Exercise 2. Data analysis

Sasha D. Hafner

04 March, 2024

R data.table solution

Overview

In this exercise you will analyze emission of methane from pig slurry samples incubated in a laboratory experiment. Multiple bottles were used as reactors in a completely crossed factorial experiment with 2 temperatures and 2 headspace gases. Measurement data were kindly provided by Frederik Dalby.

My suggestion

I recommend using response feature analysis with calculated total emission as the response variable, fitting a linear model using `lm()` in R or `ols()` in Python. But you could use other responses.

1. Read and check data

Read in the data in the file `slurry_emis.csv`. Check the data. The relevant columns are

- `reactor`: bottle key
- `ch4`: methane concentration in bottle exhaust in ppmv
- `flow`: rate of gas flow through the bottle in L/min
- `day`: time of measurement from setup in d
- `gas`: headspace gas
- `temp`: incubation temperature

Load packages

```
library(data.table)
library(rmarkdown)
library(ggplot2)
```

Function

```
source('../functions-R/dfsumm.R')
```

Load data

```
dat <- fread('../data/slurry_emis.csv')
```

Check data

```
dat
```

```
##      reactor     ch4      co2       flow day gas temp
##   1:      R1  11.374  338.300 0.06300000   5 co2   20
##   2:      R2   9.638  348.235 0.07300000   5 co2   20
##   3:      R3   5.221  320.180 0.08200000   5 co2   20
##   4:      R4   7.200  313.690 0.08100000   5 co2   20
##   5:      R5  16.000  371.500 0.08400000   5 co2   30
##  ---
## 350:     R12  59.150 1002.000 0.06121372 283  ar   20
## 351:     R13  48.320  858.300 0.06754617 283  ar   30
## 352:     R14  49.970  865.400 0.06860158 283  ar   30
## 353:     R15  45.260  837.200 0.06860158 283  ar   30
## 354:     R16 105.800  895.000 0.05910290 283  ar   30
```

```
summary(dat)
```

```
##    reactor              ch4              co2             flow
##  Length:354         Min.   :   1.95   Min.   :  74.9   Min.   :0.00000
##  Class :character   1st Qu.:  23.12   1st Qu.: 224.2   1st Qu.:0.05865
##  Mode  :character   Median :  51.34   Median : 818.6   Median :0.06555
##                     Mean   : 187.11   Mean   : 856.3   Mean   :0.06441
##                     3rd Qu.: 200.97   3rd Qu.:1164.5   3rd Qu.:0.06966
##                     Max.   :1342.00   Max.   :3578.4   Max.   :0.08400
##                     NA's   :6         NA's   :6
##       day             gas                 temp
##  Min.   :  5.0   Length:354         Min.   :20.00
##  1st Qu.: 39.0   Class :character   1st Qu.:20.00
##  Median :115.0   Mode  :character   Median :30.00
##  Mean   :123.7                      Mean   :25.06
##  3rd Qu.:200.0                      3rd Qu.:30.00
##  Max.   :283.0                      Max.   :30.00
##
```

```
dfsumm(dat)
```

```
##
##  354 rows and 7 columns
##  354 unique rows
##                      reactor     ch4     co2    flow     day       gas    temp
## Class               character numeric numeric numeric integer character integer
## Minimum                    bg    1.95    74.9       0       5        ar      20
## Maximum                    R9    1340    3580   0.084     283       co2      30
## Mean                     <NA>     187     856  0.0644     124      <NA>    25.1
## Unique (excld. NA)         17     345     321      61      22         2       2
## Missing values              0       6       6       0       0         0       0
## Sorted                  FALSE   FALSE   FALSE   FALSE    TRUE     FALSE   FALSE
##
```

```
dim(dat)
```

```
## [1] 354   7
```

```
table(dat[, reactor])
```

```
##
```

```
##   bg   R1  R10  R11  R12  R13  R14  R15  R16   R2   R3   R4   R5   R6   R7   R8   R9 
##    3   22   22   22   21   22   22   22   22   22   22   22   22   22   22   22   22
```

2. Data analysis

Use an appropriate approach to determine if the data show that incubation temperature and headspace gas affect methane emission. Quantify any effect.

Calculate methane flow rate

First we need mass-based methane concentration. From ideal gas law `pv = nRT` we get `Mn / v = p / RT`. Assume 20 degress C and 1 atm (would be good to check if these were really our data). Temperature should be at the point of flow rate measurement, right? And `gas_constant = 8.2057E-5` for the universal gas constant in atm m^3k mol^{-1} K^{-1}. So first, concentration in g/m3:

```
dat[, cmch4 := ch4 * 1.0 / 1E6 / 8.2057E-5 / (273.15 + 20)]
dat
```

```
##      reactor     ch4      co2       flow day gas temp        cmch4
##   1:      R1  11.374  338.300 0.06300000   5 co2   20 0.0004728329
##   2:      R2   9.638  348.235 0.07300000   5 co2   20 0.0004006650
##   3:      R3   5.221  320.180 0.08200000   5 co2   20 0.0002170442
##   4:      R4   7.200  313.690 0.08100000   5 co2   20 0.0002993140
##   5:      R5  16.000  371.500 0.08400000   5 co2   30 0.0006651421
##  ---                                                              
## 350:     R12  59.150 1002.000 0.06121372 283  ar   20 0.0024589473
## 351:     R13  48.320  858.300 0.06754617 283  ar   30 0.0020087292
## 352:     R14  49.970  865.400 0.06860158 283  ar   30 0.0020773220
## 353:     R15  45.260  837.200 0.06860158 283  ar   30 0.0018815208
## 354:     R16 105.800  895.000 0.05910290 283  ar   30 0.0043982523
```

And flow in g/d:

```
dat[, qch4 := cmch4 * flow / 1000 * 1400]
dat
```

```
##      reactor     ch4      co2       flow day gas temp        cmch4         qch4
##   1:      R1  11.374  338.300 0.06300000   5 co2   20 0.0004728329 4.170386e-05
##   2:      R2   9.638  348.235 0.07300000   5 co2   20 0.0004006650 4.094796e-05
##   3:      R3   5.221  320.180 0.08200000   5 co2   20 0.0002170442 2.491667e-05
##   4:      R4   7.200  313.690 0.08100000   5 co2   20 0.0002993140 3.394220e-05
##   5:      R5  16.000  371.500 0.08400000   5 co2   30 0.0006651421 7.822071e-05
##  ---                                                                           
## 350:     R12  59.150 1002.000 0.06121372 283  ar   20 0.0024589473 2.107298e-04
## 351:     R13  48.320  858.300 0.06754617 283  ar   30 0.0020087292 1.899548e-04
## 352:     R14  49.970  865.400 0.06860158 283  ar   30 0.0020773220 1.995106e-04
## 353:     R15  45.260  837.200 0.06860158 283  ar   30 0.0018815208 1.807054e-04
## 354:     R16 105.800  895.000 0.05910290 283  ar   30 0.0043982523 3.639293e-04
```

