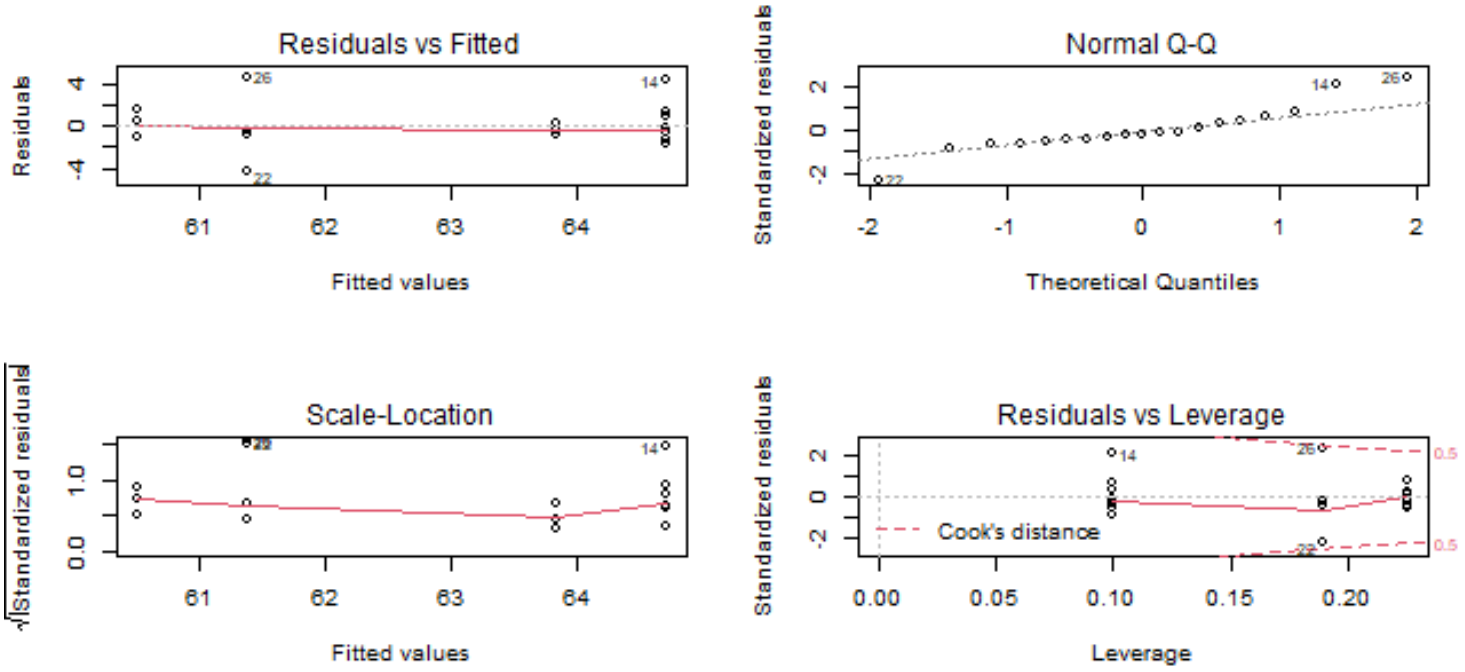
Assessing Model Assumptions

Model Assumptions

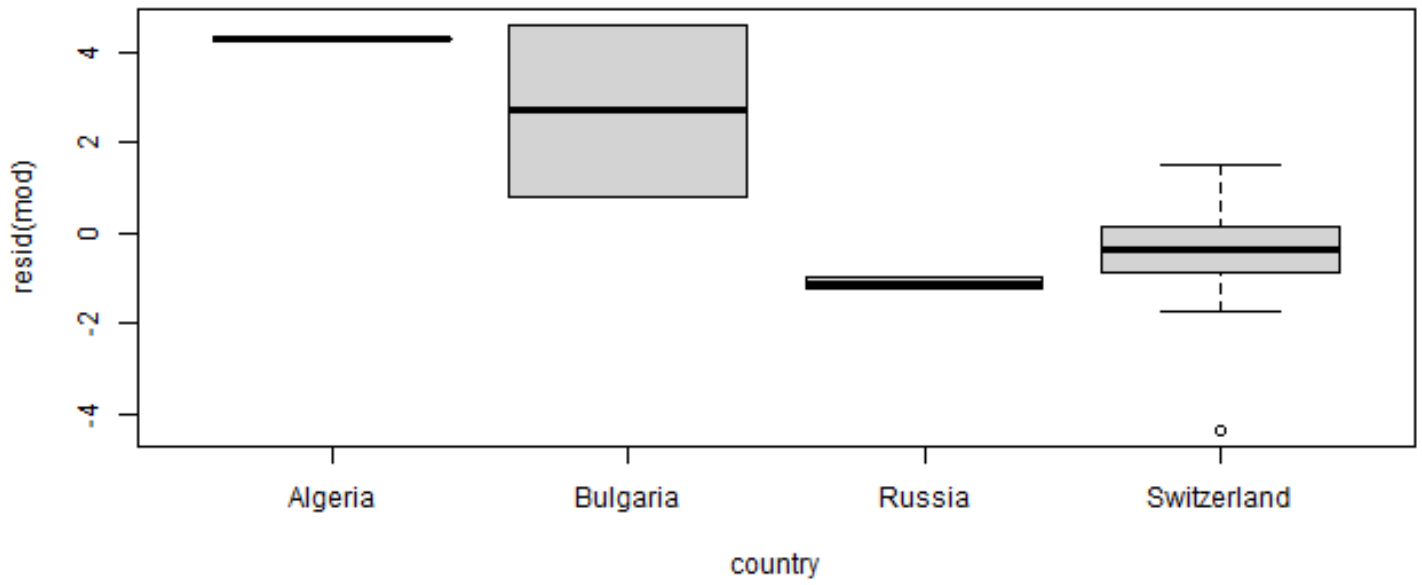
```
data("periparusater")

dat <- periparusater
mod <- lm(wing ~ sex + age, data = dat)
par(mfrow = c(2, 2))
plot(mod)
```



