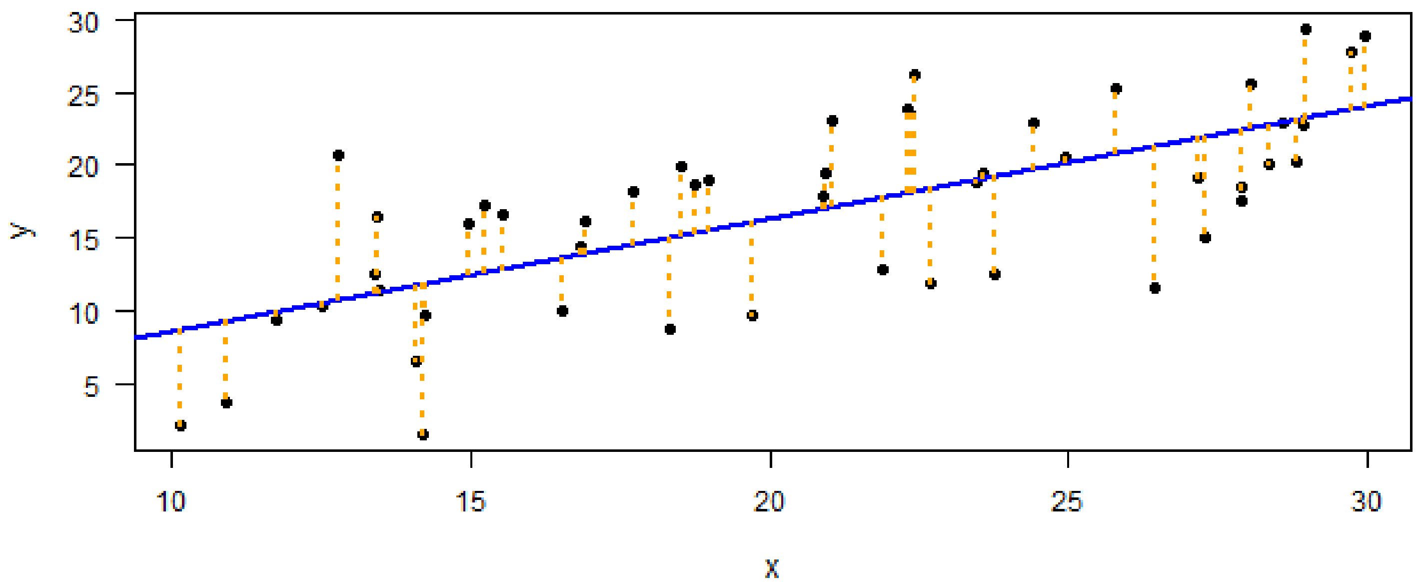
Normal Linear Models

Fitting a Linear Regression in R

```r
n <- 50 # sample size
sigma <- 5 # standard deviation of the residuals
b0 <- 2 # intercept
b1 <- 0.7 # slope
x <- runif(n, 10, 30) # sample values of the covariate
yhat <- b0 + b1*x
y <- rnorm(n, yhat, sd=sigma)

# plot the data
plot(x, y, pch=16, las=1, cex.lab=1.2)
abline(lm(y~x), lwd=2, col="blue") # insert regression line
# add residuals
segments(x, fitted(lm(y~x)), x, y, lwd=2, col="orange", lty=3)
```



Fit the model

```
mod <- lm(y~x)
mod
```

```
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
     0.8503       0.7745
```

```
summary(mod)
```

```
Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-10.3589  -3.9227   0.3873   3.6439   9.9424

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.8503     2.5438   0.334     0.74
```

```
x             0.7745     0.1179   6.567  3.4e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.904 on 48 degrees of freedom
Multiple R-squared:  0.4732,    Adjusted R-squared:  0.4622
F-statistic: 43.12 on 1 and 48 DF,  p-value: 3.399e-08
```

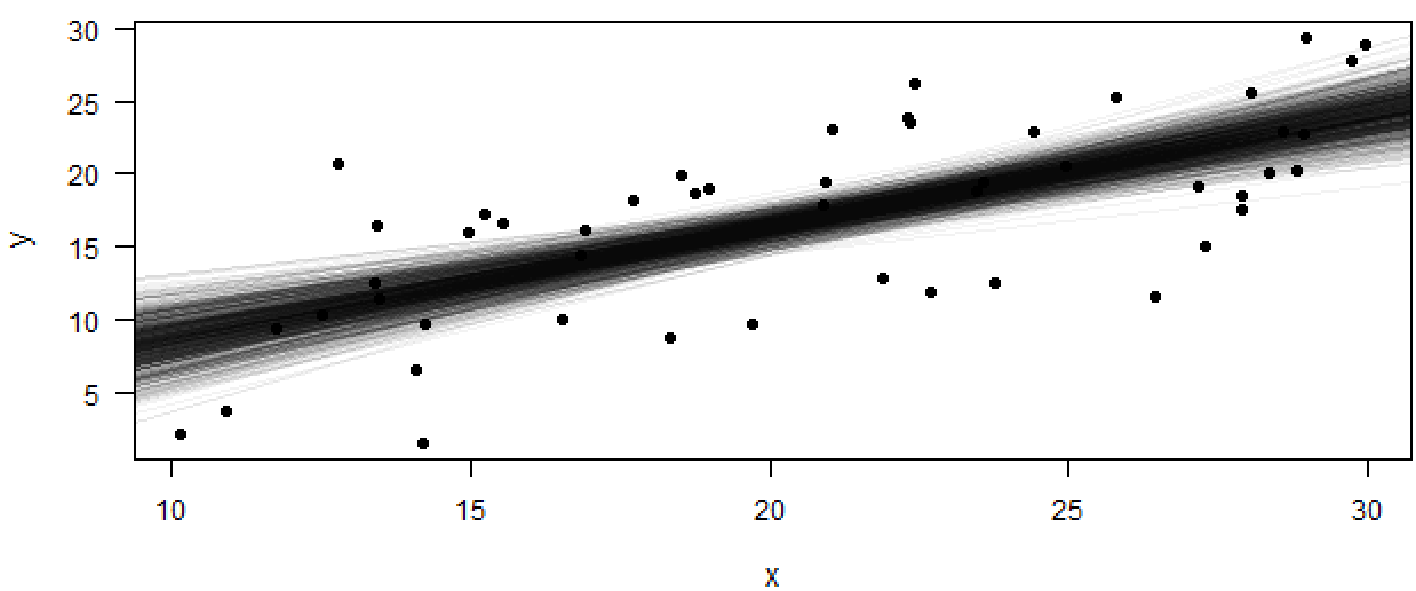
```
summary(mod)$sigma
```

```
[1] 4.904321
```