Get integration function

```
source('../functions-R/mintegrate.R')
```

Integrate

```
dim(dat)
```

```
## [1] 354   9
```

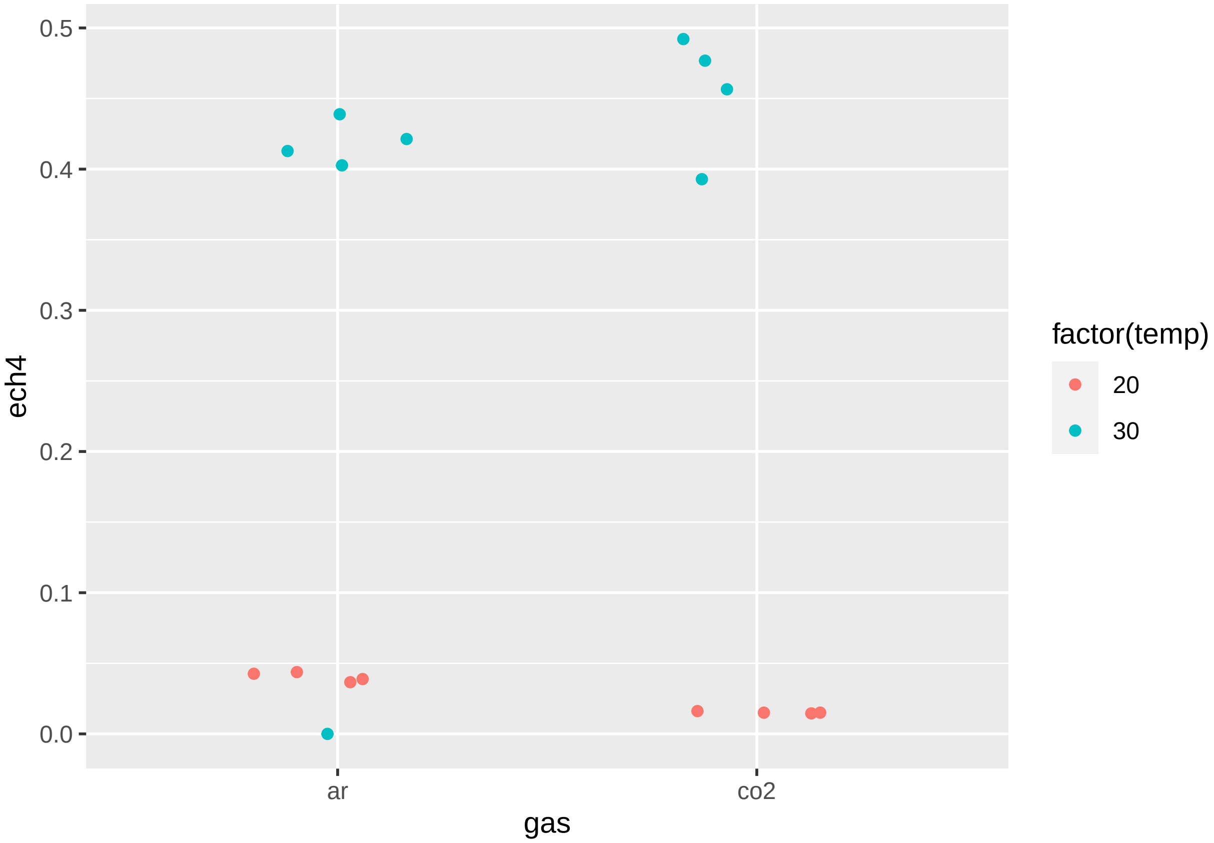
```
dat <- dat[!is.na(qch4), ]
dim(dat)
```

```
## [1] 348   9
```

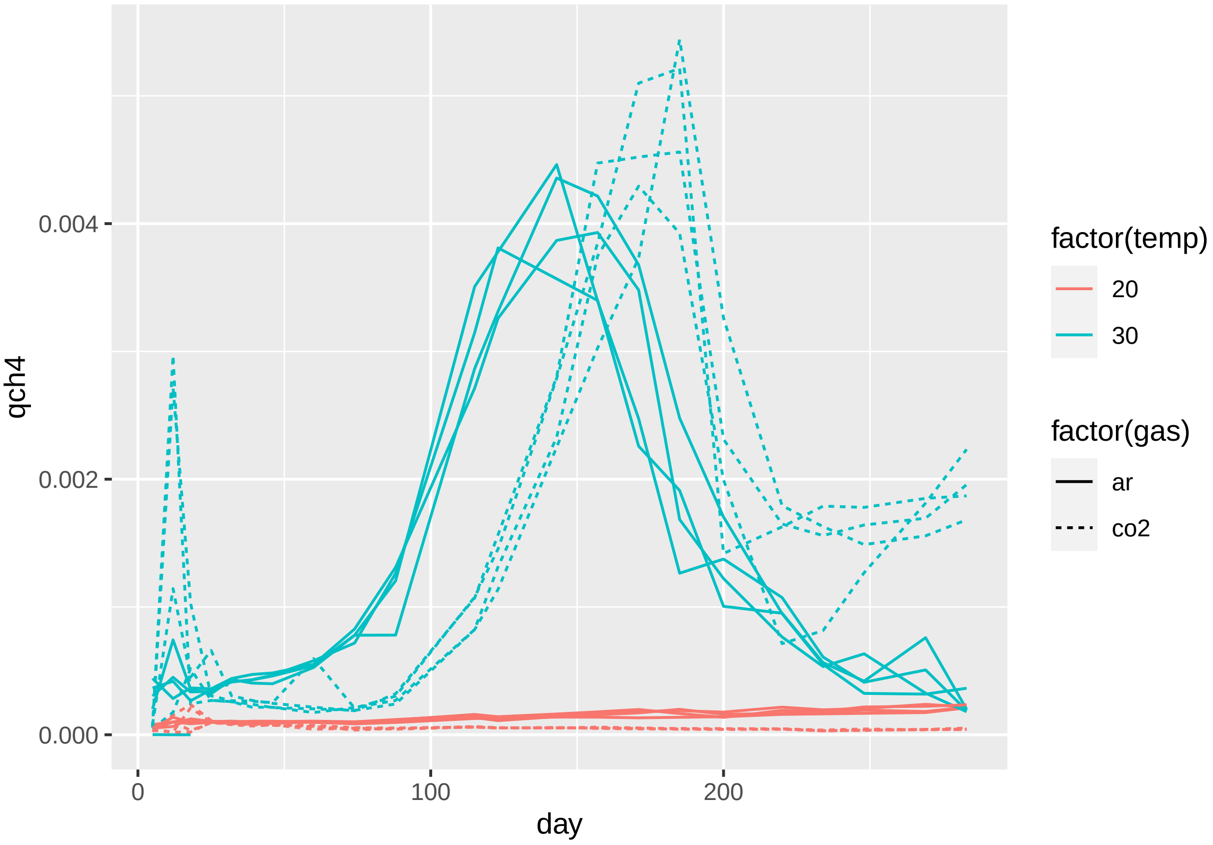
```
dat[, ech4 := mintegrate(day, qch4), by = .(reactor, gas, temp)]

dattot <- dat[, .(ech4 = mintegrate(day, qch4, value = 'total')), by = .(reactor, gas, temp)]
```