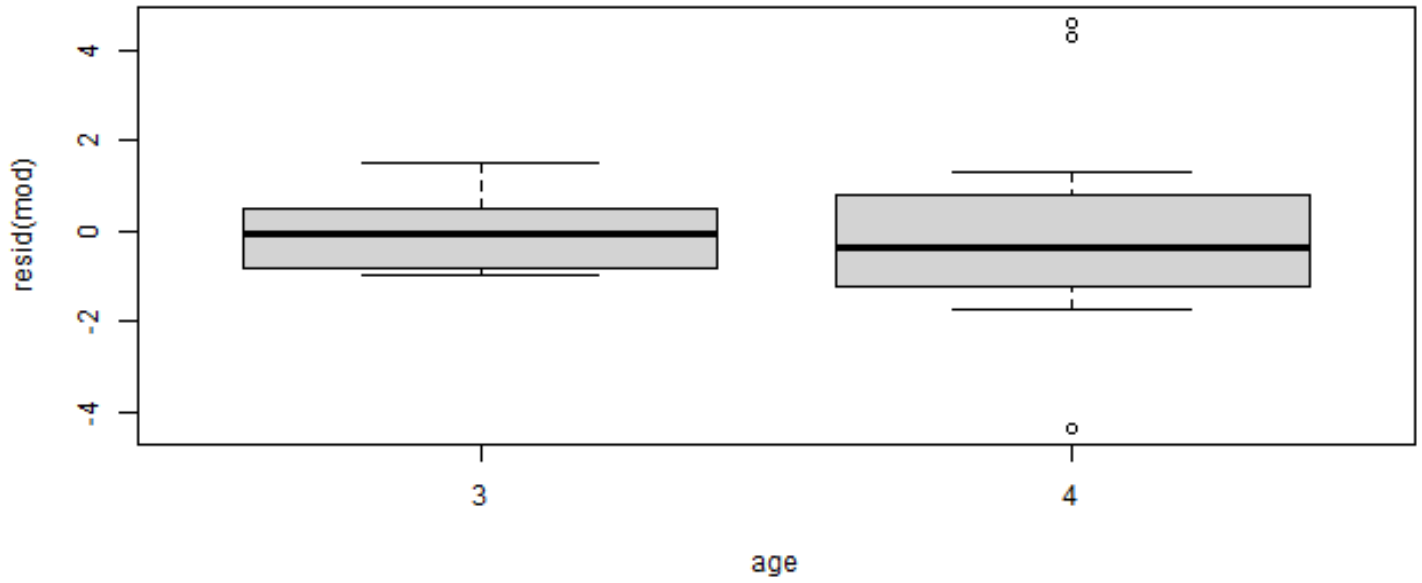

```
names(dat)
```

```
[1] "country" "age"     "sex"     "weight"  "P8"      "wing"
```

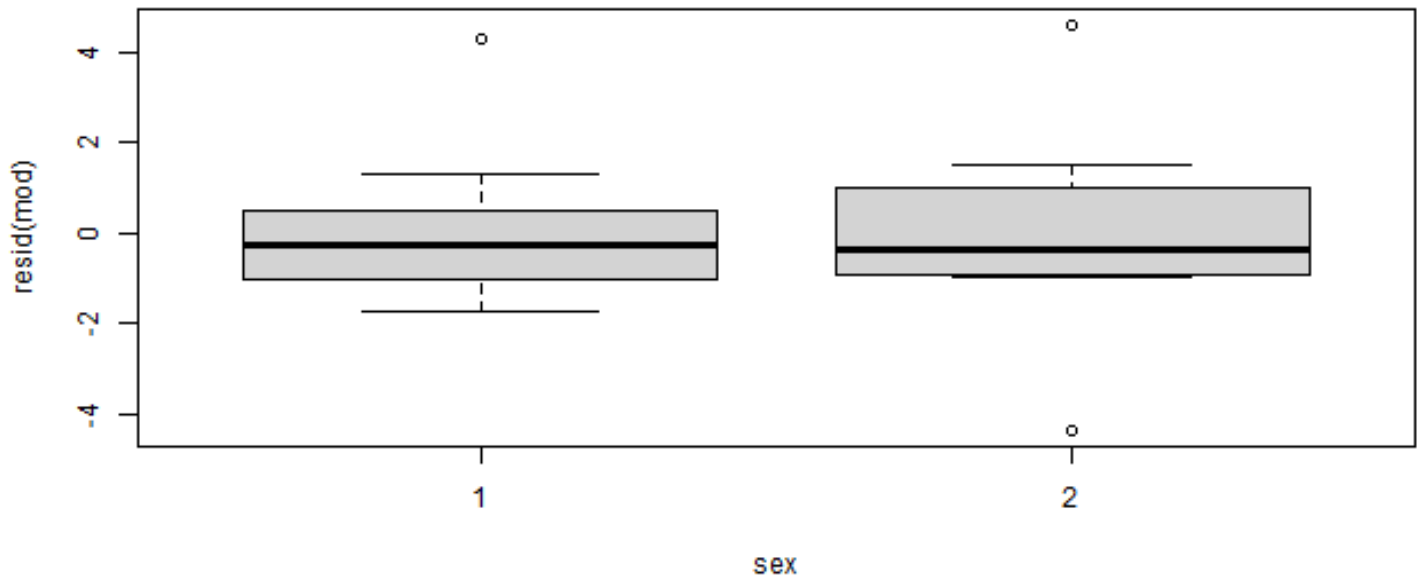
```
plot(resid(mod) ~ country, data = dat)
```