Drawing Conclusions

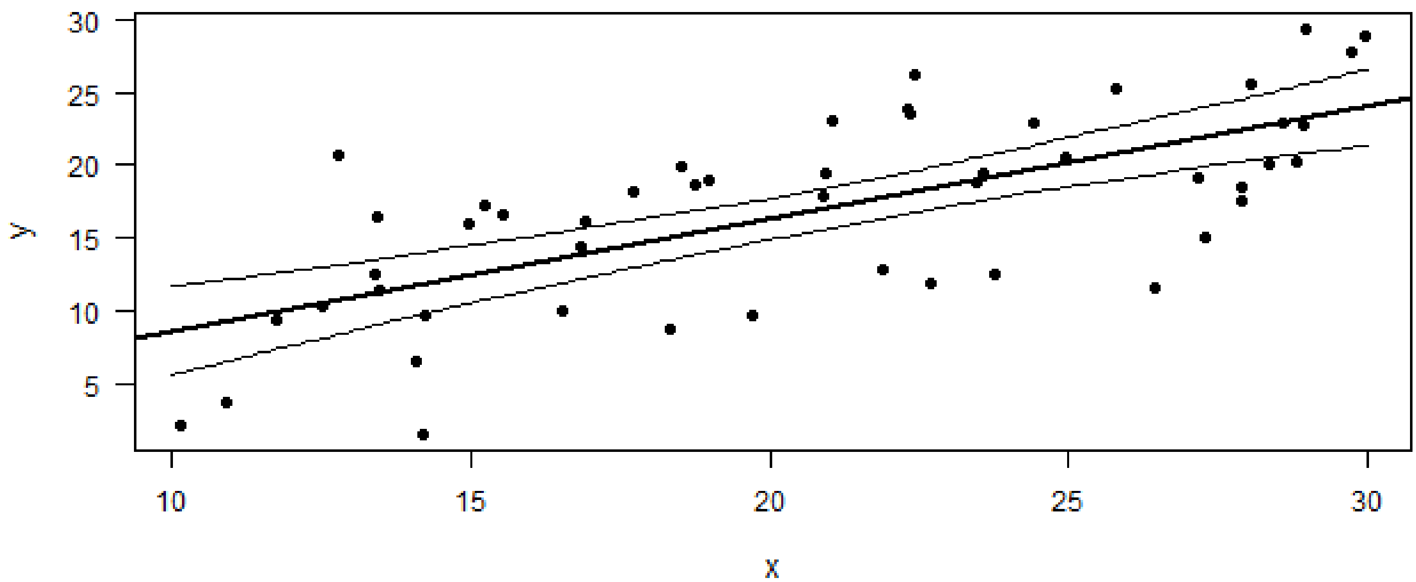
Alternative Display

```
nsim <- 1000
bsim <- sim(mod, n.sim = nsim)

plot(x, y, pch=16, las=1, cex.lab=1.2)
for(i in 1:nsim) abline(coef(bsim)[i, 1], coef(bsim)[i, 2],
                        col = rgb(0, 0, 0, 0.05))
```



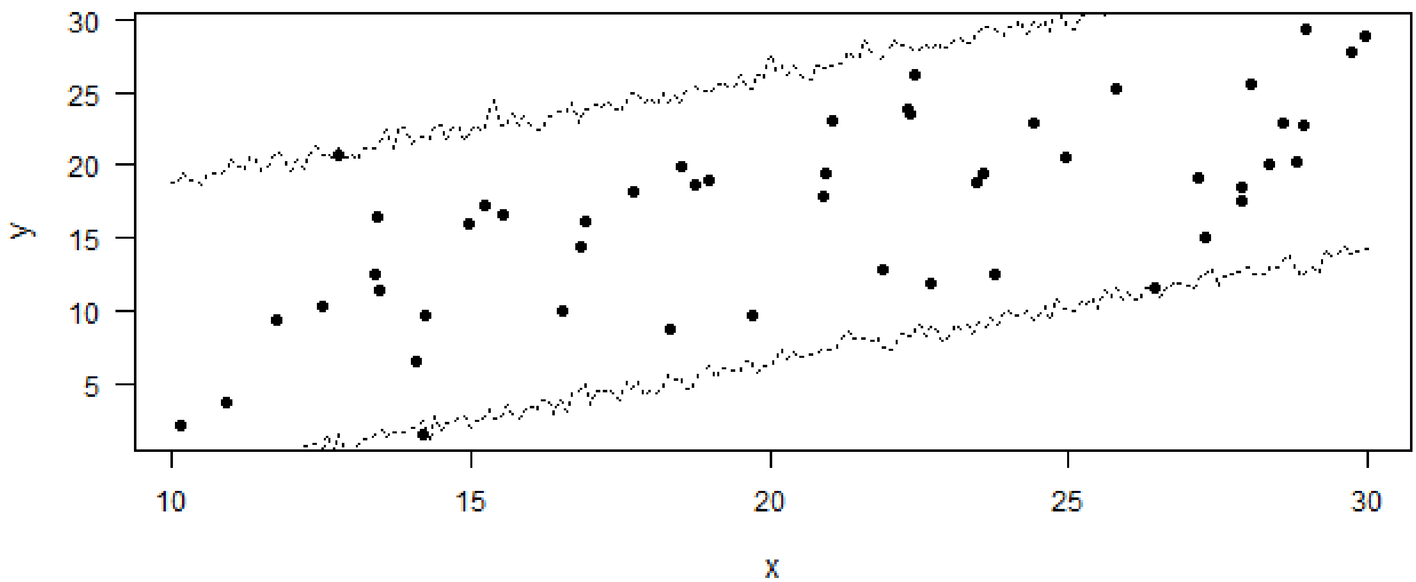

```
newdat <- data.frame(x = seq(10, 30, by=0.1))
newmodmat <- model.matrix(~x, data = newdat)
fitmat <- matrix(ncol = nsim, nrow = nrow(newdat))
```

```
for(i in 1:nsim) fitmat[, i] <- newmodmat %*% coef(bsim)[i, ]
plot(x, y, pch=16, las=1, cex.lab=1.2)
abline(mod, lwd=2)

lines(newdat$x, apply(fitmat, 1, quantile, prob = 0.025, lty=3))
lines(newdat$x, apply(fitmat, 1, quantile, prob=0.975, lty=3))
```



Predicting Future Observations

```
plot(x, y, pch=16, las=1, cex.lab=1.2)

# prepare matrix for simulated new data
newy <- matrix(ncol = nsim, nrow = nrow(newdat))
# for each simulate4d fitted value, simulate  one new y-value
for(i in 1:nsim) newy[, i] <- rnorm(nrow(newdat), mean = fitmat[, i],
                                  sd = bsim@sigma[i])

lines(newdat$x, apply(newy, 1, quantile, prob=0.025), lty=3)
lines(newdat$x, apply(newy, 1, quantile, prob=0.975), lty=3)
```




```r
sum(newy[newdat$x == 25,] > 20) / nsim
```

```
[1] 0.515
```