Plot

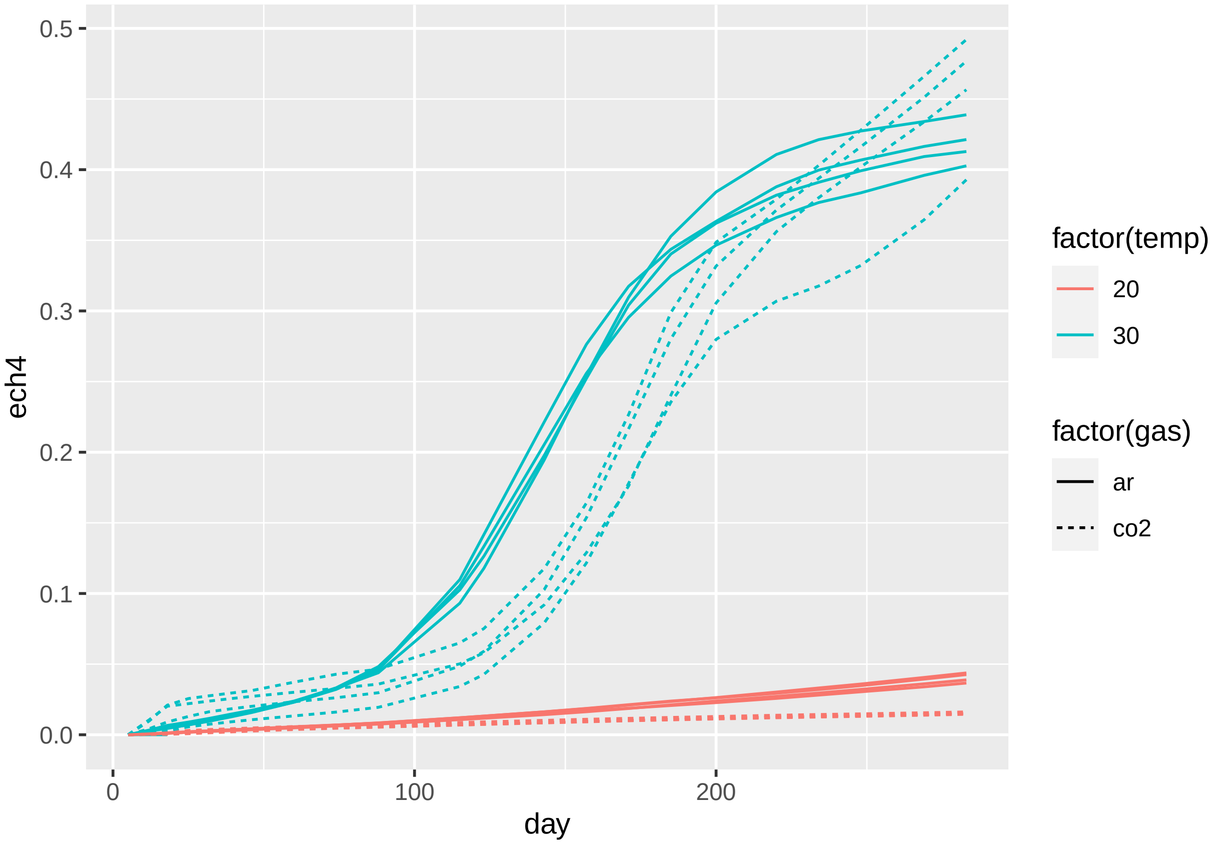
```
ggplot(dattot, aes(gas, ech4, colour = factor(temp))) +
  geom_jitter(height = 0, width = 0.2)
```



```
ggplot(dat, aes(day, qch4, colour = factor(temp), lty = factor(gas), group = reactor)) +
  geom_line()
```



```
ggplot(dat, aes(day, ech4, colour = factor(temp), lty = factor(gas), group = reactor)) +
  geom_line()
```



Replication

```
table(dattot[, .(gas, temp)])
```

```
##      temp
## gas   20 30
##   ar   4  5
##   co2  4  4
```

```
table(dattot[, .(reactor, gas, temp)])
```

```
## , , temp = 20
##
##        gas
## reactor ar co2
##     bg   0   0
##     R1   0   1
##     R10  1   0
##     R11  1   0
##     R12  1   0
##     R13  0   0
##     R14  0   0
##     R15  0   0
##     R16  0   0
##     R2   0   1
##     R3   0   1
##     R4   0   1
##     R5   0   0
##     R6   0   0
##     R7   0   0
##     R8   0   0
##     R9   1   0
##
## , , temp = 30
##
##        gas
## reactor ar co2
##     bg   1   0
##     R1   0   0
##     R10  0   0
##     R11  0   0
##     R12  0   0
##     R13  1   0
##     R14  1   0
##     R15  1   0
##     R16  1   0
##     R2   0   0
##     R3   0   0
##     R4   0   0
##     R5   0   1
##     R6   0   1
##     R7   0   1
##     R8   0   1
##     R9   0   0
```

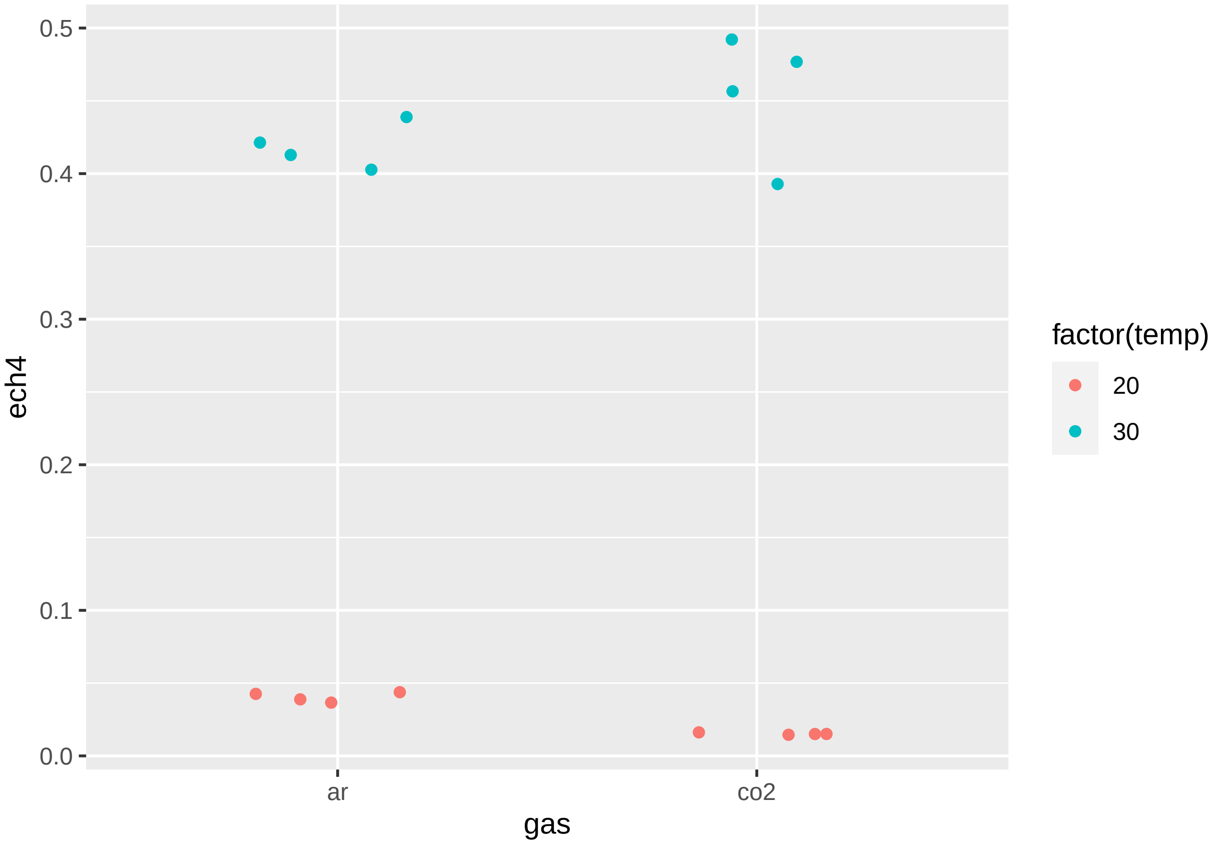
Drop background.

```
dattot <- dattot[!reactor == 'bg', ]
```

