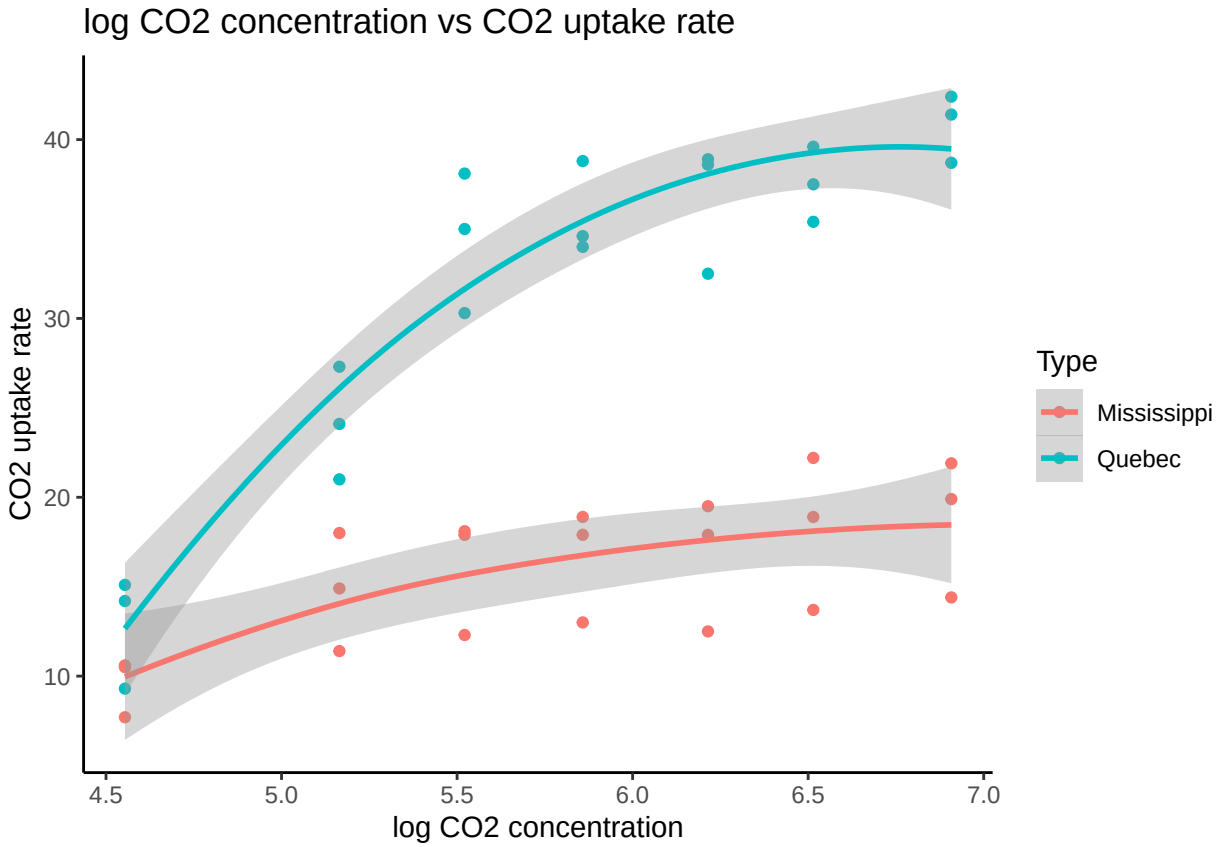
```
ggplot(d, aes(x=Type, y=uptake)) + geom_violin(draw_quantiles = c(0.25, 0.5, 0.75), aes(fill=Type))
```



- There is definitely a difference in CO2 uptake rates between sites (higher uptake rates in Quebec)

Relationship between response and explanatry variables: Scatter plot of the log CO2 concentration versus the rate of CO2 uptake

```
ggplot(d, aes(x=logconc, y=uptake, group = Type, color = Type)) +
  geom_point() +
  geom_smooth(span=2) +
  theme_classic() +
ggtitle("log CO2 concentration vs CO2 uptake rate") +
  xlab("log CO2 concentration") +
  ylab("CO2 uptake rate")
```



Higher log CO2 concentration appears positively correlated with uptake rate. There are higher overall uptake rates in Quebec. Here, you are able to see that there appears to be an interaction between location and response to CO2 concentration: Quebec plants increase their CO2 uptake rates more in response to an increase in log CO2 concentration than Mississippi plants do. Thus, the relationship between log CO2 concentration and uptake depends on location, or in other words, the association between the response and the explanatory variables depends on the level of a third variable.