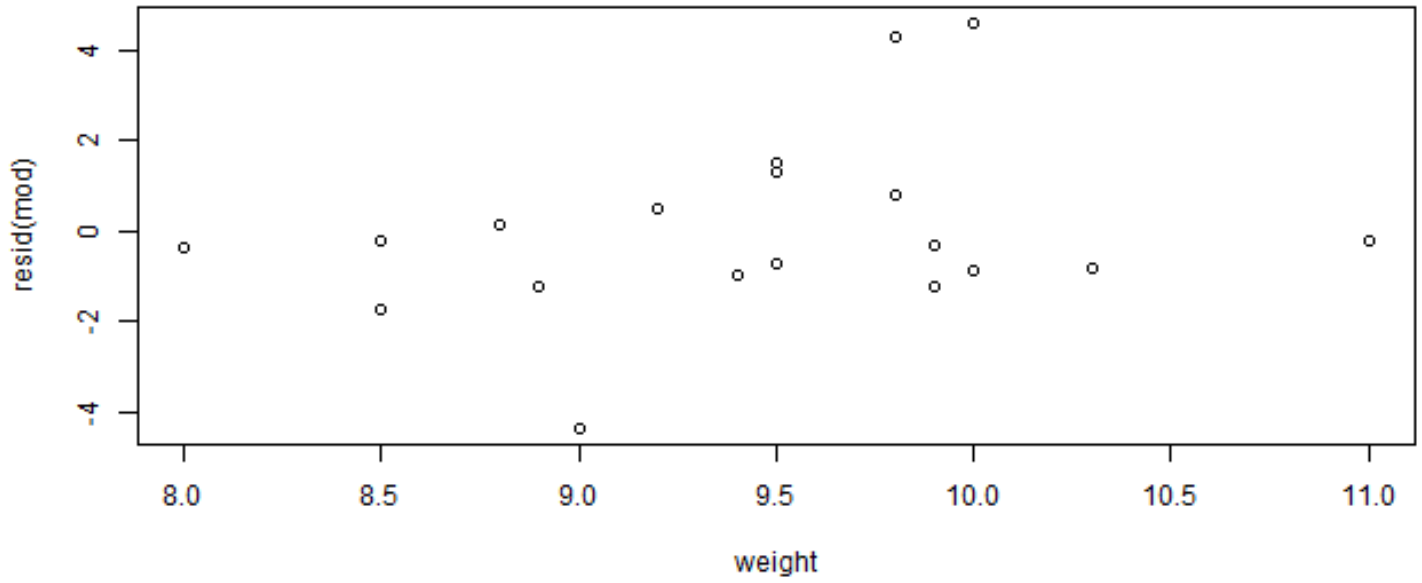

```
plot(resid(mod) ~ age, data = dat)
```




```
plot(resid(mod) ~ sex, data = dat)
```




```
plot(resid(mod) ~ weight, data = dat)
```