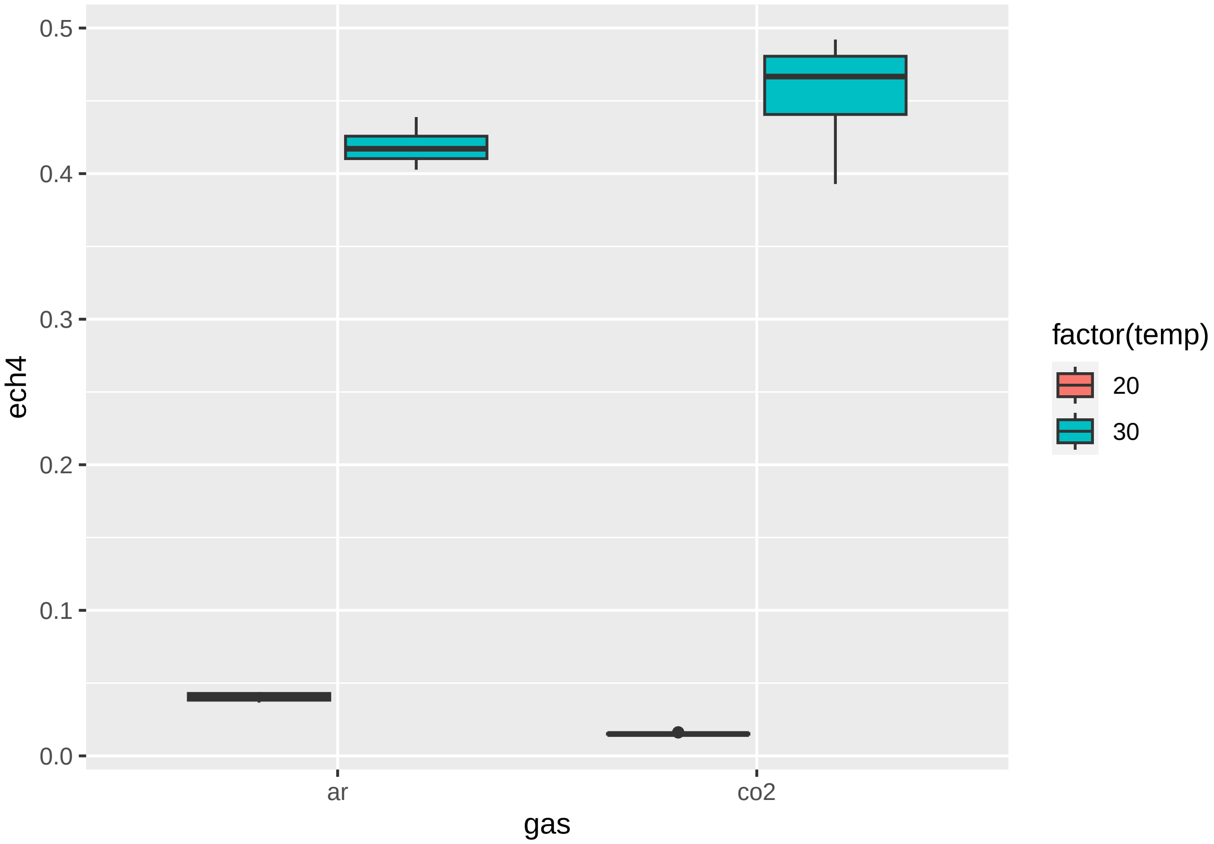
```
table(dattot[, .(gas, temp)])
```

```
##      temp
## gas   20 30
##   ar   4  4
##   co2  4  4
```

```
ggplot(dattot, aes(gas, ech4, colour = factor(temp))) +
  geom_jitter(height = 0, width = 0.2)
```

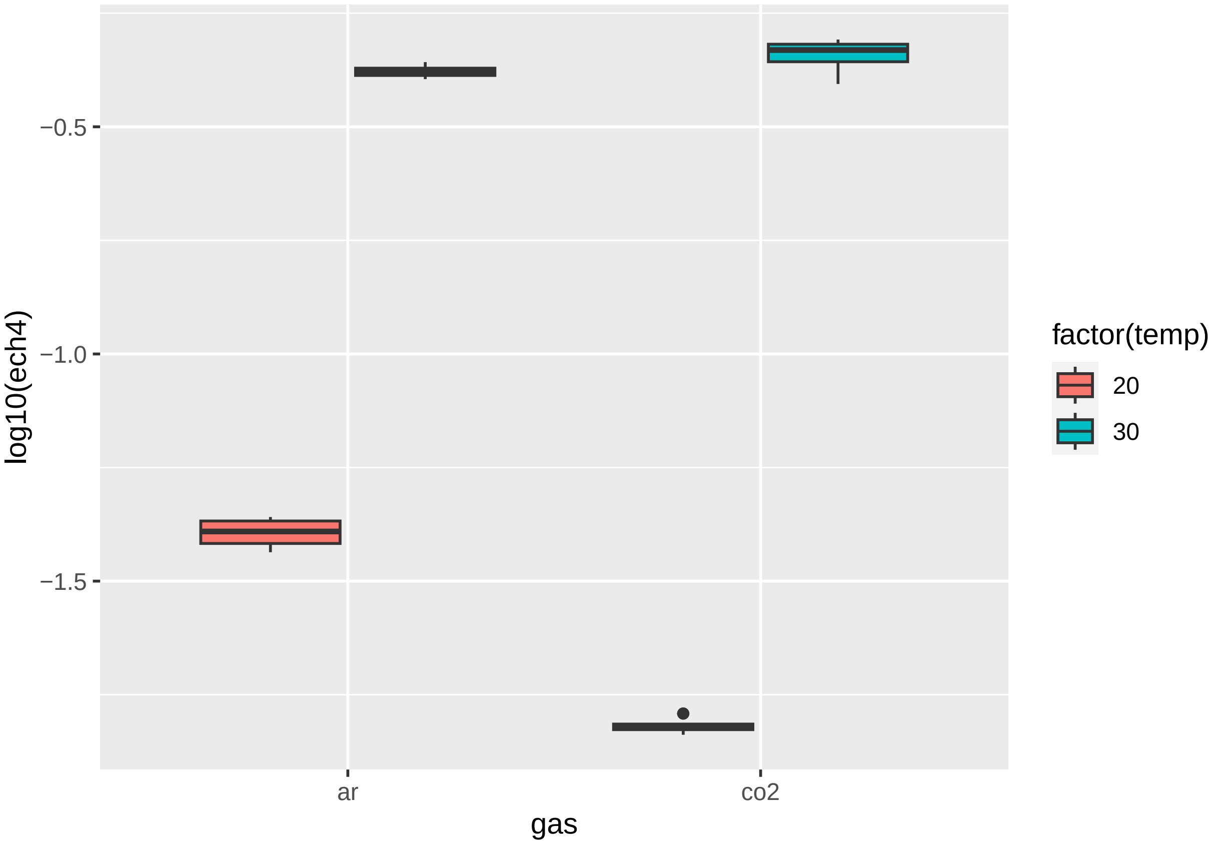


```r
ggplot(dattot, aes(gas, ech4, fill = factor(temp))) +
  geom_boxplot()
```