Frequentist Results

```r
summary(mod)
```

```

Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max 
-10.3589  -3.9227   0.3873   3.6439   9.9424 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   0.8503     2.5438   0.334     0.74    
x             0.7745     0.1179   6.567  3.4e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.904 on 48 degrees of freedom
Multiple R-squared:  0.4732,    Adjusted R-squared:  0.4622 
```

```
F-statistic: 43.12 on 1 and 48 DF,  p-value: 3.399e-08
```

Regression Variants: ANOVA, ANCOVA, and MLR

```
# data simulation
mu <- 12
sigma <- 2
b1 <- 3
b2 <- -5
n <- 90
group <- factor(rep(c(1, 2, 3), each = 30))

# simulate the y variable
simresid <- rnorm(n, mean=0, sd=sigma)
y <- mu + as.numeric(group == 2) * b1 + as.numeric(group == "3") * b2 +
   simresid
```

```
group <- factor(group) # define group as a factor
mod <- lm(y ~ group) # fit the model

mod
```

```
Call:
lm(formula = y ~ group)

Coefficients:
(Intercept)       group2       group3
     11.986        2.366       -5.010
```

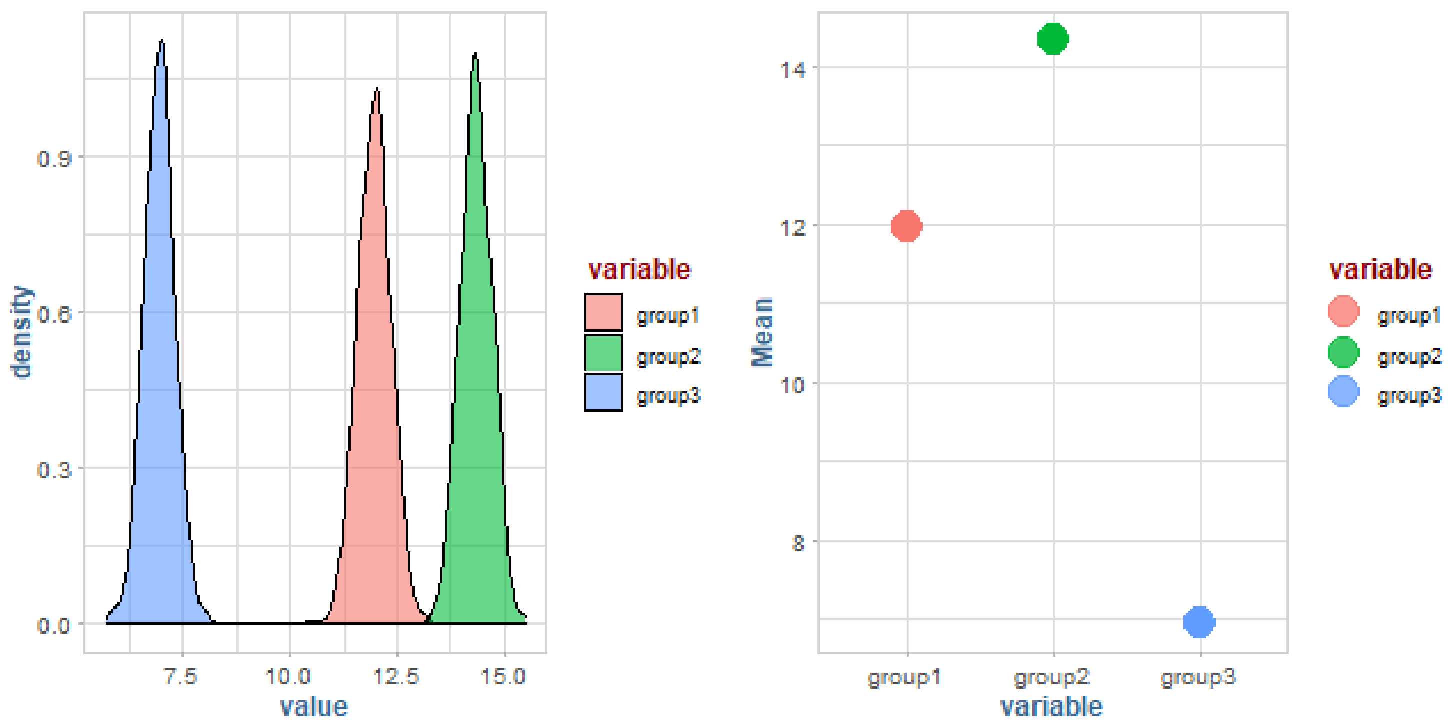
```
summary(mod)$sigma
```

```
[1] 2.003371
```

```
bsim <- sim(mod, n.sim=1000)
m.g1 <- coef(bsim)[, 1]
m.g2 <- coef(bsim)[, 1] + coef(bsim)[, 2]
m.g3 <- coef(bsim)[, 1] + coef(bsim)[, 3]

m.dat <- data.table(group1 = m.g1, group2 = m.g2, group3 = m.g3)
m.dat.long <- melt(m.dat)
```

```
Warning in melt.data.table(m.dat): id.vars and measure.vars are internally
guessed when both are 'NULL'. All non-numeric/integer/logical type columns are
considered id.vars, which in this case are columns []. Consider providing at
least one of 'id' or 'measure' vars in future.
```

```
m.dat.long[, Mean := mean(value), by = variable]

p1 <- ggplot(m.dat.long) +
   geom_density(aes(value, fill = variable), alpha = .6)

p2 <- ggplot(m.dat.long) +
   geom_point(aes(x = variable, y = Mean, color = variable), alpha = .75, size = 6)

grid.arrange(p1, p2, ncol=2)
```




```
d.g1.2 <- m.g1 - m.g2
mean(d.g1.2)
```

```
[1] -2.37292
```

```
quantile(d.g1.2, prob=c(0.025, 0.975))
```