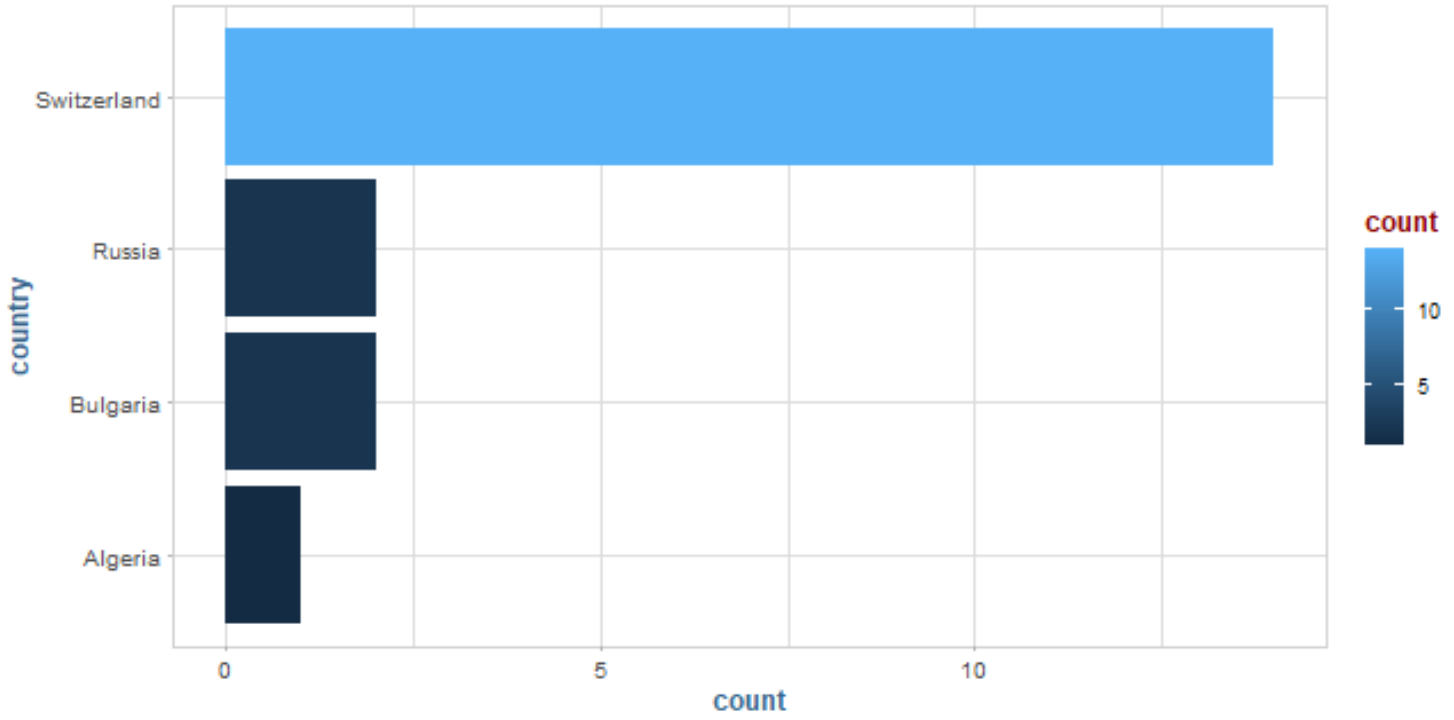



```
table(dat$country)
```

```
    Algeria    Bulgaria      Russia Switzerland
          1           2           2          14
```

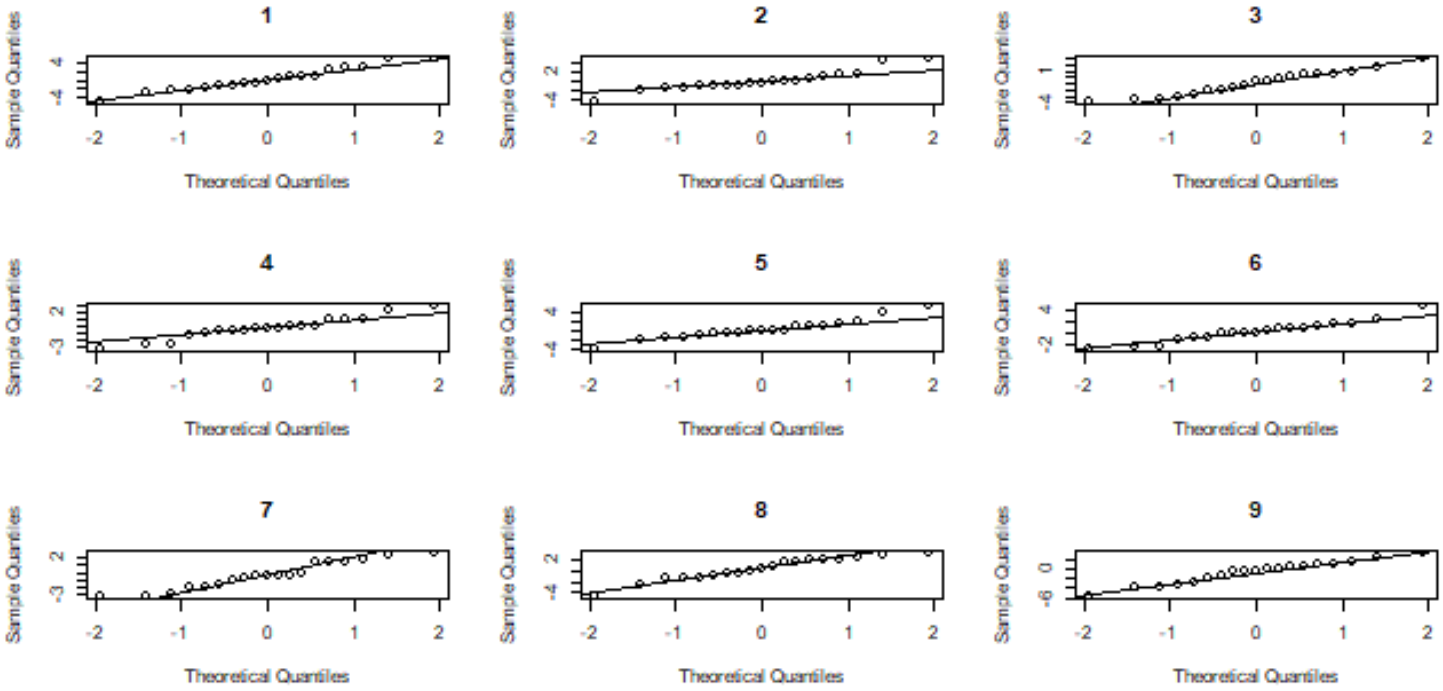
```
ggplot(dat, aes(y = country)) +
   geom_histogram(aes(fill = ..count..), stat = "count")
```

```
Warning: Ignoring unknown parameters: binwidth, bins, pad
```