Look at log transformation. Looks like constant variance assumption would be better. We can compare with and without.

```
ggplot(dattot, aes(gas, log10(ech4), fill = factor(temp))) +
  geom_boxplot()
```



Statistical comparisons

```
dattot[, temp := factor(temp)]
dattot[, gas := factor(gas)]
```

```
mod1 <- aov(ech4 ~ temp*gas, data = dattot)
summary(mod1)
```

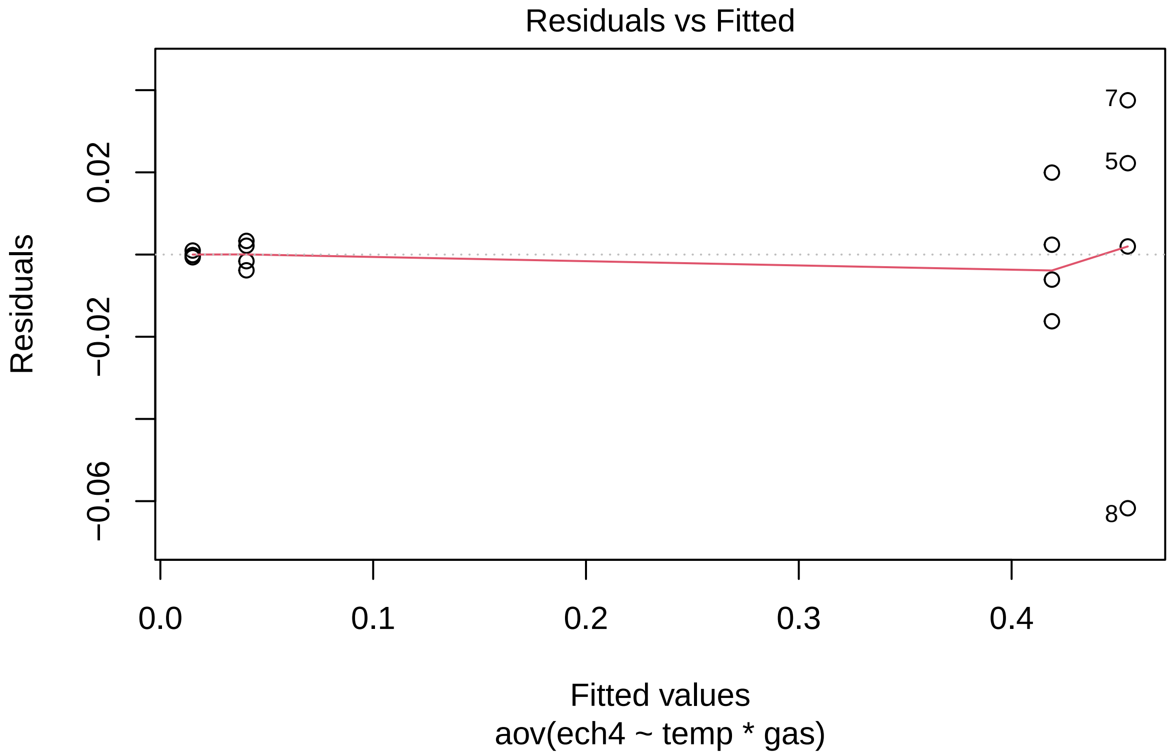
```
##             Df Sum Sq Mean Sq  F value   Pr(>F)    
## temp         1 0.6689  0.6689 1243.624 1.73e-13 ***
## gas          1 0.0001  0.0001    0.201   0.6621    
## temp:gas     1 0.0037  0.0037    6.900   0.0221 *  
## Residuals   12 0.0065  0.0005                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

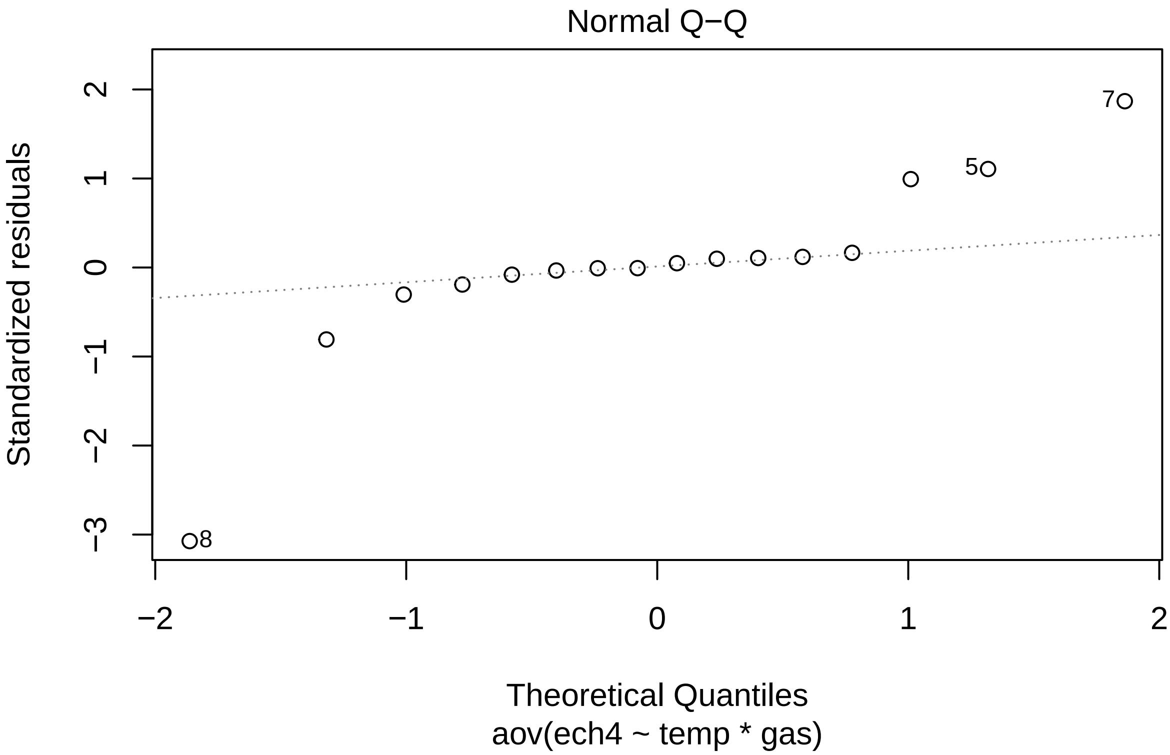
```
TukeyHSD(mod1)
```

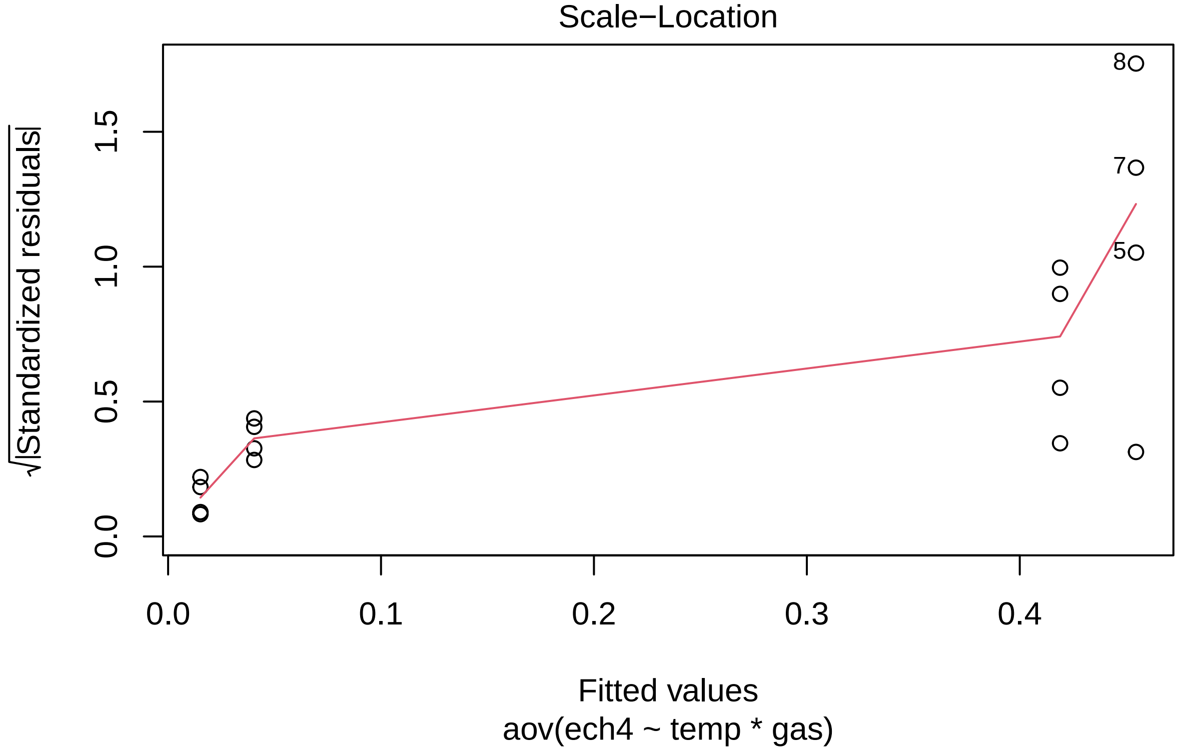
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
## 
## Fit: aov(formula = ech4 ~ temp * gas, data = dattot)
## 
## $temp
##            diff       lwr       upr p adj
## 30-20 0.4089365 0.3836708 0.4342022     0
## 
```