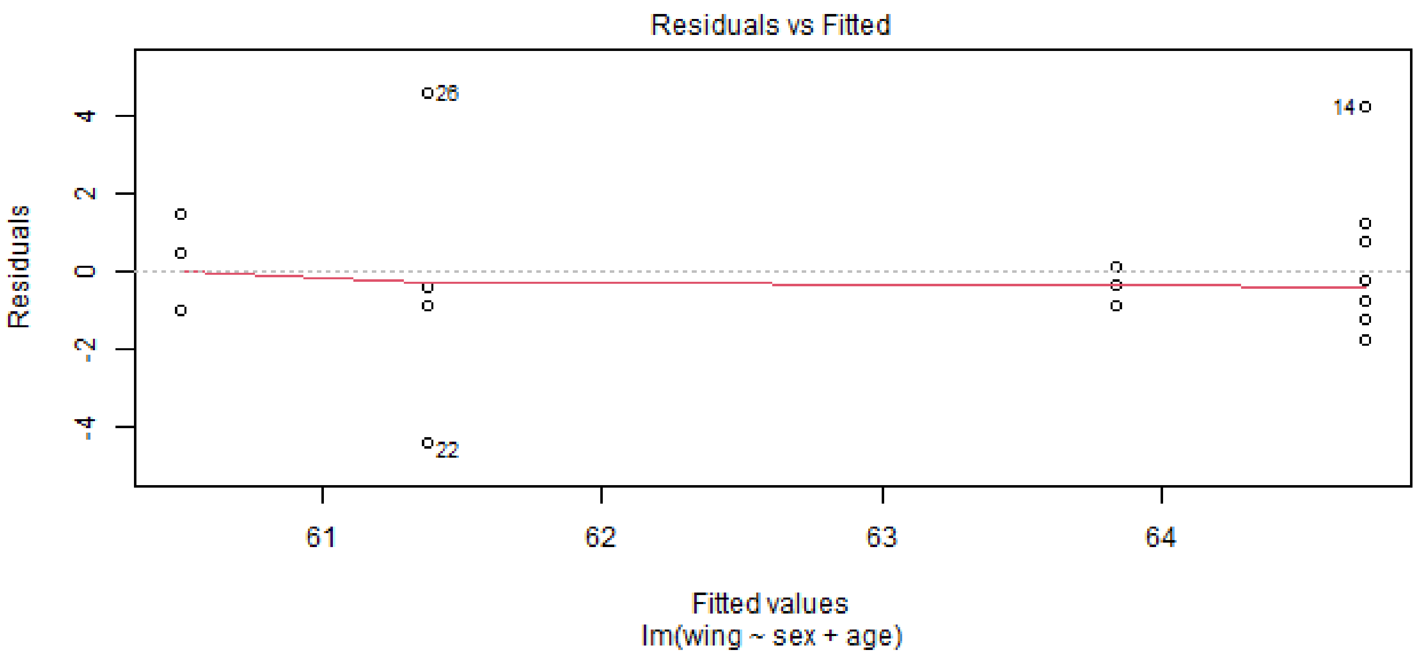
```
     2.5%     97.5%
-3.431189 -1.391826
```