The QQ

```
compareqqnorm(mod)
```




```
[1] 2
```

Temporal Autocorrelation

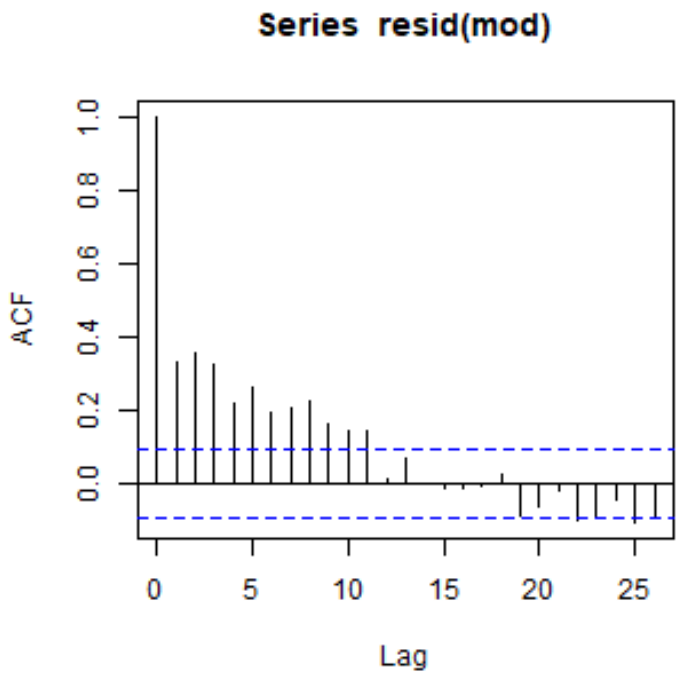
```
data("parusmajor")

dat <- parusmajor
str(dat)
```

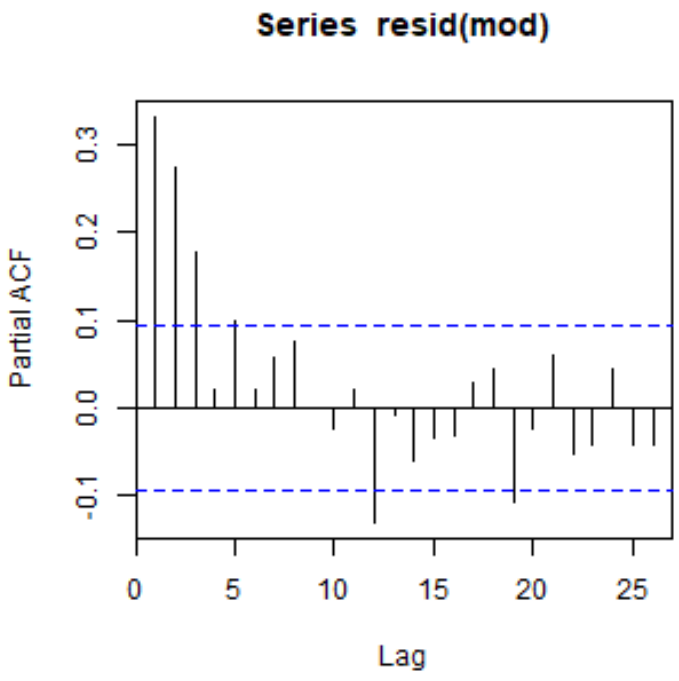
```
'data.frame':   434 obs. of  3 variables:
 $ year  : int  1996 1994 1996 2000 1988 1996 2000 2001 2005 1988 ...
 $ julian: int  266 267 267 267 268 268 268 268 268 269 ...
 $ count : int  7 1 1 3 1 3 2 1 2 2 ...
```

```
dat <- dat[order(dat$year, dat$julian), ] # order date acc, time
t.poly.jul <- poly(dat$julian, degree=2)
dat$julian.l <- t.poly.jul[, 1] # orthogonal linear trend
dat$julian.q <- t.poly.jul[, 2] # orthogonal quadratic trend

mod <- lm(count ~ year + julian.l + julian.q, data = dat )

summary(mod)
```

```
Call:
lm(formula = count ~ year + julian.l + julian.q, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-14.699  -9.835  -5.145   2.777  99.151

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -336.4073   293.6015  -1.146 0.252517
year           0.1741     0.1470   1.184 0.237001
julian.l      54.3911    16.5280   3.291 0.001081 **
julian.q     -62.7935    16.5659  -3.791 0.000172 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.53 on 430 degrees of freedom
Multiple R-squared:  0.05712,   Adjusted R-squared:  0.05054
F-statistic: 8.683 on 3 and 430 DF,  p-value: 1.323e-05
```

```
par(mfrow = c(1, 2))
acf(resid(mod))
acf(resid(mod), type = "p")
```





Autocorrelation

```
mod <- gls(count ~ year + julian.l + julian.q, data = dat,
           correlation = corAR1())

summary(mod)
```

```
Generalized least squares fit by REML
  Model: count ~ year + julian.l + julian.q
  Data: dat
       AIC      BIC    logLik
  3607.234 3631.617 -1797.617

Correlation Structure: AR(1)
 Formula: ~1
 Parameter estimate(s):
      Phi
0.3422066

Coefficients:
                Value Std.Error    t-value p-value
(Intercept) -326.8006  419.4594 -0.7790995  0.4363
year           0.1693    0.2100  0.8059450  0.4207
julian.l      48.5121   20.1315  2.4097555  0.0164
julian.q     -55.7998   22.1730 -2.5165677  0.0122

 Correlation:
         (Intr) year   juln.l
year     -1.000
julian.l -0.012  0.012
julian.q  0.062 -0.062  0.003

Standardized residuals:
       Min         Q1        Med         Q3        Max
-0.8619025 -0.5815432 -0.3272695  0.1234870  6.0001067

Residual standard error: 16.59327
Degrees of freedom: 434 total; 430 residual
```