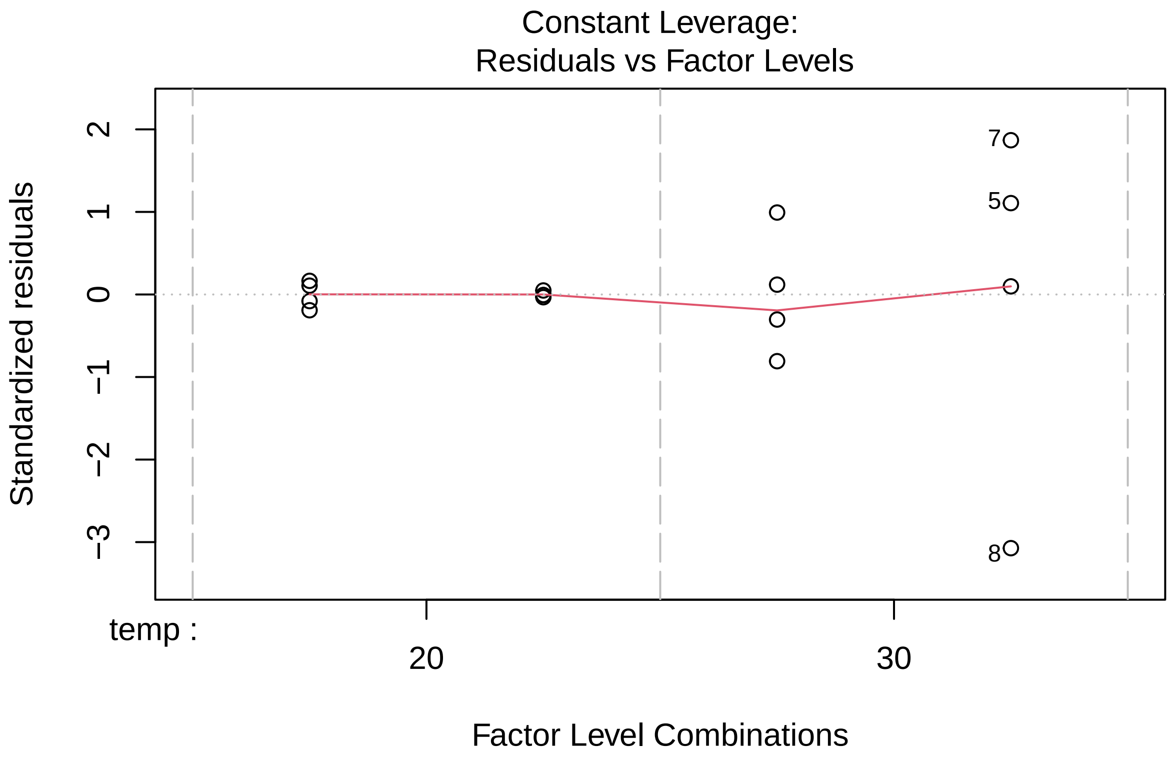
```
## $gas
##              diff         lwr        upr     p adj
## co2-ar 0.00519578 -0.02006991 0.03046147 0.6620896
##
## $`temp:gas`
##                      diff         lwr         upr     p adj
## 30:ar-20:ar    0.37847576  0.32978775  0.42716376 0.0000000
## 20:co2-20:ar  -0.02526494 -0.07395294  0.02342306 0.4454034
## 30:co2-20:ar   0.41413226  0.36544426  0.46282026 0.0000000
## 20:co2-30:ar  -0.40374070 -0.45242870 -0.35505270 0.0000000
## 30:co2-30:ar   0.03565650 -0.01303150  0.08434450 0.1856289
## 30:co2-20:co2  0.43939720  0.39070920  0.48808520 0.0000000
```

```
plot(mod1, ask = FALSE)
```




Normal Q−Q
Standardized residuals
2
1
0
−1
−2
−3
7
5
8
−2
−1
0
1
2
Theoretical Quantiles
aov(ech4 ~ temp * gas)