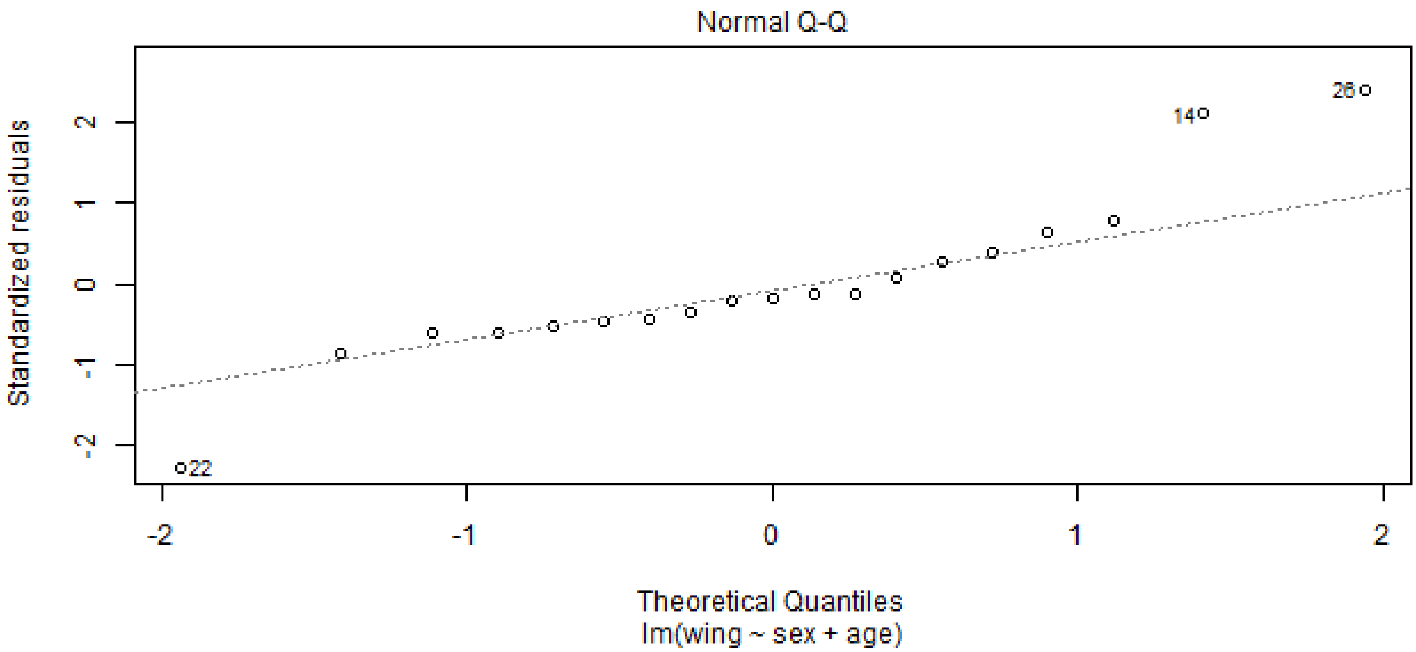
```
sum(m.g2 > m.g1) / nsim
```

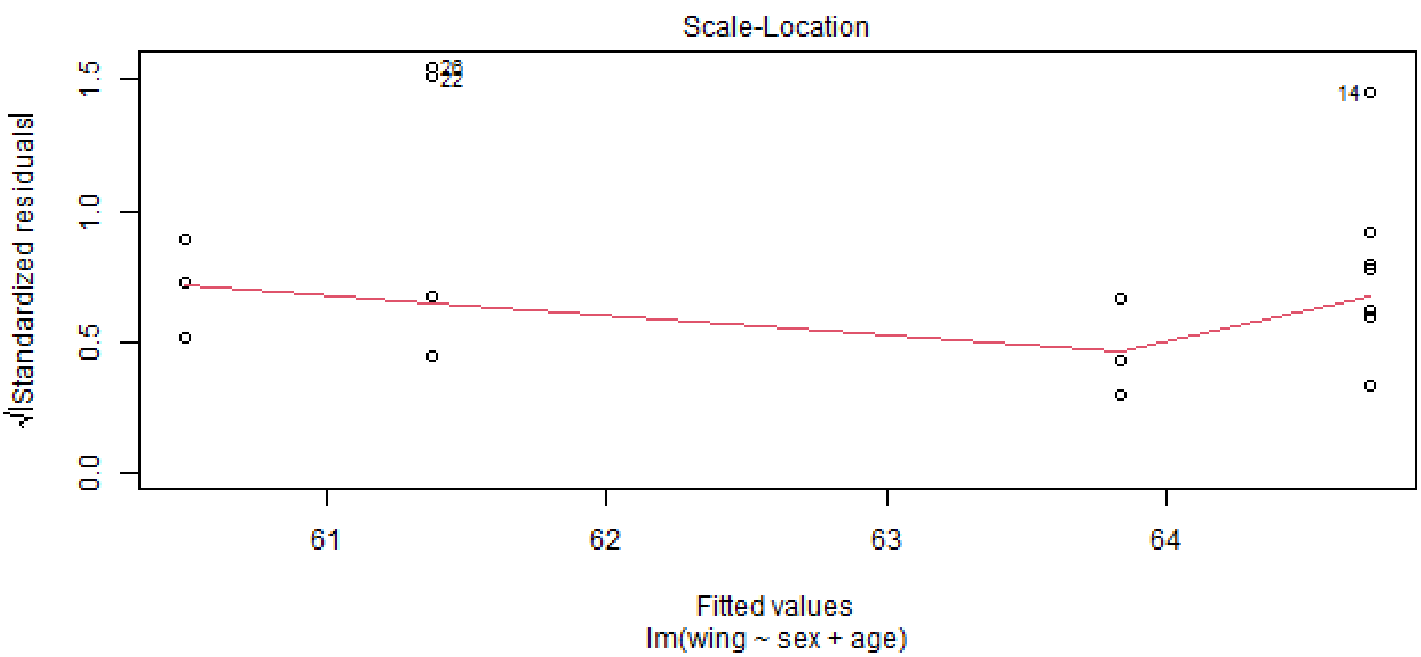
```
[1] 1
```

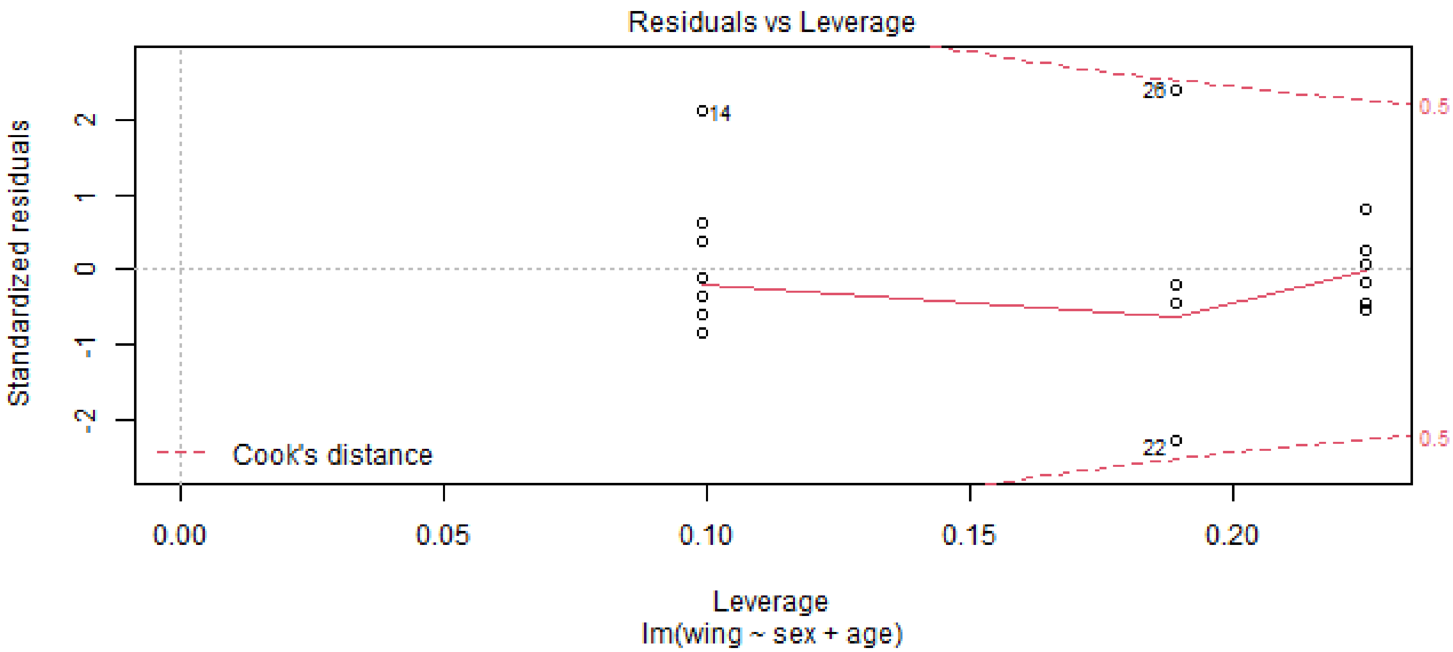
Frequentist Results

```
data("periparusater")
dat <- periparusater

mod <- lm(wing ~ sex + age, data=dat)
plot(mod)
```




Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(wing ~ sex + age)
26
14
22
-2
-1
0
1
2


Scale-Location
√|Standardized residuals|
Fitted values
lm(wing ~ sex + age)
8
26
22
14
0.0
0.5
1.0
1.5
61
62
63
64




```r
newdat <- expand.grid(sex = factor(c(1, 2)), age = factor(c(3, 4)))
newdat$fit <- model.matrix(~sex + age, dat=newdat) %*% coef(mod)

nsim <- 2000
bsim <- sim(mod, n.sim = nsim)

fitmat <- matrix(ncol = nsim, nrow = nrow(newdat))
Xmat <- model.matrix(formula(mod)[c(1,3)], dat = newdat)
for(i in 1:nsim) fitmat[, i] <- Xmat %*% bsim@coef[i, ]

alpha <- .05
intervals <- c(lower = alpha/2, upper = 1 - alpha/2)

ci <- t(apply(fitmat, 1, quantile, prob = intervals))

cbind(newdat, ci)
```

```
  sex age      fit     2.5%    97.5%
1   1   3 63.83784 61.67971 66.01047
2   2   3 60.49550 58.38732 62.74170
3   1   4 64.72072 63.26183 66.17690
4   2   4 61.37838 59.39621 63.29541
```

```r
mod2 <- lm(wing ~ sex * age, data = dat)
```

```
bsim2 <- sim(mod2, n.sim = nsim)
quantile(bsim2@coef[, 4], prob = c(0.025, 0.5, 0.975))
```

```
     2.5%       50%     97.5%
-5.776537 -1.007294  3.542257
```

```
summary(mod2)$sigma
```

```
[1] 2.18867
```

```
mean(abs(bsim2@coef[, 4]) > 0.3)
```

```
[1] 0.9045
```

```
coef(mod2)
```

```
(Intercept)        sex2        age4    sex2:age4
  63.500000   -2.666667    1.333333   -1.041667
```

```
quantile(bsim2@coef[, 2], prob = c(0.025, 0.5, 0.975))
```

```
     2.5%       50%     97.5%
-6.134490 -2.679339  1.437079
```

```
sum(bsim@coef[, 2] < 0)/nsim # for juveniles (reference level)
```

```
[1] 0.997
```

```
sum(bsim@coef[, 2] + bsim2@coef[, 4] < 0) / nsim # adults
```

```
[1] 0.9535
```