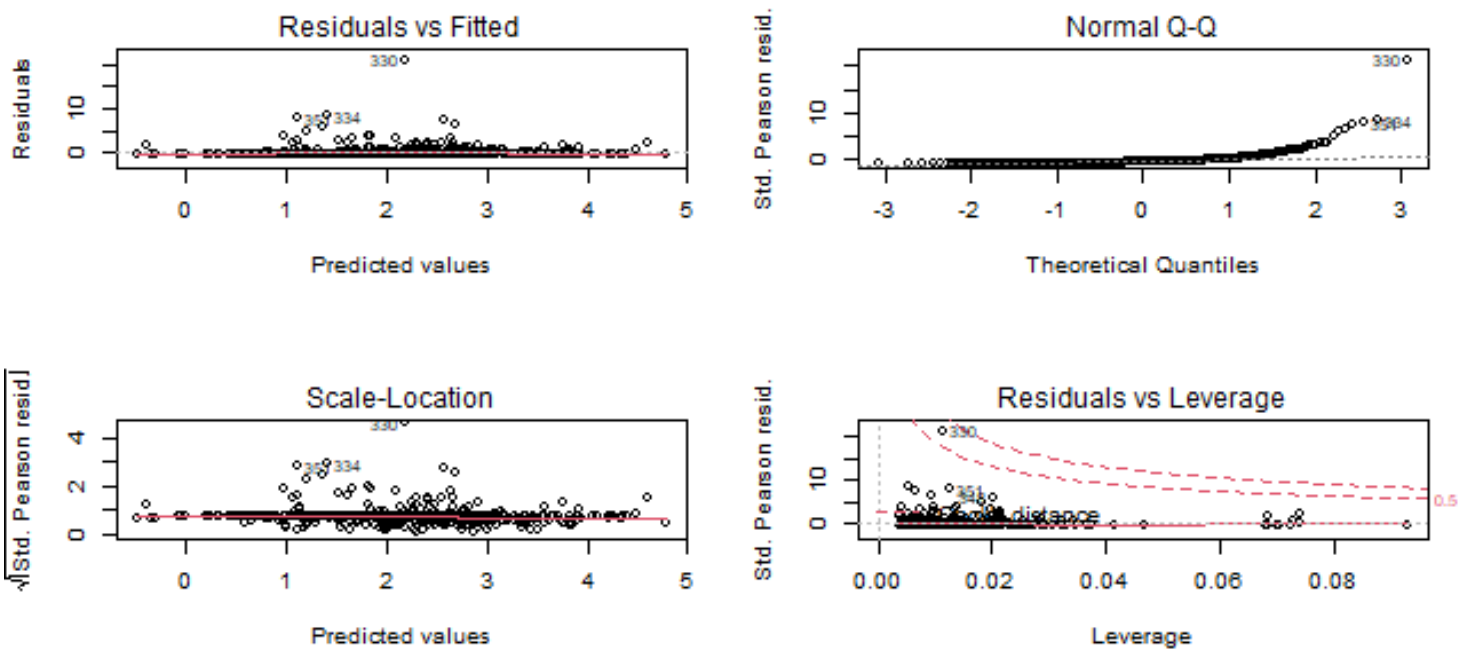
Spatial Autocorrelation

```
data("frogs")

frogs$year.z <- scale(frogs$year)
```

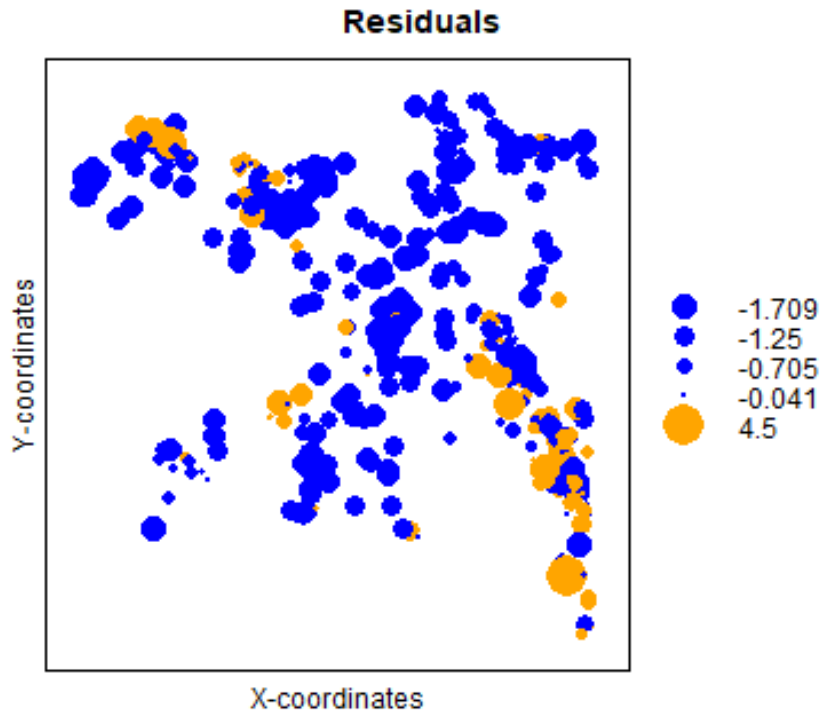
```
frogs$elevation.z <- scale(frogs$elevation)
frogs$waterarea[frogs$waterarea == 0 ] <- 0.25
frogs$waterarea.sqrt.l <- log(sqrt(frogs$waterarea))
frogs$waterarea.sqrt.l.z <- scale(frogs$waterarea.sqrt.l)
```