Scale−Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
8
7
5
0.0
0.1
0.2
0.3
0.4
Fitted values
aov(ech4 ~ temp * gas)



```
confint(mod1)
```

```
##                       2.5 %     97.5 %
## (Intercept)     0.01518798 0.06571935
## temp30          0.34274468 0.41420684
## gasco2         -0.06099602 0.01046614
## temp30:gasco2   0.01039006 0.11145282
```

Now with transformation.

```
mod2 <- aov(log10(ech4) ~ temp*gas, data = dattot)
summary(mod2)
```

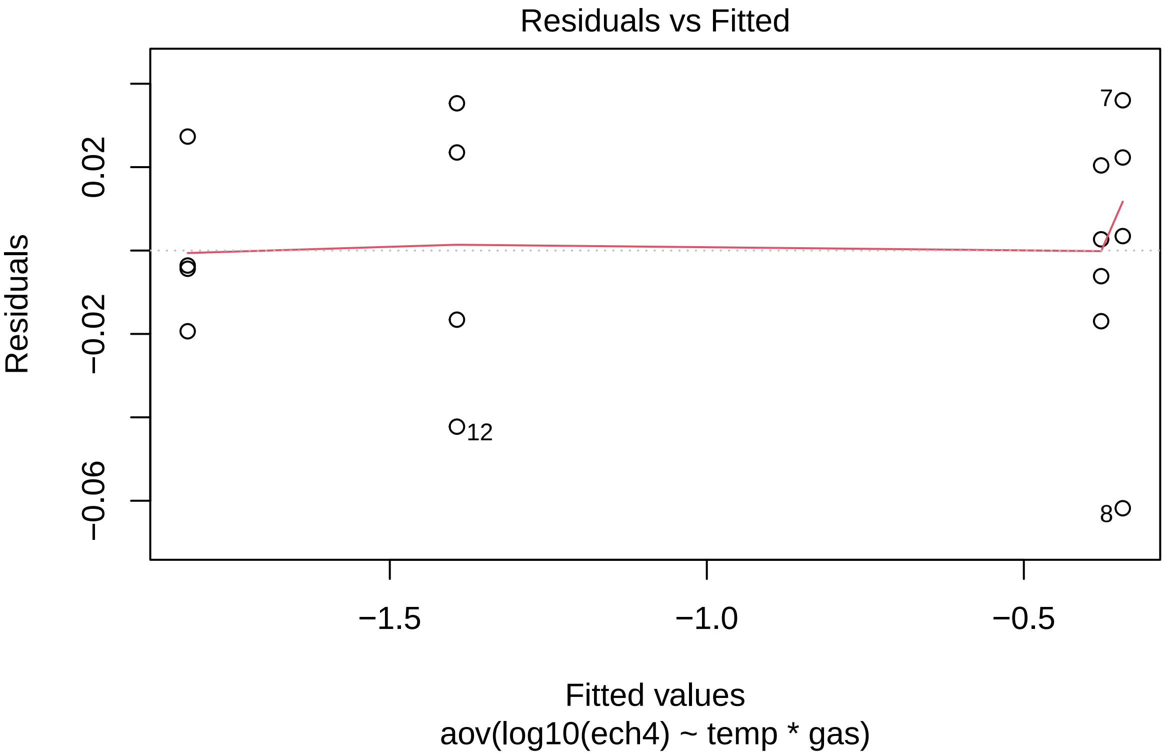
```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## temp         1  6.205   6.205  6538.4  < 2e-16 ***
## gas          1  0.153   0.153   160.7 2.61e-08 ***
## temp:gas     1  0.210   0.210   221.8 4.22e-09 ***
## Residuals   12  0.011   0.001                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

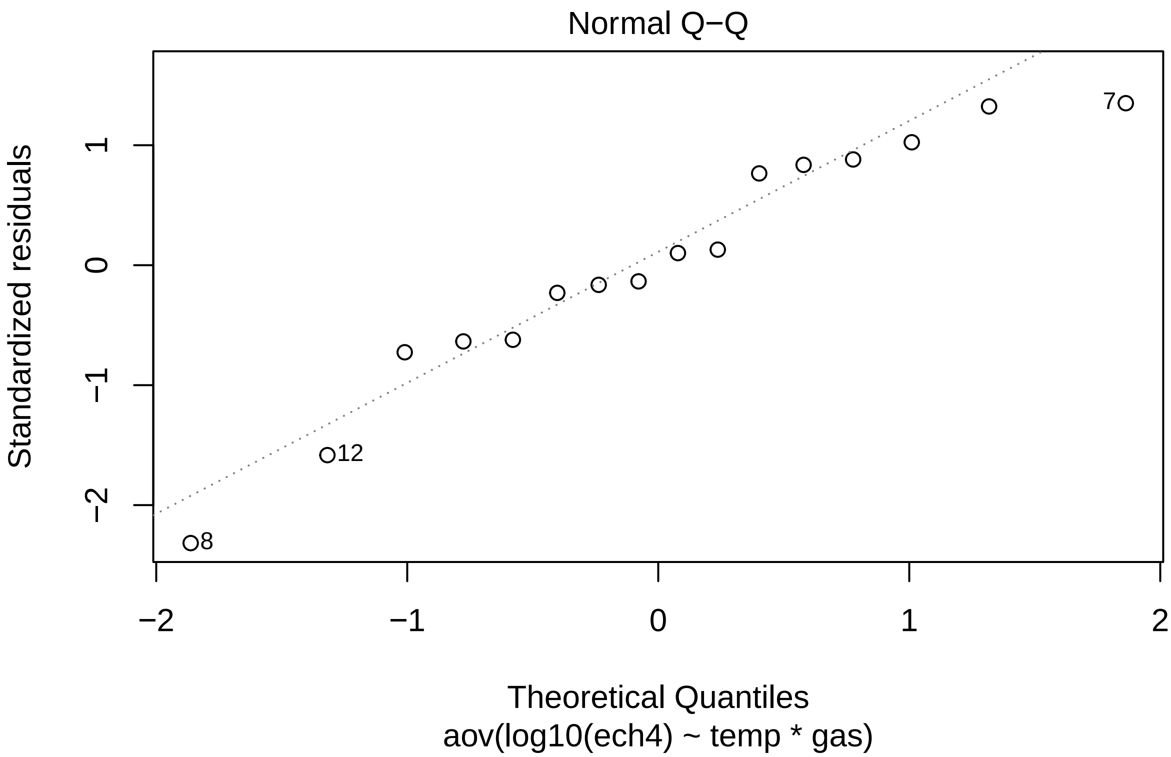
```
TukeyHSD(mod2)
```

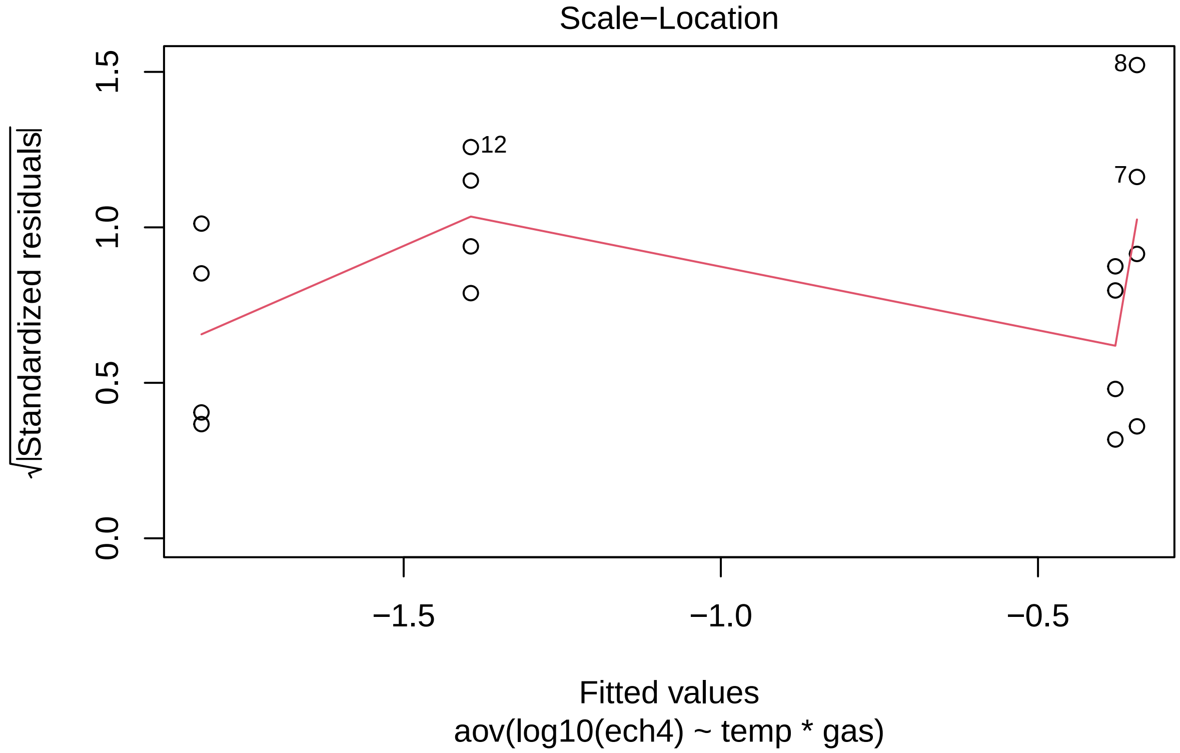
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = log10(ech4) ~ temp * gas, data = dattot)
##
## $temp
##           diff      lwr     upr p adj
## 30-20 1.245461 1.211902 1.27902     0
##
```

