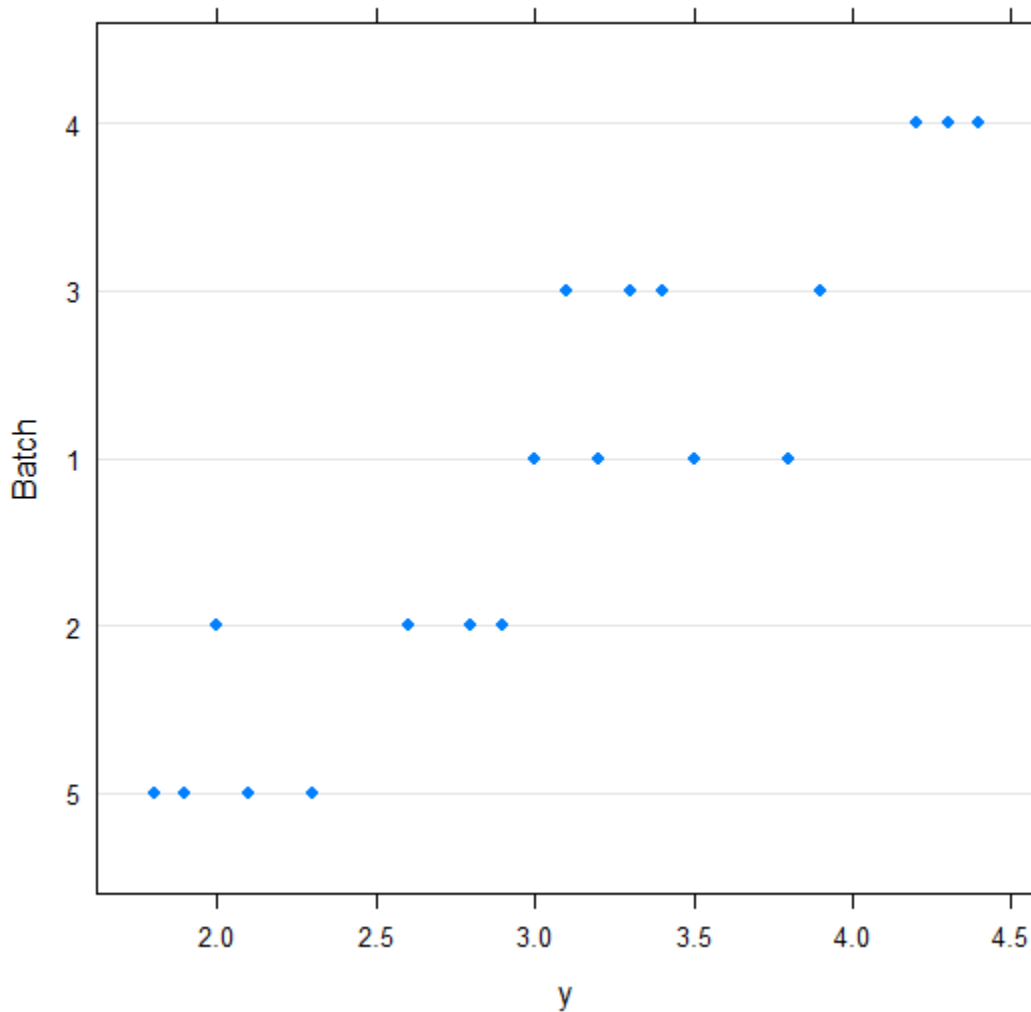


Output from drug.R

```
> # drug.R
>
> library(nlme)
> library(car)
>
> drugdata <- read.table(file="drug.dat",header=T) # read the data in
> head(drugdata)
  y batch
1 3.2    1
2 2.6    2
3 3.4    3
4 4.2    4
5 1.8    5
6 3.8    1
> drugdata$Batch <- factor(drugdata$batch)
>
> drug2 <- groupedData(y~1|Batch, data=drugdata)
>
> plot(drug2)
```



```
>
> m1 <- lme(y~1,data=drug2, random= ~1|Batch)
```

```

> summary(m1)
Linear mixed-effects model fit by REML
Data: drug2
      AIC      BIC    logLik
31.47578 34.30909 -12.73789

Random effects:
Formula: ~1 | Batch
      (Intercept)  Residual
StdDev:   0.8507595 0.3030402

Fixed effects: y ~ 1
      Value Std.Error DF   t-value p-value
(Intercept) 3.135 0.3864583 15  8.112131    0

Standardized Within-Group Residuals:
      Min      Q1      Med      Q3      Max
-1.95425202 -0.53210771 -0.01369616  0.56752282  1.59687043

Number of Observations: 20
Number of Groups: 5
> # notice that the summary of model m1 gives estimates of sigma_a and
> # sigma, not sigma^2_a and sigma^2. These estimates are given in the
> # following portion of the summary:
>
> # Random effects:
> # Formula: ~1 | Batch