Fit model

Let's compare model fit between several different models: one that has one explanatory variable (uptake rate), one that accounts for location, and one that has an interaction between uptake rate and location.

Only looks at relationship between log concentration CO2 and CO2 uptake rate

```
m1 = lm(uptake~logconc, d)
display(m1)
```

```
## lm(formula = uptake ~ logconc, data = d)
##             coef.est coef.se
## (Intercept) -18.82    11.42
## logconc       7.32     1.95
## ---
## n = 42, k = 2
## residual sd = 9.47, R-Squared = 0.26
```

Accounts for location

```
m2 = lm(uptake~logconc + Type, d)
display(m2)
```

```
## lm(formula = uptake ~ logconc + Type, data = d)
##             coef.est coef.se
## (Intercept) -26.79     5.91
## logconc       7.32     1.00
## TypeQuebec   15.94     1.50
## ---
## n = 42, k = 3
## residual sd = 4.86, R-Squared = 0.81
```

Includes interaction between uptake rate and location

```
m3 = lm(uptake~logconc*Type, d)
display(m3)
```

```
## lm(formula = uptake ~ logconc * Type, data = d)
##                     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
```

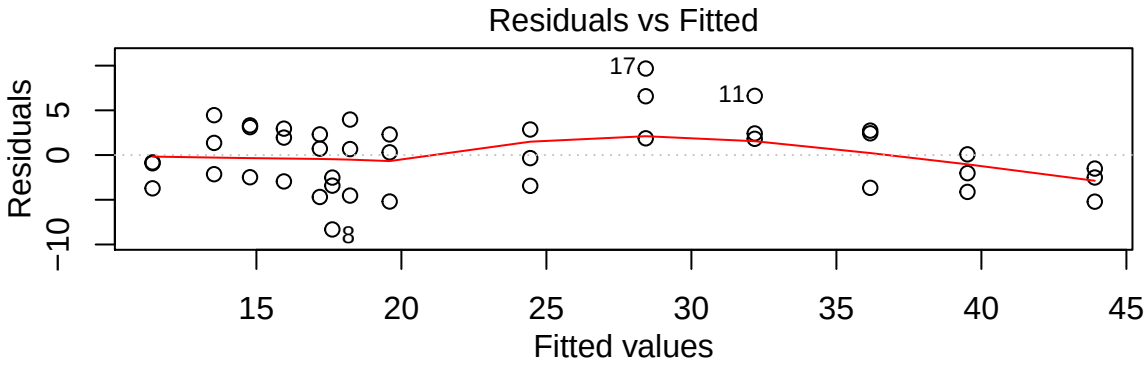
The model that includes the interaction between uptake rate and location has the best fit: its residual standard deviation is lowest and the R-squared value is highest (this model explains the most variation in the response variable). This makes sense based on our observation above that the relationship between log CO2 concentration and uptake rate depends on location.

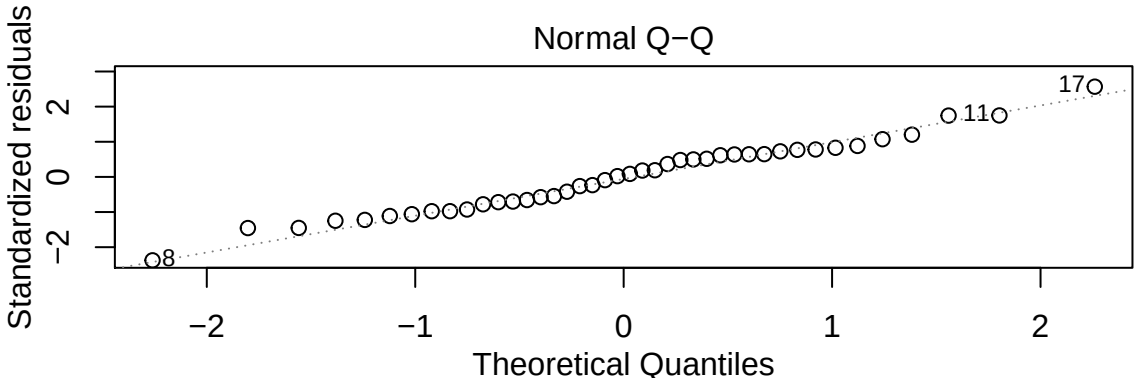
Assumptions of a linear regression model