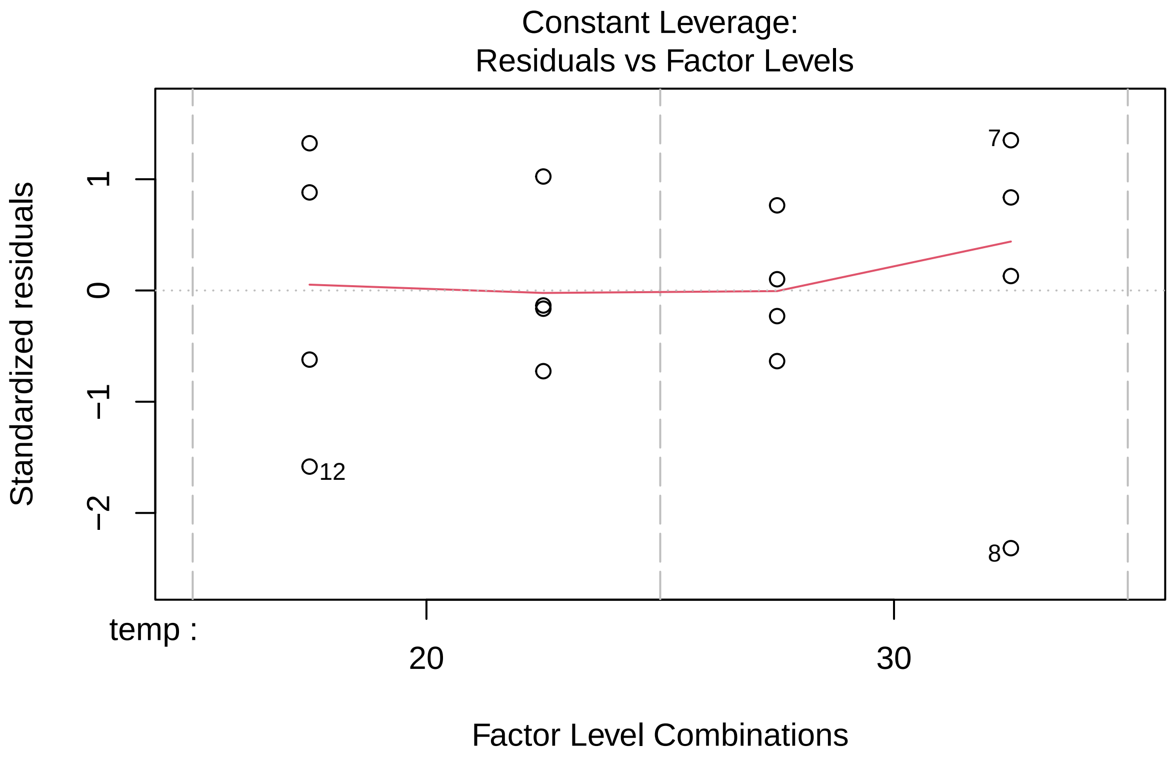
```
## $gas
##              diff        lwr        upr p adj
## co2-ar -0.1952774 -0.2288367 -0.1617181     0
##
## $`temp:gas`
##                     diff         lwr         upr     p adj
## 30:ar-20:ar    1.0160713  0.95140119  1.08074145 0.0000000
## 20:co2-20:ar  -0.4246669 -0.48933704 -0.35999678 0.0000000
## 30:co2-20:ar   1.0501834  0.98551329  1.11485355 0.0000000
## 20:co2-30:ar  -1.4407382 -1.50540836 -1.37606810 0.0000000
## 30:co2-30:ar   0.0341121 -0.03055803  0.09878223 0.4320347
## 30:co2-20:co2  1.4748503  1.41018020  1.53952046 0.0000000
```

```r
plot(mod2, ask = FALSE)
```




Normal Q−Q
Standardized residuals
1
0
−1
−2
7
12
8
−2
−1
0
1
2
Theoretical Quantiles
aov(log10(ech4) ~ temp * gas)


Scale−Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
8
12
7
−1.5
−1.0
−0.5
Fitted values
aov(log10(ech4) ~ temp * gas)



```r
confint(mod2)
```

```
##                      2.5 %     97.5 %
## (Intercept)   -1.4277061 -1.3605874
## temp30         0.9686113  1.0635313
## gasco2        -0.4721269 -0.3772069
## temp30:gasco2  0.3916604  0.5258976
```

Back-transform.

```r
10^confint(mod2) - 1
```

```
##                      2.5 %     97.5 %
## (Intercept)   -0.9626497 -0.9564074
## temp30         8.3027490 10.5752757
## gasco2        -0.6628113 -0.5804409
## temp30:gasco2  1.4641118  2.3565847
```