> #      (Intercept)  Residual
> # StdDev:   0.8507595 0.3030402
>
> # These values agree with the estimates given by SAS since
> # 0.8507595^2 = .7238, and 0.3030402^2 = .09183
>
> # Confidence intervals for sigma_a and sigma are given by the following
> # command. Note these are for sigma_a and sigma, not sigma^2_a and sigma^2
> # Furthermore, they are obtained with a different method than the one used in
> # SAS and presented in the lecture notes, so these intervals don't agree with
> # those given by PROC MIXED. E.g., you can't square the endpoints of these
> # intervals to get the intervals given by SAS. I recommend using the intervals
> # that are given by SAS and in my lecture notes rather than those given by
> # the intervals() function as illustrated below.
>
> intervals(m1,which="var-cov")
Approximate 95% confidence intervals

Random Effects:
Level: Batch
      lower      est.      upper
sd((Intercept)) 0.4161843 0.8507595 1.739113

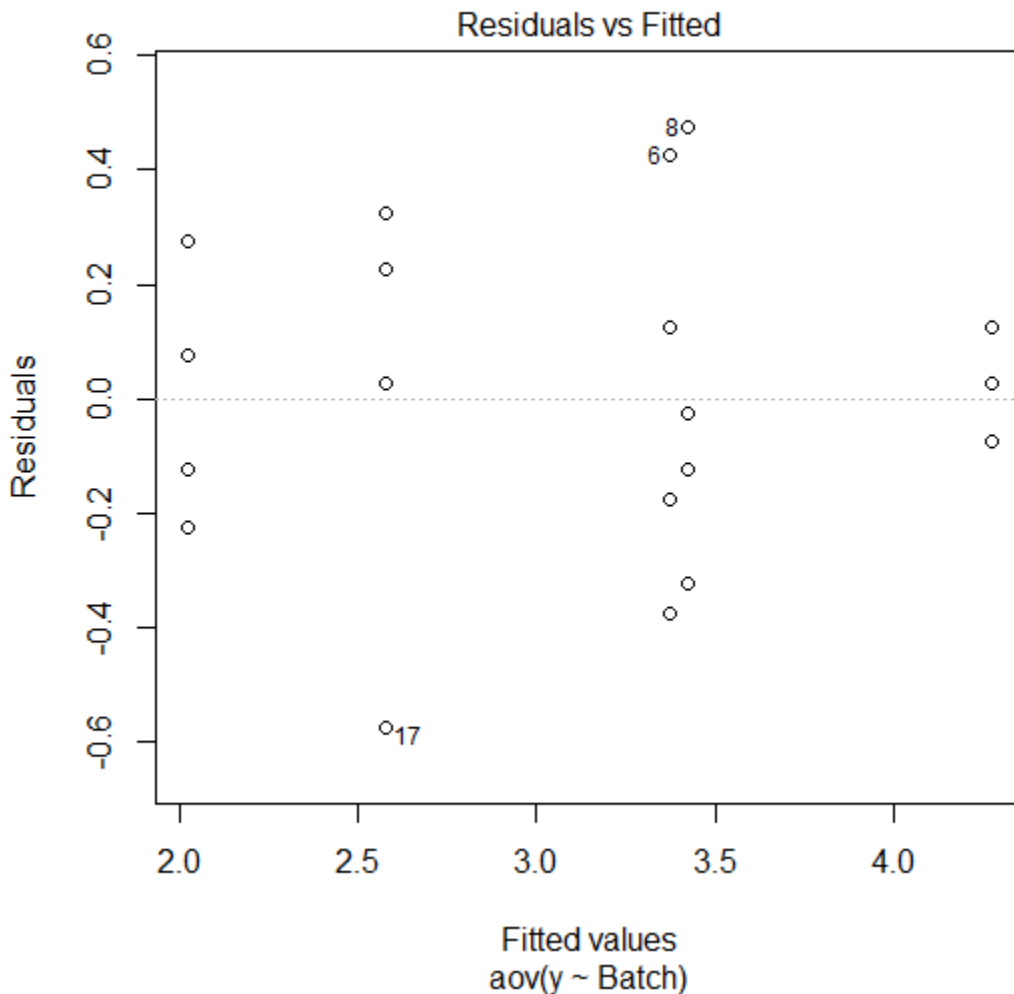
Within-group standard error:
      lower      est.      upper
0.2118814 0.3030402 0.4334186
>
> # Levene's test:
> # First fit the model as a fixed effect model
> m2 <- aov(y~Batch,drug2)