- Validity - our question is simply about the effects of one variable on the other; so in this case the regression model seems to be a valid test of this
- Linearity - relationships look fairly linear on the scatterplot
- Independence of errors
- Equal variance of errors
- Normality of errors

Let's look at the error distribution in the best fit model (m3) to test the last three assumptions:

```
par(mfrow=c(2, 1), mar=rep(3,4), mgp=c(2,1,0))
plot(m3, which=1:2)
```





- These assumptions do not appear to be badly violated

Interpreting the coefficients of the model

It's often easier to interpret coefficients of a model when you center and scale the data, especially when there are interactions. Let's do that here:

```
d.center <- mutate(d, logconc = scale(logconc, center=TRUE, scale=TRUE))
```

Then we can repeat the same regression and compare

```
m4 <- lm(uptake~logconc*Type, d.center)
display(m3)
```

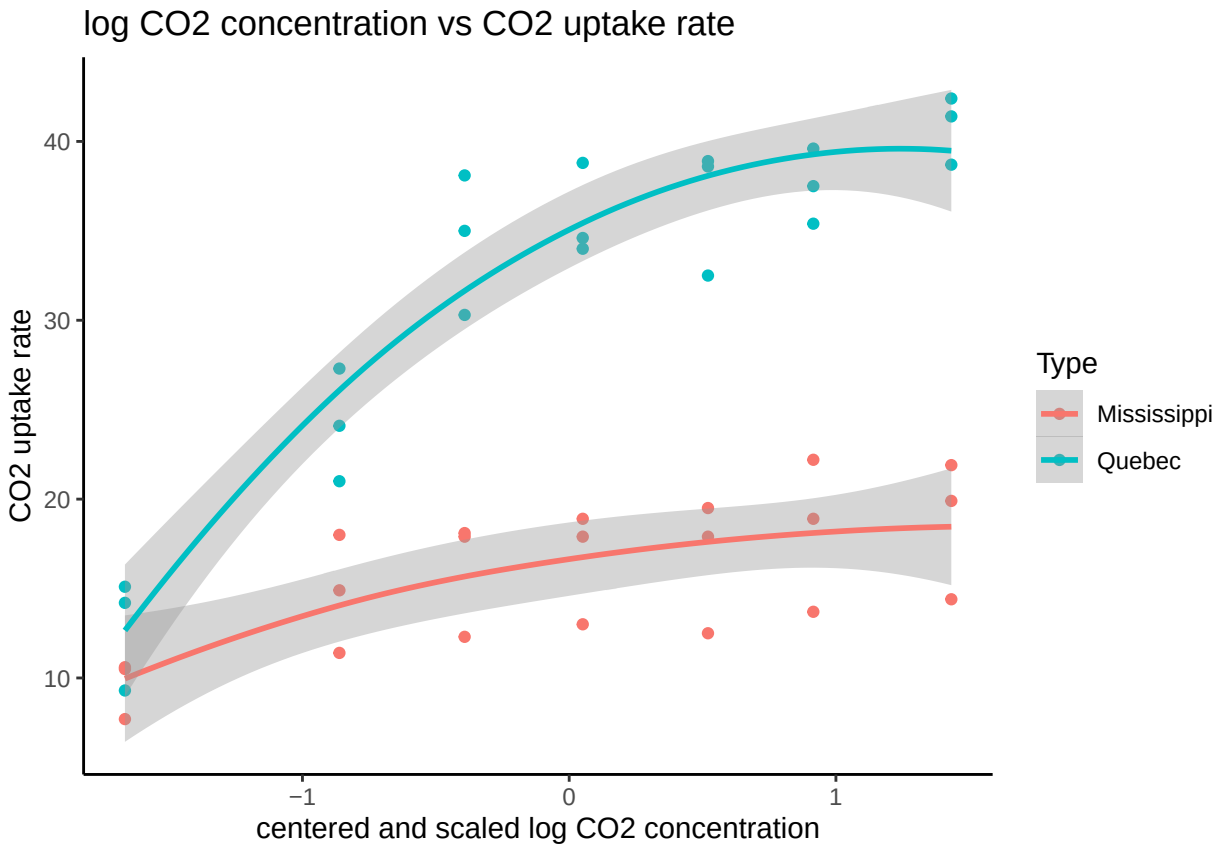
```
## lm(formula = uptake ~ logconc * Type, data = d)
##                     coef.est coef.se
## (Intercept)          -4.40     6.61
## logconc               3.47     1.13
## TypeQuebec          -28.85     9.35
## logconc:TypeQuebec    7.70     1.59
## ---
## n = 42, k = 4
## residual sd = 3.88, R-Squared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