Negative Binomial Model

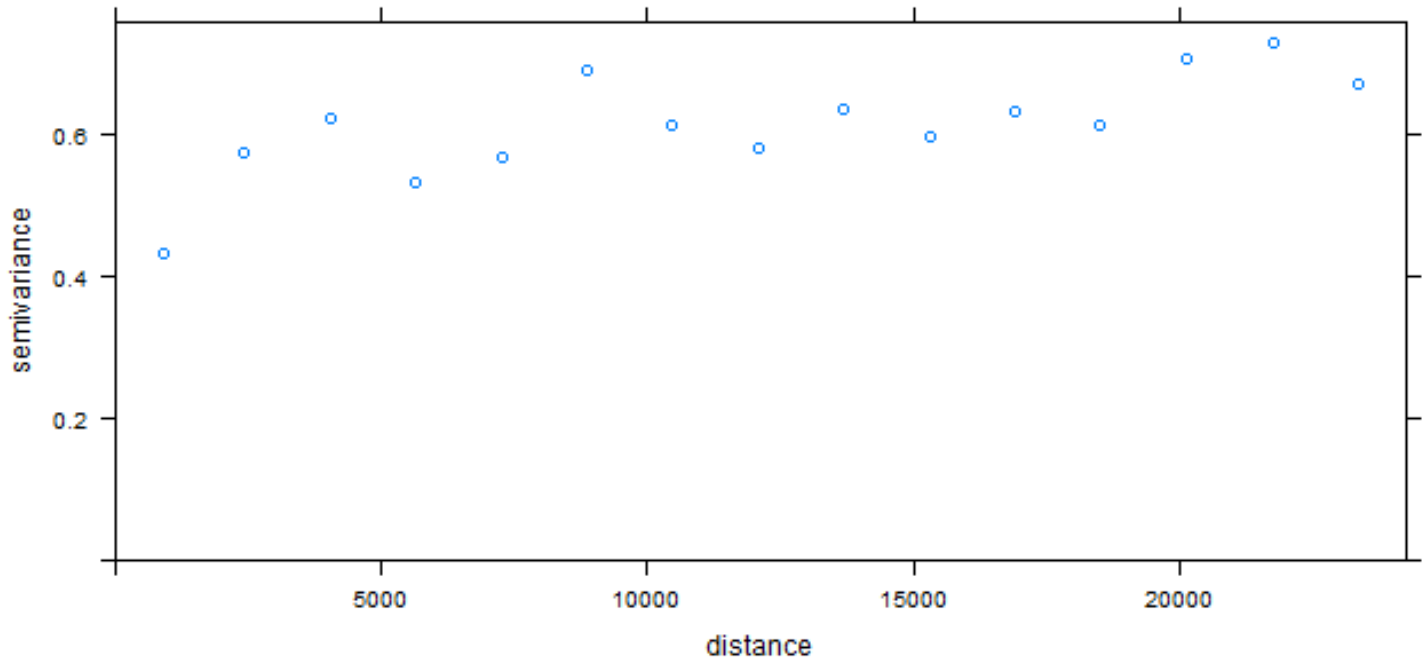
```
mod <- glm.nb(count2 ~ elevation.z + year.z + fish + vegetation + waterarea.sqrt.l.z + fish:veg

par(mfrow=c(2, 2))
plot(mod)
```