Analysis of Covariance

```
data("ellenberg")

index <- is.element(ellenberg$Species, c("Ap", "Dg"))
dat <- ellenberg[index, ] # select two species
dat <- droplevels(dat) # drop unused factor levels
str(dat) # definitions
```

```
'data.frame':   88 obs. of  29 variables:
 $ Year         : int  1952 1952 1952 1952 1952 1952 1952 1952 1952 1952 ...
 $ Soil         : Factor w/ 2 levels "Loam","Sand": 1 1 1 1 1 1 1 1 1 1 ...
 $ Water        : int  -5 5 20 35 50 65 80 95 110 125 ...
 $ Species      : Factor w/ 2 levels "Ap","Dg": 1 1 1 1 1 1 1 1 1 1 ...
 $ Mi.g         : num  NA 112.6 66.1 42.3 38.4 ...
 $ Yi.g         : num  NA 34.8 28 44.5 24.8 ...
```

```
 $ Mono.area.m2: num  NA 0.383 0.383 0.383 0.383 0.383 0.383 0.383 0.383 0.383 ...
 $ Mix.area.m2 : num  NA 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 ...
 $ Div         : int  NA 6 6 6 6 6 6 6 6 6 ...
 $ Moi.g.m2    : num  NA 294 173 110 100 ...
 $ Yoi.g.m2    : num  NA 29 23.3 37.1 20.6 ...
 $ Mo.g.m2     : num  NA 377 356 281 208 ...
 $ Yo.g.m2     : num  NA 229 291 299 314 ...
 $ RYoi        : num  NA 0.0985 0.1352 0.3357 0.2057 ...
 $ RYo         : num  NA 0.67 0.697 1.049 1.011 ...
 $ Yei.g.m2    : num  NA 49 28.8 18.4 16.7 ...
 $ Ye.g.m2     : num  NA 377 356 281 208 ...
 $ RRYo        : num  NA 0.147 0.194 0.32 0.204 ...
 $ deltaRYoi   : num  NA -0.04854 -0.05873 0.01577 0.00219 ...
 $ deltaRYo    : num  NA -0.05501 -0.05046 0.00821 0.0018 ...
 $ RYe         : num  NA 0.167 0.167 0.167 0.167 ...
 $ deltaRYe    : num  NA -0.0196 0.0273 0.1532 0.0369 ...
 $ RYT         : num  NA 0.67 0.697 1.049 1.011 ...
 $ level       : Factor w/ 1 level "species": 1 1 1 1 1 1 1 1 1 1 ...
 $ NE          : num  NA NA NA NA NA NA NA NA NA NA ...
 $ TICE        : num  NA NA NA NA NA NA NA NA NA NA ...
 $ SE          : num  NA NA NA NA NA NA NA NA NA NA ...
 $ TDCE        : num  NA NA NA NA NA NA NA NA NA NA ...
 $ DE          : num  NA NA NA NA NA NA NA NA NA NA ...
```

```r
mod <- lm(log(Yi.g) ~ Species + Water, data = dat)

head(model.matrix(mod)) # print the first 6 rows of the matrix
```

```
   (Intercept) SpeciesDg Water
24           1         0     5
25           1         0    20
26           1         0    35
27           1         0    50
28           1         0    65
29           1         0    80
```

```r
summary(mod)
```

```

Call:
lm(formula = log(Yi.g) ~ Species + Water, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.8967 -0.6418 -0.1263  0.4482  4.0191 
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.678937   0.225652  16.304  < 2e-16 ***
SpeciesDg    1.065942   0.217117   4.910 4.66e-06 ***
Water       -0.008443   0.002403  -3.513 0.000728 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.995 on 81 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.3103,    Adjusted R-squared:  0.2933
F-statistic: 18.22 on 2 and 81 DF,  p-value: 2.919e-07
```

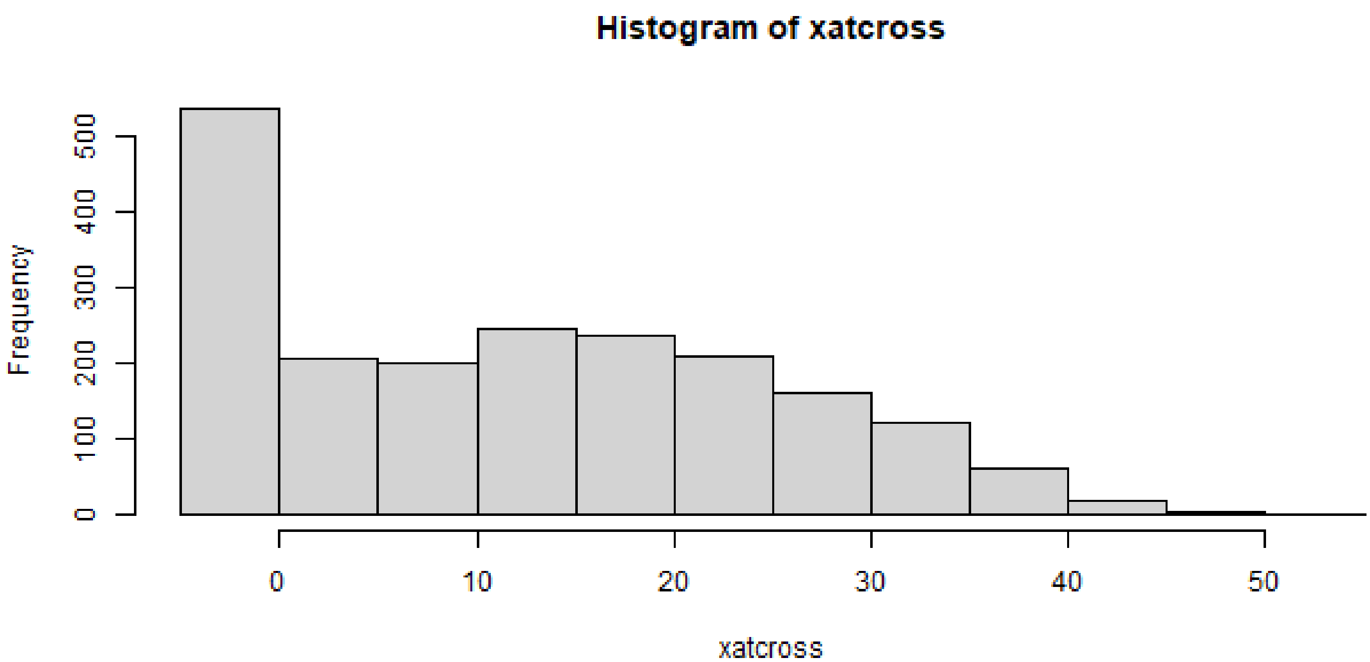
```r
mod2 <- lm(log(Yi.g) ~ Species*Water, data = dat)
summary(mod2)
```

```
Call:
lm(formula = log(Yi.g) ~ Species * Water, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-1.6158 -0.6135 -0.1063  0.5876  3.3203

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      4.330406   0.253108  17.109  < 2e-16 ***
SpeciesDg       -0.236995   0.357948  -0.662     0.51
Water           -0.017911   0.003075  -5.825 1.15e-07 ***
SpeciesDg:Water  0.018935   0.004349   4.354 3.92e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9002 on 80 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.4425,    Adjusted R-squared:  0.4215
F-statistic: 21.16 on 3 and 80 DF,  p-value: 3.454e-10
```