```
display(m4)
```

```
## lm(formula = uptake ~ logconc * Type, data = d.center)
```

```
##                     coef.est coef.se
## (Intercept)         15.81     0.85
## logconc              2.64     0.86
## TypeQuebec          15.94     1.20
## logconc:TypeQuebec  5.85     1.21
## ---
## n = 42, k = 4
## residual sd = 3.88, R-Squared = 0.88
```

```
ggplot(d.center, aes(x=logconc, y=uptake, group = Type, color = Type)) +
  geom_point() +
  geom_smooth(span=2) +
  theme_classic() +
  ggtitle("log CO2 concentration vs CO2 uptake rate") +
  xlab("centered and scaled log CO2 concentration") +
  ylab("CO2 uptake rate")
```



put here your interpretation of the coefficients of the model

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: - source-SOILTYPE: 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

```
ec = read.table("ecdata_HW1.txt", header = TRUE)
```

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?

Normal distribution —-

```
mean.ff = mean(ec$ffdate)
sigma.ff = sd(ec$ffdate)

m1ec = mle2(ffdate~dnorm(mean=mean.ff, sd=sigma.ff), data=ec, start=list(mu=10, sigma=1))
summary(m1ec)
```

```
## Maximum likelihood estimation
##
## Call:
## mle2(minuslogl = ffdate ~ dnorm(mean = mean.ff, sd = sigma.ff),
##     start = list(mu = 10, sigma = 1), data = ec)
##
## Coefficients:
##       Estimate Std. Error z value     Pr(z)
## mu          10          0     Inf < 2.2e-16 ***
## sigma        1          0     Inf < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -2 log L: 4201.236
```

Gamma distribution —-

Define the Gamma negative log-likelihood

```
gammaNLL1 <- function(shape, scale) {
  return(-sum(dgamma(ec$ffdate, shape=shape, scale=scale, log=T)))
}
```

Look up the gamma distribution and see what its moments are to get starting values:

```
shape.start <- mean(ec$ffdate)^2 / var(ec$ffdate)
scale.start <- var(ec$ffdate) / mean(ec$ffdate)

m2ec <- mle2(gammaNLL1, start=list(shape=shape.start, scale=scale.start), trace=T)
```

```
## Warning in dgamma(ec$ffdate, shape = shape, scale = scale, log = T): NaNs
## produced

## Warning in dgamma(ec$ffdate, shape = shape, scale = scale, log = T): NaNs
```

```
## produced

## Warning in dgamma(ec$ffdate, shape = shape, scale = scale, log = T): NaNs
## produced
```

```
summary(m2ec)
```

```
## Maximum likelihood estimation
##
## Call:
## mle2(minuslogl = gammaNLL1, start = list(shape = shape.start,
##     scale = scale.start), trace = T)
##
## Coefficients:
##        Estimate Std. Error z value     Pr(z)
## shape 27.269826   1.507633  18.088 < 2.2e-16 ***
## scale  1.231522   0.068714  17.922 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -2 log L: 4221.783
```

Now compare models:

```
AIC(m1ec, m2ec)
```

```
##        AIC df
## 1 4205.236  2
## 2 4225.783  2
```

- Normal distribution fits better

Calculate the log-likelihood for the normal distribution at the fitted values of the parameters. Show (graphically or in numbers) that the log-likelihood of the data becomes more negative if you shift the mean parameter value away from its maximum-likelihood value.

Calculate the log-likelihood —-

```
logLik(m1ec) # -2100.618
```

```
## 'log Lik.' -2100.618 (df=2)
```

```
logLik(m2ec) # -2110.891
```

```
## 'log Lik.' -2110.891 (df=2)
```

As expected, the log-likelihood for the better fitting model (m2ec) is higher.

Shift the mean parameter to 40:

```
m3ec = mle2(ffdate~dnorm(mean=40, sd=sigma.ff), data=ec, start=list(mu=10, sigma=1))
logLik(m3ec) # -2440.377
```

```
## 'log Lik.' -2440.377 (df=2)
```

Log-likelihood goes down to -2440.377.