```

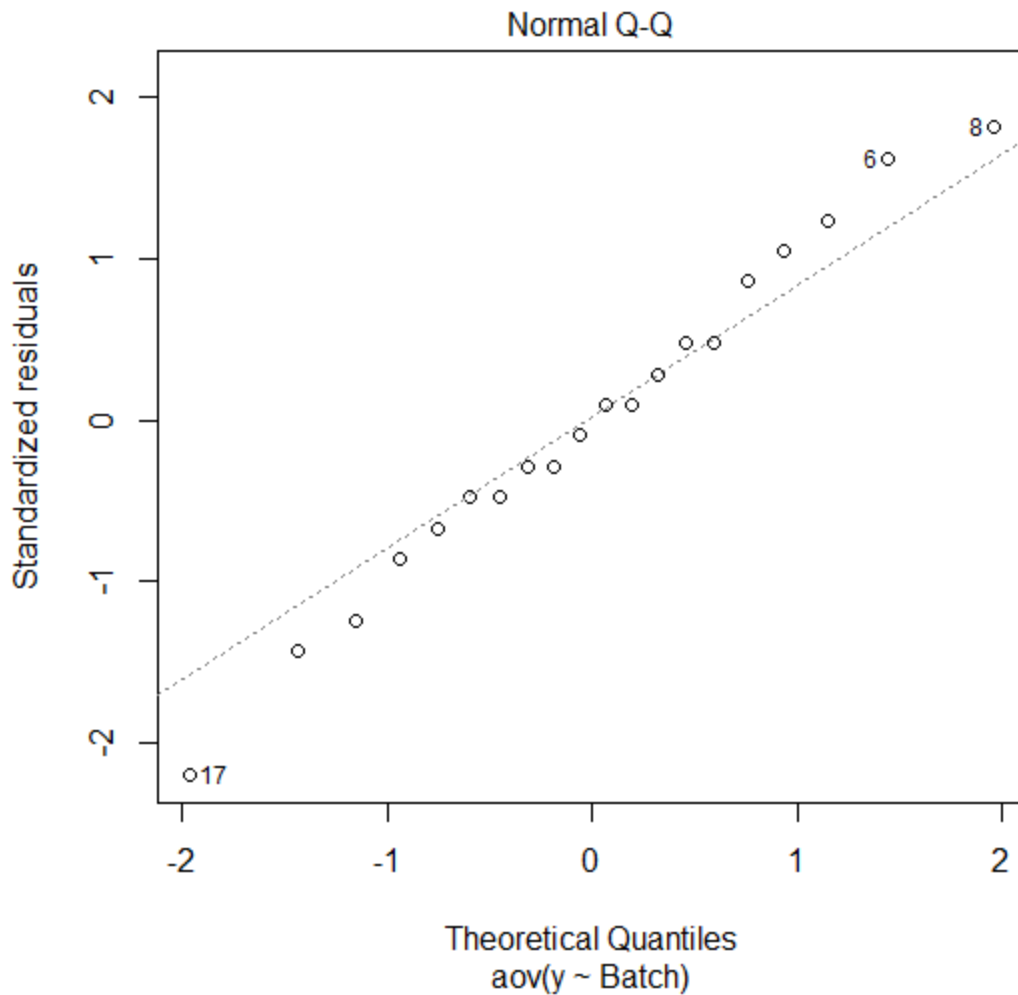
```

>
> # then conduct the test:
> leveneTest(m2) # agrees with SAS
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  4  0.8116 0.5371
      15
>
> # If you use the plot function and give it a fitted linear model as its
> # argument it produces several different types of residual plots. There are
> # actually 6 such plots that plot can produce when it is given a fitted linear
> # model as its argument. Specific plots among these 6 can be chosen with the
> # which argument. See ?plot.lm for details. Here, I ask for plots of the
> # residuals versus fitted values and a quantile-quantile (Q-Q) plot to check
> # normality of the residuals. Raw residuals are used for the resid vs fitted
> # and standardized residuals are used for the Q-Q plot.
>
> plot(m2,which=c(1,2),add.smooth=F)
Waiting to confirm page change...