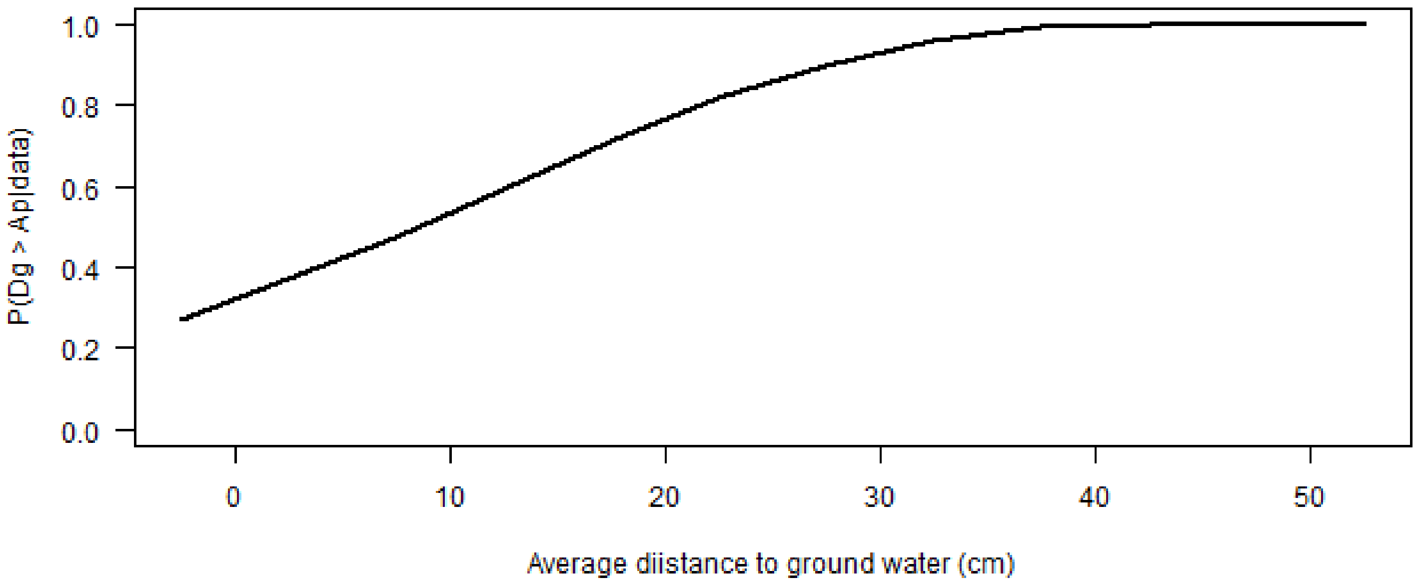
```r
nsim <- 2000
bsim <- sim(mod2, n.sim = nsim)
coefs <- coef(bsim)

xatcross <- crosspoint(coefs[, 1], coefs[, 3],
                       coefs[, 1] + coefs[, 2], coefs[, 3] + coefs[, 4])[ , 1]

xatcross[xatcross< (-5)] <- -5
```

```
th <- hist(xatcross, breaksw = seq(-5.5, 140.5, by=5))
```

```
Warning in plot.window(xlim, ylim, "", ...): "breaksw" is not a graphical
parameter

Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
"breaksw" is not a graphical parameter

Warning in axis(1, ...): "breaksw" is not a graphical parameter

Warning in axis(2, ...): "breaksw" is not a graphical parameter
```




```
plot(th$mids, cumsum(th$counts)/2000, type = "l", lwd=2, las=1,
     ylim=c(0, 1), ylab="P(Dg > Ap|data)", xlab="Average diistance to ground water (cm)")
```



Multiple Regression and Collinearity

```r
data(mdat)
mod <- lm(y ~ x1 + x2, data = mdat)
summary(mod)
```

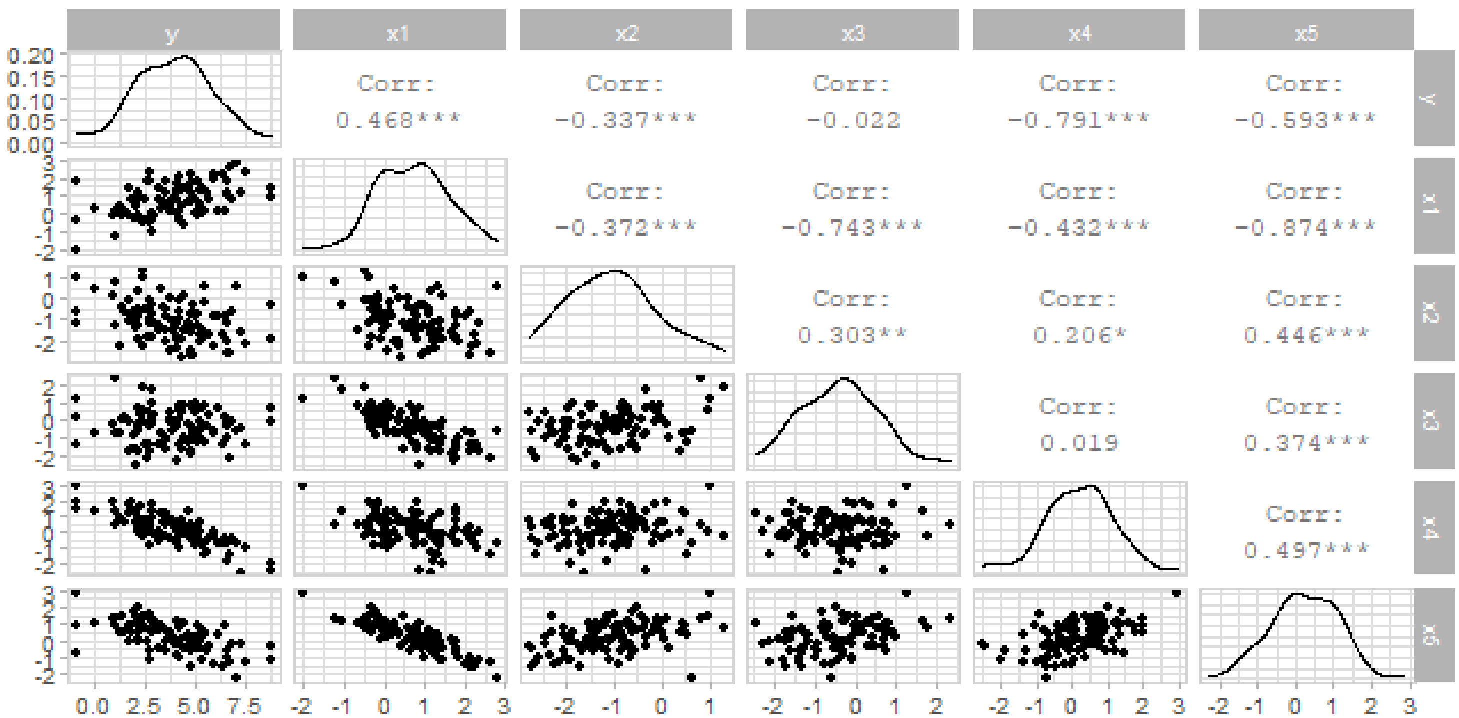
```

Call:
lm(formula = y ~ x1 + x2, data = mdat)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.5306 -1.0652 -0.0037  1.0613  4.9743 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.8867     0.2699  10.695  < 2e-16 ***
x1            0.8716     0.2076   4.199 5.96e-05 ***
x2           -0.4009     0.2011  -1.993    0.049 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.704 on 97 degrees of freedom
```

```
Multiple R-squared:  0.2497,    Adjusted R-squared:  0.2343
F-statistic: 16.14 on 2 and 97 DF,  p-value: 8.868e-07
```