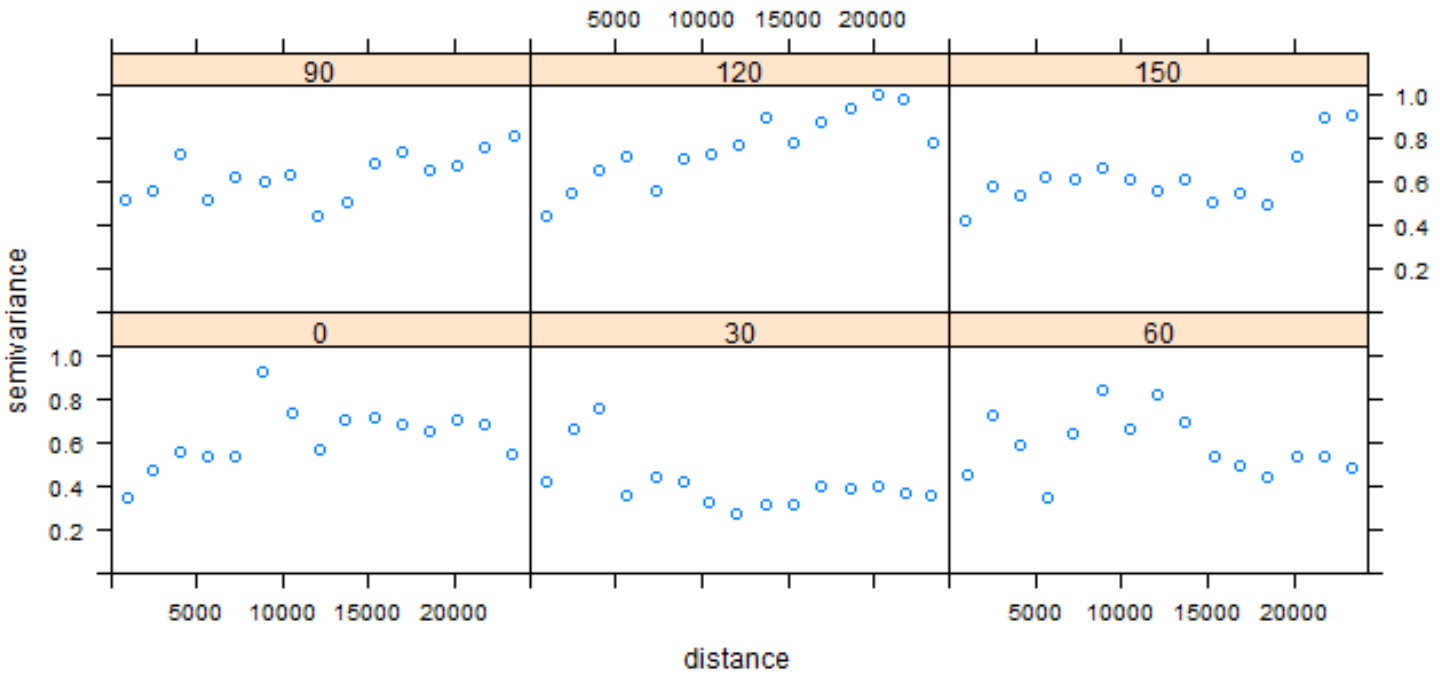

```
spdata <- data.frame(resid = resid(mod), x = frogs$x, y = frogs$y)
coordinates(spdata) <- c("x", "y")
bubble(spdata, "resid", col = c("blue", "orange"), main = "Residuals",
       xlab = "X-coordinates", ylab = "Y-coordinates")
```




```
vario.mod <- variogram(resid(mod) ~ 1, spdata)
plot(vario.mod)
```

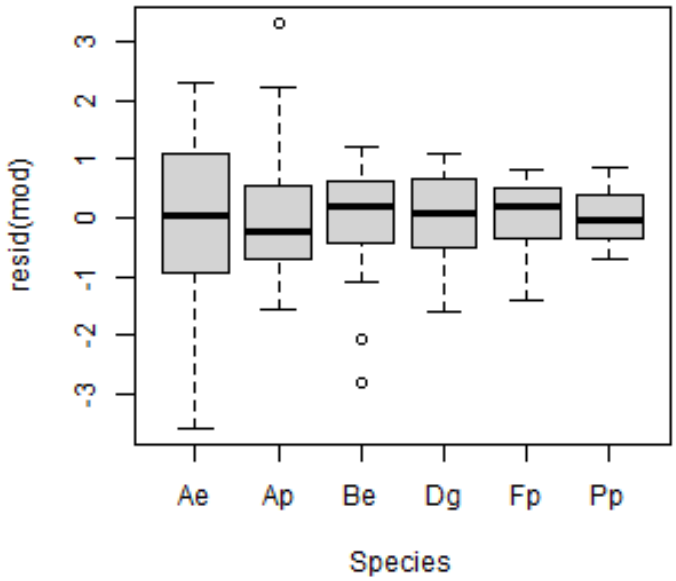


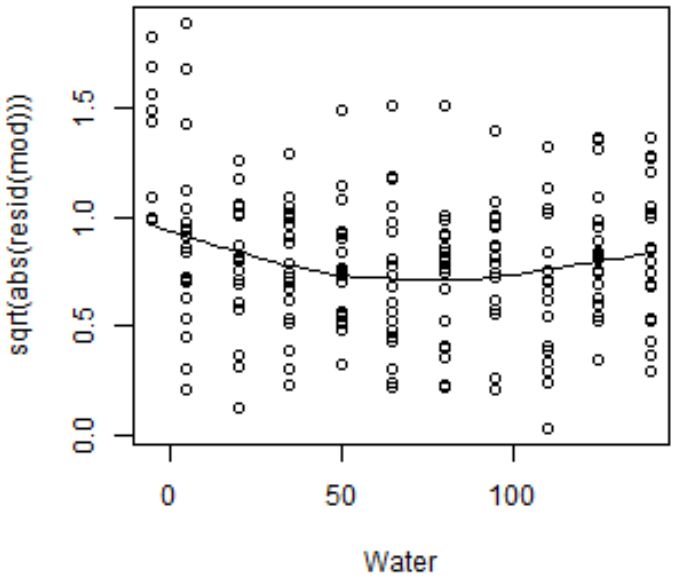

```
vario.mod.6dir <- variogram(resid(mod) ~ 1, spdata, alpha = seq(0, 150, by = 30))
plot(vario.mod.6dir)
```



Hetroscedasticity

```
data(ellenberg)
dat <- ellenberg[complete.cases(ellenberg[c("Yi.g", "Water",
                                           "Species")]), ]
mod <- lm(log(Yi.g) ~ Water + Species + Water:Species, dat)
par(mfrow=c(1, 2))
plot(resid(mod) ~ Species, dat)
scatter.smooth(dat$Water, sqrt(abs(resid(mod))), xlab = "Water")
```


resid(mod)
3
2
1
0
-1
-2
-3
Ae
Ap
Be
Dg
Fp
Pp
Species


sqrt(abs(resid(mod)))
1.5
1.0
0.5
0.0
0
50
100
Water