```

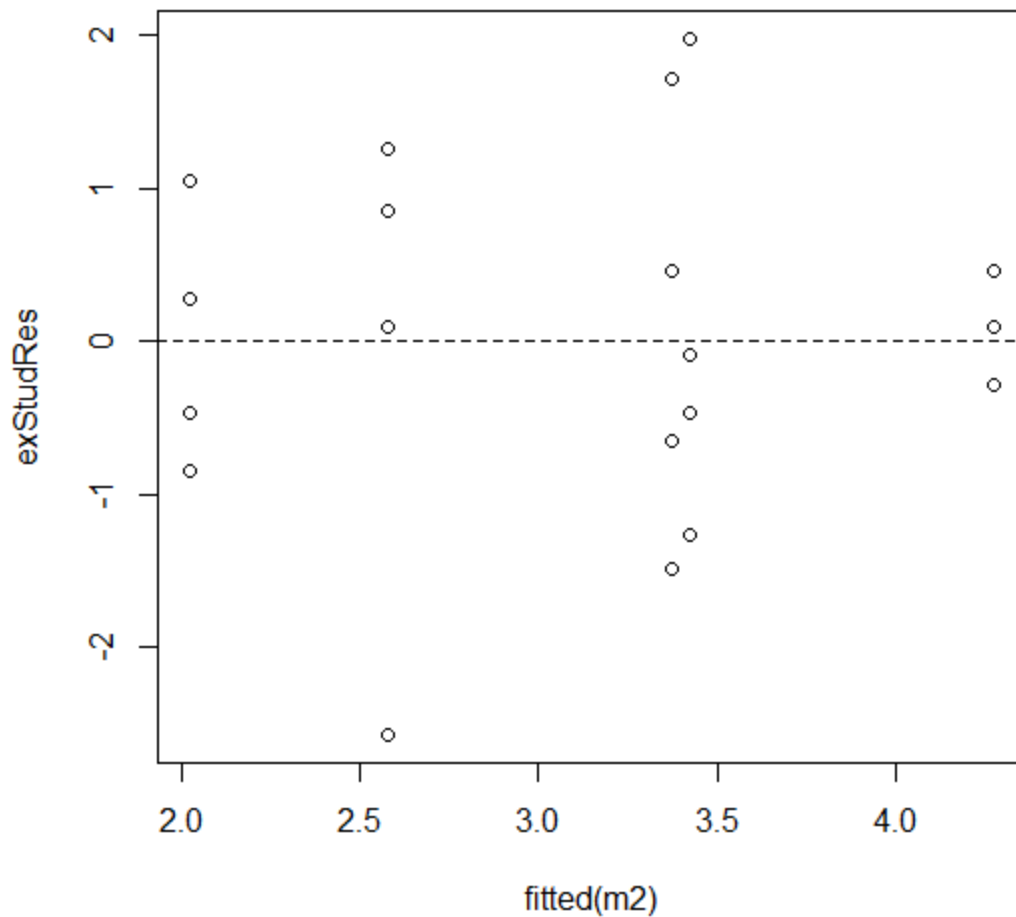


Waiting to confirm page change...



```
>
> # If we prefer to look at externally Studentized residuals, we can do that too
> exStudRes <- rstudent(m2) # extract the externally Studentized residis from m2
> plot(fitted(m2),exStudRes)# alternative syntax: plot(exStudRes~fitted(m2))
> title(main="Externally Studentized residis vs fitted values, model m2")
> abline(h=0,lty=2)
```

Externally Studentized residvs vs fitted values, model m2



```
>  
> # The Shapiro-Wilk test can be conducted to test normality of residuals.  
> # Here I used the internally Studentized residus, which can be obtained using  
> # the rstandard() function  
> shapiro.test(rstandard(m2))
```

Shapiro-Wilk normality test

```
data: rstandard(m2)  
W = 0.988, p-value = 0.9944
```

```
>
```