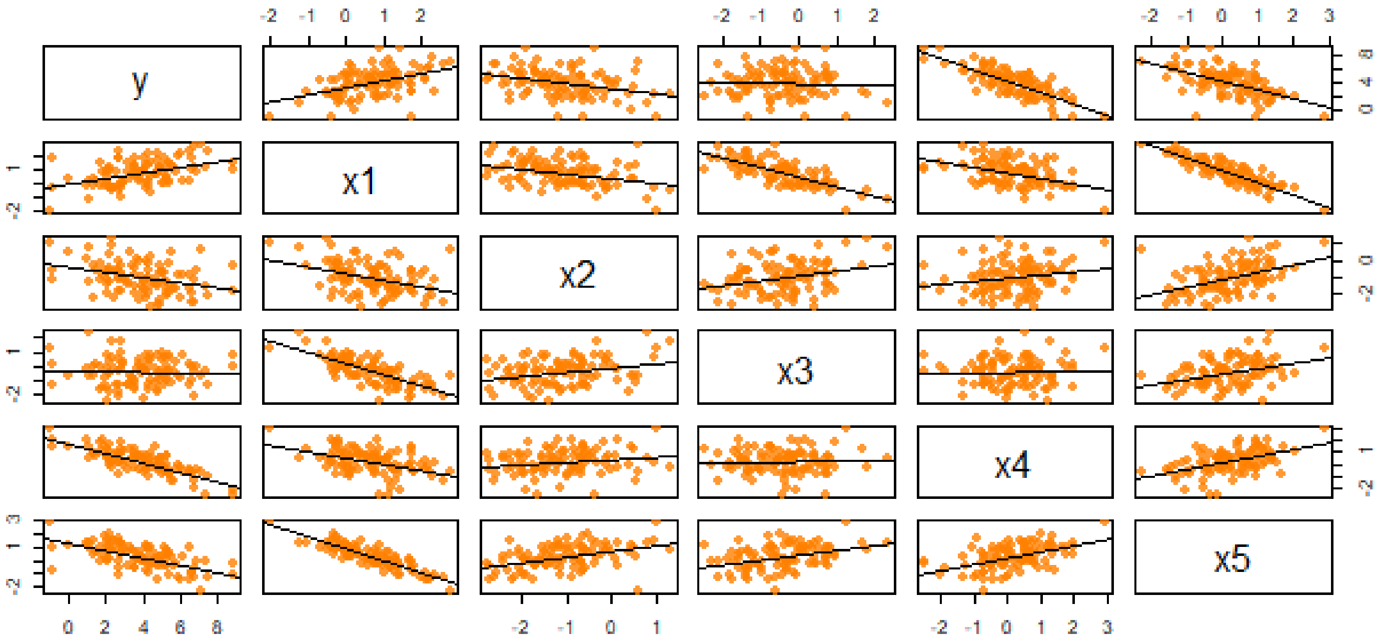
```r
cor(mdat[, 2:6])
```

```
           x1         x2         x3         x4         x5
x1  1.0000000 -0.3717495 -0.7429106 -0.4324213 -0.8743843
x2 -0.3717495  1.0000000  0.3025238  0.2061209  0.4463830
x3 -0.7429106  0.3025238  1.0000000  0.0191966  0.3736499
x4 -0.4324213  0.2061209  0.0191966  1.0000000  0.4966910
x5 -0.8743843  0.4463830  0.3736499  0.4966910  1.0000000
```

```r
ggpairs(mdat)
```




```r
own.graph <- function(x, y) {
   points(x, y, pch=16, col=rgb(1, 0.5, 0.0, 0.8))
   abline(lm(y~x))
}
pairs(mdat, panel = own.graph)
```



Ordered Factors and Constants

```r
data("swallows")
levels(swallows$nesting_aid)
```

```
[1] "artif_nest" "both"       "none"       "support"
```

```r
str(swallows)
```

```
'data.frame':	27 obs. of  6 variables:
 $ farm       : int  1 2 3 4 5 6 7 8 9 10 ...
 $ nhirrus    : int  3 2 2 0 0 1 2 6 1 2 ...
 $ ndelurb    : int  3 0 10 0 3 0 0 0 2 0 ...
 $ ncows      : int  45 8 25 0 0 30 18 35 0 10 ...
 $ nesting_aid: Factor w/ 4 levels "artif_nest","both",..: 1 3 4 3 2 4 2 4 4 3 ...
 $ ndaysempty : int  0 5 0 60 60 0 0 0 60 3 ...
```

```r
contrasts(swallows$nesting_aid)
```

```
           both none support
artif_nest    0    0       0
both          1    0       0
none          0    1       0
support       0    0       1
```

```
swallows$nesting_aid <- factor(swallows$nesting_aid, levels =
                                 c("none", "support", "artif_nest", "both"),
                                 ordered = T)
levels(swallows$nesting_aid)
```

```
[1] "none"       "support"    "artif_nest" "both"
```

```
contrasts(swallows$nesting_aid)
```

```
             .L   .Q         .C
[1,] -0.6708204  0.5 -0.2236068
[2,] -0.2236068 -0.5  0.6708204
[3,]  0.2236068 -0.5 -0.6708204
[4,]  0.6708204  0.5  0.2236068
```